IP Spoofing Prevention

ICMP.py | udp.py | t.py | tcppacketcreation.py | ipspoofingprevention.py

home > kali > ipspoofingprevention.py > ...

```python
from scapy.all import sniff, IP

def packet_callback(packet):
    if IP in packet:
        source_ip = packet[IP].src
        dest_ip = packet[IP].dst

        # Check if the source IP is from a range it shouldn't be
        # Replace "10.0.0.0/8" with appropriate network range
        if source_ip.startswith("10.") and dest_ip != "192.168.56.1":
            print(f"Possible IP spoofing: Source IP {source_ip} not valid for this network")

sniff(prn=packet_callback, store=0)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Possible IP spoofing: Source IP 10.0.2.15 not valid for this network
Possible IP spoofing: Source IP 10.0.2.15 not valid for this network
```

These can be implemented in practice using router configurations or firewall rules, for example:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
┌──(kali㉿kali)-[~]
└─$ # Egress Filtering: Deny any outgoing packet that has a source IP address not in your network range.
sudo iptables -A OUTPUT ! -s 10.0.2.0/24 -j DROP
[sudo] password for kali:

┌──(kali㉿kali)-[~]
└─$ # Ingress Filtering: Deny any incoming packet that has a source IP address from your internal range.
sudo iptables -A INPUT -s 10.0.2.0/24 -j DROP


┌──(kali㉿kali)-[~]
└─$
```

Ingress Filtering: This is applied by network routers or firewalls to check incoming traffic and block packets with source addresses that shouldn't come from outside the network.

Egress Filtering: This is used to inspect outgoing traffic from the network and prevent packets with forged source IP addresses from leaving the network. This ensures that no one inside your network is spoofing an IP address.