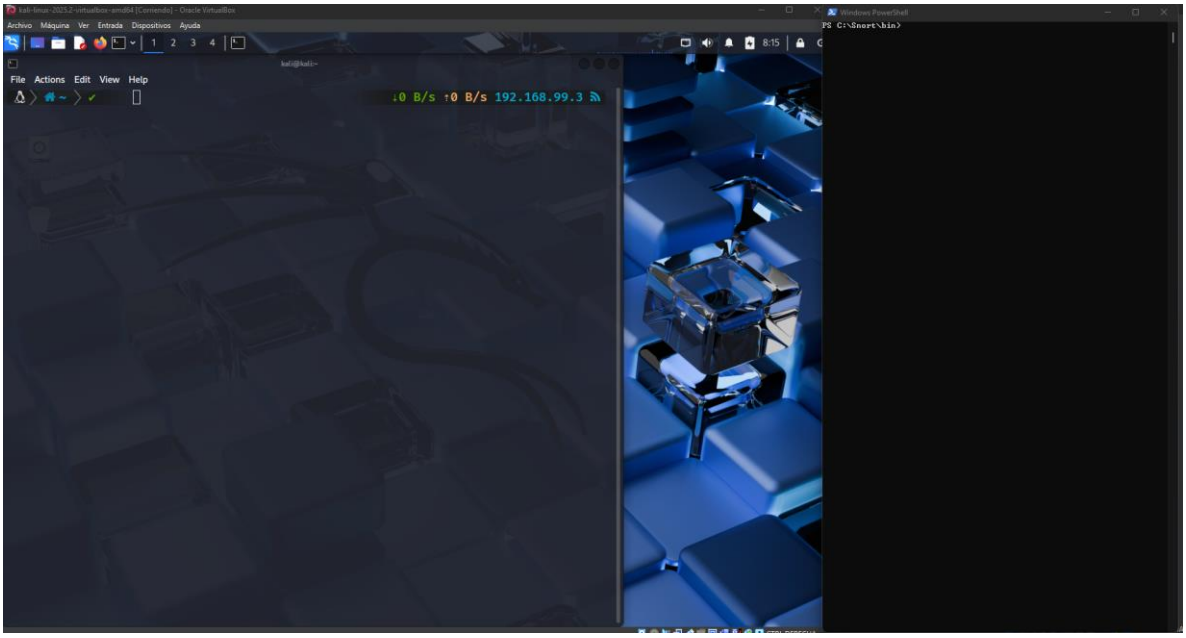
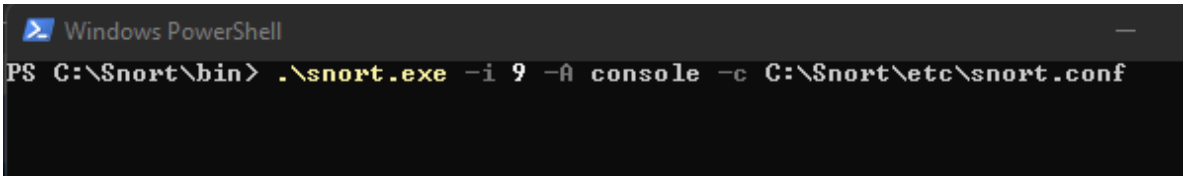
RECONOCIMIENTO DE HOST A NIVEL ORGANIZACIONAL

1. Escenario a utilizar

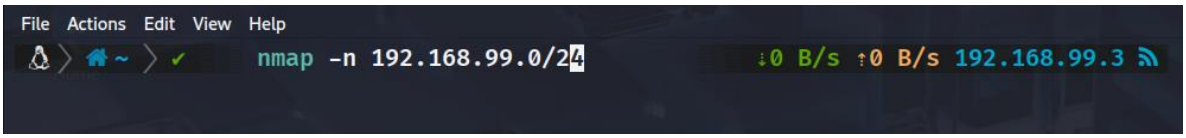


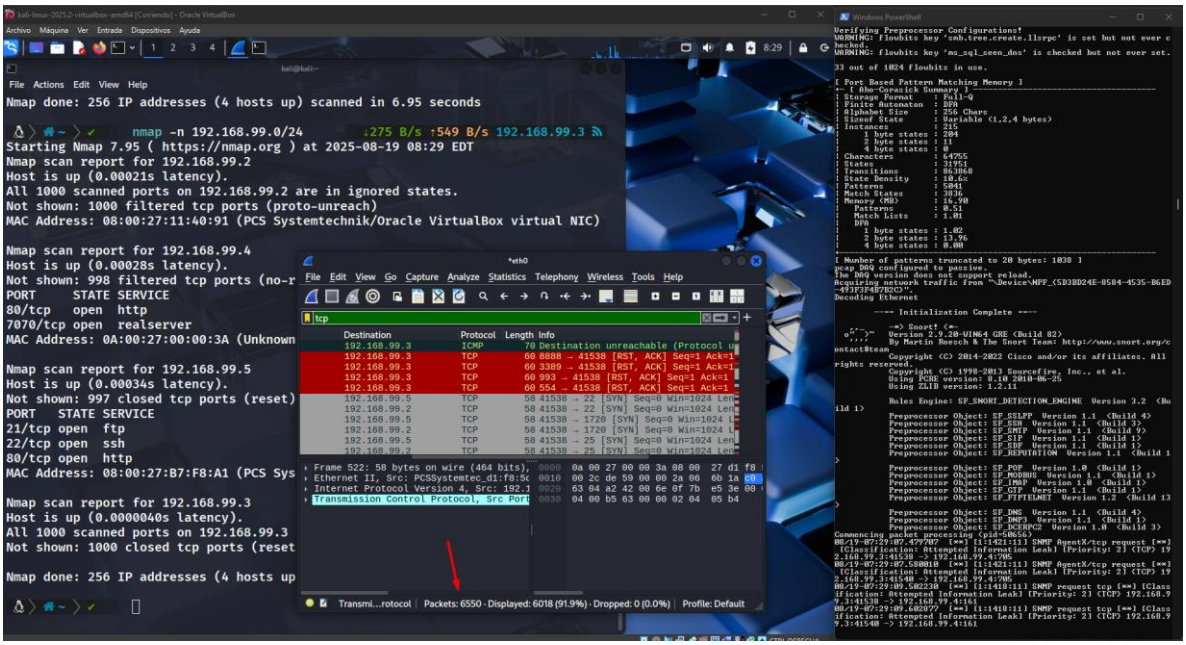
2. Utilizar snort para que empiece a monitorizar todo lo que se lleve a cabo en la infraestructura de red



