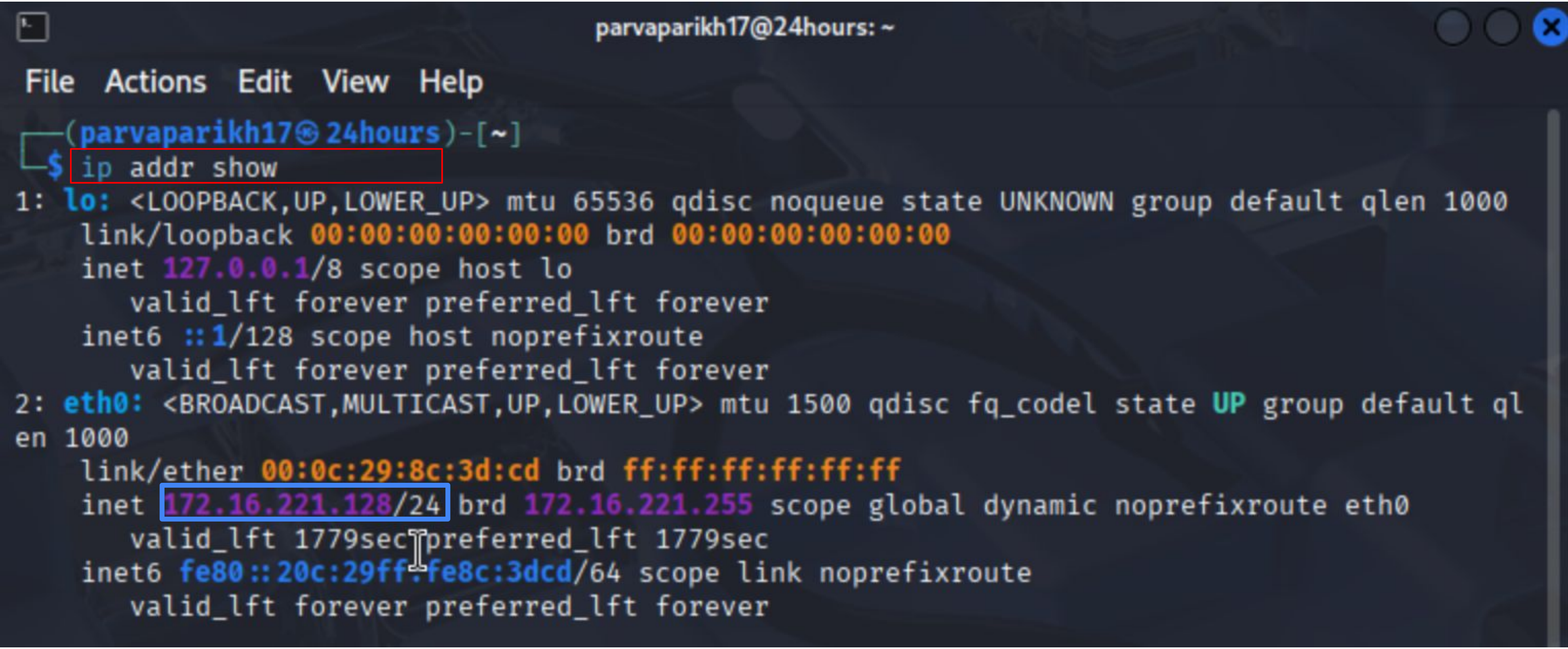
Integrated Network Reconnaissance and Traffic Analysis Toolkit

A multi-phase cybersecurity project combining automated scanning, host enumeration, and packet-level monitoring using Metasploit, Nmap, Bash scripting, and Wireshark.

Part 1: Environment Preparation and Active Network Scanning

This phase covers IP address validation, initializing the Metasploit environment, launching a full TCP SYN scan, and enumerating discovered hosts and services.

