MOBILE AND DIGITAL FORENSICS

IT-428

PROJECT REPORT

SUBMITTED BY:

DHRUV CHANDEL: 2K18/IT/046

KARTIK BANSAL: 2K18/IT/064

SUBMITTED TO:

Ananya Pandey

Department of Information Technology

https://github.com/dhruvchandel/Reverse_shell

1. Introduction

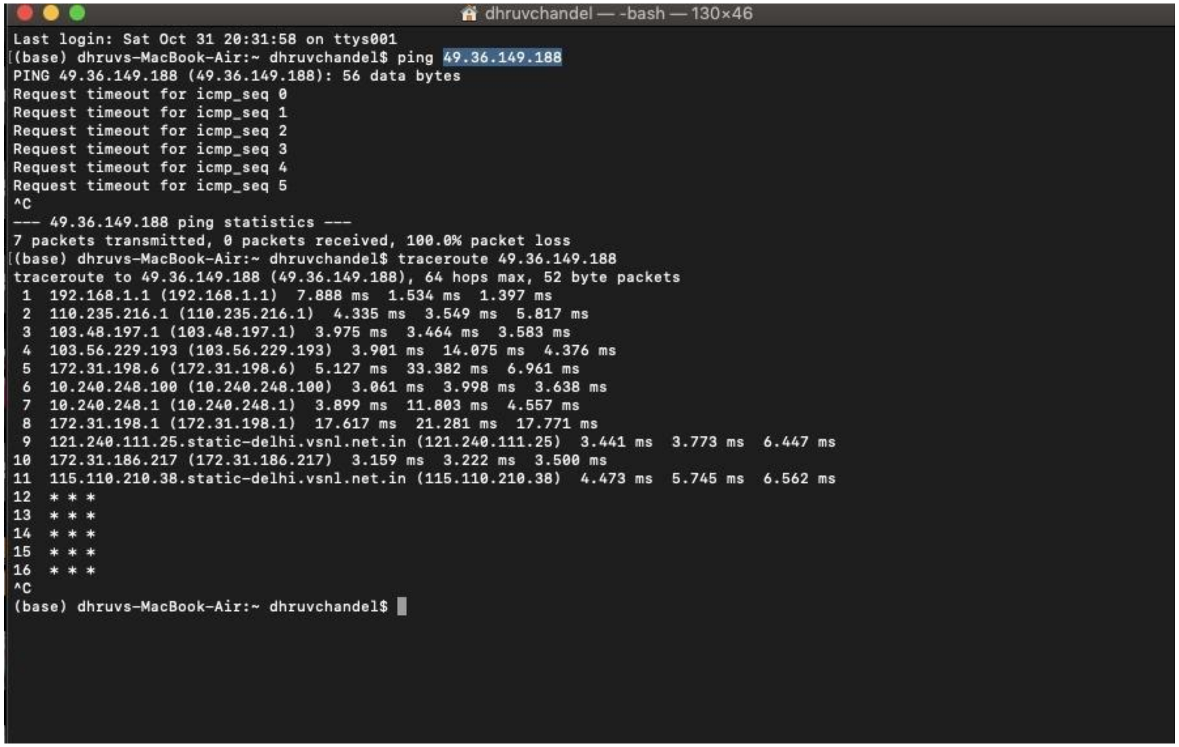
- In computing, a shell is a computer program which exposes an operating system's services to a human user or other program. In general, operating system shells use either a command line interface or graphical user interface, depending on a computer's role and particular operation.
- A reverse shell is a type of shell in which the target machine (Client) communicates back to the attacking machine (Server). The attacking machine has a listener port on which it receives the connection, which by using, code or command execution is achieved.
- A firewall usually blocks incoming connections on open ports, but does not block outgoing traffic and since in reverse shell the connection is initiated by client itself; therefore, reverse shell can easily bypass firewall and server/host can easily execute commands on client/target machine.

2. Idea

- Reverse shell is a powerful tool **as they are** the only way to perform remote maintenance on hosts behind a NAT, so they have legitimate administrative **uses**. However, they can also be **used** by cybercriminals to execute operating system commands on hosts protected from incoming connections by a firewall or other network security systems.
- So, our Aim is to implement a multi-client reverse shell program using socket programming in python which has various features which are described in further sections.

