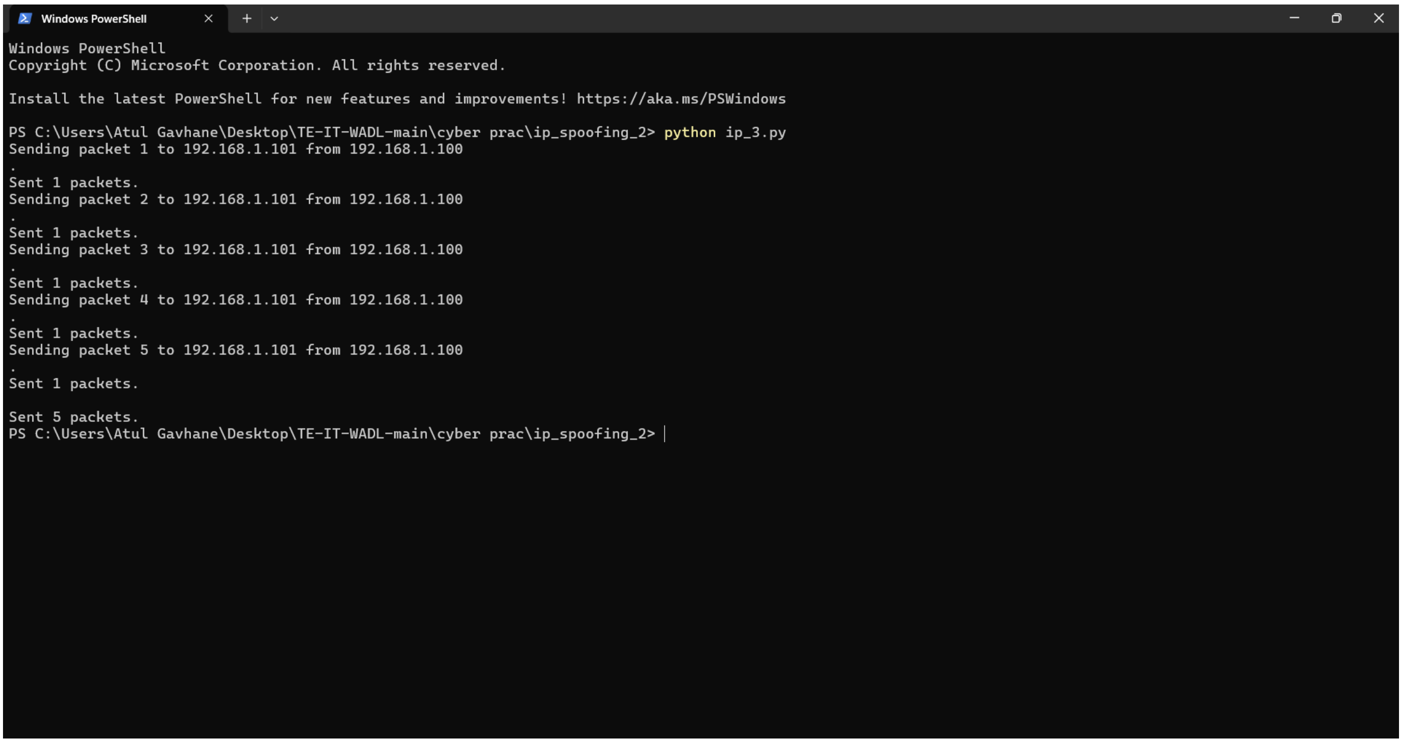

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Atul Gavhane\Desktop\TE-IT-WADL-main\cyber prac\ip_spoofing_2> python ip_3.py
Sending packet 1 to 192.168.1.101 from 192.168.1.100
.
Sent 1 packets.
Sending packet 2 to 192.168.1.101 from 192.168.1.100
.
Sent 1 packets.
Sending packet 3 to 192.168.1.101 from 192.168.1.100
.
Sent 1 packets.
Sending packet 4 to 192.168.1.101 from 192.168.1.100
.
Sent 1 packets.
Sending packet 5 to 192.168.1.101 from 192.168.1.100
.
Sent 1 packets.

Sent 5 packets.
PS C:\Users\Atul Gavhane\Desktop\TE-IT-WADL-main\cyber prac\ip_spoofing_2>
```

Great! Now that the code runs successfully, I'll provide you with a comprehensive explanation of the code, the steps to run it during your external practicals, and how to explain it to your examiners.

Code with Detailed Explanation:

```
from scapy.all import IP, ICMP, send
import time

# Fake source IP (spoofed)
fake_source_ip = "192.168.1.100"  # Replace with any spoofed IP

# Target IP (destination)
target_ip = "192.168.1.101"       # Replace with the actual test target

# Create the spoofed packet
packet = IP(src=fake_source_ip, dst=target_ip) / ICMP()

# Send the packet 5 times and print status
sent_count = 0
for i in range(5):
    print(f"Sending packet {i + 1} to {target_ip} from {fake_source_ip}")
    send(packet)  # Send the packet
    sent_count += 1  # Increment sent packet counter
    time.sleep(1)  # 1-second delay between each send