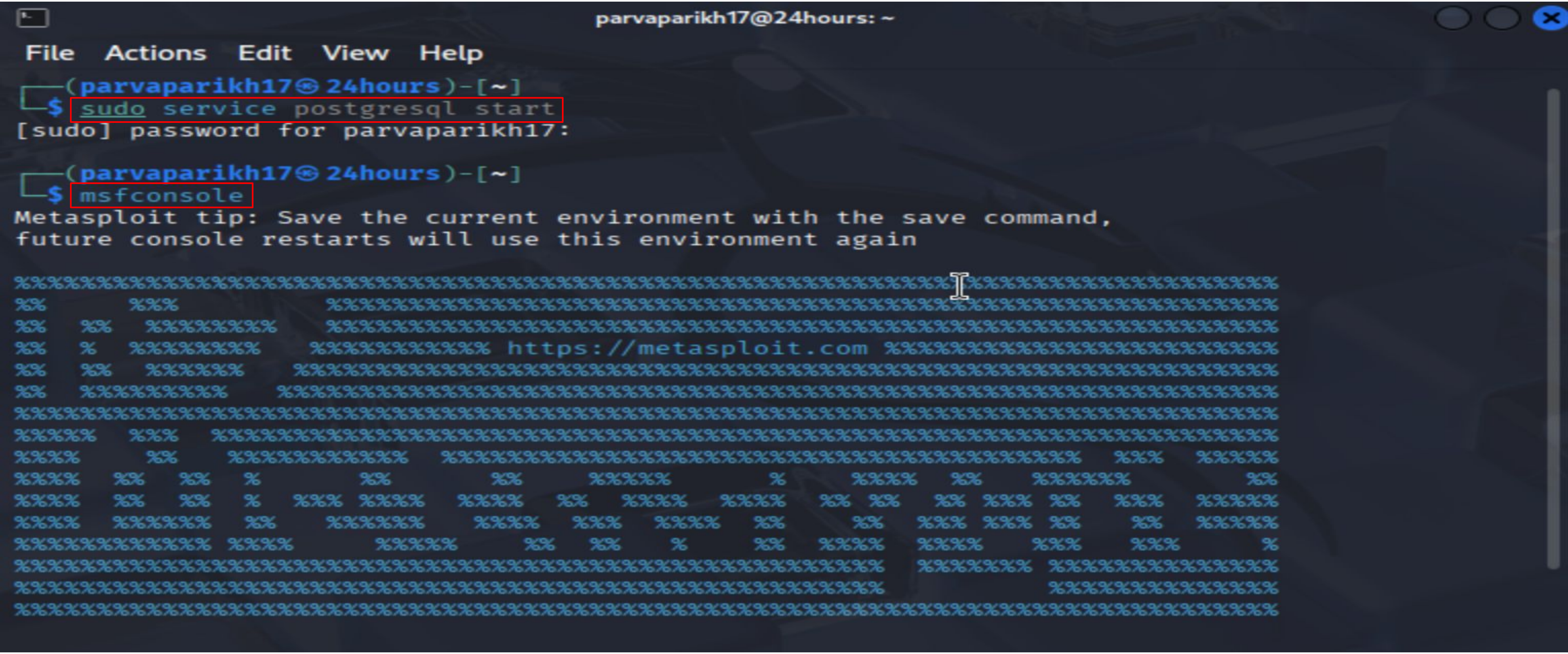
Kali VM IP Address Configuration



```
┌──(parvaparikh17㉿24hours)-[~]
└─$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default ql
en 1000
    link/ether 00:0c:29:8c:3d:cd brd ff:ff:ff:ff:ff:ff
    inet 172.16.221.128/24 brd 172.16.221.255 scope global dynamic noprefixroute eth0
       valid_lft 1779sec preferred_lft 1779sec
    inet6 fe80::20c:29ff:fe8c:3dcd/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Displays the Kali Linux VM's assigned IP address 172.16.221.128, confirming its position in the 172.16.221.0/24 subnet used for internal scanning.

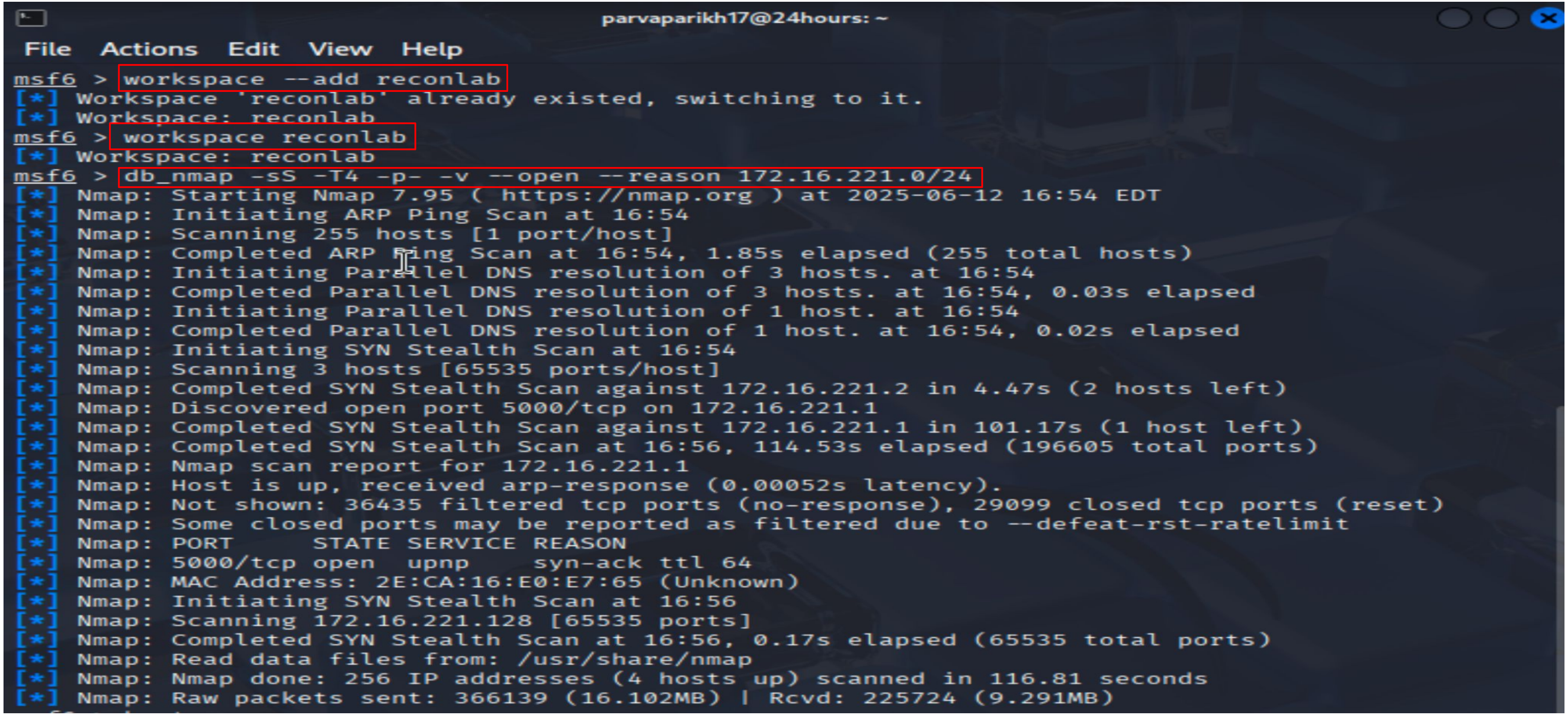
Launching PostgreSQL and Metasploit Console



```
┌──(parvaparikh17㉿24hours)-[~]
└─$ sudo service postgresql start
[sudo] password for parvaparikh17:

┌──(parvaparikh17㉿24hours)-[~]
└─$ msfconsole
Metasploit tip: Save the current environment with the save command,
future console restarts will use this environment again

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%     %%%         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  %%  %%%%%%%%   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  %  %%%%%%%%   %%%%%%%%%%% https://metasploit.com %%%%%%%%%%%%%%%%%%%%%%%%
%%  %%  %%%%%%   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  %%%%%%%%%   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%  %%%  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%    %%   %%%%%%%%%%%  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  %%%  %%%%%
%%%%  %%  %%  %      %%      %%    %%%%%      %    %%%%  %%   %%%%%%       %%
%%%%  %%  %%  %  %%% %%%%  %%%%  %%  %%%%  %%%%  %% %%  %% %%% %%  %%%  %%%%%
%%%%  %%%%%%  %%   %%%%%%   %%%%  %%%  %%%%  %%    %%  %%% %%% %%   %%  %%%%%
%%%%%%%%%%%% %%%%     %%%%%    %%  %%   %    %%  %%%%  %%%%   %%%   %%%     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  %%%%%%% %%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%          %%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

PostgreSQL is started to support Metasploit's database-backed workspace. Metasploit is then launched to begin the reconnaissance workflow.