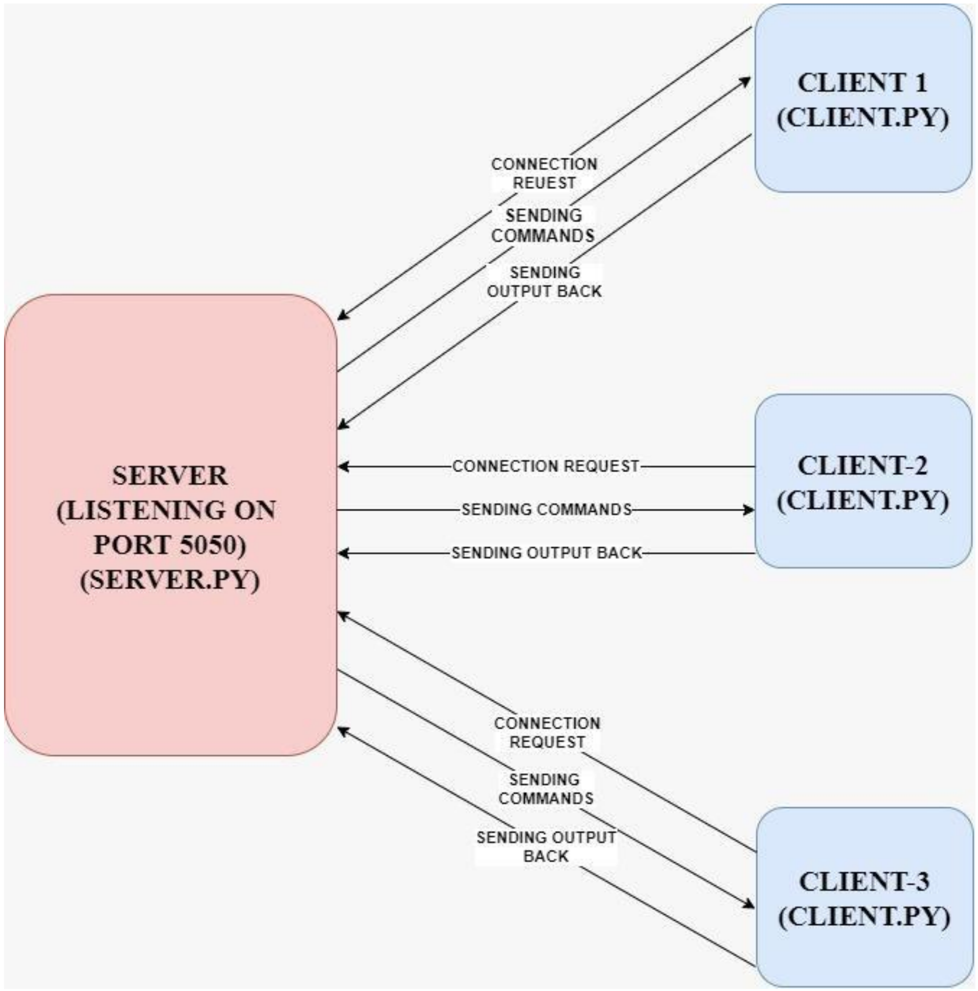
3. DESIGN

- Connection Initiation Request
  - LAN Connection: When ‘server’ and ‘client’ are present on same LAN then, connection can easily be established using private IP of the server only.
  - Connection Outside LAN: when ‘server’ and ‘client’ are present on different networks and want to connect using public Internet (through ISP), then public IP of server needs to be accessible by the client.



```
Last login: Sat Oct 31 20:31:58 on ttys001
(base) dhruvs-MacBook-Air:~ dhruvchandel$ ping 49.36.149.188
PING 49.36.149.188 (49.36.149.188): 56 data bytes
Request timeout for icmp_seq 0
Request timeout for icmp_seq 1
Request timeout for icmp_seq 2
Request timeout for icmp_seq 3
Request timeout for icmp_seq 4
Request timeout for icmp_seq 5
^C
--- 49.36.149.188 ping statistics ---
7 packets transmitted, 0 packets received, 100.0% packet loss
(base) dhruvs-MacBook-Air:~ dhruvchandel$ traceroute 49.36.149.188
traceroute to 49.36.149.188 (49.36.149.188), 64 hops max, 52 byte packets
 1  192.168.1.1 (192.168.1.1)  7.888 ms  1.534 ms  1.397 ms
 2  110.235.216.1 (110.235.216.1)  4.335 ms  3.549 ms  5.817 ms
 3  103.48.197.1 (103.48.197.1)  3.975 ms  3.464 ms  3.583 ms
 4  103.56.229.193 (103.56.229.193)  3.901 ms  14.075 ms  4.376 ms
 5  172.31.198.6 (172.31.198.6)  5.127 ms  33.382 ms  6.961 ms
 6  10.240.248.100 (10.240.248.100)  3.061 ms  3.998 ms  3.638 ms
 7  10.240.248.1 (10.240.248.1)  3.899 ms  11.803 ms  4.557 ms
 8  172.31.198.1 (172.31.198.1)  17.617 ms  21.281 ms  17.771 ms
 9  121.240.111.25.static-delhi.vsnl.net.in (121.240.111.25)  3.441 ms  3.773 ms  6.447 ms
10  172.31.186.217 (172.31.186.217)  3.159 ms  3.222 ms  3.500 ms
11  115.110.210.38.static-delhi.vsnl.net.in (115.110.210.38)  4.473 ms  5.745 ms  6.562 ms
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
^C
(base) dhruvs-MacBook-Air:~ dhruvchandel$
```