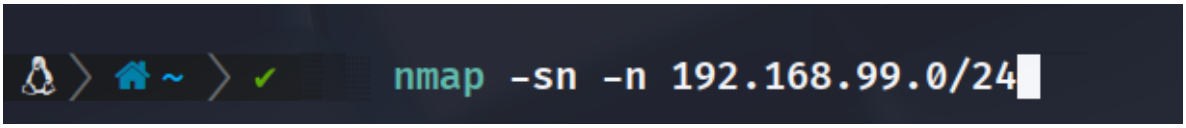
Lanzar un nmap básico

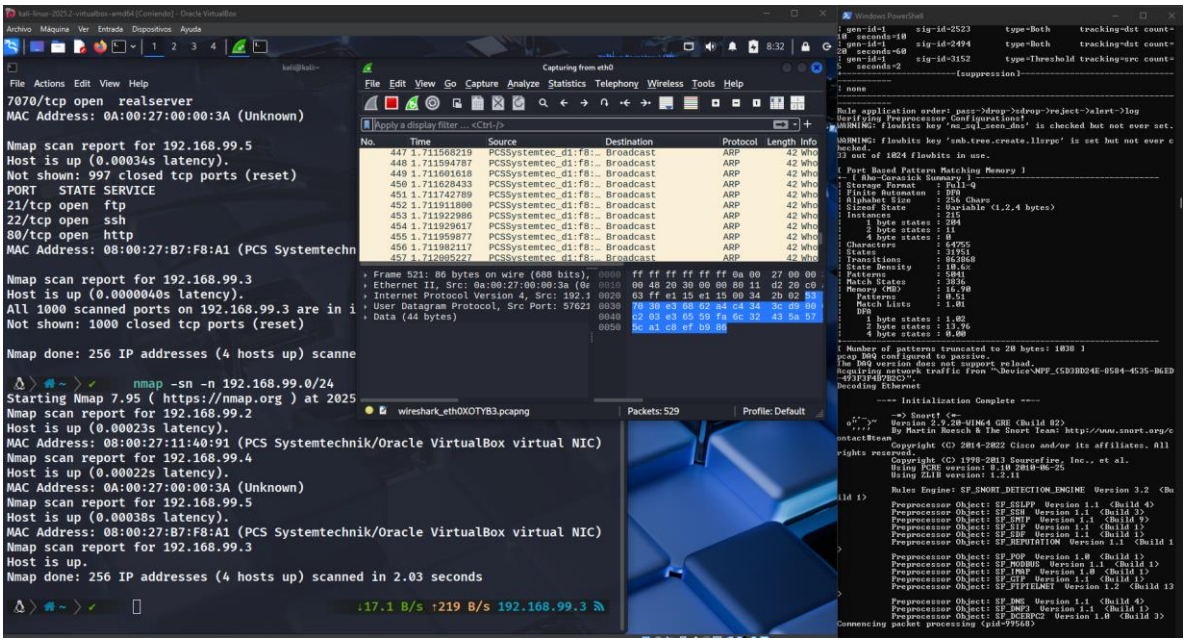
nmap -n 192.168.99.0/24



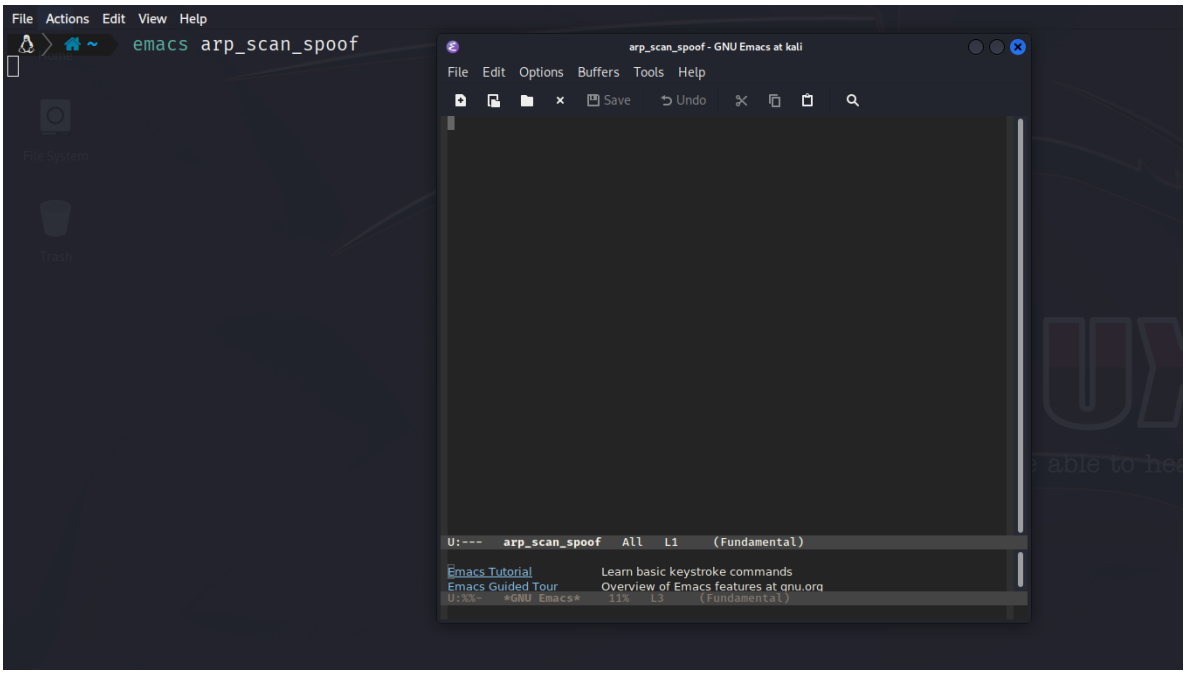


