COMPENG 4DN4 Lab 1 Report

Aaron Pinto
pintoa9

Raeed Hassan
hassam41

February 12, 2023

TCP: WireShark

The connection setup packets are shown in Figure 1. We can see that the connection sequence begins with the 3-way handshake. The source/receiving computer sends a SYN packet to the `compeng4dn4.mooo.com` server, with a source port of 25078, destination port of 50008, window size of 64240 bytes, and max segment size of 1460 bytes. The length is 0 as this is a SYN packet so no data is transmitted in this stage. The server responds to the source computer with a packet with the SYN and ACK bits set, with the same windows and max segment sizes. Finally, the receiving computer sends an ACK packet back to the server with the sequence number incremented by 1. The sequence number is incremented by 1 as the length of the packet was 0.

Once the handshake is completed successfully, we can see that the data transfer starts after the GET request packet. The server acknowledges this and starts sending the image data in another packet of length 1460 bytes. It then sends another packet of the same length with the PUSH flag set, which signals to the receiving computer that it must not wait for more data from the sending TCP device before passing the data to the receiving process. There is a coupling between the push function and the use of buffers of data that cross the TCP/user interface. Each time a PUSH flag is associated with data placed into the receiving user's buffer, the buffer is returned to the user for processing even if the buffer is not filled. If data arrives that fills the user's buffer before a PUSH is seen, the data is passed to the user in buffer size units.

This acknowledgement and data transfer sequence continues in packets 171, 172, 176, 177, 181, 182, 186, 187, 191, 192, 196, 197, 201, 202, 207, 208, 212, 213, 214, 218, 222, 223, 227, 228, 232, 233, and 237 until the server returns an HTTP 200 OK code in 237, which signals that the GET request has been completed successfully. Each time a data packet is transmitted by the server, the sequence number of the server packet increases by the length of the previous packet. For example, from packet 212 to 213, the sequence number increased from 23361 to 23361 + 1460 = 24821. When the receiving computer receives this packet, it responds with a packet that contains the SEQ number of the requested ACK value, and it sets its own ACK value to the previous packet SEQ number + the length of the previous packet. For example, from packet 214 to 215, the sequence number went from 26281 to 560, which is what the server requested as the ACK, and the ACK went from 560 to 26281 + 1460 = 27741. This is shown in Figure 2. The server then sends a FIN-ACK packet, which means that it received the previous packet and wants to close the connection. We do not see the source computer send a FIN-ACK back because that only happens when you close the browser tab with the image.

The image that was downloaded from the server was `351.jpeg` and can be seen in Figure 3.