    - ✦ But usually due to firewalls deployed by most ISPs on their internal Networks public IPs are not accessible, and also most ISPs don’t allow port-forwarding on their routers.
    - ✦ So, to tackle above stated problem, we can use a third-party VPN server (which allows port forwarding) to act as an intermediate to pass information from server to client and vice-versa and successfully (using VPN tunnelling). *E.g.: OpenVPN*
- Sending Commands From Server to Client
  - Interactive shell -> We created an interactive shell (KD) that provides various commands and functionalities for multi-client support which are mentioned in further section.



- Receiving infrastructure (client to server)

- We used a send and receive infrastructure while transferring of files and commands in which we send an acknowledgement for each message received at either end and thus ensured that entire message sent from one end is received at other end successfully.

4. FEATURES

- **Interactive Shell Commands**
  - Provides interactive shell for easy management and handling of multiple clients from single server.
  - Commands :
    - List
    - Help
    - Select <client_id>
    - exit
- **Webcam Feed**
  - It returns a webcam feed of the client machine (Pre-set at dur=6s and fps=20)
    - Command = webcam
- **Screenshot**
  - Displays .png format screenshot of the client machine ○ Command = ss
- **File Transfer**
  - File Transfer to and fro between client and server.
  - Command (to retrieve file from client machine at the given file path) = getfile filepath

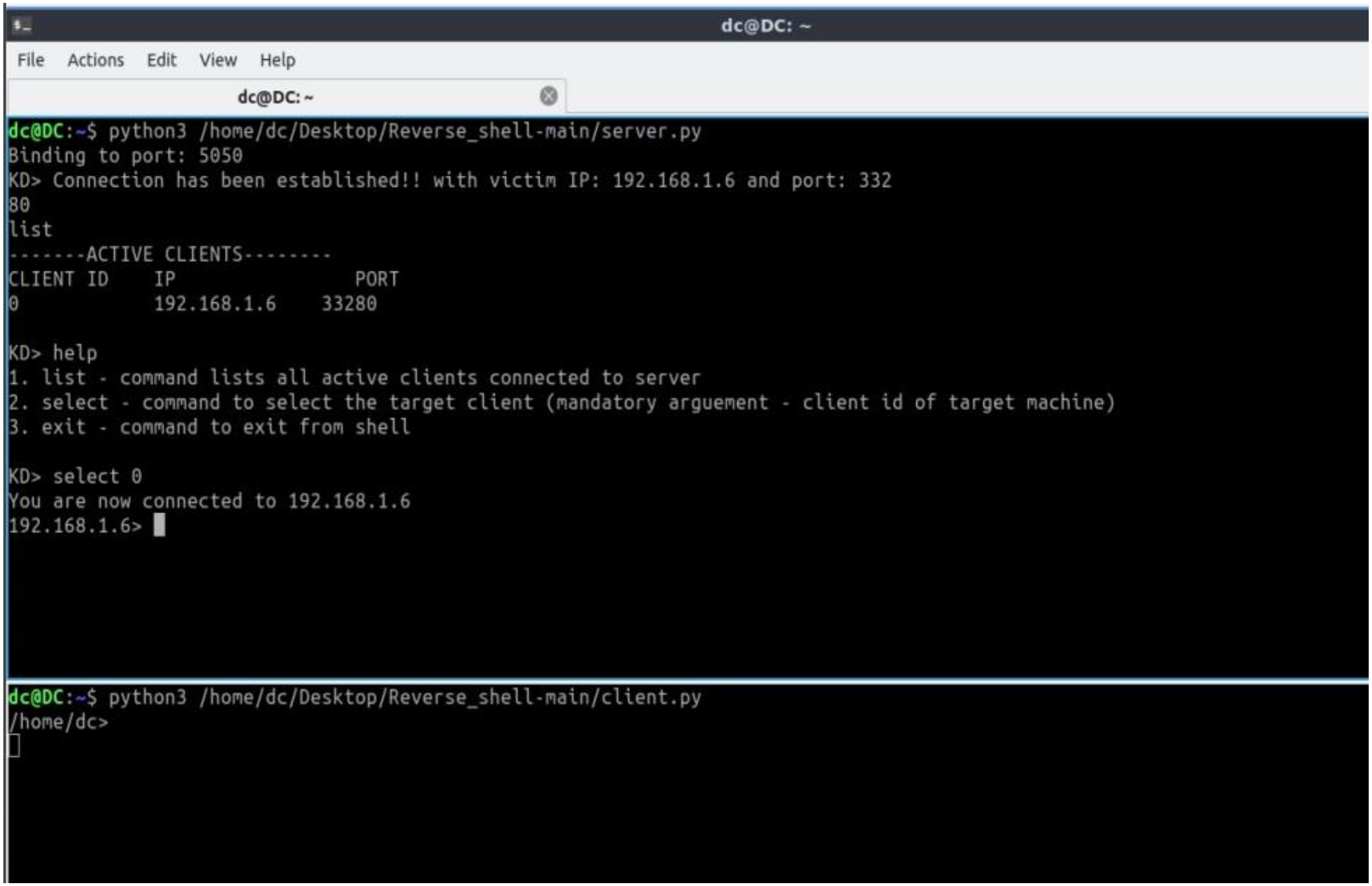
  - Command (to send file from server to client at the given file path) = sendfile filepath
- **Live Screen Feed**
  - It returns a screen feed of the client machine (Pre-set at dur=6s and fps=20)
  - Command = rec
- **System Information**
  - Returns Host name, OS version, chip information etc.
  - Command = sysinfo
- **Shell Commands**
  - Execute terminal/cmd commands on the client machine.
- Keylogger
  - Stores keyboard key's presses in logs.txt file upto preset 50 character/until 'esc' is pressed.
  - Command = keylogger