Ir a la literatura de nmap y buscar el sn que no hace escaneo de puertos





emacs arp_scan_spoof



M-x

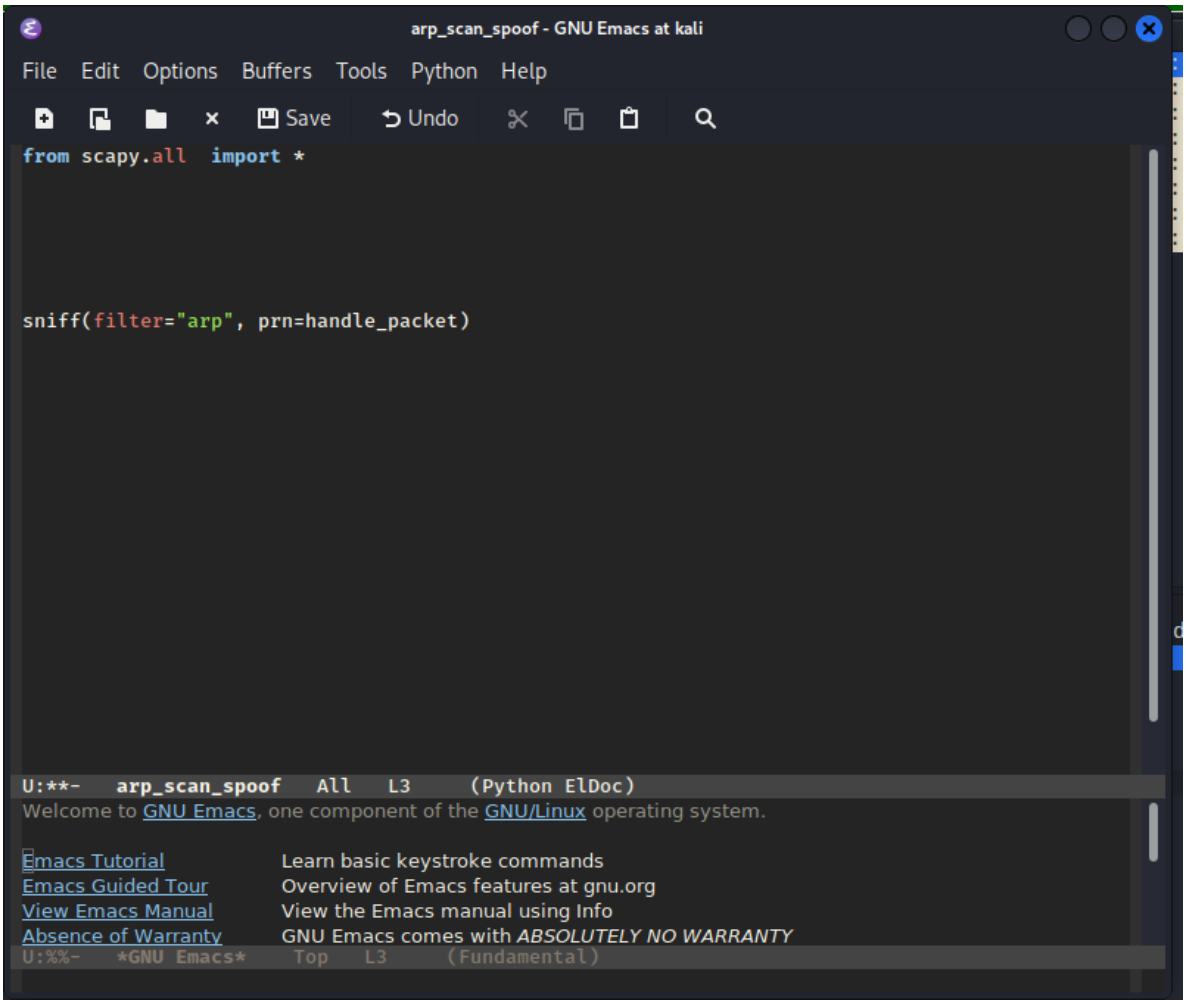
alt

Python-mode

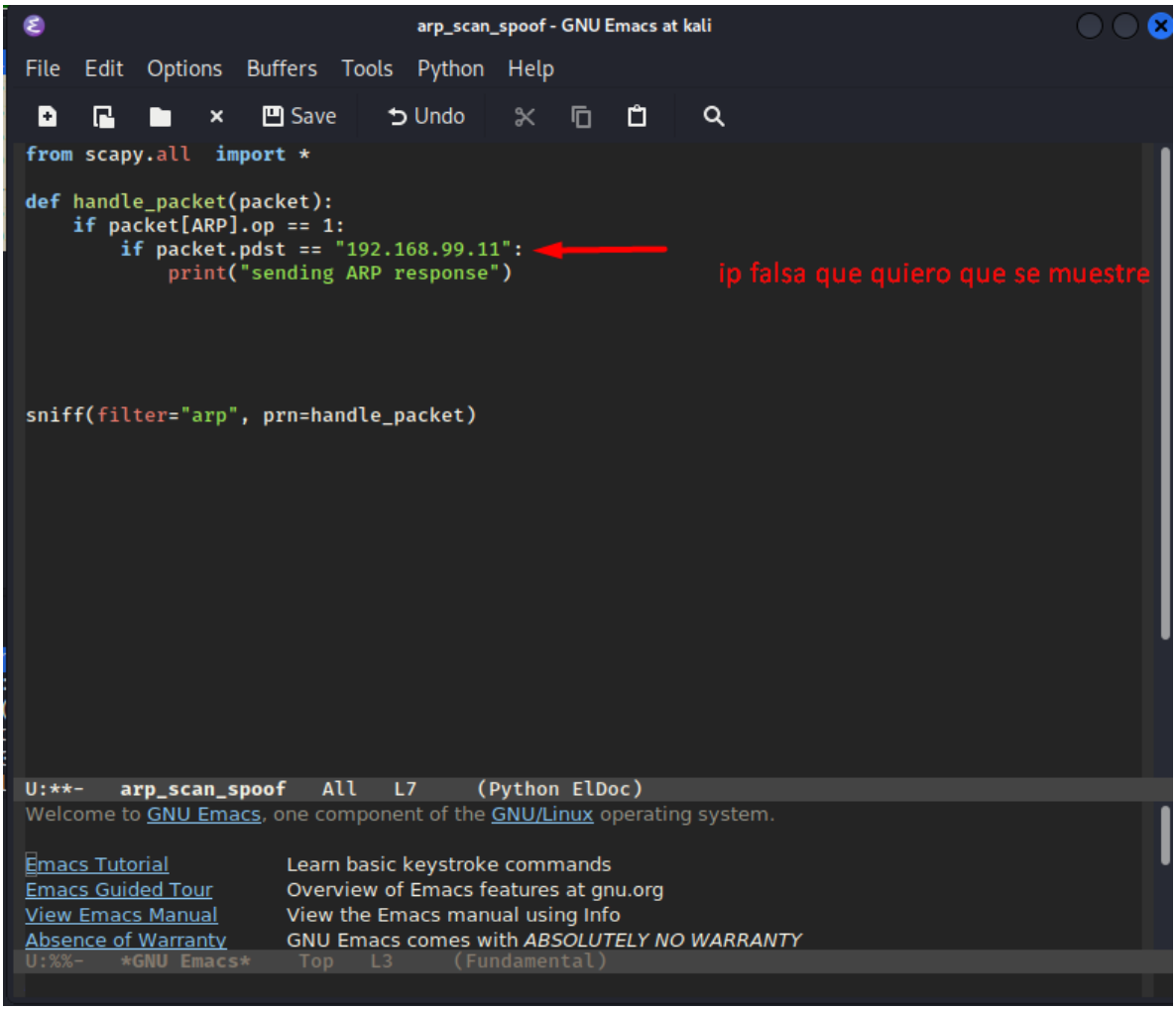
Vamos que scapy importe todo