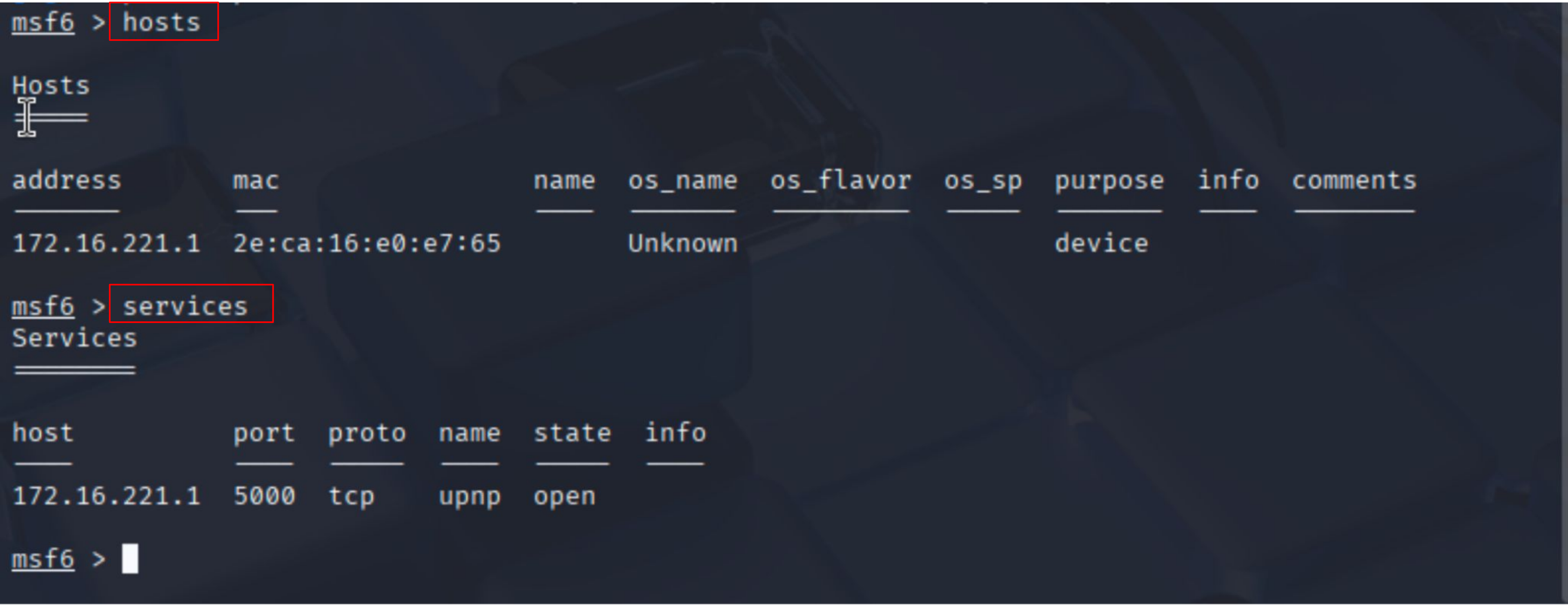
Full TCP SYN Scan in Metasploit Workspace



```
msf6 > workspace --add reconlab
[*] Workspace 'reconlab' already existed, switching to it.
[*] Workspace: reconlab
msf6 > workspace reconlab
[*] Workspace: reconlab
msf6 > db_nmap -sS -T4 -p- -v --open --reason 172.16.221.0/24
[*] Nmap: Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-12 16:54 EDT
[*] Nmap: Initiating ARP Ping Scan at 16:54
[*] Nmap: Scanning 255 hosts [1 port/host]
[*] Nmap: Completed ARP Ping Scan at 16:54, 1.85s elapsed (255 total hosts)
[*] Nmap: Initiating Parallel DNS resolution of 3 hosts. at 16:54
[*] Nmap: Completed Parallel DNS resolution of 3 hosts. at 16:54, 0.03s elapsed
[*] Nmap: Initiating Parallel DNS resolution of 1 host. at 16:54
[*] Nmap: Completed Parallel DNS resolution of 1 host. at 16:54, 0.02s elapsed
[*] Nmap: Initiating SYN Stealth Scan at 16:54
[*] Nmap: Scanning 3 hosts [65535 ports/host]
[*] Nmap: Completed SYN Stealth Scan against 172.16.221.2 in 4.47s (2 hosts left)
[*] Nmap: Discovered open port 5000/tcp on 172.16.221.1
[*] Nmap: Completed SYN Stealth Scan against 172.16.221.1 in 101.17s (1 host left)
[*] Nmap: Completed SYN Stealth Scan at 16:56, 114.53s elapsed (196605 total ports)
[*] Nmap: Nmap scan report for 172.16.221.1
[*] Nmap: Host is up, received arp-response (0.00052s latency).
[*] Nmap: Not shown: 36435 filtered tcp ports (no-response), 29099 closed tcp ports (reset)
[*] Nmap: Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
[*] Nmap: PORT     STATE SERVICE REASON
[*] Nmap: 5000/tcp open  upnp    syn-ack ttl 64
[*] Nmap: MAC Address: 2E:CA:16:E0:E7:65 (Unknown)
[*] Nmap: Initiating SYN Stealth Scan at 16:56
[*] Nmap: Scanning 172.16.221.128 [65535 ports]
[*] Nmap: Completed SYN Stealth Scan at 16:56, 0.17s elapsed (65535 total ports)
[*] Nmap: Read data files from: /usr/share/nmap
[*] Nmap: Nmap done: 256 IP addresses (4 hosts up) scanned in 116.81 seconds
[*] Nmap: Raw packets sent: 366139 (16.102MB) | Rcvd: 225724 (9.291MB)
```

Created a new reconlab workspace and ran a full TCP SYN scan on 172.16.221.0/24, capturing host, port, and service data for all devices in the subnet.

Discovered Hosts and Open Services



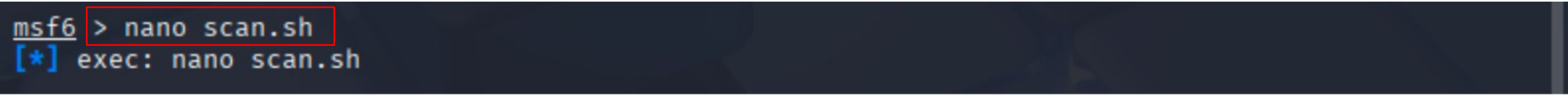
```
msf6 > hosts

Hosts
=====

address       mac                name  os_name  os_flavor  os_sp  purpose  info  comments
-------       ---                ----  -------  ---------  -----  -------  ----  --------
172.16.221.1  2e:ca:16:e0:e7:65        Unknown                    device

msf6 > services
Services
========

host          port  proto  name  state  info
----          ----  -----  ----  -----  ----
172.16.221.1  5000  tcp    upnp  open

msf6 >
```

Used hosts and services commands in Metasploit to confirm that 172.16.221.1 is live and has port 5000/tcp open running a UPNP service.

Part 2: Recon Automation and Scripted Execution

Conducted automation of Metasploit scan process and the integration of traffic capture through Wireshark

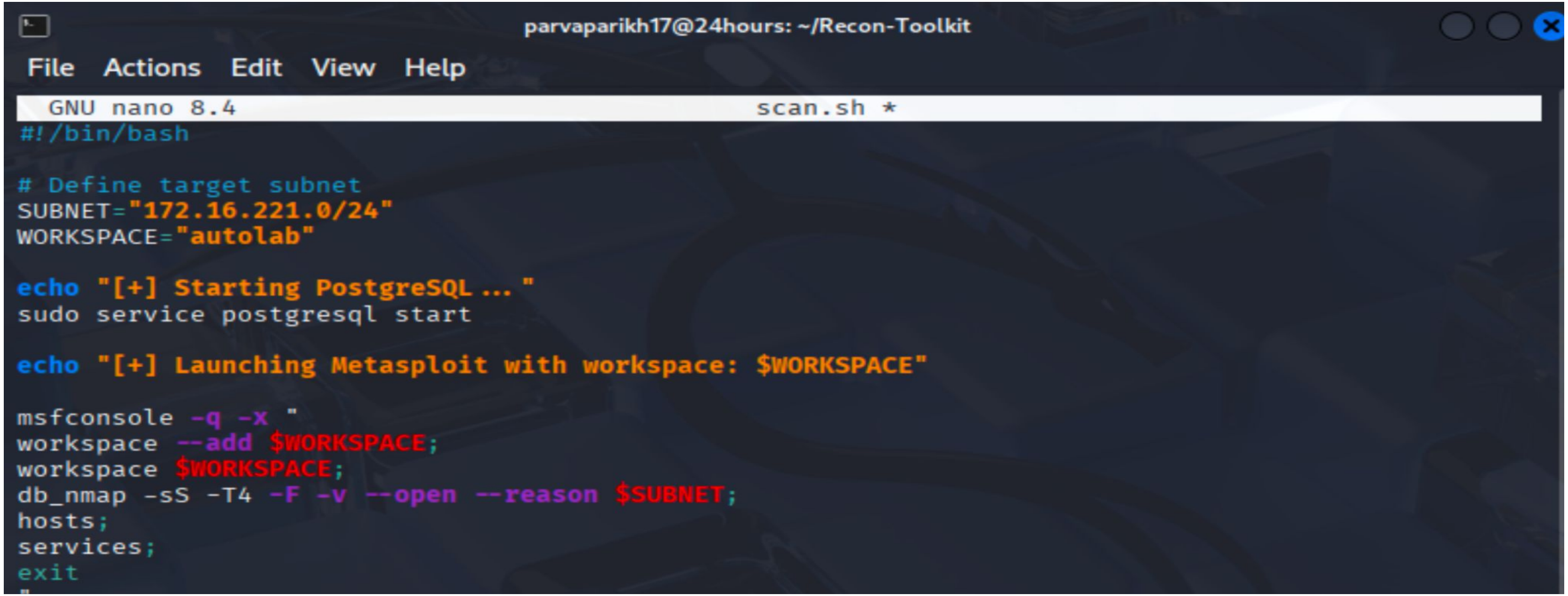
Opening Recon Toolkit Directory



```
msf6 > nano scan.sh
[*] exec: nano scan.sh
```

