IoT Security – Autumn 2024
Lab 7: Port Scanning on IoT Devices

Manh Bui
School of Electrical and Data Engineering
Email: DucManh.Bui@uts.edu.au

Objectives

- This lab provides an overview of **scanning network traffic** the Nmap tool
- The flow of implementation:
  - Part 1: Perform a Nmap network discovery scan
  - Part 2: Compare a Nmap TCP default port scan and a full scan
  - Part 3: Perform a Nmap UDP scan
  - Part 4: Perform Nmap OS and Service Foot Printing

Analysis of cyberattacks on IoT devices

- IoT systems are **vulnerable** to various cyberattacks (Attacks on IoT devices, IoT web apps, IoT servers,…)
- Some known attacks are Denial of Service (**DoS**), Distributed Denial of Service (**DDoS**), Malware, Side-channel,…

How does cybercrime identify the vulnerabilities?

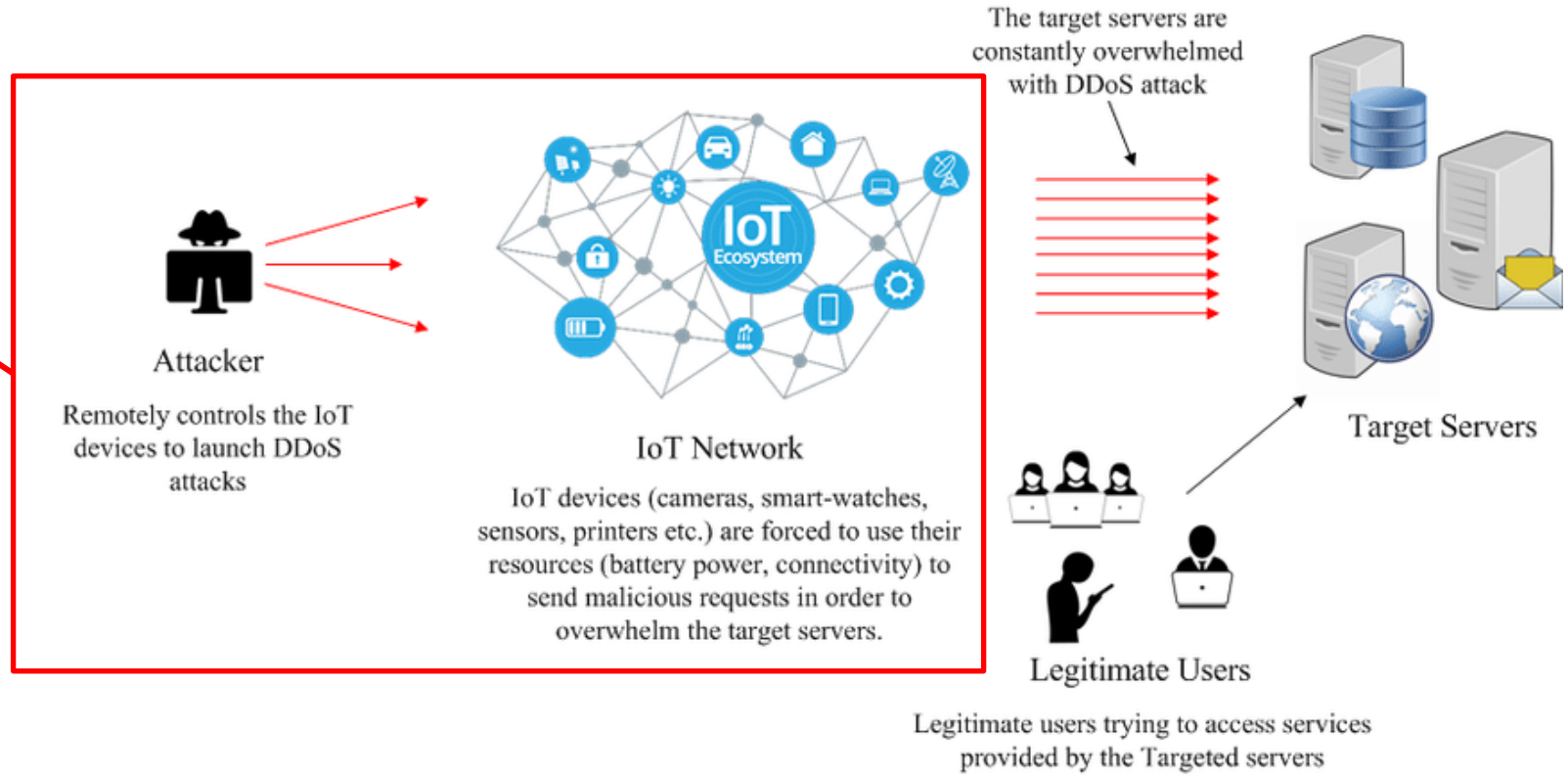


Figure 1: Example of DDoS attacks on IoT environment

Network Scanning

- Network Scanning is the process of **collecting and analysing** machine networking
- It aims to identify servers, end devices, running services, open ports, software versions, and operating systems…
- It can be used for legal and illegal purposes:
  - Legal: Maintaining and enhancing networking infrastructure
  - Illegal: Analyze network vulnerabilities
- Types of scanning techniques: **Port Scanning**, Vulnerability Scanning, IP Scanning, OS Detection,…

What is a Port?

- Port is a virtual point where network connections start and end
- Ports are assigned **port numbers** which, conjunct with an IP address, form vital information that each internet service provider (ISP) **uses to fulfill requests**
- **Port numbers** range from **0 to 65535**
- Ports **numbered 0 to 1023** are **"well-known" ports**, which are typically reserved for **internet usage**
- Some of the most frequently used ports:
  - Port 20 (UDP): File Transfer Protocol (FTP) used for transferring data
  - Port 22 (TCP): Secure Shell (SSH) protocol for FTP, port forwarding and secure logins
  - Port 23 (TCP): Telnet protocol used for unencrypted communication
  - Port 53 (UDP): The DNS which translates the internet domain to machine-readable IP
  - Port 80 (TCP): The World Wide Web HTTP

Port Scanning

- A port scan is a common technique hackers use to **discover open doors** or **weak points in a network**
- A port scan attack helps cyber criminals **find open ports** and **figure out whether they are receiving or sending data**
- Some port scanning methods: Ping scans, Vanilla scan, **SYN scan**, **UDP scan**, XMAS and FIN scans, FTP bounce scan, Sweep scan
- Tools for port scanning: Wireshark, **Nmap**,…

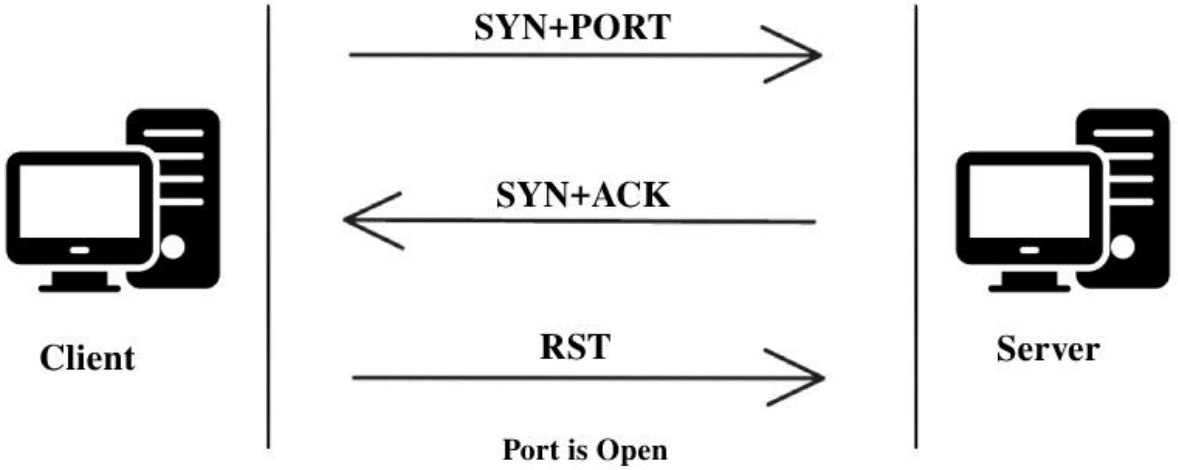


Figure 2: Example of SYN scan

Nmap tool

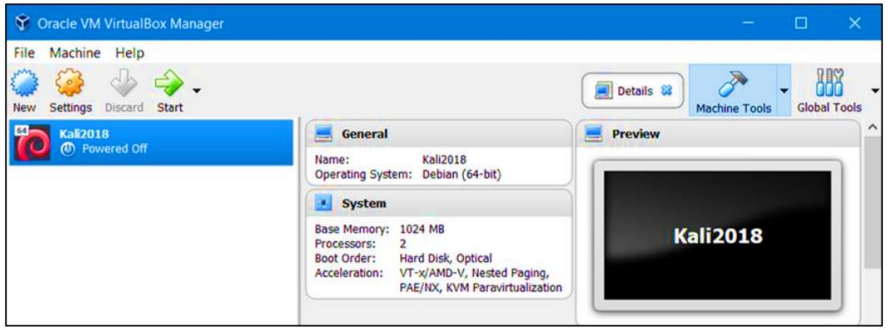
- Nmap, or “Network mapper,” is a free and open source network port scanner.
- Nmap provides information on
  - Every active IP
  - Your network as whole
  - Vulnerabilities
- Nmap is used to scan
  - Enterprise-scale networks
  - Small business networks
  - Connected devices
  - IoT device and traffic
- Nmap common functions
  - Ping scanning
  - Port scanning
  - Host/OS scanning
  - Scan top ports
  - Output to files
  - Disable DNS resolution

Required Resources

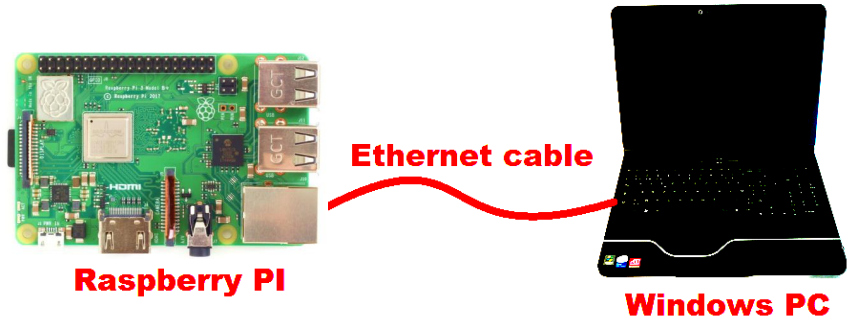
Part1: Raspberry Pi 3 Model B or later

Part2: 8GB Micro SD card (minimum required)