```
from scapy.all  import *
```

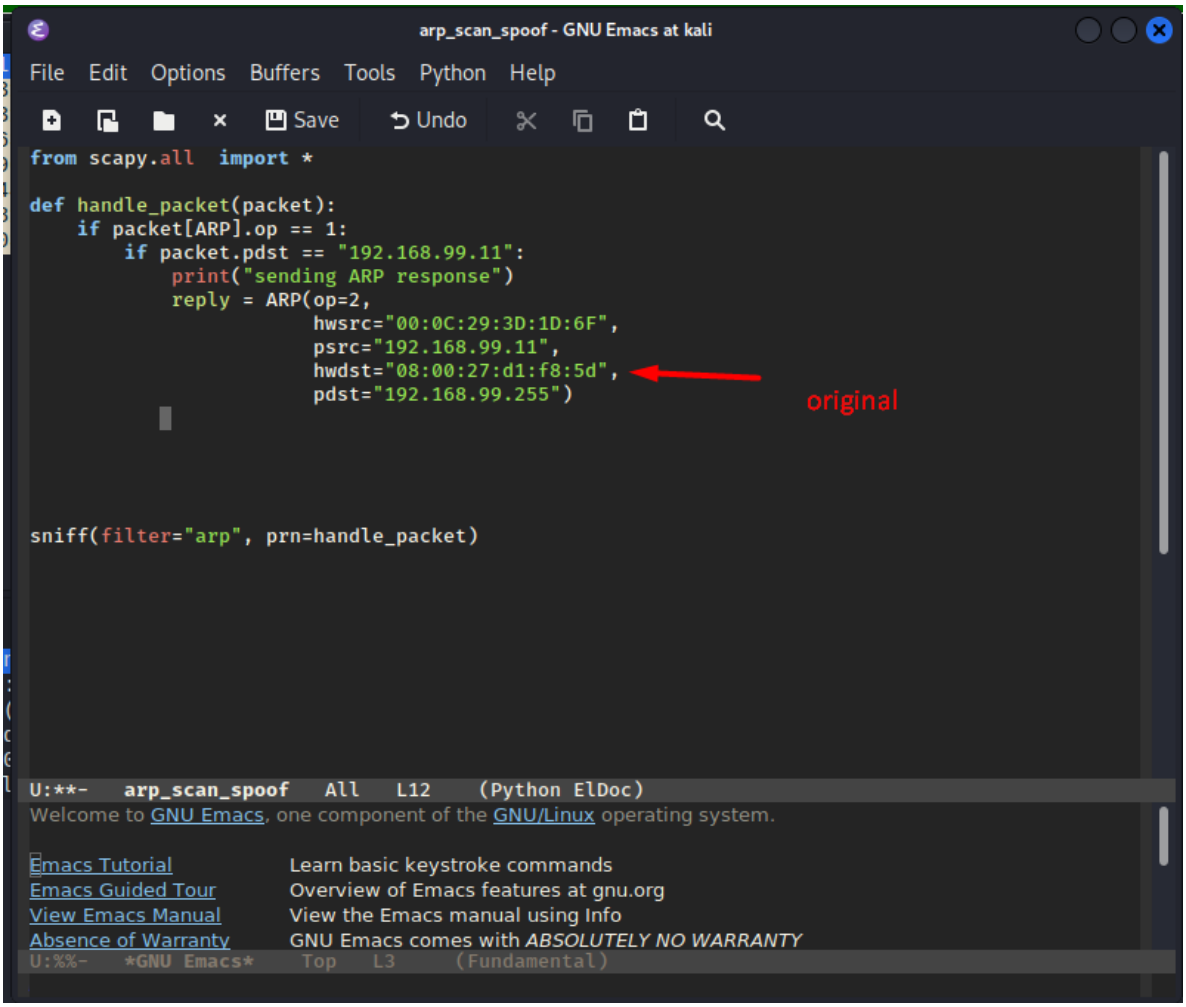
Se póne a scapy a hacer sniffing por el filtro ARP y si es asi que llame a una función llamada handle_packet