Table I: Implementation Environment, an example

Programming Language(s)	Python 3
Operating System	Cross-Platform (Windows/Mac/Linux)
Library Packages or APIs used (if any)	Subprocess, threading, os, pyautogui, opencv, pyscreenshot, socket, sys, queue, platform
Interface Design (GUI / Web / Other)	Interactive Shell
Servers Used (if any)	OpenVpn (Non LAN connections)

https://github.com/dhruvchandel/Reverse_shell

5. RESULTS



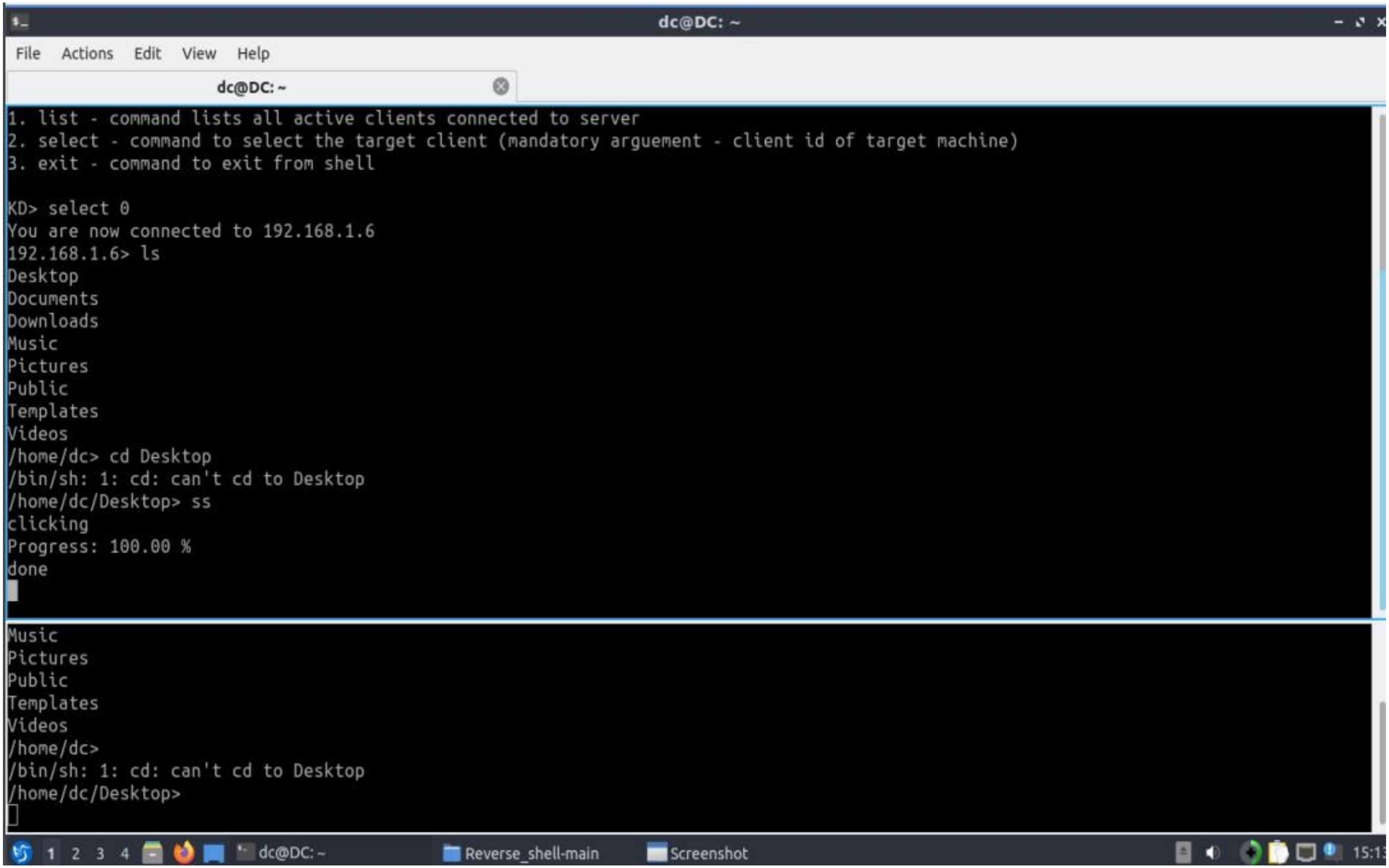
```
dc@DC:~$ python3 /home/dc/Desktop/Reverse_shell-main/server.py
Binding to port: 5050
KD> Connection has been established!! with victim IP: 192.168.1.6 and port: 332
80
list
-------ACTIVE CLIENTS--------
CLIENT ID    IP                PORT
0            192.168.1.6    33280

KD> help
1. list - command lists all active clients connected to server
2. select - command to select the target client (mandatory arguement - client id of target machine)
3. exit - command to exit from shell

KD> select 0
You are now connected to 192.168.1.6
192.168.1.6>
```

```
dc@DC:~$ python3 /home/dc/Desktop/Reverse_shell-main/client.py
/home/dc>
```

