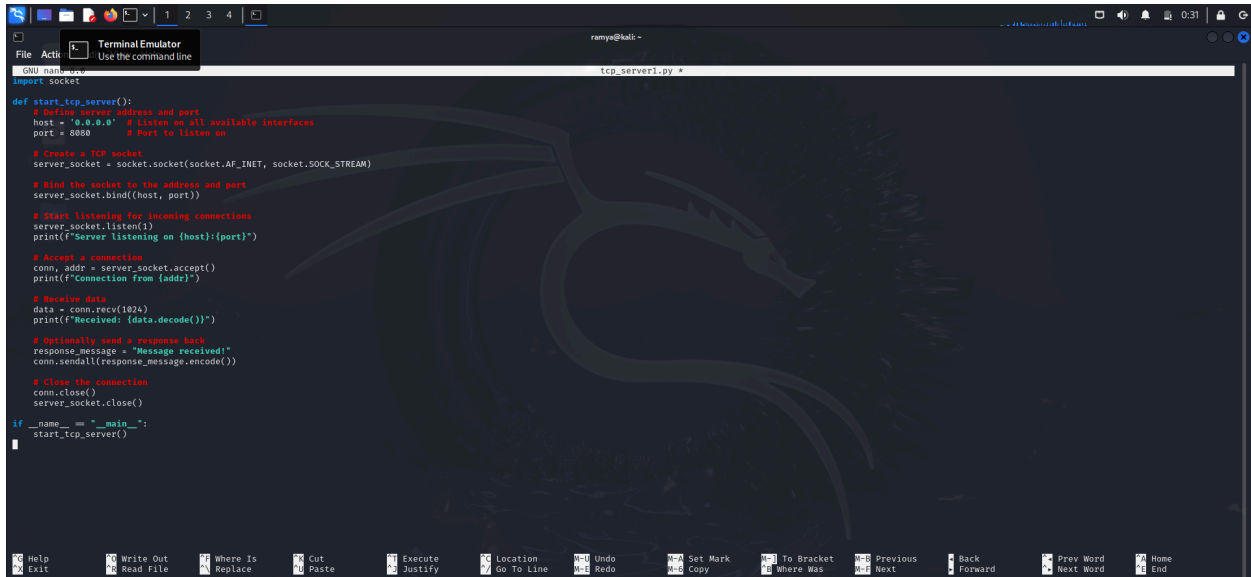


TCP Packet Creation with socket



The screenshot shows a terminal window titled "Terminal Emulator" with a dark background and a dragon logo. The user is editing a file named "tcp_server1.py" using the nano text editor. The code defines a function "start_tcp_server()" that sets up a TCP server on all interfaces at port 8888. It includes comments for each step: defining address and port, creating a TCP socket, binding the socket, listening for connections, accepting a connection, receiving data, and optionally sending a response. The main block calls "start_tcp_server()" if the script is run directly.

```
import socket

def start_tcp_server():
    # Define server address and port
    host = '0.0.0.0' # Listen on all available interfaces
    port = 8888      # Port to listen on

    # Create a TCP socket
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    # Bind the socket to the address and port
    server_socket.bind((host, port))

    # Start listening for incoming connections
    server_socket.listen(1)
    print(f"Server listening on {host}:{port}")

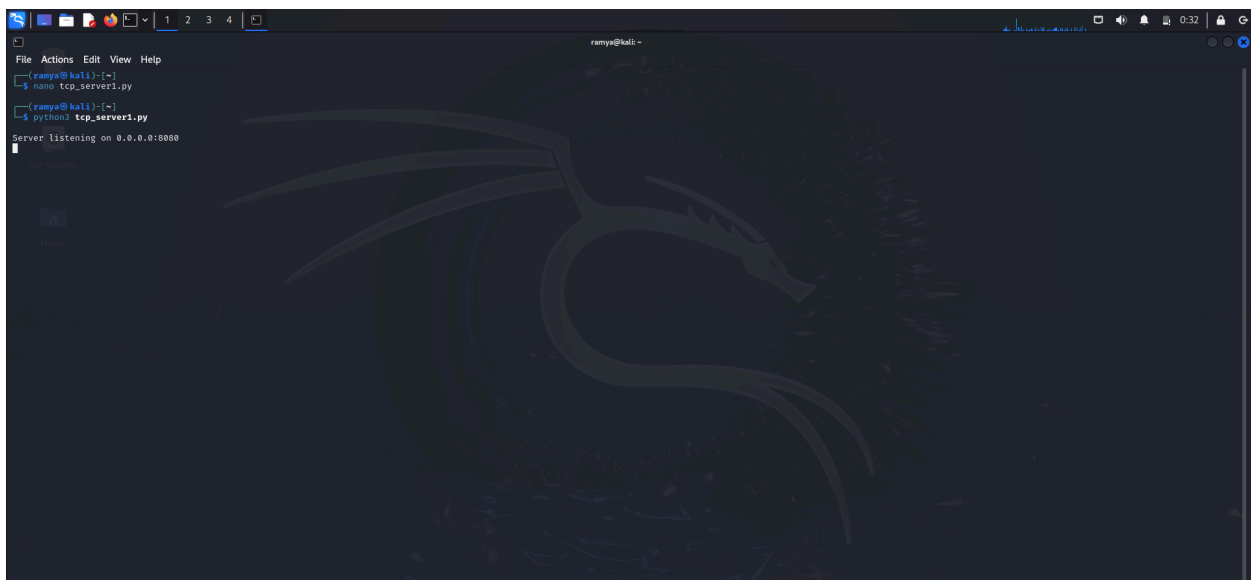
    # Accept a connection
    conn, addr = server_socket.accept()
    print(f"Connection from {addr}")

    # Receive data
    data = conn.recv(1024)
    print(f"Received: {data.decode()}")

    # Optionally send a response back
    response_message = "Message received!"
    conn.sendall(response_message.encode())

    # Close the connection
    conn.close()
    server_socket.close()

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



The screenshot shows the same terminal window after running the script. The user has executed "nano tcp_server1.py" and then "python3 tcp_server1.py". The output shows the server is now listening on 0.0.0.0:8888.

```
(ramya@kali)-[~]
$ nano tcp_server1.py
(ramya@kali)-[~]
$ python3 tcp_server1.py
Server listening on 0.0.0.0:8888
```

```
ramya@kali: ~  
File Actions Edit View Help  
GNU nano 3.0 tcp_client1.py *  
import socket  
  
def send_tcp_packet(ip, port, message):  
    # Create a new TCP socket  
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)  
    try:  
        # Connect to the specified IP and port  
        sock.connect((ip, port))  
        # Send the message  
        sock.sendall(message.encode('utf-8'))  
        print(f'Sent packet to {ip}:{port} with message: {message}')  
        # Optionally, receive a response  
        response = sock.recv(1024) # Receive up to 1024 bytes  
        print(f'Received response: {response.decode('utf-8')}')  
    except Exception as e:  
        print(f'An error occurred: {e}')  
    finally:  
        # Close the connection  
        sock.close()  
  
if __name__ == '__main__':  
    target_ip = '192.168.100.5' # Updated IP address  
    target_port = 8080 # Port to connect to  
    message = 'Hello, TCP World!' # Message to send  
    send_tcp_packet(target_ip, target_port, message) # Call the function
```

```
ramya@kali: ~  
File Actions Edit View Help  
ramya@kali:~$ nano tcp_client1.py  
ramya@kali:~$ python3 tcp_client1.py  
Sent packet to 192.168.100.5:8080 with message: Hello, TCP World!  
Received response: Message received!  
ramya@kali:~$
```