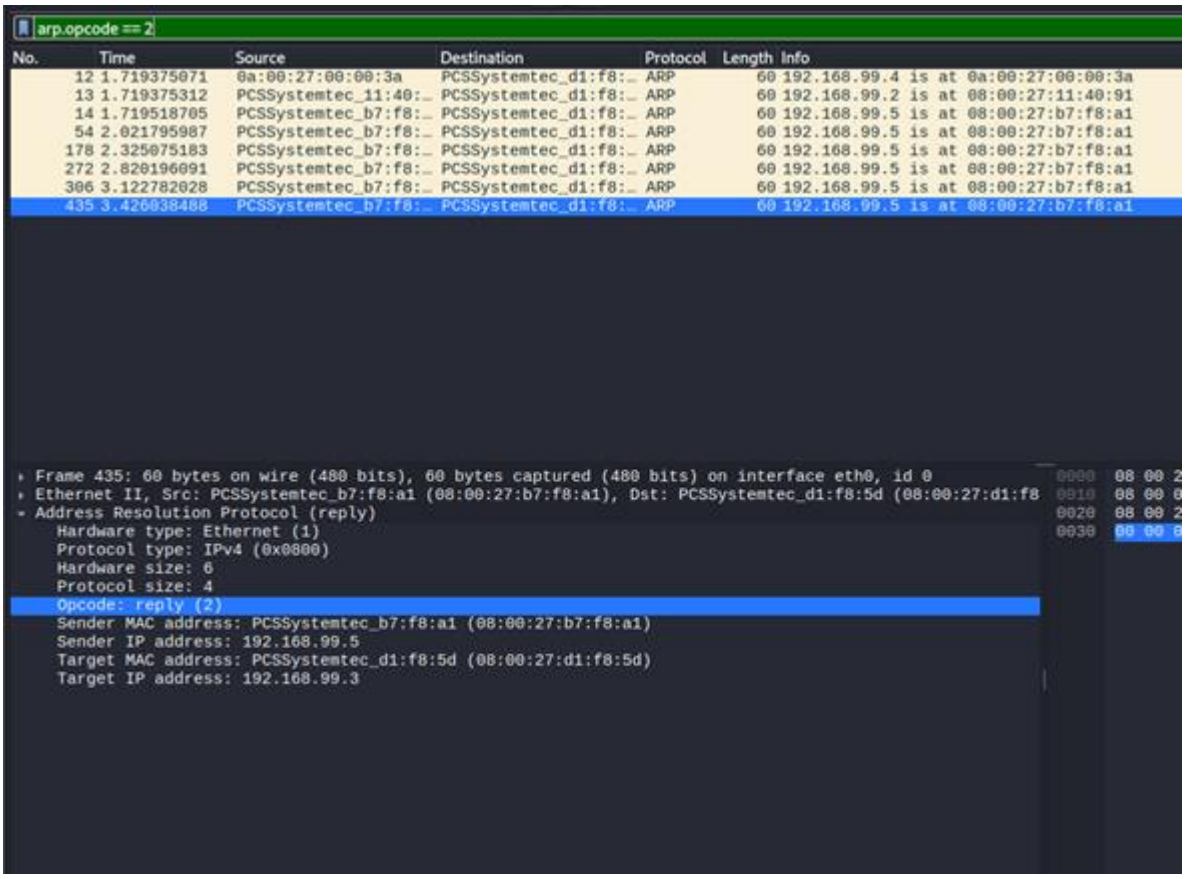
Ahora se deberá configurar para que crea que es otra función