Navigated to the custom Recon Toolkit directory and prepared to edit the scan automation file.

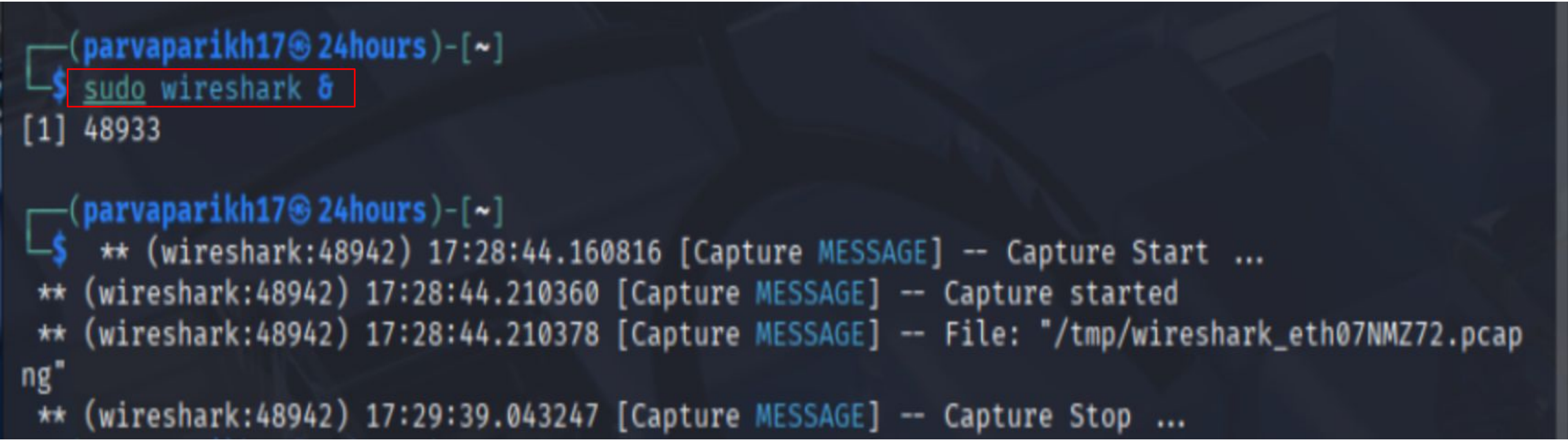
Writing scan.sh Script for Automation



```
  GNU nano 8.4                                scan.sh *
#!/bin/bash

# Define target subnet
SUBNET="172.16.221.0/24"
WORKSPACE="autolab"

echo "[+] Starting PostgreSQL ... "
sudo service postgresql start

echo "[+] Launching Metasploit with workspace: $WORKSPACE"

msfconsole -q -x "
workspace --add $WORKSPACE;
workspace $WORKSPACE;
db_nmap -sS -T4 -F -v --open --reason $SUBNET;
hosts;
services;
exit
"
```

Displays the full script used to automate PostgreSQL startup, workspace creation, and subnet scanning from within Metasploit.

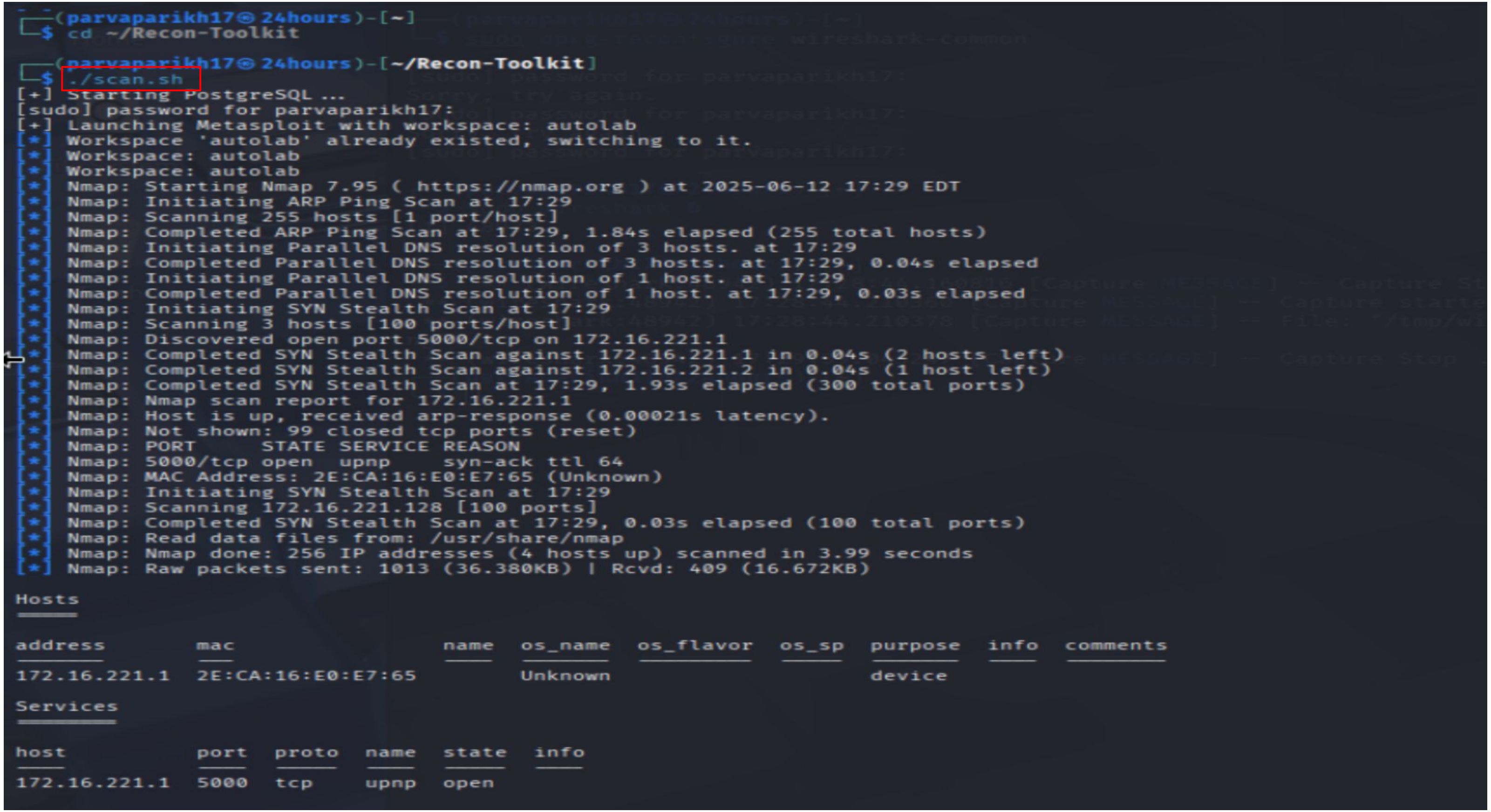
Launching Wireshark for Packet Capture



```
┌──(parvaparikh17㉿24hours)-[~]
└─$ sudo wireshark &
[1] 48933

┌──(parvaparikh17㉿24hours)-[~]
└─$  ** (wireshark:48942) 17:28:44.160816 [Capture MESSAGE] -- Capture Start ...
 ** (wireshark:48942) 17:28:44.210360 [Capture MESSAGE] -- Capture started
 ** (wireshark:48942) 17:28:44.210378 [Capture MESSAGE] -- File: "/tmp/wireshark_eth07NMZ72.pcap
ng"
 ** (wireshark:48942) 17:29:39.043247 [Capture MESSAGE] -- Capture Stop ...
```