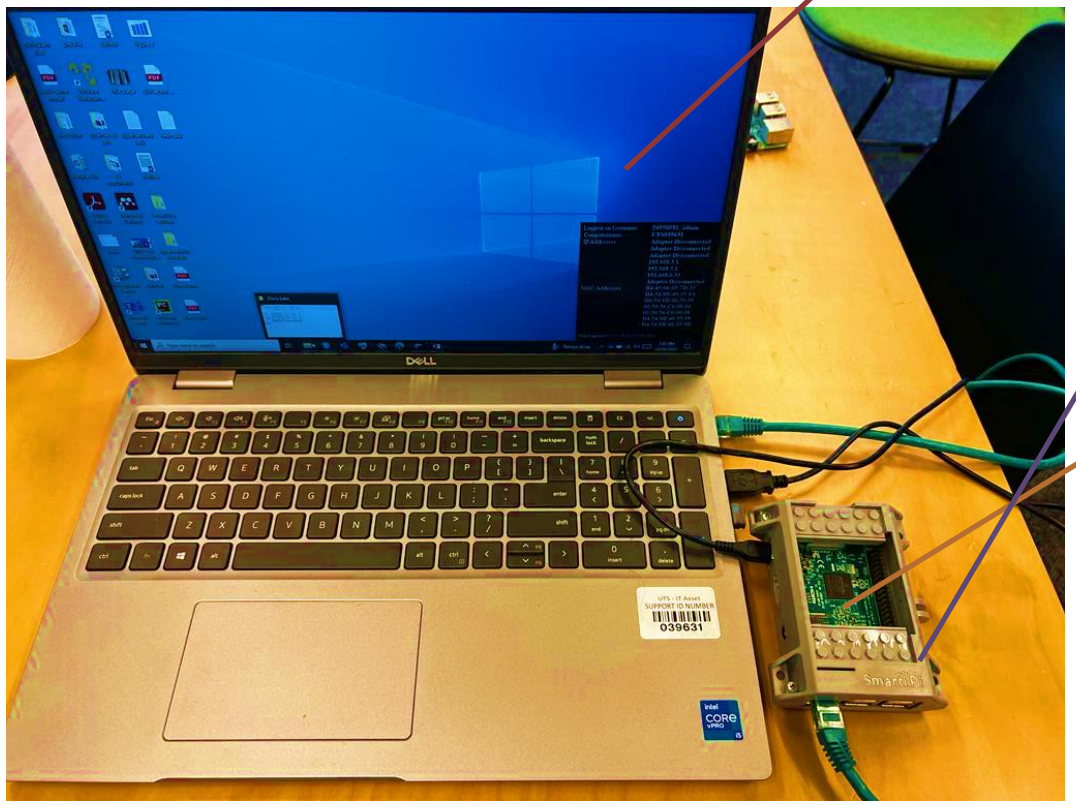
Part3: PC with IoTSec Kali VM



Part4: Network connectivity between PC and Raspberry Pi



Environment Setup



PC with IoTSec Kali VM

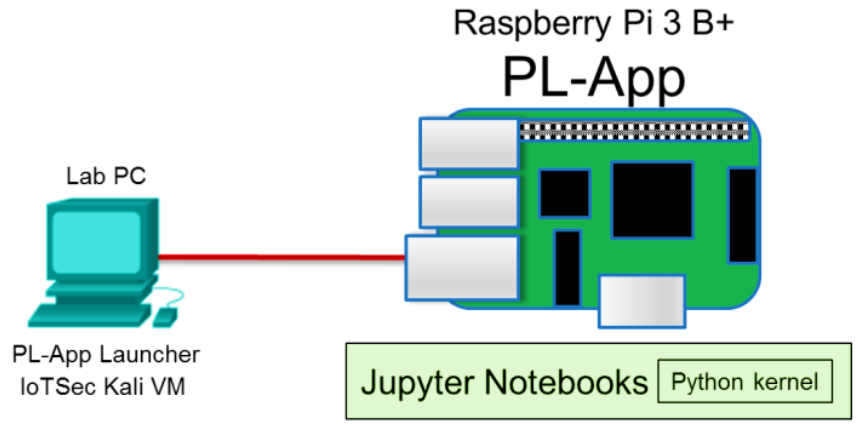
Sd card inserted

Raspberry Pi 3

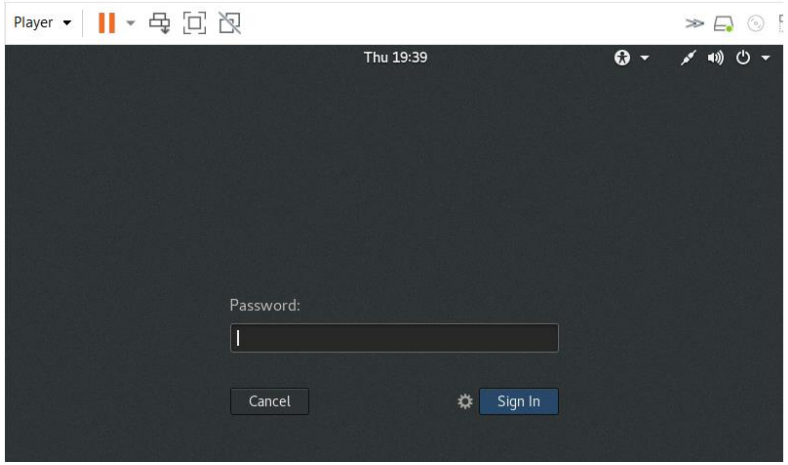
Part 1: Perform a Nmap Network Discovery Scan

Step 1: Use Kali to perform a host discovery scan

a. Set up the topology by connecting the Raspberry Pi to the PC.

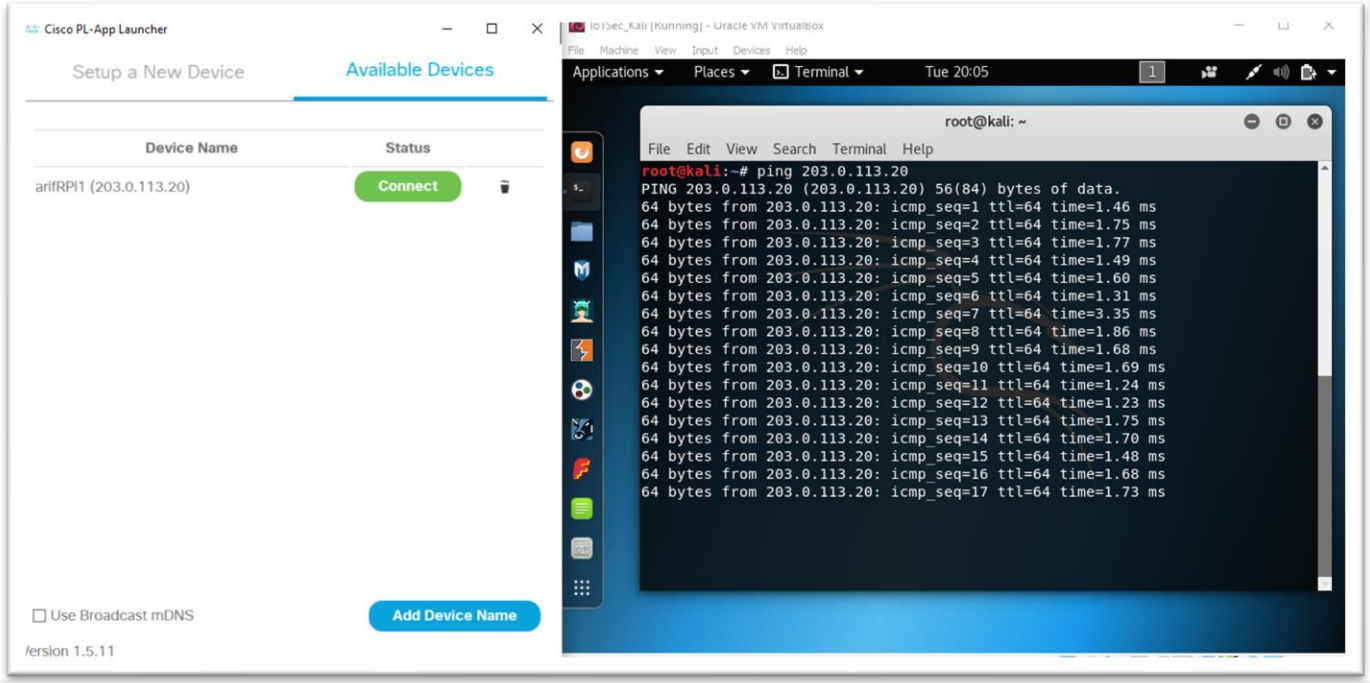


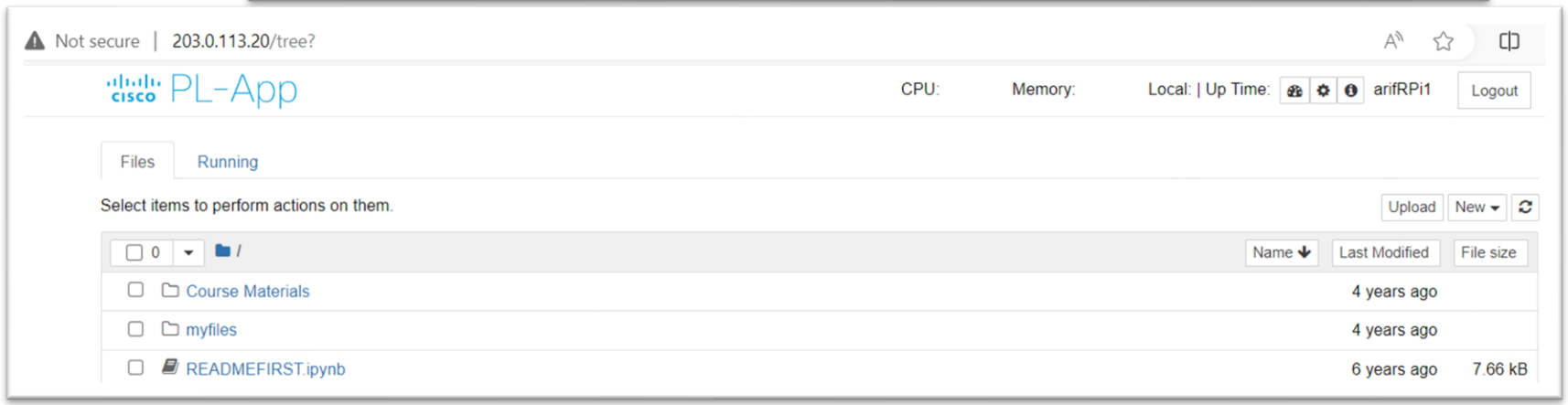
b. Start the IoTSec Kali VM and log in.



Part 1: Perform a Nmap Network Discovery Scan

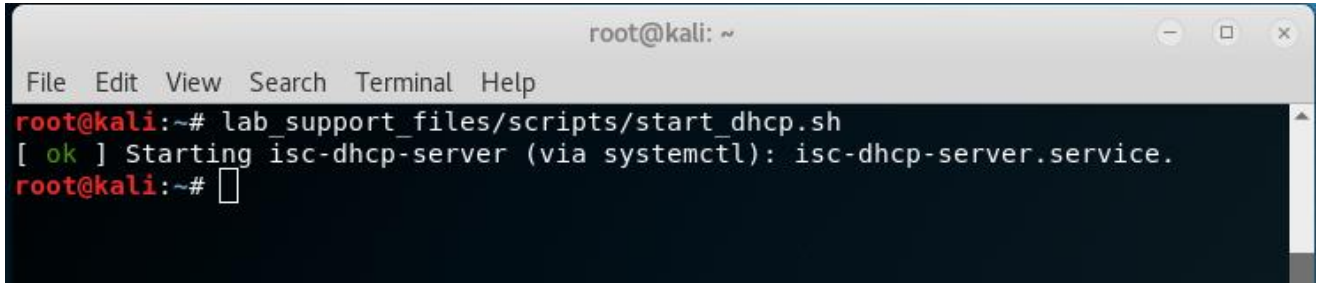
a. Set up the topology by connecting the Raspberry Pi to the PC.