https://github.com/dhruvchandel/Reverse_shell

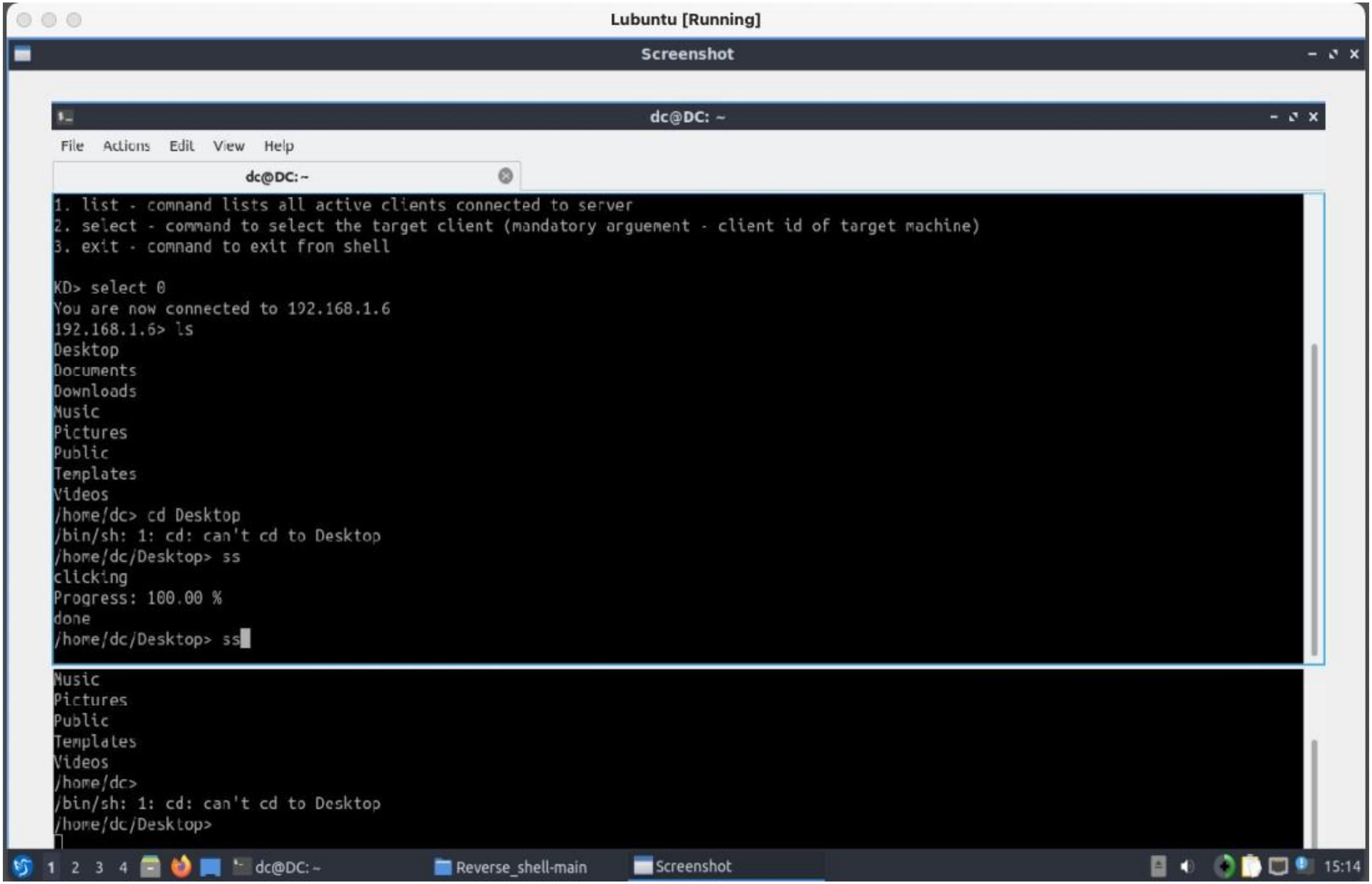


```
1. list - command lists all active clients connected to server
2. select - command to select the target client (mandatory arguement - client id of target machine)
3. exit - command to exit from shell

KD> select 0
You are now connected to 192.168.1.6
192.168.1.6> ls
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
/home/dc> cd Desktop
/bin/sh: 1: cd: can't cd to Desktop
/home/dc/Desktop> ss
clicking
Progress: 100.00 %
done
```

```
Music
Pictures
Public
Templates
Videos
/home/dc>
/bin/sh: 1: cd: can't cd to Desktop
/home/dc/Desktop>
```