Wireshark is started with elevated privileges to monitor live traffic during recon scans, saving the capture as a .pcap file.

Executing scan.sh for Automated Recon



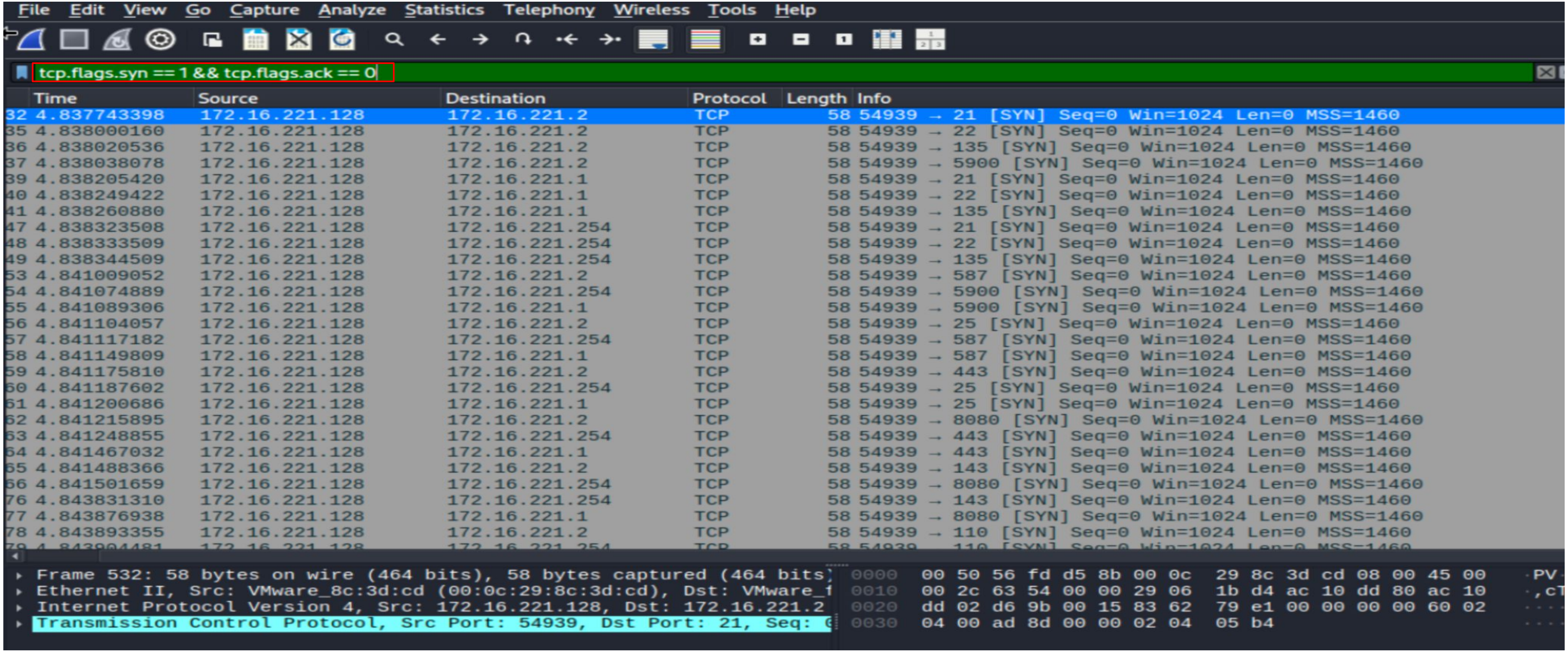
```
┌──(parvaparikh17㉿24hours)-[~]
└─$ cd ~/Recon-Toolkit

┌──(parvaparikh17㉿24hours)-[~/Recon-Toolkit]
└─$ ./scan.sh
[+] Starting PostgreSQL ...
[sudo] password for parvaparikh17:
[+] Launching Metasploit with workspace: autolab
[*] Workspace 'autolab' already existed, switching to it.
[*] Workspace: autolab
[*] Workspace: autolab
[*] Nmap: Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-12 17:29 EDT
[*] Nmap: Initiating ARP Ping Scan at 17:29
[*] Nmap: Scanning 255 hosts [1 port/host]
[*] Nmap: Completed ARP Ping Scan at 17:29, 1.84s elapsed (255 total hosts)
[*] Nmap: Initiating Parallel DNS resolution of 3 hosts. at 17:29
[*] Nmap: Completed Parallel DNS resolution of 3 hosts. at 17:29, 0.04s elapsed
[*] Nmap: Initiating Parallel DNS resolution of 1 host. at 17:29
[*] Nmap: Completed Parallel DNS resolution of 1 host. at 17:29, 0.03s elapsed
[*] Nmap: Initiating SYN Stealth Scan at 17:29
[*] Nmap: Scanning 3 hosts [100 ports/host]
[*] Nmap: Discovered open port 5000/tcp on 172.16.221.1
[*] Nmap: Completed SYN Stealth Scan against 172.16.221.1 in 0.04s (2 hosts left)
[*] Nmap: Completed SYN Stealth Scan against 172.16.221.2 in 0.04s (1 host left)
[*] Nmap: Completed SYN Stealth Scan at 17:29, 1.93s elapsed (300 total ports)
[*] Nmap: Nmap scan report for 172.16.221.1
[*] Nmap: Host is up, received arp-response (0.00021s latency).
[*] Nmap: Not shown: 99 closed tcp ports (reset)
[*] Nmap: PORT     STATE SERVICE REASON
[*] Nmap: 5000/tcp open  upnp    syn-ack ttl 64
[*] Nmap: MAC Address: 2E:CA:16:E0:E7:65 (Unknown)
[*] Nmap: Initiating SYN Stealth Scan at 17:29
[*] Nmap: Scanning 172.16.221.128 [100 ports]
[*] Nmap: Completed SYN Stealth Scan at 17:29, 0.03s elapsed (100 total ports)
[*] Nmap: Read data files from: /usr/share/nmap
[*] Nmap: Nmap done: 256 IP addresses (4 hosts up) scanned in 3.99 seconds
[*] Nmap: Raw packets sent: 1013 (36.380KB) | Rcvd: 409 (16.672KB)

Hosts
=====

address        mac                name  os_name  os_flavor  os_sp  purpose  info  comments
-------        ---                ----  -------  ---------  -----  -------  ----  --------
172.16.221.1   2E:CA:16:E0:E7:65        Unknown                    device

Services
========

host          port  proto  name  state  info
----          ----  -----  ----  -----  ----
172.16.221.1  5000  tcp    upnp  open
```

The automation script is executed, launching a full scan and logging host/service output, enabling reproducible recon workflows.