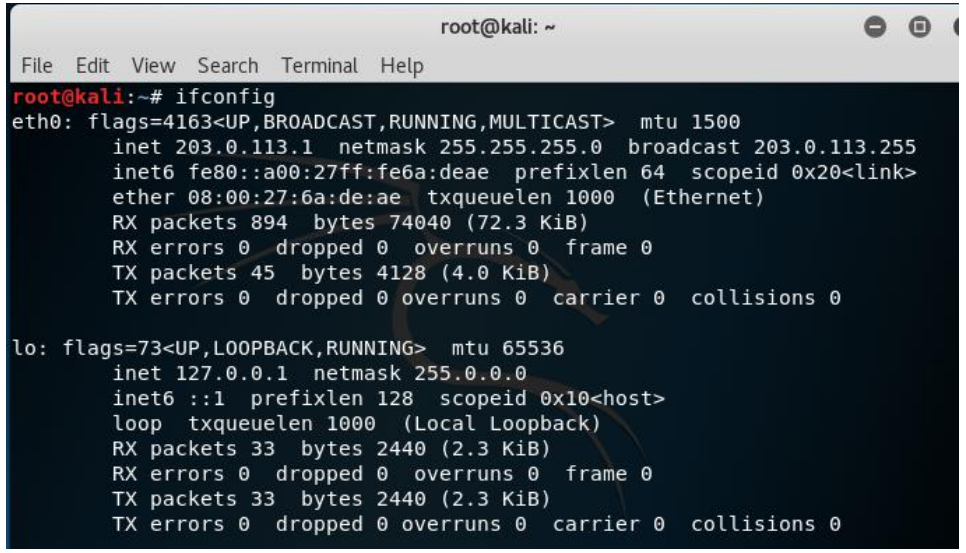


Part 1: Perform a Nmap Network Discovery Scan

c. Open a terminal and start the DHCP server on the Kali VM.
root@kali:~# lab_support_files/scripts/start_dhcp.sh



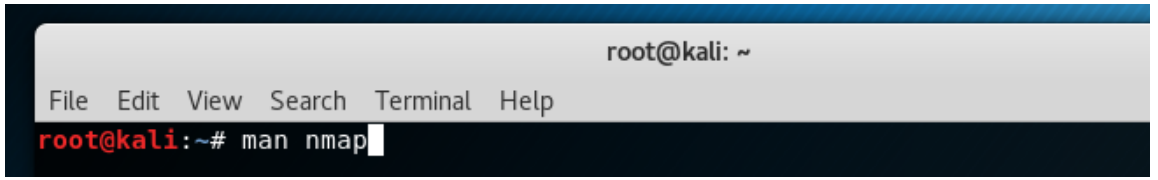
d. Verify that Kali VM is assigned an IP address on eth0.
root@kali:~# ifconfig

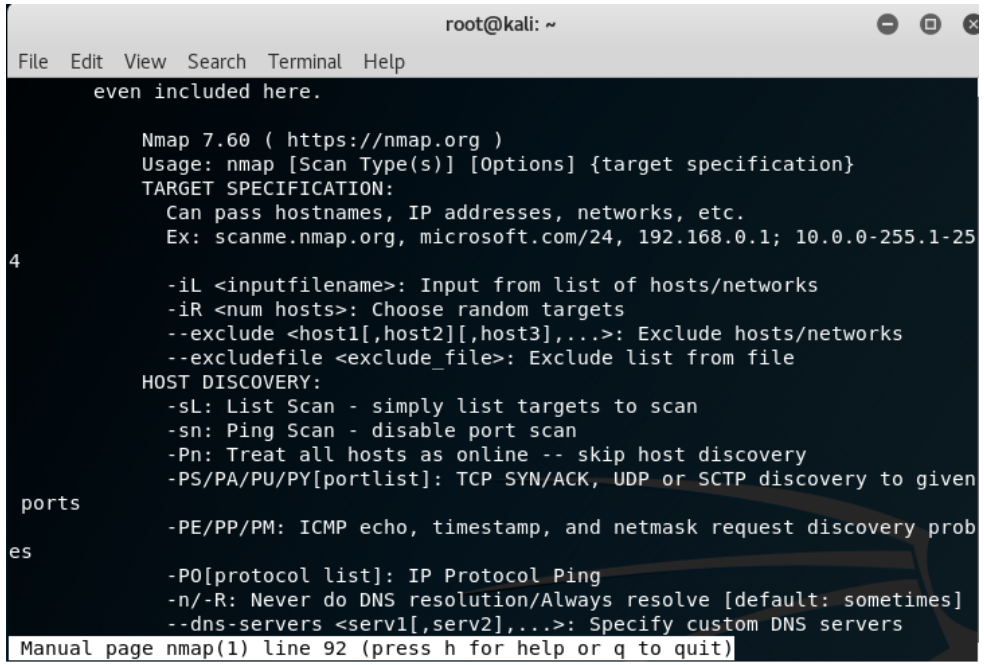


Part 1: Perform a Nmap Network Discovery Scan

e. Open the man page for nmap in Kali VM and review the options that are available in Nmap.

root@kali:~# man nmap

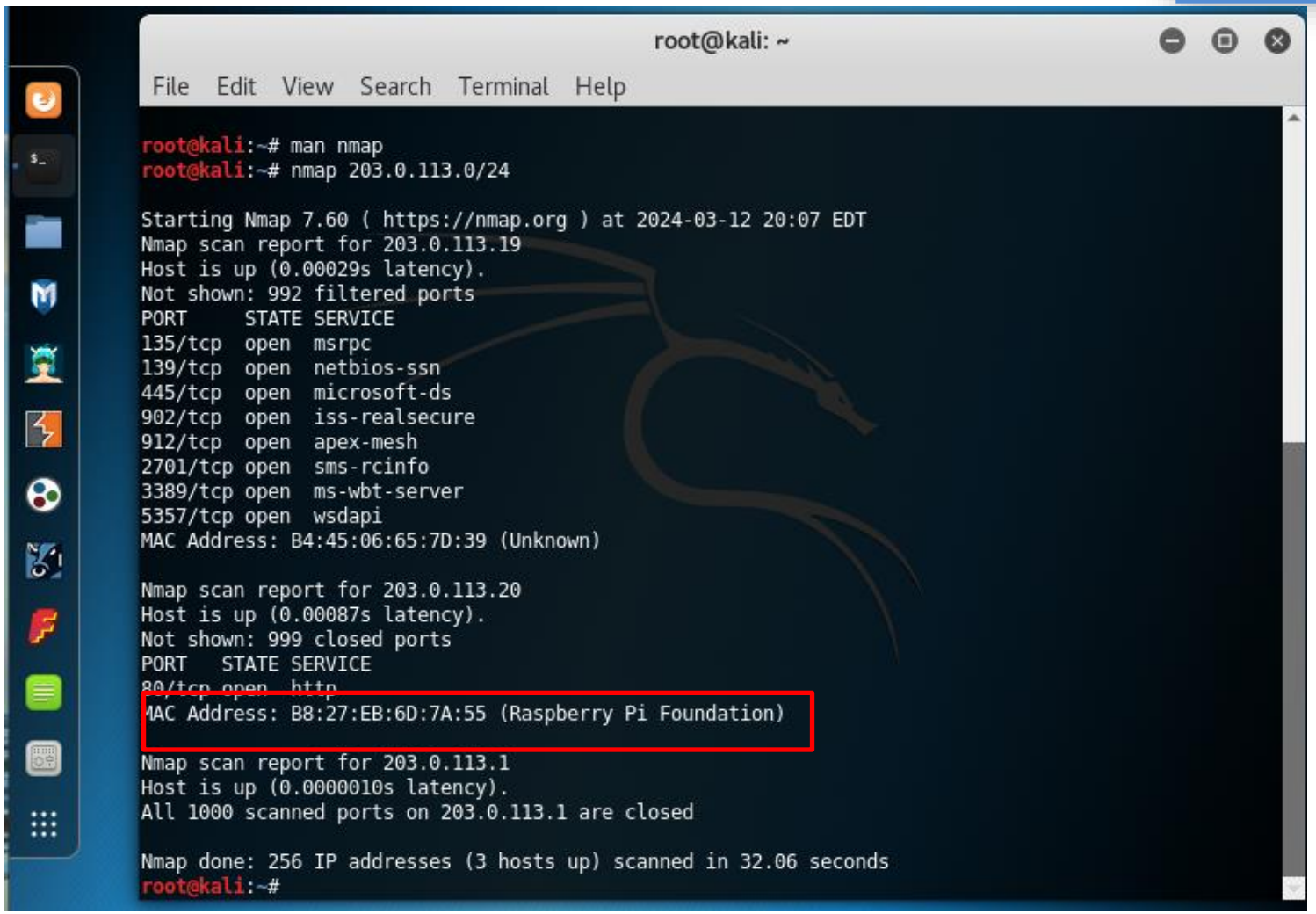




Part 1: Perform a Nmap Network Discovery Scan

f. Perform a scan on your network by specifying the network address and bit mask.

root@kali:~# nmap 203.0.113.0/24

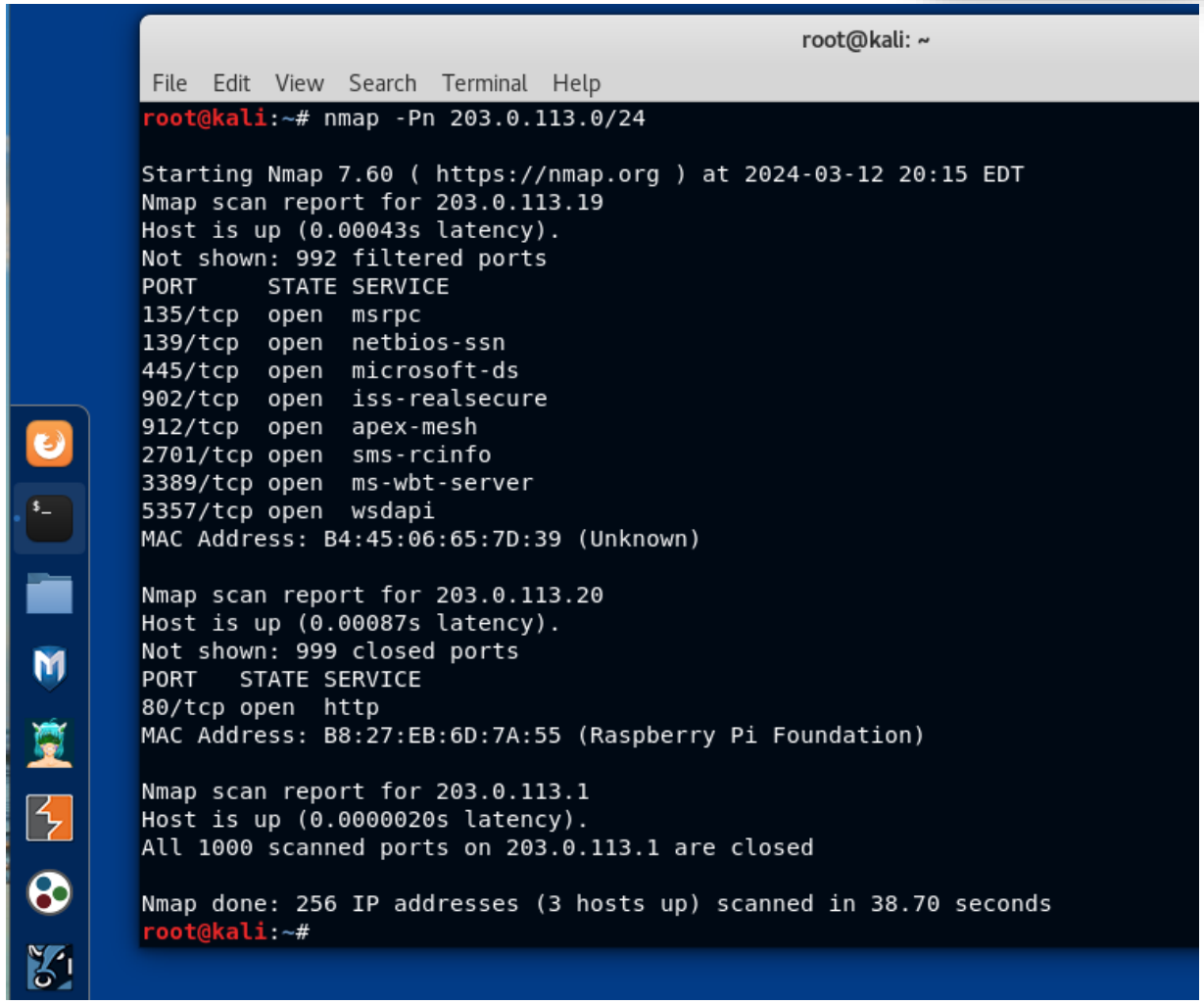


Part 1: Perform a Nmap Network Discovery Scan

g. Sometimes a device would not reply to Nmap's initial network discovery scan because of a firewall, IDS/IPS system etc. Instead of relying on the initial scan to discover hosts that are alive for further scanning.

