

## Arpspoof Warning + Protection

### Part 1: Warning System

We have decided to implement three detection methods:

- Counting the number of ARP reply packets and ARP request packets, other than requests from other devices and replies from our device.
- Validating MAC address – we checked that the MAC address in the reply packet is actually belongs to the IP in that packet.
- We checked the ARP table for any duplicates.

There is no false positive.

### Part 2: Protection (Added challenge)

First, we tried to drop all ARP replies while sniffing them in a lower level before they were dropped and then we tried to add manually packets that seemed correct. However, this implementation didn't work as expected – the dropping wasn't working properly. Therefore, we have decided to make the ARP table permanent. The protection checks the ARP table, and if there is a row that is not permanent, update it to permanent status. For each new ARP reply packet, we verify that it's correct, and only then add it to the table. In order to verify a packet, we use the same methods of detection, with a slight change in the implementation of the duplicates function.

Here we can see that once the ping runs, the ARP table shows the address in permanent status:

```
(kali㉿kali)-[~]
└─$ ping 10.0.0.23 -c3
PING 10.0.0.23 (10.0.0.23) 56(84) bytes of data:
64 bytes from 10.0.0.23: icmp_seq=1 ttl=64 time=1.01 ms
64 bytes from 10.0.0.23: icmp_seq=2 ttl=64 time=0.625 ms
64 bytes from 10.0.0.23: icmp_seq=3 ttl=64 time=0.685 ms

— 10.0.0.23 ping statistics —
3 packets transmitted, 3 received, 0% packet loss, time 2032ms
rtt min/avg/max/mdev = 0.625/0.772/1.006/0.167 ms

(kali㉿kali)-[~]
└─$ ip n
10.0.0.23 dev eth0 lladdr 00:0c:29:28:72:ea PERMANENT
10.0.0.138 dev eth0 lladdr 00:b8:c2:62:bb:aa PERMANENT
fe80::2b8:c2ff:fe62:bbaa dev eth0 lladdr 00:b8:c2:62:bb:aa router STALE
```