Part 3: Packet-Level Network Visibility with Wireshark

This phase captures and analyzes live network traffic generated by the Nmap scan. Using Wireshark, filters are applied to isolate SYN packets, host-specific communication, closed port responses, and traffic to an open port—providing full visibility into how network reconnaissance appears at the packet level.

Filtered View of SYN Packets (Scan Initiation)


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

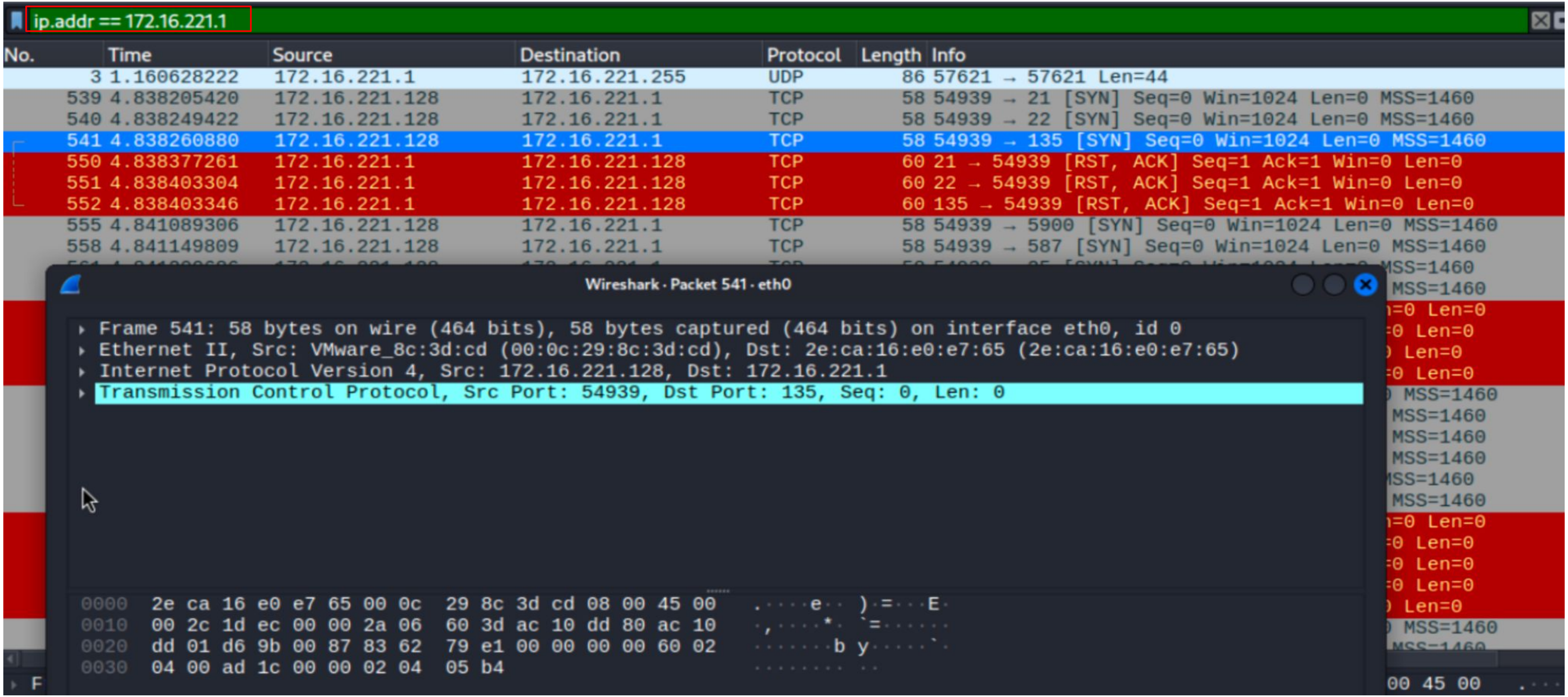
```
tcp.flags.syn == 1 && tcp.flags.ack == 0
```

```
   Time           Source           Destination      Protocol Length Info
32 4.837743398    172.16.221.128   172.16.221.2     TCP      58 54939 → 21 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
35 4.838000160    172.16.221.128   172.16.221.2     TCP      58 54939 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36 4.838020536    172.16.221.128   172.16.221.2     TCP      58 54939 → 135 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
37 4.838038078    172.16.221.128   172.16.221.2     TCP      58 54939 → 5900 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
39 4.838205420    172.16.221.128   172.16.221.1     TCP      58 54939 → 21 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
40 4.838249422    172.16.221.128   172.16.221.1     TCP      58 54939 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
41 4.838260880    172.16.221.128   172.16.221.1     TCP      58 54939 → 135 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
47 4.838323508    172.16.221.128   172.16.221.254   TCP      58 54939 → 21 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
48 4.838333509    172.16.221.128   172.16.221.254   TCP      58 54939 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
49 4.838344509    172.16.221.128   172.16.221.254   TCP      58 54939 → 135 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
53 4.841009052    172.16.221.128   172.16.221.2     TCP      58 54939 → 587 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
54 4.841074889    172.16.221.128   172.16.221.254   TCP      58 54939 → 5900 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
55 4.841089306    172.16.221.128   172.16.221.1     TCP      58 54939 → 5900 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
56 4.841104057    172.16.221.128   172.16.221.2     TCP      58 54939 → 25 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
57 4.841117182    172.16.221.128   172.16.221.254   TCP      58 54939 → 587 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
58 4.841149809    172.16.221.128   172.16.221.1     TCP      58 54939 → 587 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
59 4.841175810    172.16.221.128   172.16.221.2     TCP      58 54939 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
60 4.841187602    172.16.221.128   172.16.221.254   TCP      58 54939 → 25 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
61 4.841200686    172.16.221.128   172.16.221.1     TCP      58 54939 → 25 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
62 4.841215895    172.16.221.128   172.16.221.2     TCP      58 54939 → 8080 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
63 4.841248855    172.16.221.128   172.16.221.254   TCP      58 54939 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
64 4.841467032    172.16.221.128   172.16.221.1     TCP      58 54939 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
65 4.841488366    172.16.221.128   172.16.221.2     TCP      58 54939 → 143 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
66 4.841501659    172.16.221.128   172.16.221.254   TCP      58 54939 → 8080 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
76 4.843831310    172.16.221.128   172.16.221.254   TCP      58 54939 → 143 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
77 4.843876938    172.16.221.128   172.16.221.1     TCP      58 54939 → 8080 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
78 4.843893355    172.16.221.128   172.16.221.2     TCP      58 54939 → 110 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
79 4.843904481    172.16.221.128   172.16.221.254   TCP      58 54939 → 110 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
```

```
▸ Frame 532: 58 bytes on wire (464 bits), 58 bytes captured (464 bits)   0000  00 50 56 fd d5 8b 00 0c  29 8c 3d cd 08 00 45 00   ·PV
▸ Ethernet II, Src: VMware_8c:3d:cd (00:0c:29:8c:3d:cd), Dst: VMware_f   0010  00 2c 63 54 00 00 29 06  1b d4 ac 10 dd 80 ac 10   ·,cT
▸ Internet Protocol Version 4, Src: 172.16.221.128, Dst: 172.16.221.2   0020  dd 02 d6 9b 00 15 83 62  79 e1 00 00 00 00 60 02   ····
▸ Transmission Control Protocol, Src Port: 54939, Dst Port: 21, Seq: (   0030  04 00 ad 8d 00 00 02 04  05 b4                     ····
```