we can use Nmap to scan the network by assuming all hosts are alive.

root@kali:~# nmap -Pn 203.0.113.0/24 – answer will vary



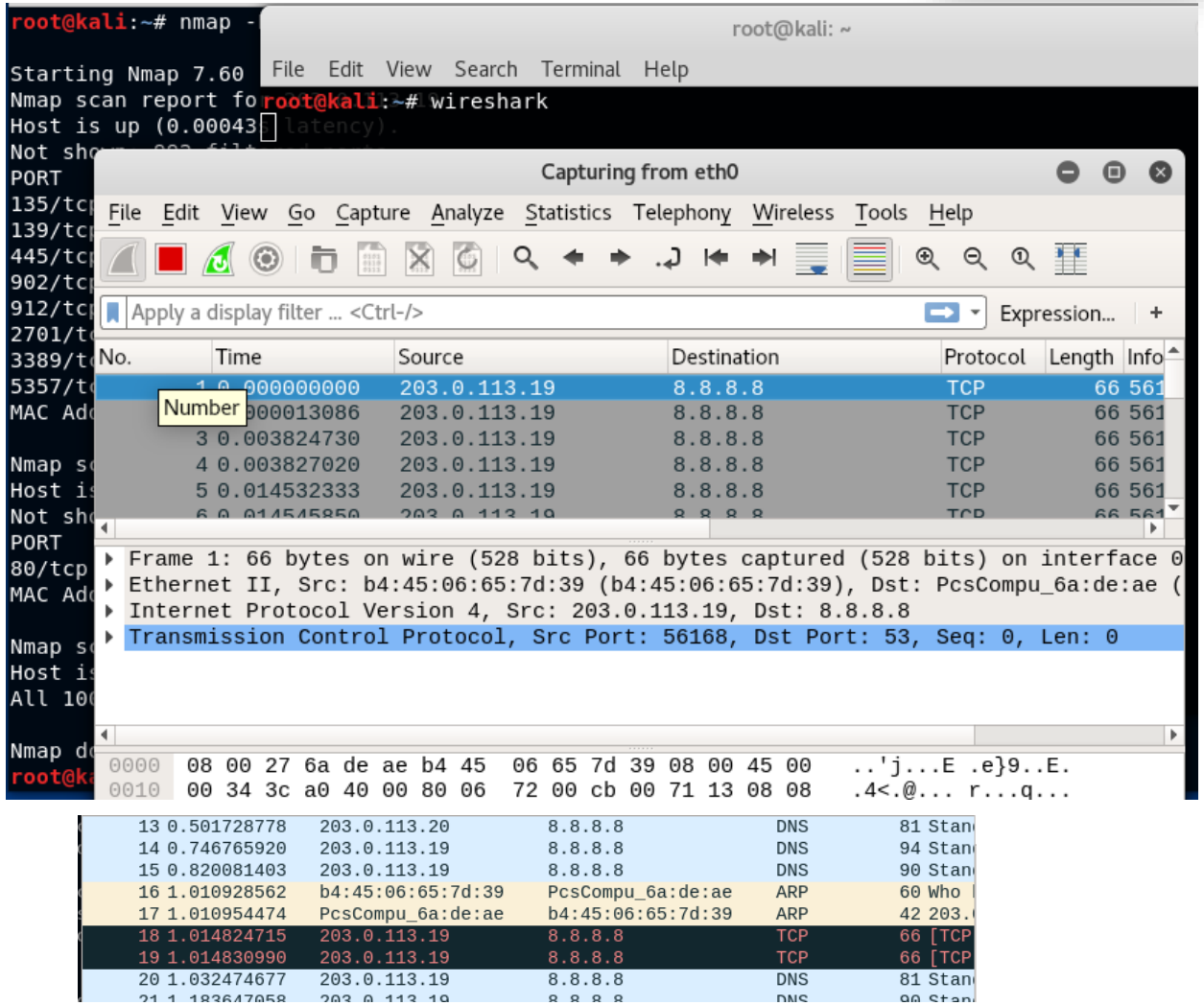
Part 2: Compare a Nmap TCP Default Scan and Full Scan

Step 1: Using Wireshark to display Nmap scans

a. In a Kali VM terminal, start Wireshark. Wireshark is used to monitor the traffic while scanning the network using Nmap. Click **OK** for the warning message regarding running Wireshark as a root user.

root@kali:~# wireshark

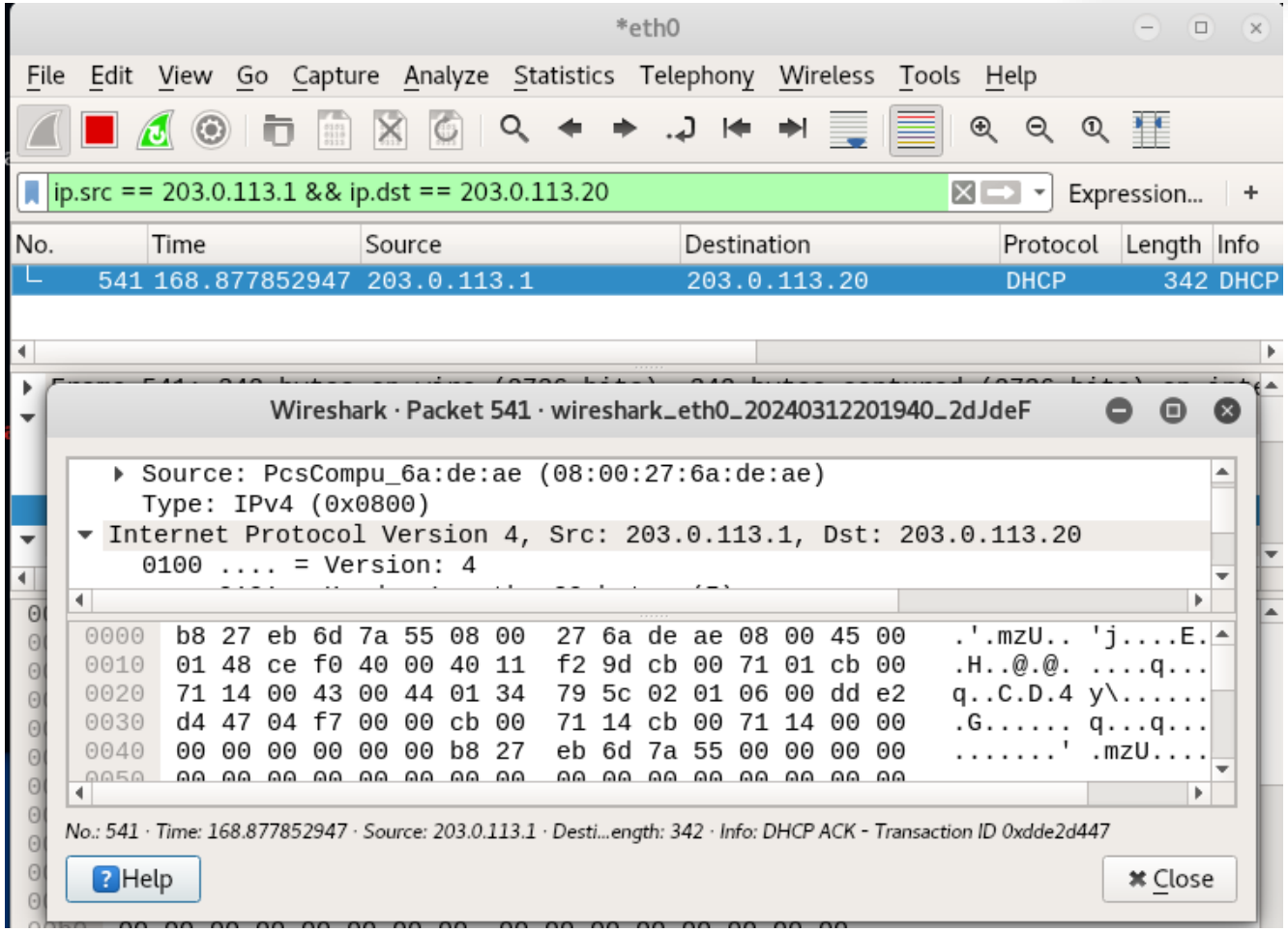
b. Select the **eth0** interface and click **Capture** to start capturing packets



Part 2: Compare a Nmap TCP Default Scan and Full Scan

c. There can be a lot of traffic on the network. A display filter is applied to limit the number of captured packets displayed to just those that you are interested in. The interesting traffic is between the IP address of the Kali VM and the Raspberry Pi. Replace IP address of the Raspberry Pi with the IP address Apply the following display filter in Wireshark using IP address of Kali VM as the source address and IP address of your Raspberry Pi as the destination address.

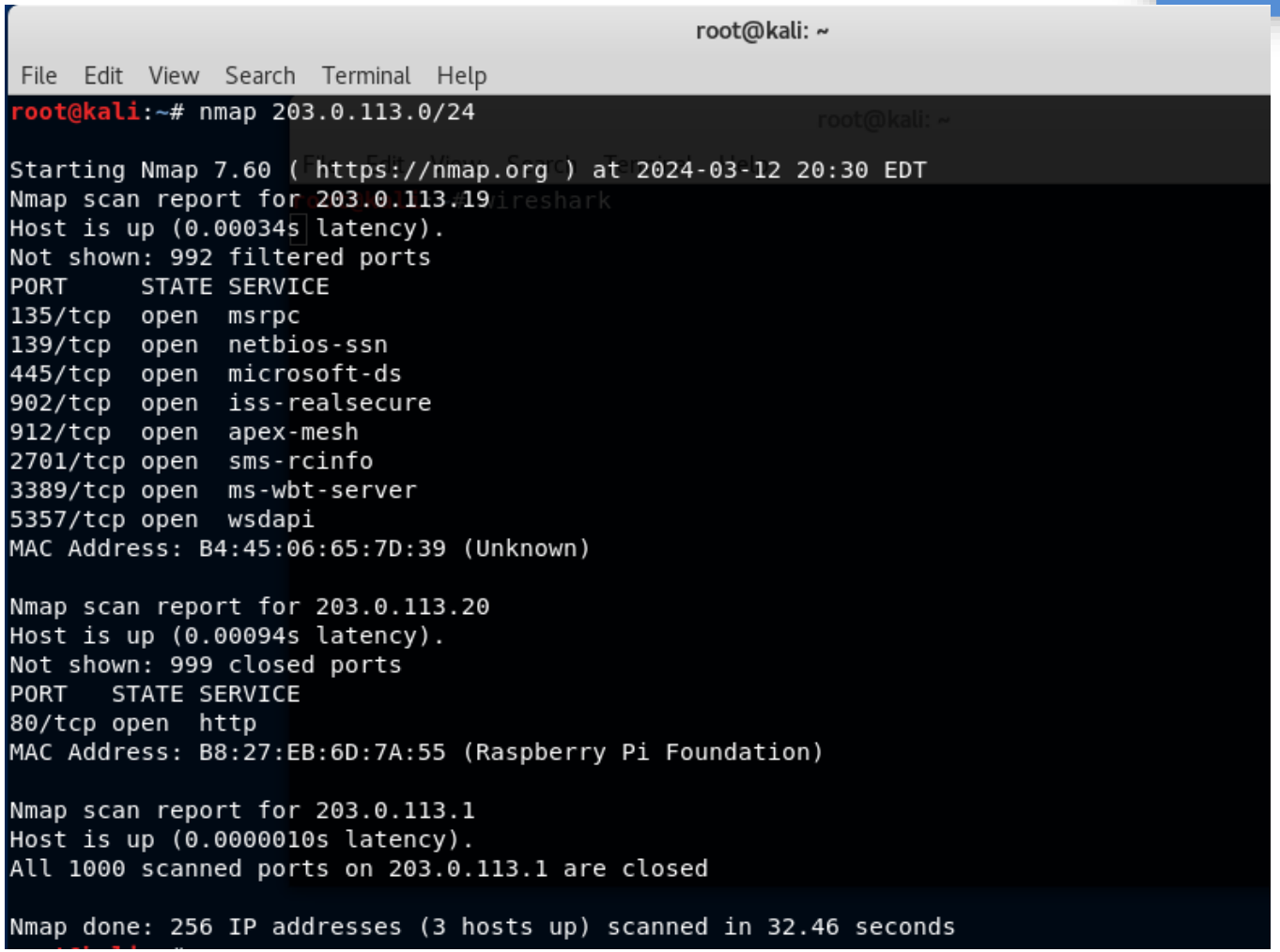
ip.src == 203.0.113.1 && ip.dst == 203.0.113.20



Part 2: Compare a Nmap TCP Default Scan and Full Scan-Cont.