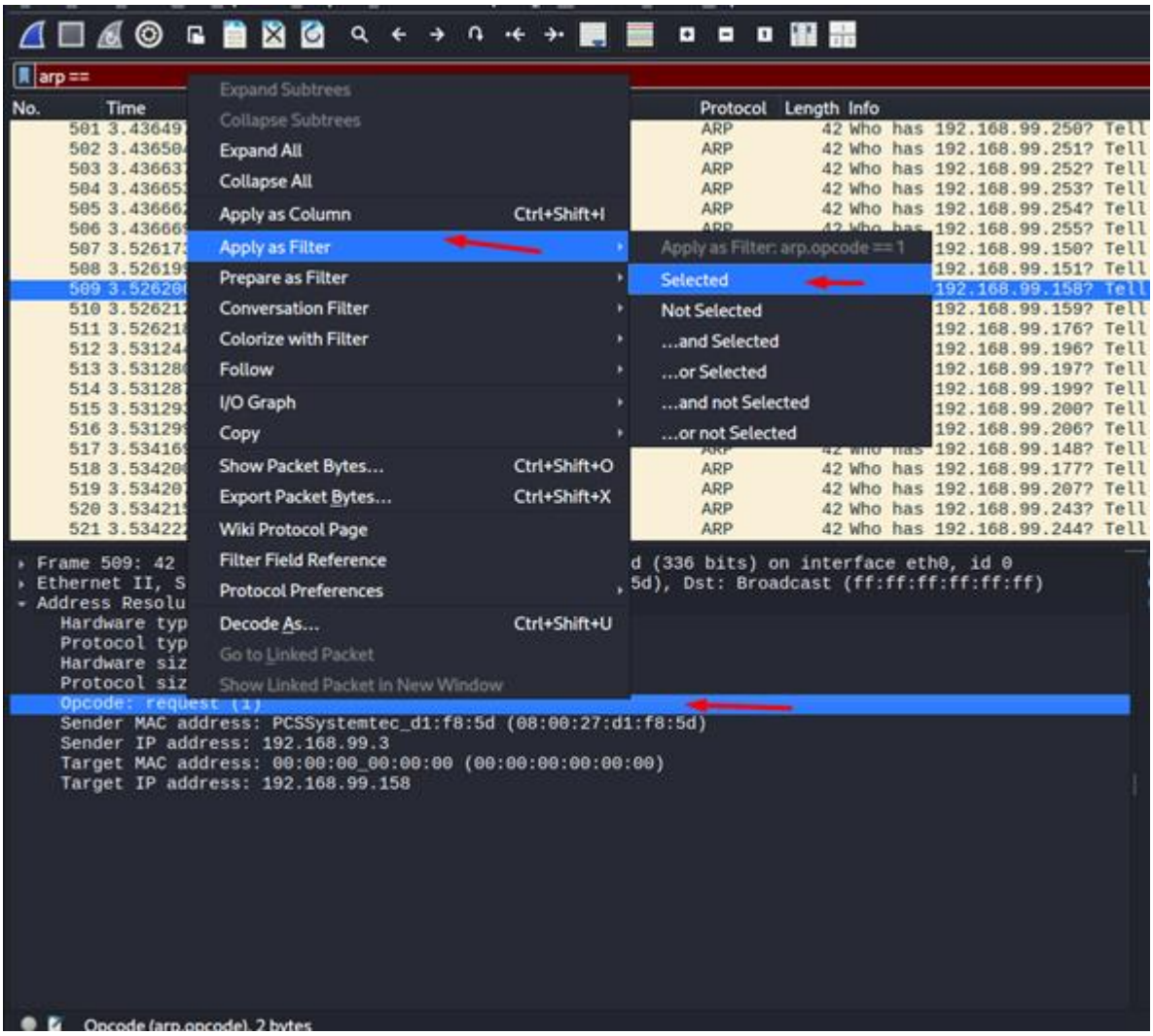


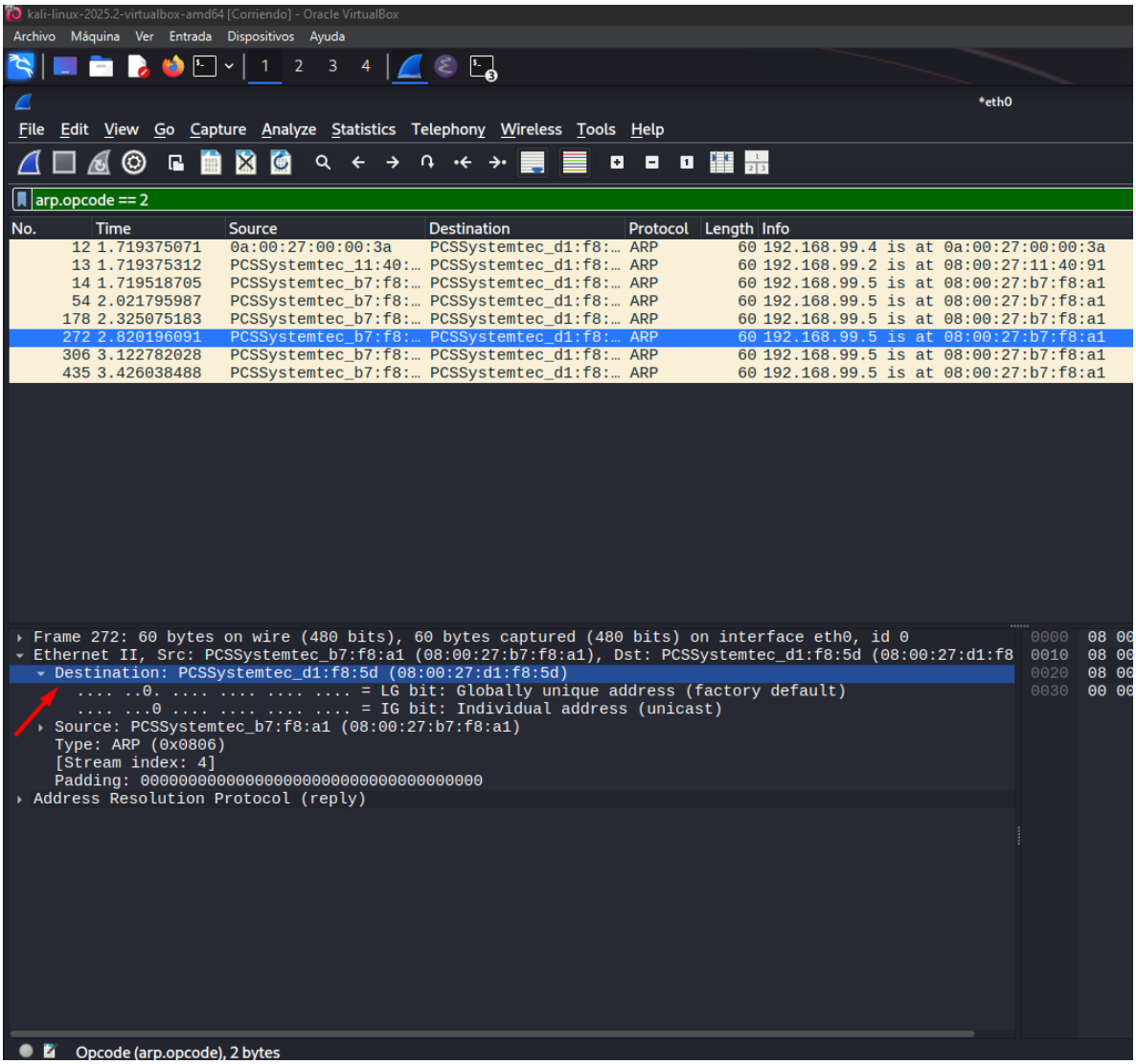
Ahora formar el paquete ARP porque ya será la respuesta vamos a ir a wire