Applied the filter **tcp.flags.syn == 1 && tcp.flags.ack == 0** to isolate TCP SYN packets sent by Nmap, indicating attempted connections to various destination ports.

Isolating Traffic to Target Host (172.16.221.1)



```
ip.addr == 172.16.221.1
```

```
No.   Time         Source           Destination      Protocol Length Info
    3 1.160628222  172.16.221.1     172.16.221.255   UDP         86 57621 → 57621 Len=44
  539 4.838205420  172.16.221.128   172.16.221.1     TCP         58 54939 → 21 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  540 4.838249422  172.16.221.128   172.16.221.1     TCP         58 54939 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  541 4.838260880  172.16.221.128   172.16.221.1     TCP         58 54939 → 135 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  550 4.838377261  172.16.221.1     172.16.221.128   TCP         60 21 → 54939 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  551 4.838403304  172.16.221.1     172.16.221.128   TCP         60 22 → 54939 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  552 4.838403346  172.16.221.1     172.16.221.128   TCP         60 135 → 54939 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  555 4.841089306  172.16.221.128   172.16.221.1     TCP         58 54939 → 5900 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  558 4.841149809  172.16.221.128   172.16.221.1     TCP         58 54939 → 587 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
```

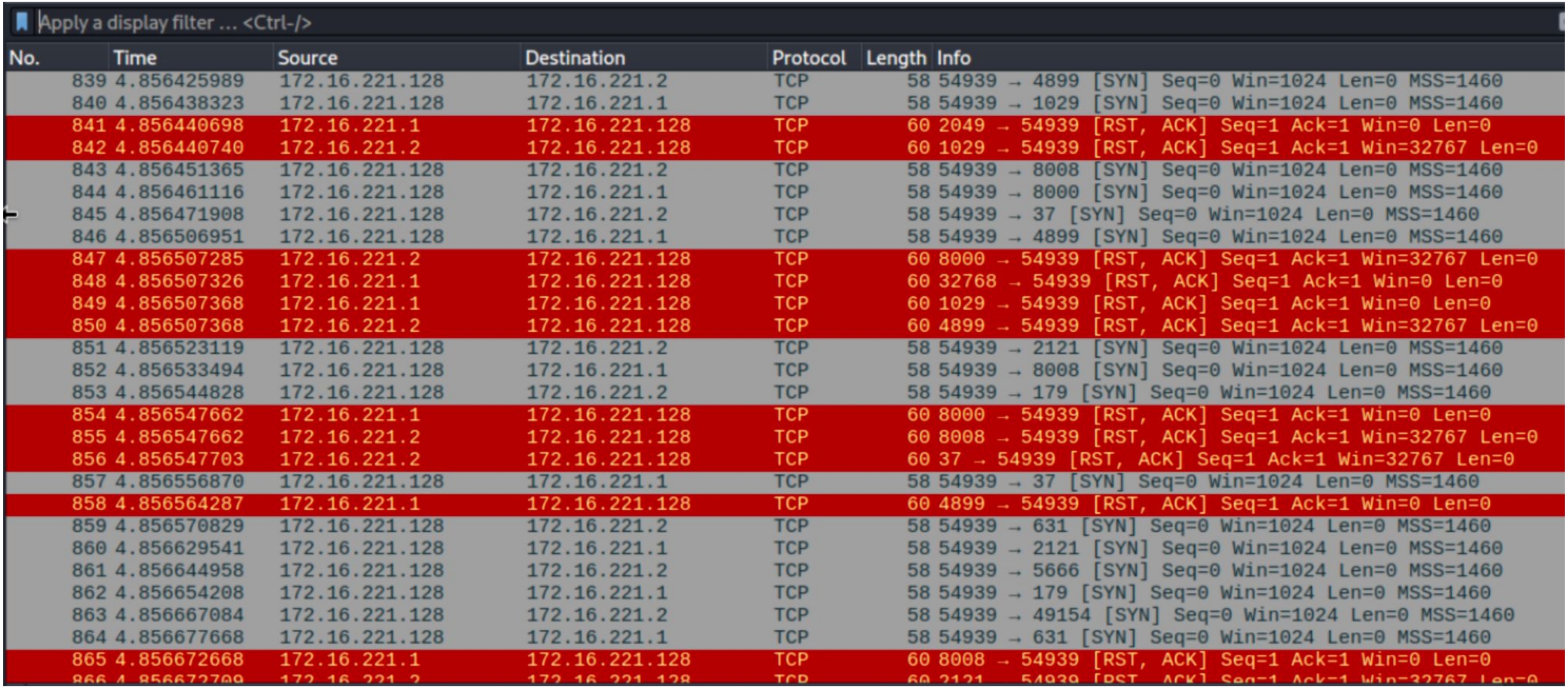
Wireshark · Packet 541 · eth0

```
▸ Frame 541: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface eth0, id 0
▸ Ethernet II, Src: VMware_8c:3d:cd (00:0c:29:8c:3d:cd), Dst: 2e:ca:16:e0:e7:65 (2e:ca:16:e0:e7:65)
▸ Internet Protocol Version 4, Src: 172.16.221.128, Dst: 172.16.221.1
▸ Transmission Control Protocol, Src Port: 54939, Dst Port: 135, Seq: 0, Len: 0
```

```
0000  2e ca 16 e0 e7 65 00 0c  29 8c 3d cd 08 00 45 00   .····e·· )·=···E·
0010  00 2c 1d ec 00 00 2a 06  60 3d ac 10 dd 80 ac 10   ·,····*· `=······
0020  dd 01 d6 9b 00 87 83 62  79 e1 00 00 00 00 60 02   ·······b y·····`·
0030  04 00 ad 1c 00 00 02 04  05 b4                     ········ ··
```

Used **ip.addr == 172.16.221.1** to view all traffic between the attacker and the discovered host. The expanded pane shows a SYN packet to port 135.