Step 2: Nmap TCP default scan

a. In the terminal, enter the following command to start the Nmap scanning.
root@kali:~# nmap 203.0.113.0/24



Part 2: Compare a Nmap TCP Default Scan and Full Scan-Cont.

After Nmap reports the result of the scan, stop the Wireshark capture. Click the arrow next to the display filter field to filter the results of the scan. Review the Wireshark output. Which TCP flag is Nmap using to discover the open ports?

Review the Wireshark output. Which TCP flag is Nmap using to discover the open ports?

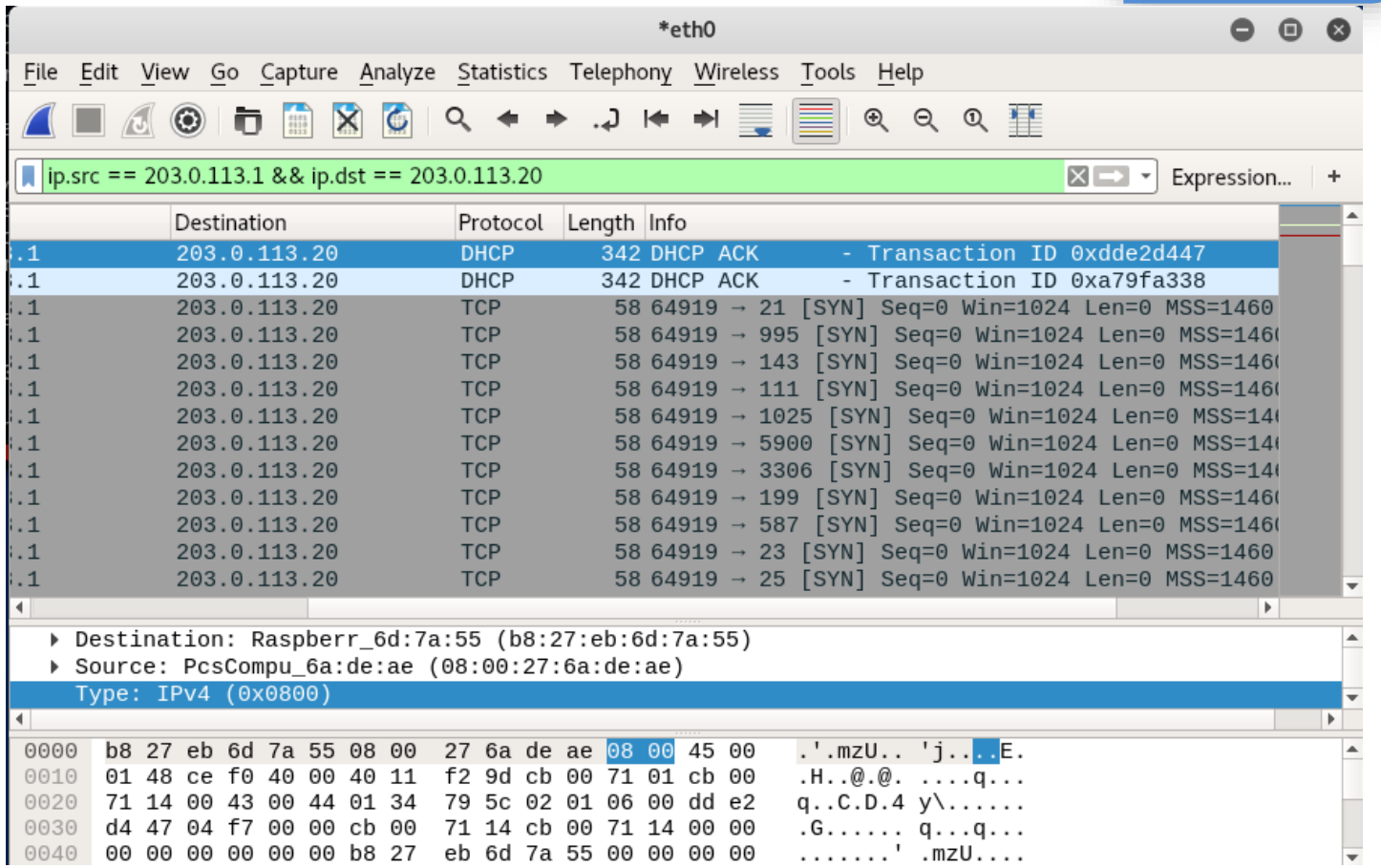
SYN

Notice the ports that are being tested. The default Nmap scan does not test all ports. How many ports does a default Nmap scan test? (Do a web search if necessary.)

Only 1,000 of the most common ports.

Look at the results of the nmap scan in the terminal. What ports are identified?

Answers may vary. Ports 22 and 80 should be open, but others may also be found.



Part 2: Compare a Nmap TCP Default Scan and Full Scan-Cont.

Step 3: Nmap TCP full TCP port scan

a. Nmap by default only scan a limited number of TCP ports. We would like to scan all 65535 TCP ports.

b. Start a new Wireshark capture. Click **Continue without Saving** when prompted to save the capture. to start a new capture.

c. Enter the nmap command to scan all the TCP ports in Kali VM.

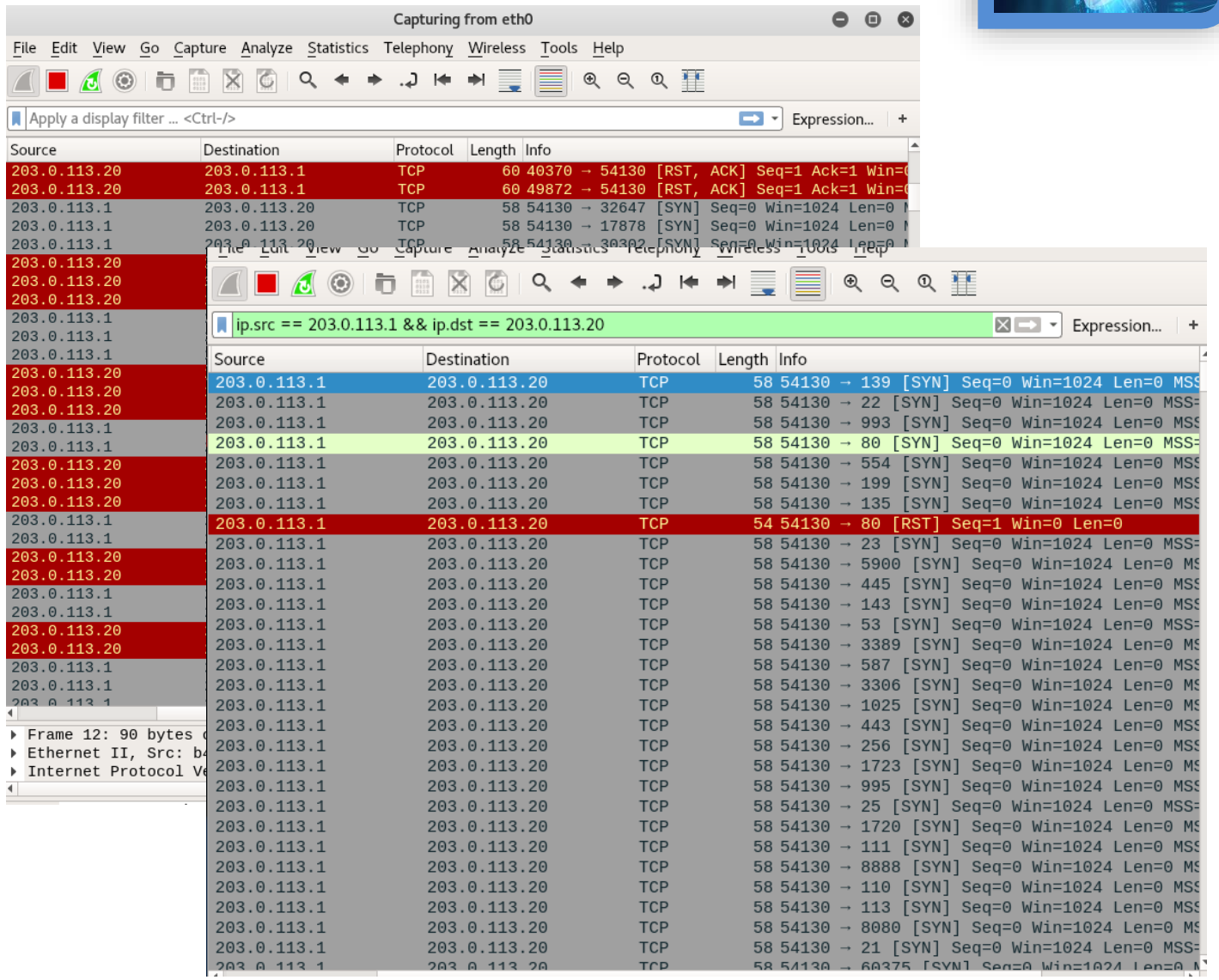
root@kali:~# nmap -p 1-65535 203.0.113.20

```
root@kali:~# nmap -p 1-65535 203.0.113.20

Starting Nmap 7.60 ( https://nmap.org ) at 2024-03-12 20:38 EDT
Nmap scan report for 203.0.113.20
Host is up (0.00067s latency).
Not shown: 65534 closed ports
PORT   STATE SERVICE
80/tcp open  http
MAC Address: B8:27:EB:6D:7A:55 (Raspberry Pi Foundation)

Nmap done: 1 IP address (1 host up) scanned in 29.45 seconds
root@kali:~#
```

Part 2: Compare a Nmap TCP Default Scan and Full Scan-Cont.

d. Watch the Wireshark capture screen. Notice you are sending TCP packets just as before, but the number of ports being scanned has increased.

e. Stop the Wireshark capture when finished and clear the filter.



Part 3: Perform a Nmap UDP scan-Cont.

