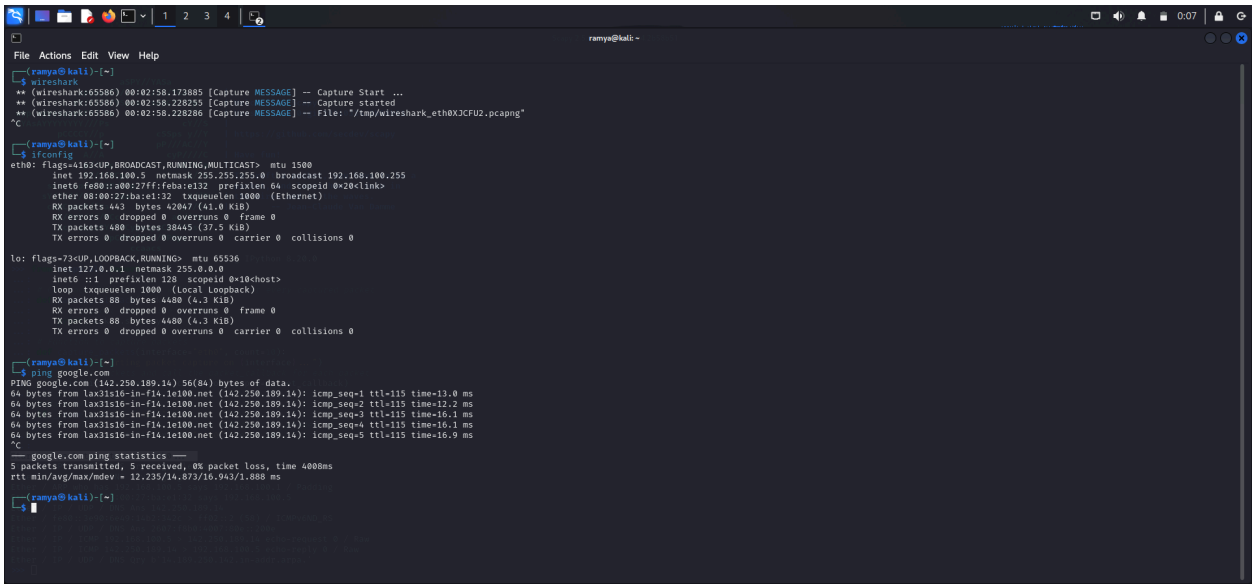
BakkaRamyasree

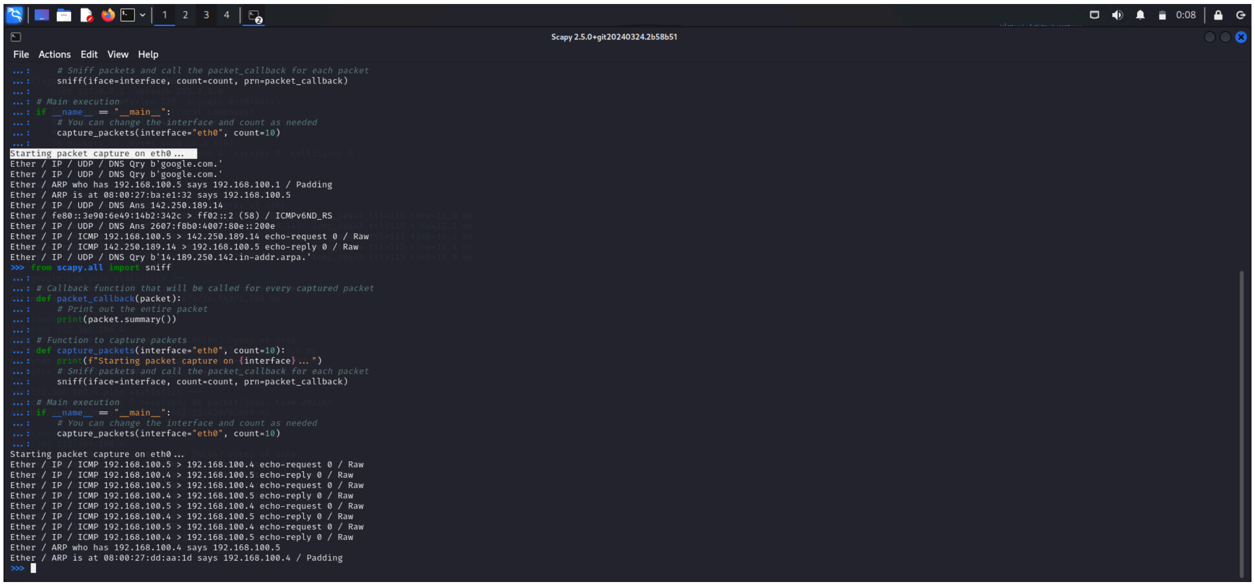
Packet Capture using scapy

Steps:

- Open scapy in kali linux
- Start writing the code give
- And run the code it starts to capture the packets
- Open one more terminal and start to ping like i used google.com and to ping my ubuntu machine ip