https://github.com/dhruvchandel/Reverse_shell

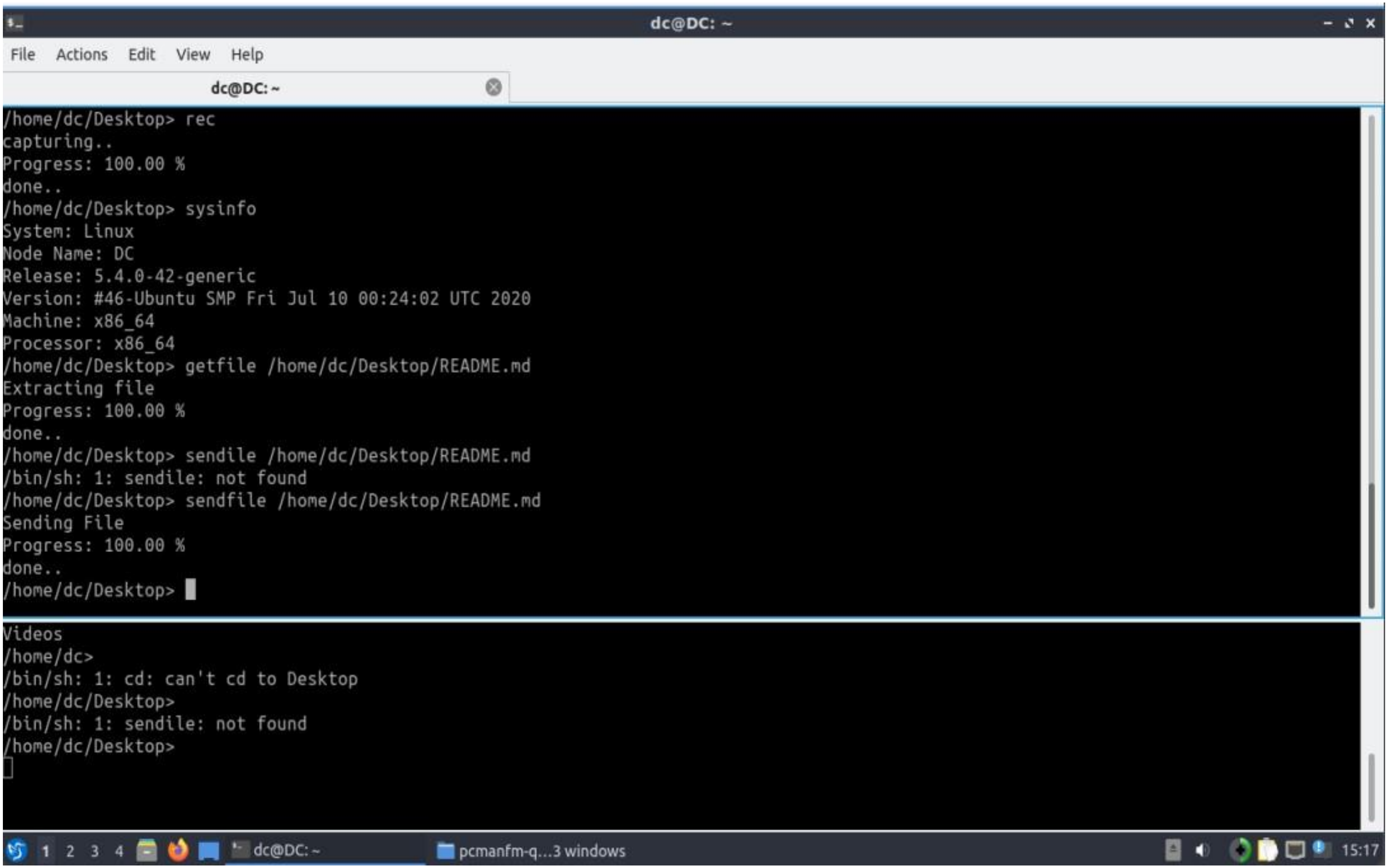


```
1. list - comnand lists all active clients connected to server
2. select - command to select the target client (mandatory arguement - client id of target machine)
3. exit - comnand to exit fron shell

KD> select 0
You are now connected to 192.168.1.6
192.168.1.6> ls
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
/home/dc> cd Desktop
/bin/sh: 1: cd: can't cd to Desktop
/home/dc/Desktop> ss
clicking
Progress: 100.00 %
done
/home/dc/Desktop> ss
```

```
Music
Pictures
Public
Templates
Videos
/home/dc>
/bin/sh: 1: cd: can't cd to Desktop
/home/dc/Desktop>
```