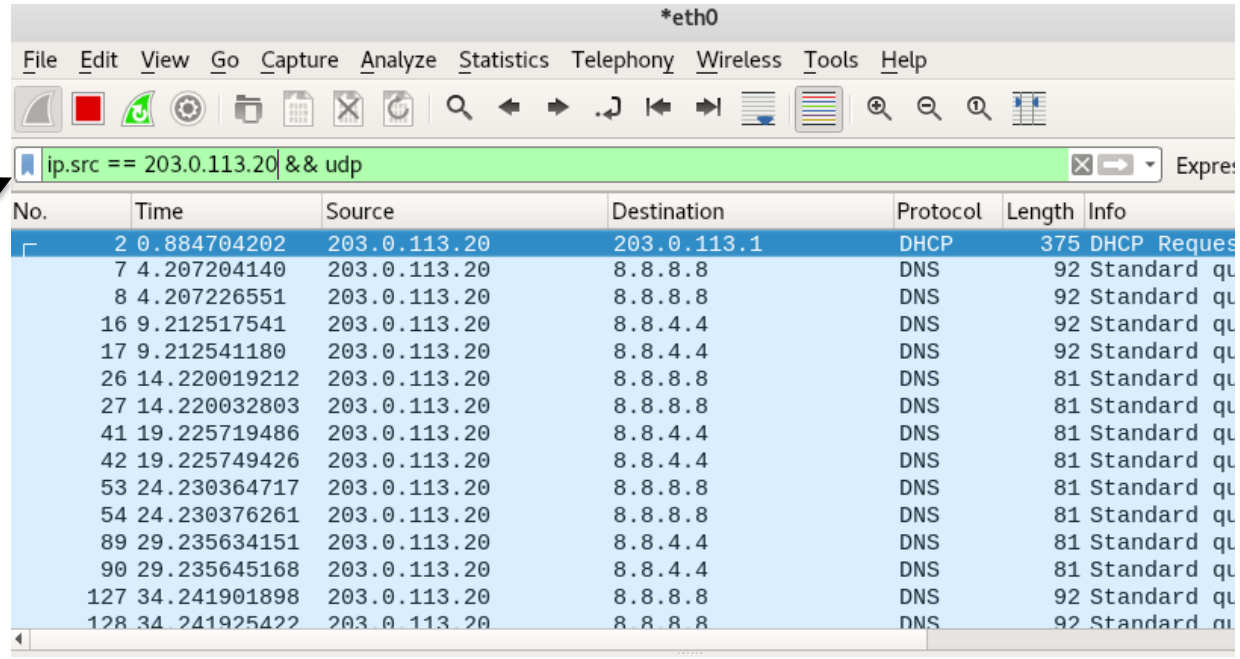
UDP Header Format

Source Port	Destination Port
Length	Checksum

Step 1: UDP scan with a new Wireshark filter.

a. Start a new Wireshark capture. Click Continue without Saving when prompted to save the capture.

b. Apply the following filter in Wireshark:
ip.src == 203.0.113.20 && udp



This will allow us to see the UDP traffic generated by nmap to the Raspberry PI. Notice in the UDP header above there are no flags. We can only send UDP packets and receive a possible “port unreachable or destination unreachable” message meaning the port is closed.

Part 3: Perform a Nmap UDP scan-Cont.

c. In the Kali VM terminal, enter the command nmap with -sU option scanning the IP address of your Raspberry Pi.

root@kail:~# nmap -sU -F 203.0.113.13

```
root@kali:~# nmap -sU -F 203.0.113.20
Starting Nmap 7.60 ( https://nmap.org ) at 2024-03-12 20:52 EDT
Nmap scan report for 203.0.113.20
Host is up (0.0016s latency).
Not shown: 98 closed ports
PORT     STATE         SERVICE
68/udp   open|filtered dhcpc
5353/udp open|filtered zeroconf
MAC Address: B8:27:EB:6D:7A:55 (Raspberry Pi Foundation)

Nmap done: 1 IP address (1 host up) scanned in 114.96 seconds
```

d. It will take a few minutes, but look at the Wireshark capture. As you scroll down the packets, you will see some "Destination unreachable" on different ports indicating the port is closed.

	Source	Destination	Protocol	Length	Info
10	203.0.113.20	203.0.113.1	ICMP	70	Destination unreachable (Port unreachabl
84	203.0.113.20	203.0.113.1	ICMP	94	Destination unreachable (Port unreachabl
13	203.0.113.20	8.8.8.8	DNS	81	Standard query 0x5b97 A 3.debian.pool.nt
43	203.0.113.20	8.8.8.8	DNS	81	Standard query 0x6797 AAAA 3.debian.pool
62	203.0.113.20	203.0.113.1	ICMP	70	Destination unreachable (Port unreachabl
22	203.0.113.20	203.0.113.1	ICMP	70	Destination unreachable (Port unreachabl
62	203.0.113.20	203.0.113.1	ICMP	70	Destination unreachable (Port unreachabl
60	203.0.113.20	203.0.113.1	ICMP	70	Destination unreachable (Port unreachabl

Part 3: Perform a Nmap UDP scan-Cont.

How many UDP ports are there? How many UDP ports does Nmap scan, by default? (Use web search as necessary.)

_____

There are 65,535 UDP ports. Nmap scans only the most commonly used 1,000.

What are the UDP ports that are open from the scan? What protocols use these UDP ports?

_____

_____

Answers may vary. Ports 68 and 5353 are most likely. The bootp and Multicast DNS use these ports. These services are used by PL-App.

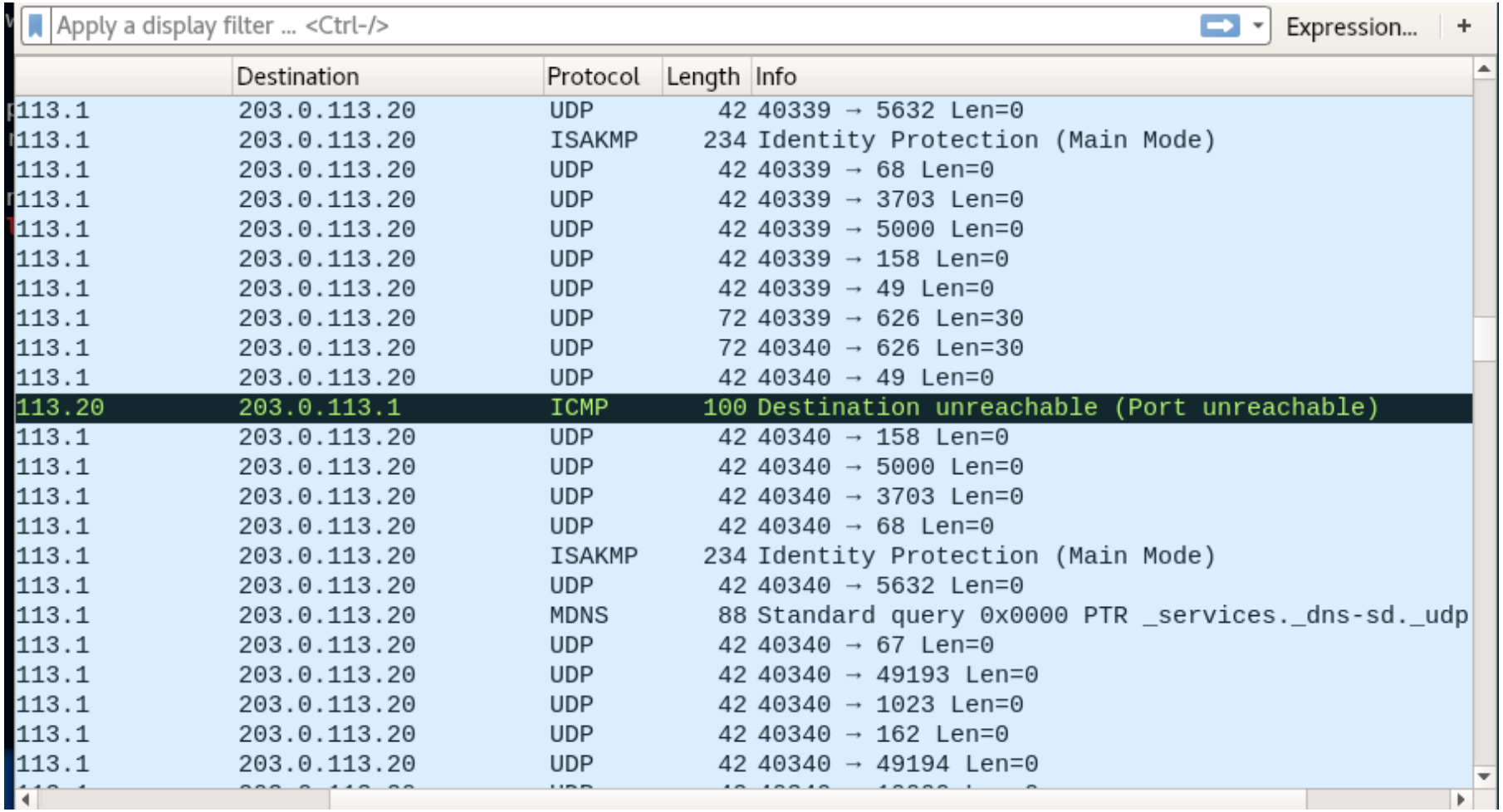
What is the meaning of the -F option? Try the same scan without the -F. What is the difference?

_____

_____

The -F option stands for fast. Only the most common 100 ports are scanned. This makes the scan much faster.

Part 3: Perform a Nmap UDP scan-Cont.



	Destination	Protocol	Length	Info
113.1	203.0.113.20	UDP	42	40339 → 5632 Len=0
113.1	203.0.113.20	ISAKMP	234	Identity Protection (Main Mode)
113.1	203.0.113.20	UDP	42	40339 → 68 Len=0
113.1	203.0.113.20	UDP	42	40339 → 3703 Len=0
113.1	203.0.113.20	UDP	42	40339 → 5000 Len=0
113.1	203.0.113.20	UDP	42	40339 → 158 Len=0
113.1	203.0.113.20	UDP	42	40339 → 49 Len=0
113.1	203.0.113.20	UDP	72	40339 → 626 Len=30
113.1	203.0.113.20	UDP	72	40340 → 626 Len=30
113.1	203.0.113.20	UDP	42	40340 → 49 Len=0
113.20	203.0.113.1	ICMP	100	Destination unreachable (Port unreachable)
113.1	203.0.113.20	UDP	42	40340 → 158 Len=0
113.1	203.0.113.20	UDP	42	40340 → 5000 Len=0
113.1	203.0.113.20	UDP	42	40340 → 3703 Len=0
113.1	203.0.113.20	UDP	42	40340 → 68 Len=0
113.1	203.0.113.20	ISAKMP	234	Identity Protection (Main Mode)
113.1	203.0.113.20	UDP	42	40340 → 5632 Len=0
113.1	203.0.113.20	MDNS	88	Standard query 0x0000 PTR _services._dns-sd._udp
113.1	203.0.113.20	UDP	42	40340 → 67 Len=0
113.1	203.0.113.20	UDP	42	40340 → 49193 Len=0
113.1	203.0.113.20	UDP	42	40340 → 1023 Len=0
113.1	203.0.113.20	UDP	42	40340 → 162 Len=0
113.1	203.0.113.20	UDP	42	40340 → 49194 Len=0

e. Stop the Wireshark capture when finished and clear the filter.