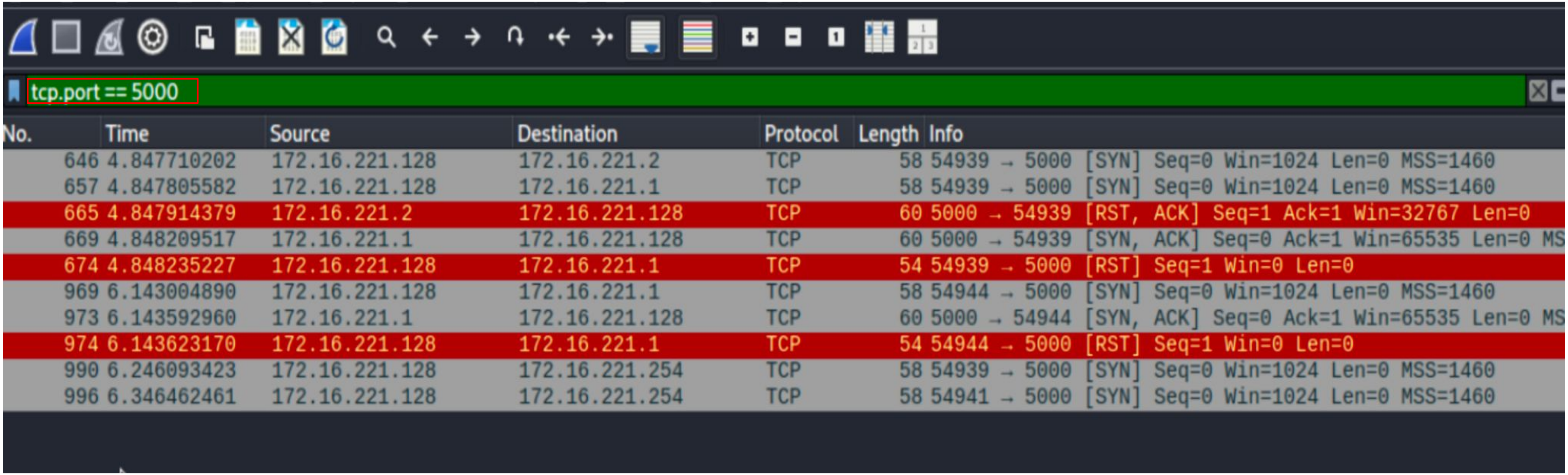
Unfiltered View of Network Scan Traffic



```
No.  Time         Source          Destination     Protocol Length Info
 839 4.856425989  172.16.221.128  172.16.221.2    TCP      58 54939 → 4899 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
 840 4.856438323  172.16.221.128  172.16.221.1    TCP      58 54939 → 1029 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
 841 4.856440698  172.16.221.1    172.16.221.128  TCP      60 2049 → 54939 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 842 4.856440740  172.16.221.2    172.16.221.128  TCP      60 1029 → 54939 [RST, ACK] Seq=1 Ack=1 Win=32767 Len=0
 843 4.856451365  172.16.221.128  172.16.221.2    TCP      58 54939 → 8008 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
 844 4.856461116  172.16.221.128  172.16.221.1    TCP      58 54939 → 8000 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
 845 4.856471908  172.16.221.128  172.16.221.2    TCP      58 54939 → 37 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
 846 4.856506951  172.16.221.128  172.16.221.1    TCP      58 54939 → 4899 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
 847 4.856507285  172.16.221.2    172.16.221.128  TCP      60 8000 → 54939 [RST, ACK] Seq=1 Ack=1 Win=32767 Len=0
 848 4.856507326  172.16.221.1    172.16.221.128  TCP      60 32768 → 54939 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 849 4.856507368  172.16.221.1    172.16.221.128  TCP      60 1029 → 54939 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 850 4.856507368  172.16.221.2    172.16.221.128  TCP      60 4899 → 54939 [RST, ACK] Seq=1 Ack=1 Win=32767 Len=0
 851 4.856523119  172.16.221.128  172.16.221.2    TCP      58 54939 → 2121 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
 852 4.856533494  172.16.221.128  172.16.221.1    TCP      58 54939 → 8008 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
 853 4.856544828  172.16.221.128  172.16.221.2    TCP      58 54939 → 179 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
 854 4.856547662  172.16.221.1    172.16.221.128  TCP      60 8000 → 54939 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 855 4.856547662  172.16.221.2    172.16.221.128  TCP      60 8008 → 54939 [RST, ACK] Seq=1 Ack=1 Win=32767 Len=0
 856 4.856547703  172.16.221.2    172.16.221.128  TCP      60 37 → 54939 [RST, ACK] Seq=1 Ack=1 Win=32767 Len=0
 857 4.856556870  172.16.221.128  172.16.221.1    TCP      58 54939 → 37 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
 858 4.856564287  172.16.221.1    172.16.221.128  TCP      60 4899 → 54939 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 859 4.856570829  172.16.221.128  172.16.221.2    TCP      58 54939 → 631 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
 860 4.856629541  172.16.221.128  172.16.221.1    TCP      58 54939 → 2121 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
 861 4.856644958  172.16.221.128  172.16.221.2    TCP      58 54939 → 5666 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
 862 4.856654208  172.16.221.128  172.16.221.1    TCP      58 54939 → 179 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
 863 4.856667084  172.16.221.128  172.16.221.2    TCP      58 54939 → 49154 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
 864 4.856677668  172.16.221.128  172.16.221.1    TCP      58 54939 → 631 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
 865 4.856672668  172.16.221.1    172.16.221.128  TCP      60 8008 → 54939 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 866 4.856672709  172.16.221.2    172.16.221.128  TCP      60 2121 → 54939 [RST, ACK] Seq=1 Ack=1 Win=32767 Len=0
```

Displays raw, unfiltered TCP traffic during the Nmap scan, capturing SYN, SYN-ACK, and RST-ACK packets to provide a complete timeline of scanning behavior.

Filtering by Open Port 5000 Traffic



tcp.port == 5000

```
No.   Time         Source          Destination     Protocol Length Info
  646 4.847710202  172.16.221.128  172.16.221.2    TCP      58 54939 → 5000 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  657 4.847805582  172.16.221.128  172.16.221.1    TCP      58 54939 → 5000 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  665 4.847914379  172.16.221.2    172.16.221.128  TCP      60 5000 → 54939 [RST, ACK] Seq=1 Ack=1 Win=32767 Len=0
  669 4.848209517  172.16.221.1    172.16.221.128  TCP      60 5000 → 54939 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MS
  674 4.848235227  172.16.221.128  172.16.221.1    TCP      54 54939 → 5000 [RST] Seq=1 Win=0 Len=0
  969 6.143004890  172.16.221.128  172.16.221.1    TCP      58 54944 → 5000 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  973 6.143592960  172.16.221.1    172.16.221.128  TCP      60 5000 → 54944 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MS
  974 6.143623170  172.16.221.128  172.16.221.1    TCP      54 54944 → 5000 [RST] Seq=1 Win=0 Len=0
  990 6.246093423  172.16.221.128  172.16.221.254  TCP      58 54939 → 5000 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  996 6.346462461  172.16.221.128  172.16.221.254  TCP      58 54941 → 5000 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
```

Applied the filter tcp.port == 5000 to focus on communication involving the open UPNP service discovered earlier, helping verify valid responses.

This project demonstrates a full-spectrum approach to internal network reconnaissance, combining attacker-side scanning with defender-side traffic analysis. By leveraging Metasploit, Nmap, and Wireshark in a controlled environment, I was able to identify active hosts, enumerate services, and observe scan behavior at the packet level. The addition of automation through scripting further streamlined the process, making it efficient and repeatable. Altogether, this work reflects both a practical understanding of offensive recon techniques and an analytical ability to interpret how those actions appear from a defensive perspective.