1 2 3 4
0:08
ramya@kali: ~
File Actions Edit View Help
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 88 bytes 4480 (4.3 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 88 bytes 4480 (4.3 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
┌──(ramya㉿kali)-[~]
└─$ ping google.com
PING google.com (142.250.189.14) 56(84) bytes of data.
64 bytes from lax31s16-in-f14.1e100.net (142.250.189.14): icmp_seq=1 ttl=115 time=13.0 ms
64 bytes from lax31s16-in-f14.1e100.net (142.250.189.14): icmp_seq=2 ttl=115 time=12.2 ms
64 bytes from lax31s16-in-f14.1e100.net (142.250.189.14): icmp_seq=3 ttl=115 time=16.1 ms
64 bytes from lax31s16-in-f14.1e100.net (142.250.189.14): icmp_seq=4 ttl=115 time=16.1 ms
64 bytes from lax31s16-in-f14.1e100.net (142.250.189.14): icmp_seq=5 ttl=115 time=16.9 ms
^C
--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 12.235/14.873/16.943/1.888 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=15.4 ms
64 bytes from 192.168.100.4: icmp_seq=2 ttl=64 time=0.757 ms
64 bytes from 192.168.100.4: icmp_seq=3 ttl=64 time=0.809 ms
^C
--- 192.168.100.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2011ms
rtt min/avg/max/mdev = 0.757/5.662/15.420/6.899 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=0.639 ms
64 bytes from 192.168.100.4: icmp_seq=2 ttl=64 time=0.691 ms
64 bytes from 192.168.100.4: icmp_seq=3 ttl=64 time=1.37 ms
64 bytes from 192.168.100.4: icmp_seq=4 ttl=64 time=0.788 ms
^C
--- 192.168.100.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3042ms
rtt min/avg/max/mdev = 0.639/0.871/1.368/0.291 ms
┌──(ramya㉿kali)-[~]
└─$


1 2 3 4
0:08
Scapy 2.5.0+git20240324.2b58b51
File Actions Edit View Help
...:   # Sniff packets and call the packet_callback for each packet
...:   sniff(iface=interface, count=count, prn=packet_callback)
...:
...: # Main execution
...: if __name__ == "__main__":
...:   # You can change the interface and count as needed
...:   capture_packets(interface="eth0", count=10)
...:
Starting packet capture on eth0 ...
Ether / IP / UDP / DNS Qry b'google.com.'
Ether / IP / UDP / DNS Qry b'google.com.'
Ether / ARP who has 192.168.100.5 says 192.168.100.1 / Padding
Ether / ARP is at 08:00:27:ba:e1:32 says 192.168.100.5
Ether / IP / UDP / DNS Ans 142.250.189.14
Ether / fe80::3e90:6e49:14b2:342c > ff02::2 (58) / ICMPv6ND_RS
Ether / IP / UDP / DNS Ans 2607:f8b0:4007:80e::200e
Ether / IP / ICMP 192.168.100.5 > 142.250.189.14 echo-request 0 / Raw
Ether / IP / ICMP 142.250.189.14 > 192.168.100.5 echo-reply 0 / Raw
Ether / IP / UDP / DNS Qry b'14.189.250.142.in-addr.arpa.'
>>> from scapy.all import sniff
...:
...: # Callback function that will be called for every captured packet
...: def packet_callback(packet):
...:   # Print out the entire packet
...:   print(packet.summary())
...:
...: # Function to capture packets
...: def capture_packets(interface="eth0", count=10):
...:   print(f"Starting packet capture on {interface} ... ")
...:   # Sniff packets and call the packet_callback for each packet
...:   sniff(iface=interface, count=count, prn=packet_callback)
...:
...: # Main execution
...: if __name__ == "__main__":
...:   # You can change the interface and count as needed
...:   capture_packets(interface="eth0", count=10)
...:
Starting packet capture on eth0 ...
Ether / IP / ICMP 192.168.100.5 > 192.168.100.4 echo-request 0 / Raw
Ether / IP / ICMP 192.168.100.4 > 192.168.100.5 echo-reply 0 / Raw
Ether / IP / ICMP 192.168.100.5 > 192.168.100.4 echo-request 0 / Raw
Ether / IP / ICMP 192.168.100.4 > 192.168.100.5 echo-reply 0 / Raw
Ether / IP / ICMP 192.168.100.5 > 192.168.100.4 echo-request 0 / Raw
Ether / IP / ICMP 192.168.100.4 > 192.168.100.5 echo-reply 0 / Raw
Ether / IP / ICMP 192.168.100.5 > 192.168.100.4 echo-request 0 / Raw
Ether / IP / ICMP 192.168.100.4 > 192.168.100.5 echo-reply 0 / Raw
Ether / ARP who has 192.168.100.4 says 192.168.100.5
Ether / ARP is at 08:00:27:dd:aa:1d says 192.168.100.4 / Padding
>>>