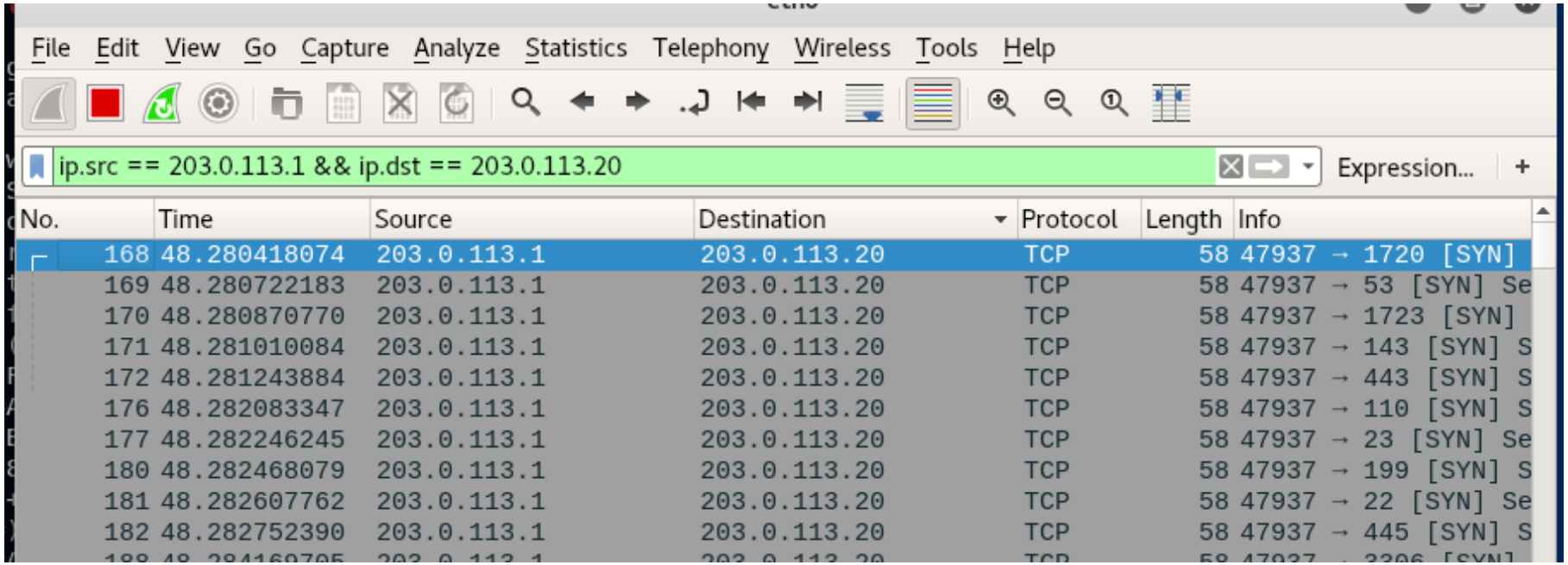
Part 4: Perform Nmap OS and Service Foot Printing

Step 1: Use Nmap to find a device operating system.

a. Start a new Wireshark capture. Click Continue without Saving when prompted to save the capture.

b. Apply the following filter in Wireshark using the IP address of Kali VM as the source address and the IP address of your Raspberry Pi as the destination address.

ip.src == 203.0.113.1 && ip.dst == 203.0.113.20/28



Part 4: Perform Nmap OS and Service Foot Printing-cont

c. In the Kali VM terminal, enter the command nmap using the -O option scanning the IP address of your Raspberry Pi.

root@kali:~# nmap -0 203.0.113.20

What operating system did Nmap guess?

```
root@kali:~# nmap -O 203.0.113.20
Starting Nmap 7.60 ( https://nmap.org ) at 2024-03-12 21:01 EDT
Nmap scan report for 203.0.113.20
Host is up (0.0016s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
80/tcp open  http
MAC Address: B8:27:EB:6D:7A:55 (Raspberry Pi Foundation)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.60%E=4%D=3/12%OT=80%CT=1%CU=38585%PV=N%DS=1%DC=D%G=Y%M=B827EB%T
OS:M=65F0FAF3%P=x86_64-pc-linux-gnu)SEQ(SP=105%GCD=1%ISR=108%TI=Z%CI=Z%II=I
OS:%TS=A)OPS(O1=M5B4ST11NW7%O2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7%O
OS:5=M5B4ST11NW7%O6=M5B4ST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6
OS:=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M5B4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O
OS:%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=
OS:0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%
OS:S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(
OS:R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=
OS:N%T=40%CD=S)

Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.99 seconds
```

Part 4: Perform Nmap OS and Service Foot Printing-cont

Look at the packets in Wireshark. What protocols were used to determine the OS?

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.src == 203.0.113.1 && ip.dst == 203.0.113.20 Expression... +

Destination	Protocol	Length	Info
203.0.113.20	TCP	70	[TCP Port numbers reused] 39532 → 80 [SYN]
203.0.113.20	TCP	54	39532 → 80 [RST] Seq=1 Win=0 Len=0
203.0.113.20	ICMP	162	Echo (ping) request id=0x95e2, seq=295/99
203.0.113.20	ICMP	192	Echo (ping) request id=0x95e3, seq=296/10
203.0.113.20	UDP	342	39465 → 38585 Len=300
203.0.113.20	TCP	66	[TCP Port numbers reused] 39539 → 80 [SYN,
203.0.113.20	TCP	54	39539 → 80 [RST] Seq=1 Win=0 Len=0
203.0.113.20	TCP	74	39541 → 80 [<None>] Seq=2295984856 Win=131
203.0.113.20	TCP	74	[TCP Port numbers reused] 39542 → 80 [FIN,
203.0.113.20	TCP	74	39543 → 80 [ACK] Seq=2295984856 Ack=415449
203.0.113.20	TCP	74	[TCP Port numbers reused] 39544 → 1 [SYN]
203.0.113.20	TCP	74	39545 → 1 [ACK] Seq=2295984856 Ack=4154496
203.0.113.20	TCP	74	[TCP Retransmission] 39546 → 1 [FIN, PSH,
203.0.113.20	TCP	74	39541 → 80 [<None>] Seq=2295984856 Win=131
203.0.113.20	TCP	74	[TCP Retransmission] 39542 → 80 [FIN, SYN,
203.0.113.20	TCP	74	39541 → 80 [<None>] Seq=2295984856 Win=131
203.0.113.20	TCP	74	[TCP Retransmission] 39542 → 80 [FIN, SYN,
203.0.113.20	TCP	74	39541 → 80 [<None>] Seq=2295984856 Win=131
203.0.113.20	TCP	74	[TCP Retransmission] 39542 → 80 [FIN, SYN,
203.0.113.20	TCP	74	[TCP Port numbers reused] 39527 → 80 [SYN]
203.0.113.20	TCP	54	39527 → 80 [RST] Seq=1 Win=0 Len=0
203.0.113.20	TCP	74	[TCP Port numbers reused] 39528 → 80 [SYN]
203.0.113.20	TCP	54	39528 → 80 [RST] Seq=1 Win=0 Len=0

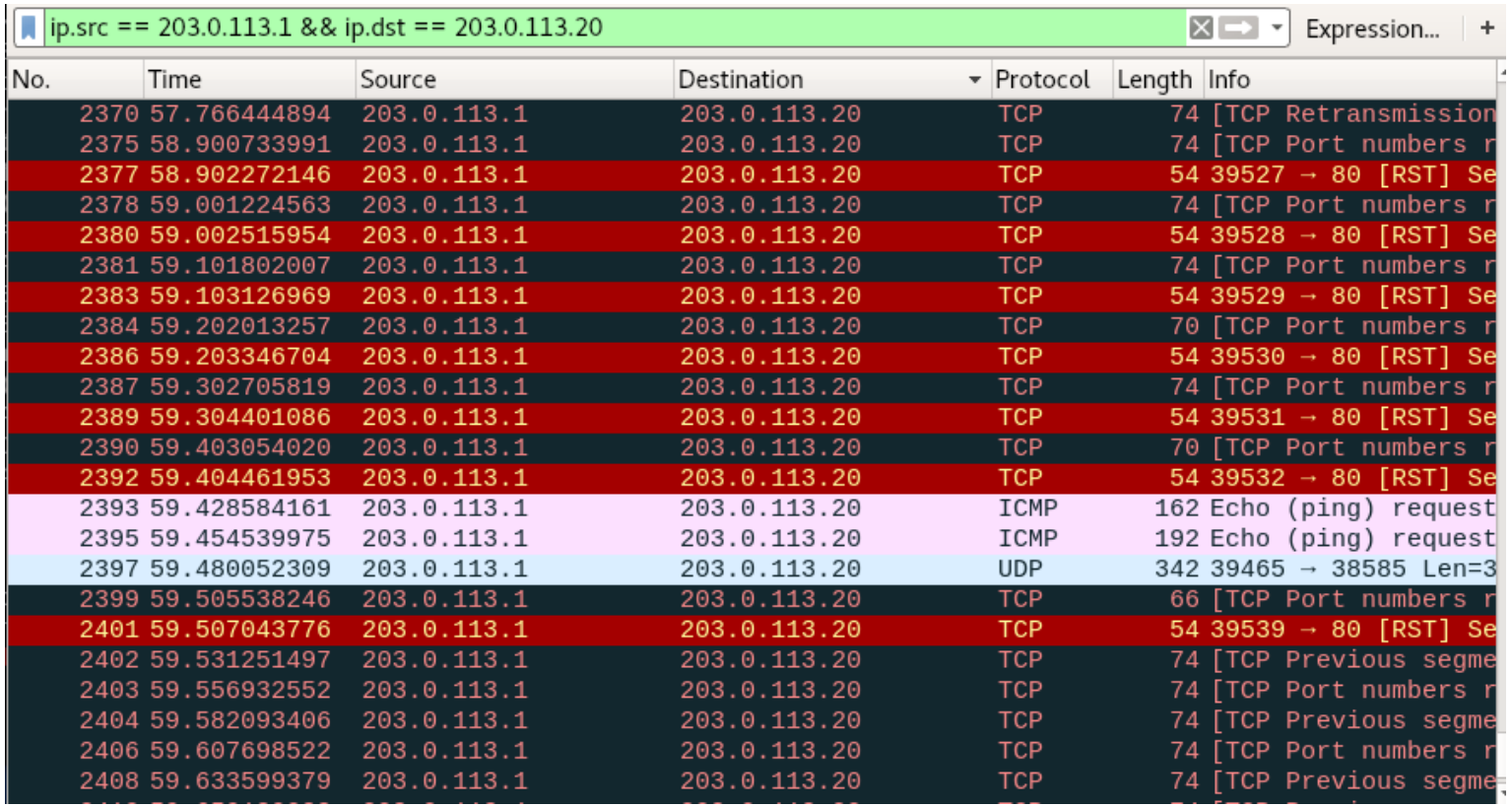
ip.src == 203.0.113.1 && ip.dst == 203.0.113.20 Expression... +

Destination	Protocol	Length	Info
203.0.113.20	TCP	74	[TCP Retransmission] 39542 → 80 [FIN, SYN, PSH, URG] S
203.0.113.20	TCP	74	[TCP Port numbers reused] 39527 → 80 [SYN] Seq=0 Win=1
203.0.113.20	TCP	54	39527 → 80 [RST] Seq=1 Win=0 Len=0
203.0.113.20	TCP	74	[TCP Port numbers reused] 39528 → 80 [SYN] Seq=0 Win=6
203.0.113.20	TCP	54	39528 → 80 [RST] Seq=1 Win=0 Len=0
203.0.113.20	TCP	74	[TCP Port numbers reused] 39529 → 80 [SYN] Seq=0 Win=4
203.0.113.20	TCP	54	39529 → 80 [RST] Seq=1 Win=0 Len=0
203.0.113.20	TCP	70	[TCP Port numbers reused] 39530 → 80 [SYN] Seq=0 Win=4
203.0.113.20	TCP	54	39530 → 80 [RST] Seq=1 Win=0 Len=0
203.0.113.20	TCP	74	[TCP Port numbers reused] 39531 → 80 [SYN] Seq=0 Win=1
203.0.113.20	TCP	54	39531 → 80 [RST] Seq=1 Win=0 Len=0
203.0.113.20	TCP	70	[TCP Port numbers reused] 39532 → 80 [SYN] Seq=0 Win=5
203.0.113.20	TCP	54	39532 → 80 [RST] Seq=1 Win=0 Len=0
203.0.113.20	ICMP	162	Echo (ping) request id=0x08c8, seq=295/9985, ttl=58 (
203.0.113.20	ICMP	192	Echo (ping) request id=0x08c9, seq=296/10241, ttl=58
203.0.113.20	UDP	342	39465 → 38585 Len=300
203.0.113.20	TCP	66	[TCP Port numbers reused] 39539 → 80 [SYN, ECN, CWR, R
203.0.113.20	TCP	54	39539 → 80 [RST] Seq=1 Win=0 Len=0
203.0.113.20	TCP	74	[TCP Previous segment not captured] 39541 → 80 [<None>
203.0.113.20	TCP	74	[TCP Port numbers reused] 39542 → 80 [FIN, SYN, PSH, U
203.0.113.20	TCP	74	[TCP Previous segment not captured] 39543 → 80 [ACK] S
203.0.113.20	TCP	74	[TCP Port numbers reused] 39544 → 1 [SYN] Seq=0 Win=31
203.0.113.20	TCP	74	[TCP Previous segment not captured] 39545 → 1 [ACK] Se