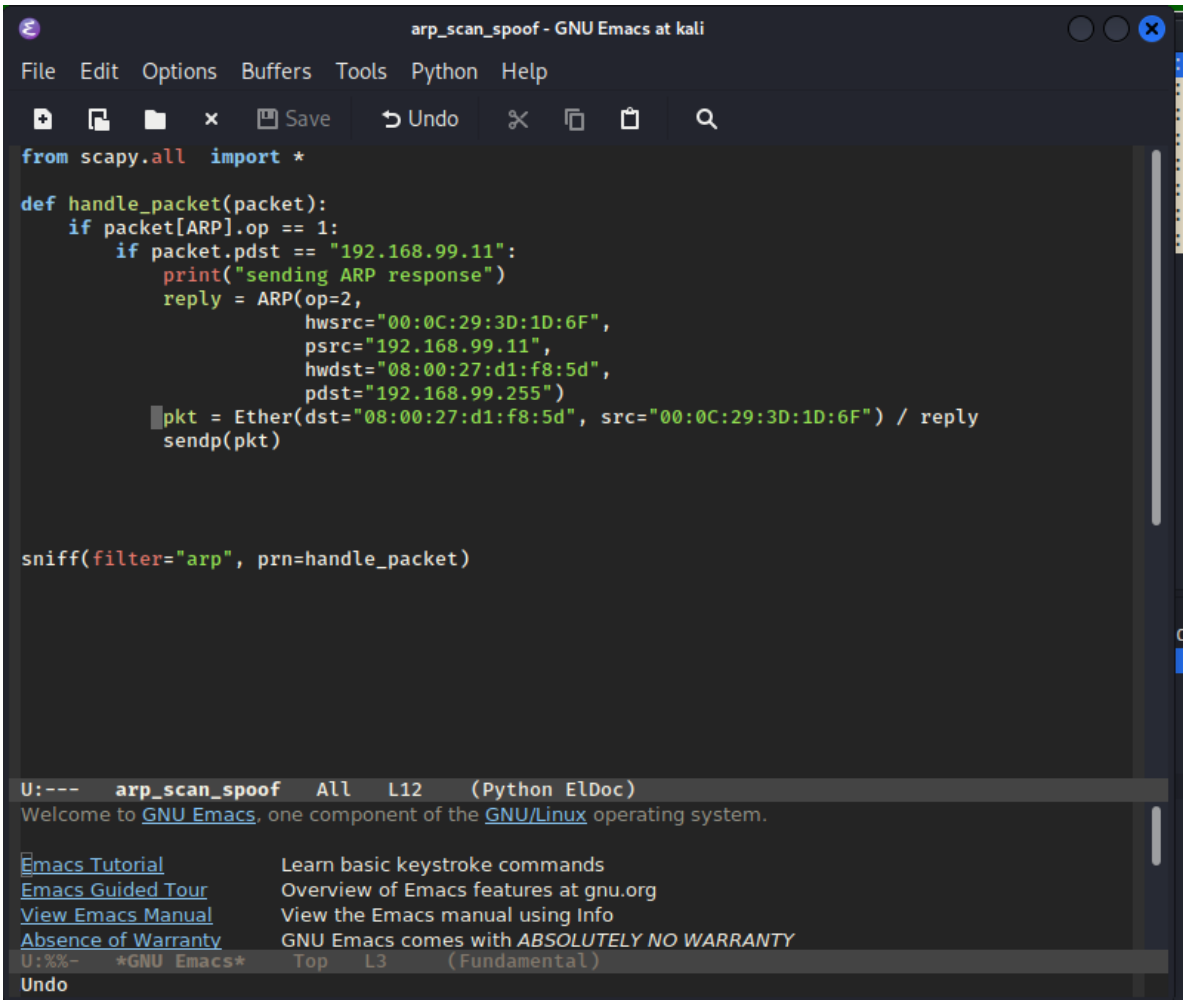
Y copiamos el destino que le responde para eso