print(f"\nSent {sent_count} packets.")  # Final confirmation of packet sending
```

Code Explanation:

1. **Imports**:
   - from scapy.all import IP, ICMP, send:
     - IP: This is used to create an IP packet where we can specify the source and destination IP addresses.
     - ICMP: This is used to add an ICMP layer to the packet, which is typically used for "ping" requests.
     - send: This function sends the crafted packet over the network.

2. **Variable Declarations**:
    - fake_source_ip = "192.168.1.100": Here, you specify the **spoofed source IP address**. In real-world scenarios, this could be any IP address you want to fake as the sender.
    - target_ip = "192.168.1.101": The **target IP address** is the address where you want to send the ICMP packets (the destination machine).
3. **Packet Creation**:
    - packet = IP(src=fake_source_ip, dst=target_ip) / ICMP(): This line creates the actual **ICMP packet**. The source address is set to fake_source_ip, and the destination address is target_ip. The ICMP layer is added to this IP packet.
4. **Sending the Packets**:
    - sent_count = 0: Initializes a counter to keep track of how many packets have been sent.
    - The for loop runs 5 times (range(5)), sending 5 packets.
        - Each iteration sends the crafted packet (send(packet)).
        - After sending each packet, it waits for 1 second (time.sleep(1)) before sending the next one to avoid overwhelming the network.
        - Each packet sent increments the sent_count by 1.
    - After all packets are sent, the script prints the total number of packets sent: Sent 5 packets.

Steps to Run the Code During External Practicals:

1. **Pre-requisite Setup**:
    - **Scapy Installation**: Make sure Scapy is installed on your machine. If not, run the following command to install it:
    - pip install scapy
    - **Administrator Privileges**: Since Scapy uses raw packet manipulation, you need to run the script with **administrator privileges** (or root on Linux/macOS).
        - On Windows: Open Command Prompt as Administrator.
        - On Linux/macOS: Open Terminal and use sudo to run the script.
2. **Prepare the Script**:
    - Open a **text editor** (like Notepad or VS Code) and paste the code above into a new file.
    - Save the file as ip_spoofing.py.
3. **Edit the IP Addresses**:
    - Replace the fake_source_ip with any IP address you want to spoof. For example, you can use 192.168.1.100.
    - Replace target_ip with the **actual IP address** of the machine you are sending the packets to (this could be a test machine on the same network).
4. **Running the Script**:
    - Open **Command Prompt** (Windows) or **Terminal** (Linux/macOS) as Administrator.
    - Navigate to the directory where your script is saved. For example:
    - cd C:\Users\Atul Gavhane\Desktop\TE-IT-WADL-main\cyber prac\ip_spoofing_2
    - Run the script with Python:
    - python ip_spoofing.py
5. **Expected Output**: The script will print the following, showing each packet being sent:
6. Sending packet 1 to 192.168.1.101 from 192.168.1.100
7. Sending packet 2 to 192.168.1.101 from 192.168.1.100
8. Sending packet 3 to 192.168.1.101 from 192.168.1.100
9. Sending packet 4 to 192.168.1.101 from 192.168.1.100
10. Sending packet 5 to 192.168.1.101 from 192.168.1.100
11.
12. Sent 5 packets.

How to Explain This Practical to Your Examiners:

1. **Introduction**:

  - **Brief Explanation**: Start by explaining that the script demonstrates **IP Spoofing**, a technique where the source IP address in the packet header is forged to make the packet appear as though it came from a different device.
  - Mention that this is commonly used in **network testing**, **penetration testing**, and also by malicious actors in **Denial of Service (DoS)** attacks.
2. **Step-by-Step Explanation**:
   - **Packet Crafting**: Explain how the script uses the Scapy library to craft an **ICMP packet** (like a ping) with a spoofed source IP (fake_source_ip) and a target destination IP (target_ip).
   - **Looping to Send Multiple Packets**: Explain that the for loop ensures that the script sends multiple packets (5 in this case) to the target machine. This simulates a form of attack like a small-scale DDoS.
   - **Sending Packets**: Describe how the send(packet) function is used to send each crafted packet over the network. Mention that this is an example of **packet injection**.
   - **Sleep Between Sends**: Explain that time.sleep(1) introduces a 1-second delay between each packet being sent, which mimics real traffic flow.
3. **Demonstrating the Result**:
   - **Final Output**: After running the script, explain that the console output shows how many packets were sent. In this case, it will show that 5 packets were sent from the spoofed IP to the target IP.
4. **Use Case and Ethical Considerations**:
   - Explain that while **IP spoofing** has legitimate uses in network diagnostics (e.g., penetration testing), it is **illegal and unethical** to use it without permission on networks you don’t own or have explicit authorization to test.
5. **Monitor and Verify**:
   - If possible, show the **target machine's response** using network monitoring tools (e.g., **Wireshark** or **tcpdump**) to capture the incoming packets and verify that the packets are indeed being sent to the target IP.

Key Points to Mention During the Practical Explanation:

- **What is IP Spoofing?**
  - Explain that IP spoofing is the creation of Internet Protocol (IP) packets with a fake source address, often used to hide the identity of the sender or impersonate another system.
- **Legitimate Use**:
  - Mention that in ethical hacking and network testing, spoofing is used to test how networks respond to malicious traffic, ensuring systems are secure against attacks.
- **Scapy and Packet Crafting**:
  - Discuss how Scapy makes it easy to craft, manipulate, and send network packets. This makes it a popular tool for penetration testers and network engineers.
- **Precautions and Ethics**:
  - Stress the importance of using IP spoofing **only in controlled environments** where testing is allowed, as it can cause network disruptions and illegal activity if misused.

With this, you should be well-prepared to explain the code during your external practical exam. Let me know if you need further assistance!