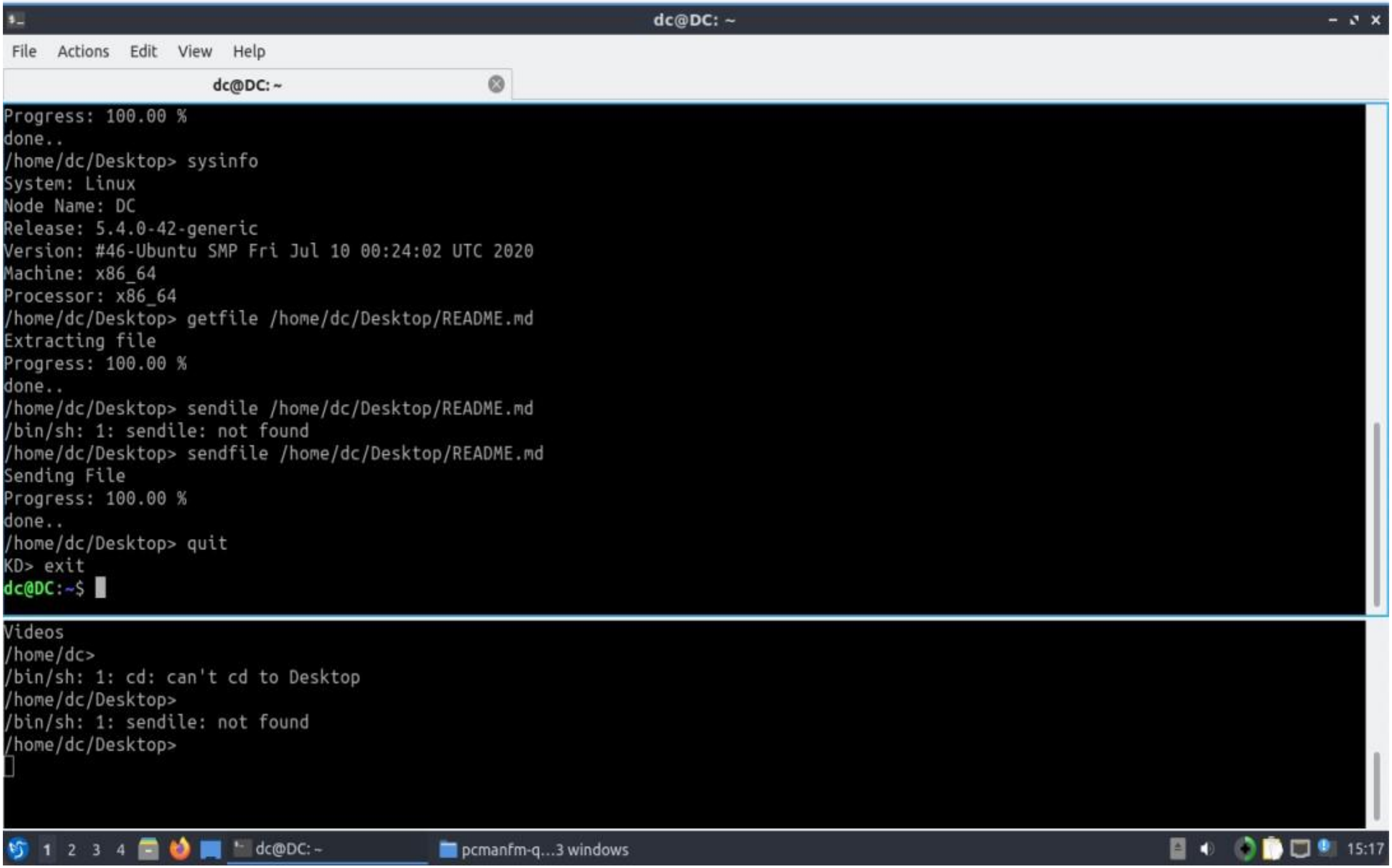
https://github.com/dhruvchandel/Reverse_shell



```
/home/dc/Desktop> rec
capturing..
Progress: 100.00 %
done..
/home/dc/Desktop> sysinfo
System: Linux
Node Name: DC
Release: 5.4.0-42-generic
Version: #46-Ubuntu SMP Fri Jul 10 00:24:02 UTC 2020
Machine: x86_64
Processor: x86_64
/home/dc/Desktop> getfile /home/dc/Desktop/README.md
Extracting file
Progress: 100.00 %
done..
/home/dc/Desktop> sendile /home/dc/Desktop/README.md
/bin/sh: 1: sendile: not found
/home/dc/Desktop> sendfile /home/dc/Desktop/README.md
Sending File
Progress: 100.00 %
done..
/home/dc/Desktop> 
```

```
Videos
/home/dc>
/bin/sh: 1: cd: can't cd to Desktop
/home/dc/Desktop>
/bin/sh: 1: sendile: not found
/home/dc/Desktop>
```

https://github.com/dhruvchandel/Reverse_shell



```
Progress: 100.00 %
done..
/home/dc/Desktop> sysinfo
System: Linux
Node Name: DC
Release: 5.4.0-42-generic
Version: #46-Ubuntu SMP Fri Jul 10 00:24:02 UTC 2020
Machine: x86_64
Processor: x86_64
/home/dc/Desktop> getfile /home/dc/Desktop/README.md
Extracting file
Progress: 100.00 %
done..
/home/dc/Desktop> sendile /home/dc/Desktop/README.md
/bin/sh: 1: sendile: not found
/home/dc/Desktop> sendfile /home/dc/Desktop/README.md
Sending File
Progress: 100.00 %
done..
/home/dc/Desktop> quit
KD> exit
dc@DC:~$
```

```
Videos
/home/dc>
/bin/sh: 1: cd: can't cd to Desktop
/home/dc/Desktop>
/bin/sh: 1: sendile: not found
/home/dc/Desktop>
```

https://github.com/dhruvchandel/Reverse_shell

6. Future work

- Provide a more interactive and easier to use GUI for easy access to the functionalities.
- Provide high speed servers to provide live webcam/screen feed instead of recorded ones.
- Provide keyboard/mouse input from host to client machine.

7. References

- https://en.wikipedia.org/wiki/Reverse_connection#:~:text=In%20a%20normal%20forward%20connection,firewall%20and%20router%20security%20restrictions.