Part 4: Perform Nmap OS and Service Foot Printing-Cont

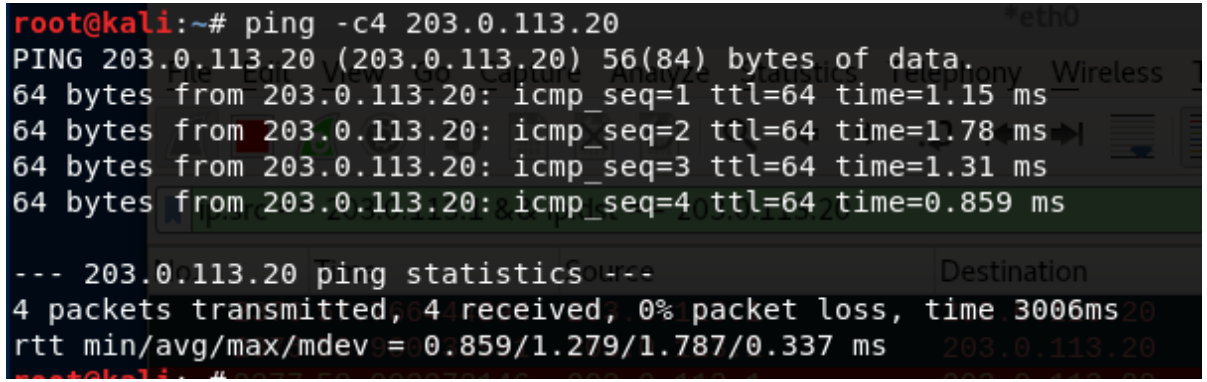
d. We can sometimes identify a device by looking at the time to live (TTL) field of a local ping response, which varies by device OS. See the table below:

Operating System	TTL Response time
Cisco	255
Windows	128
Linux	64



Part 4: Perform Nmap OS and Service Foot Printing-cont

In the Kali VM terminal, enter the command to ping your Raspberry Pi with 4 ICMP packets.
root@kali:~# ping -c4 203.0.113.20



What is the response time? Is it consistent with the Nmap identification?

The response time is 64 which is consistent with a Linux OS response.

e. Stop the Wireshark capture when finished.

Part 4: Perform Nmap OS and Service Foot Printing-Cont

Step 2: Use Nmap to find services versions

Nmap has a built-in database of about 2,200 well-known services that is used to help identify application service versions.

a. Start a new Wireshark capture. Click Continue without Saving when prompted to save the capture. Apply the same display filter as the previous step.
b. In the Kali VM, enter the nmap command with the -A option scanning the IP address of your Raspberry Pi.

root@kali:~# nmap -A 203.0.113.20

```
root@kali:~# nmap -A 203.0.113.20

Starting Nmap 7.60 ( https://nmap.org ) at 2024-03-12 21:15 EDT
Nmap scan report for 203.0.113.20
Host is up (0.0015s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE VERSION
80/tcp open  http    Tornado httpd 6.0.3
| http-robots.txt: 1 disallowed entry
| /
```

Part 4: Perform Nmap OS and Service Foot Printing-cont

What are the identified applications running on the ports? Complete the table.

Port Number	Application service identified
22	SSH-2.0-OpenSSH_7.4p1
80	Tornado httpd 4.5.1
88	nginx 1.10.3
89	Node.js Express framework

Answers may vary, but ports 22 and 80 should at least be open.

```
root@kali:~# nmap -A 203.0.113.20

Starting Nmap 7.60 ( https://nmap.org ) at 2024-03-12 21:15 EDT
Nmap scan report for 203.0.113.20
Host is up (0.0015s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE VERSION
80/tcp open  http    Tornado httpd 6.0.3
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: TornadoServer/6.0.3
| http-title: Jupyter Notebook
|_Requested resource was /login?next=%2Ftree%3F
MAC Address: B8:27:EB:6D:7A:55 (Raspberry Pi Foundation)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/
TCP/IP fingerprint:
OS:SCAN(V=7.60%E=4%D=3/12%OT=80%CT=1%CU=43066%PV=N%DS=1%DC=D%G=Y%M=B827EB%T
OS:M=65F0FE47%P=x86_64-pc-linux-gnu)SEQ(SP=107%GCD=1%ISR=10C%TI=Z%CI=Z%II=I
OS:%TS=A)OPS(O1=M5B4ST11NW7%O2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7%O
OS:5=M5B4ST11NW7%O6=M5B4ST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6
OS:=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M5B4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O
OS:%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=
OS:0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%
OS:S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(
OS:R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=
OS:N%T=40%CD=S)

Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   1.54 ms 203.0.113.20

OS and Service detection performed. Please report any incorrect results at https://nmap.org/subm
Nmap done: 1 IP address (1 host up) scanned in 32.11 seconds
```

Part 4: Perform Nmap OS and Service Foot Printing

What is different in the port identification from the Nmap TCP scan in Part 2 of the lab and this application service scan? What implications does this have for IoT security?

In the normal TCP scan, only protocols are identified. In this scan, details about the running services are also shown. These details can be used to identify versions of software services that have well-known vulnerabilities.

What additional functions does this scan also perform?

Nmap also performs a traceroute, OS detection, and SSH details.

Part 4: Perform Nmap OS and Service Foot Printing

	Time	Source	Destination	Protocol	Length	Info
3835	213.198317868	203.0.113.1	203.0.113.20	TCP	58	54138 → 3351 [SYN] Seq=0
3836	213.198462745	203.0.113.1	203.0.113.20	TCP	58	54138 → 1805 [SYN] Seq=0
3837	213.198592697	203.0.113.20	203.0.113.1	TCP	60	1093 → 54138 [RST, ACK]
3838	213.198596354	203.0.113.20				
3839	213.198658131	203.0.113.1				
3840	213.198798868	203.0.113.1				
3841	213.199036833	203.0.113.20				
3842	213.199039155	203.0.113.20				
3843	213.199039889	203.0.113.20				
3844	213.199101400	203.0.113.1				
3845	213.199238389	203.0.113.1				
3846	213.199381729	203.0.113.1				
3847	213.199485204	203.0.113.20				
3848	213.199487616	203.0.113.20				
3849	213.199488431	203.0.113.20				
3850	213.199547265	203.0.113.1				
3851	213.199684284	203.0.113.1				
3852	213.199840838	203.0.113.1				
3853	213.199944988	203.0.113.20				
3854	213.199947705	203.0.113.20				
3855	213.199949049	203.0.113.20				
3856	213.200007783	203.0.113.1				
3857	213.200227273	203.0.113.1				

	Source	Destination	Protocol	Length	Info
491694292	203.0.113.1	203.0.113.20	TCP	74	[TCP Port numbers reused] 36951
493327608	203.0.113.20	203.0.113.1	TCP	74	80 → 36951 [SYN, ACK] Seq=0 Ack=1
493354777	203.0.113.1	203.0.113.20	TCP	54	36951 → 80 [RST] Seq=1 Win=0 Len=
527777912	203.0.113.19	224.0.0.251	MDNS	80	Standard query 0x0000 A arifRPi1.
596127488	203.0.113.1	203.0.113.20	TCP	74	[TCP Port numbers reused] 36952
597157836	203.0.113.20	203.0.113.1	TCP	74	80 → 36952 [SYN, ACK] Seq=0 Ack=1
597181143	203.0.113.1	203.0.113.20	TCP	54	36952 → 80 [RST] Seq=1 Win=0 Len=
598523573	203.0.113.19	20.189.173.2	TCP	66	57216 → 443 [SYN] Seq=0 Win=64240
598731481	203.0.113.19	8.8.8.8	DNS	91	Standard query 0xa7f7 AAAA teams.
596625539	203.0.113.1	203.0.113.20	TCP	74	[TCP Port numbers reused] 36953
598279808	203.0.113.20	203.0.113.1	TCP	74	80 → 36953 [SYN, ACK] Seq=0 Ack=1
598312774	203.0.113.1	203.0.113.20	TCP	54	36953 → 80 [RST] Seq=1 Win=0 Len=
797259177	203.0.113.1	203.0.113.20	TCP	70	[TCP Port numbers reused] 36954
798890979	203.0.113.20	203.0.113.1	TCP	74	80 → 36954 [SYN, ACK] Seq=0 Ack=1
798922745	203.0.113.1	203.0.113.20	TCP	54	36954 → 80 [RST] Seq=1 Win=0 Len=
898388265	203.0.113.1	203.0.113.20	TCP	74	[TCP Port numbers reused] 36955
900032986	203.0.113.20	203.0.113.1	TCP	74	80 → 36955 [SYN, ACK] Seq=0 Ack=1
900066332	203.0.113.1	203.0.113.20	TCP	54	36955 → 80 [RST] Seq=1 Win=0 Len=
998860401	203.0.113.1	203.0.113.20	TCP	70	[TCP Port numbers reused] 36956
999655087	203.0.113.20	203.0.113.1	TCP	70	80 → 36956 [SYN, ACK] Seq=0 Ack=1
999672788	203.0.113.1	203.0.113.20	TCP	54	36956 → 80 [RST] Seq=1 Win=0 Len=
924640944	203.0.113.1	203.0.113.20	ICMP	162	Echo (ping) request id=0xcba3, s
926049039	203.0.113.20	203.0.113.1	ICMP	162	Echo (ping) reply id=0xcba3, s

c. Stop the Wireshark capture when finished.

References

1. What is a computer port? | Ports in networking: https://www.cloudflare.com/learning/network-layer/what-is-a-computer-port/
2. Oracle Agile Engineering Data Management Security Guide: https://docs.oracle.com/cd/E89228_03/otn/pdf/install/html_edmsc/output/chapter_6.htm
3. Computer Ports and Cables

https://elluminetpress.com/2017/08/computer-ports-and-cables/

4. Port Scanning an IoT Device

https://www.youtube.com/watch?v=1XisyWvY0SY


Thank
You
köszönöm
Kiitos
Maake
Asante
Shukria
Dhanyavadagalu
Vinaka
Dankscheen
спасибо
Kam Sah Hammida
شكرا
Manana
Dankon
감사합니다
Dank Je
Mauruuru
Biyan
Matondo
Terima Kasih
Taiku
Tack
Blagodaram
Ngiyabonga
Dziekuje
Chokrane
Diolch i Chi
Grazie
Juspaxar
Arigato
Mochchakkeram
Gracias
Khap
Tingki
நன்றி
Bedankt
Dakujem
धन्यवाद
cảm ơn bạn
Paldies
Gratias Tibi
Ua Tsaug Rau Koj
Grazas
Obrigado
Děkuji
Nirringrazzjak
Di Ou Mèsi
Kia Ora
Kop Khun
ありがとう
Suksama
Hvala
Rahmat
Nuwun
Matur
谢
Welalin
Djiere Dieuf
Misaotra
Danke
Merci
Go Raibh Maith Agat
Eskerrik Asko
ขอขอบคุณคุณ
Salamat
Najis Tuke