arp.opcode == 2
No. Time Source Destination Protocol Length Info
12 1.719375071 0a:00:27:00:00:3a PCSSystemtec_d1:f8:… ARP 60 192.168.99.4 is at 0a:00:27:00:00:3a
13 1.719375312 PCSSystemtec_11:40:… PCSSystemtec_d1:f8:… ARP 60 192.168.99.2 is at 08:00:27:11:40:91
14 1.719518705 PCSSystemtec_b7:f8:… PCSSystemtec_d1:f8:… ARP 60 192.168.99.5 is at 08:00:27:b7:f8:a1
54 2.021795987 PCSSystemtec_b7:f8:… PCSSystemtec_d1:f8:… ARP 60 192.168.99.5 is at 08:00:27:b7:f8:a1
178 2.325075183 PCSSystemtec_b7:f8:… PCSSystemtec_d1:f8:… ARP 60 192.168.99.5 is at 08:00:27:b7:f8:a1
272 2.820196091 PCSSystemtec_b7:f8:… PCSSystemtec_d1:f8:… ARP 60 192.168.99.5 is at 08:00:27:b7:f8:a1
306 3.122782028 PCSSystemtec_b7:f8:… PCSSystemtec_d1:f8:… ARP 60 192.168.99.5 is at 08:00:27:b7:f8:a1
435 3.426038488 PCSSystemtec_b7:f8:… PCSSystemtec_d1:f8:… ARP 60 192.168.99.5 is at 08:00:27:b7:f8:a1
Frame 435: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, id 0
Ethernet II, Src: PCSSystemtec_b7:f8:a1 (08:00:27:b7:f8:a1), Dst: PCSSystemtec_d1:f8:5d (08:00:27:d1:f8
Address Resolution Protocol (reply)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender MAC address: PCSSystemtec_b7:f8:a1 (08:00:27:b7:f8:a1)
Sender IP address: 192.168.99.5
Target MAC address: PCSSystemtec_d1:f8:5d (08:00:27:d1:f8:5d)
Target IP address: 192.168.99.3
0000 08 00 2
0010 08 00 0
0020 08 00 2
0030 00 00 0


arp ==
No.
Time
Protocol
Length
Info
Expand Subtrees
Collapse Subtrees
Expand All
Collapse All
Apply as Column
Ctrl+Shift+I
Apply as Filter
Prepare as Filter
Conversation Filter
Colorize with Filter
Follow
I/O Graph
Copy
Show Packet Bytes...
Ctrl+Shift+O
Export Packet Bytes...
Ctrl+Shift+X
Wiki Protocol Page
Filter Field Reference
Protocol Preferences
Decode As...
Ctrl+Shift+U
Go to Linked Packet
Show Linked Packet in New Window
Apply as Filter: arp.opcode == 1
Selected
Not Selected
...and Selected
...or Selected
...and not Selected
...or not Selected
Frame 509: 42
Ethernet II, S
Address Resolu
Hardware typ
Protocol typ
Hardware siz
Protocol siz
Opcode: request (1)
Sender MAC address: PCSSystemtec_d1:f8:5d (08:00:27:d1:f8:5d)
Sender IP address: 192.168.99.3
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
Target IP address: 192.168.99.158
Opcode (arp.opcode), 2 bytes



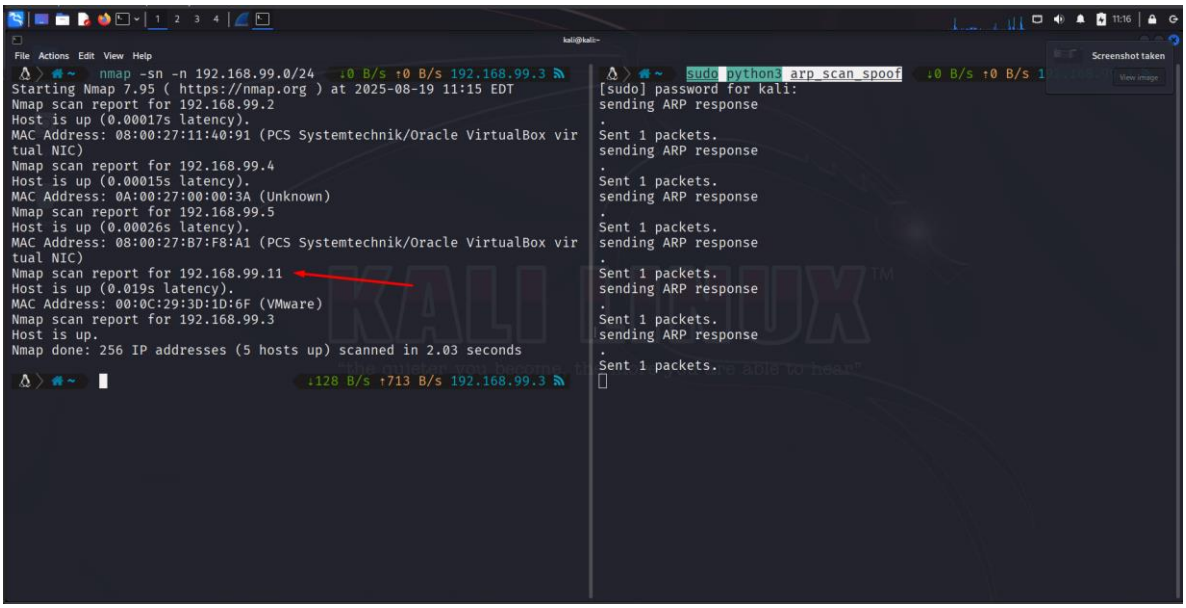
Por último crear la capa ethernet con el destino mac original y el source es el inventado



Se guarda se pone en terminal y a probar

sudo python3 arp_scan_spoof

nmap -sn -n 192.168.99.0/24



ESCANEO AVANZADO DE PUERTOS

nmap -n -PS 192.168.99.0/24