Figure 1: WireShark Packet Capture 1st Half

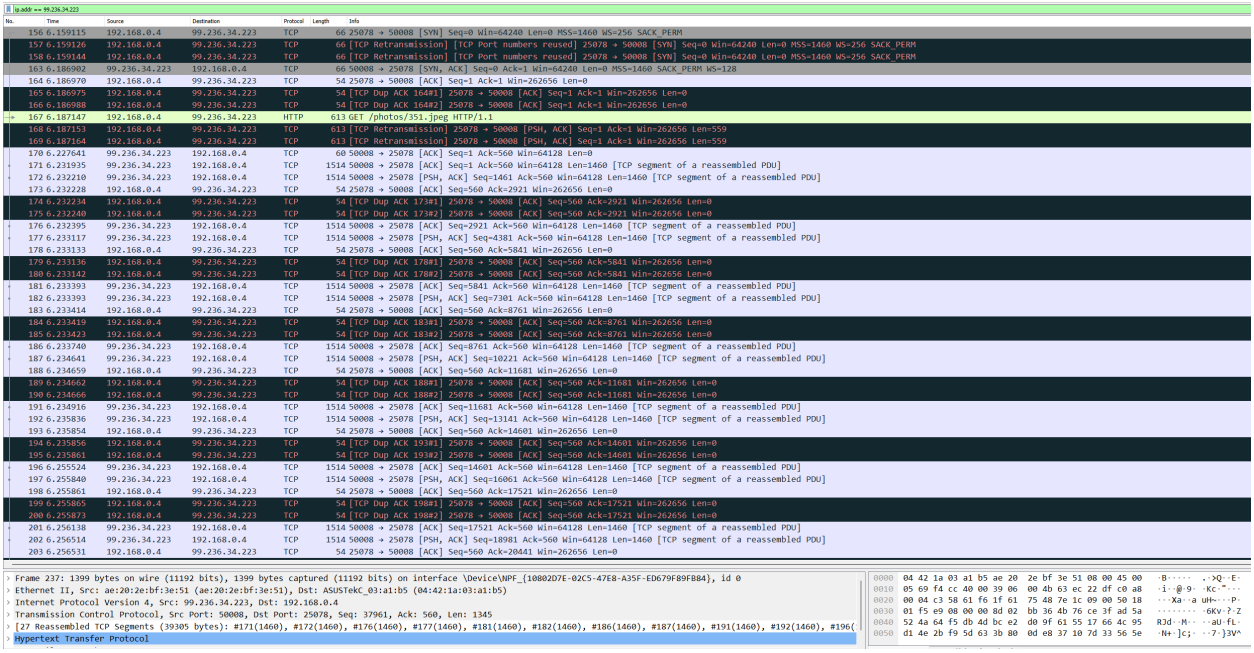


Figure 2: WireShark Packet Capture 2nd Half

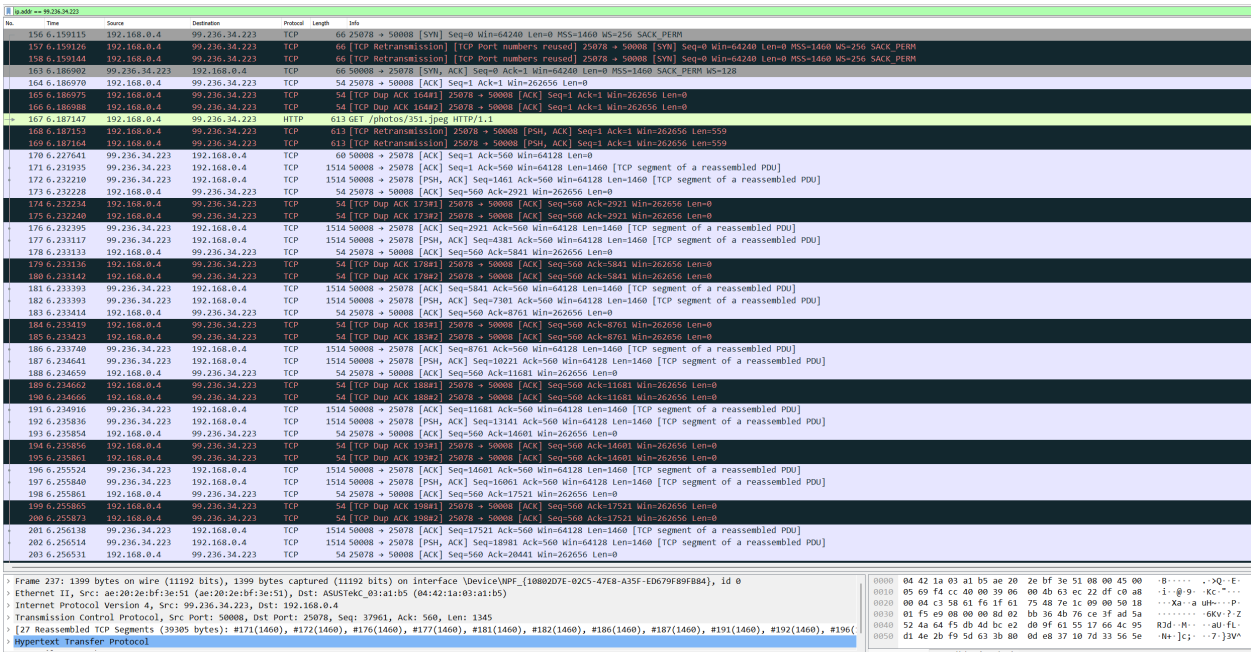


Figure 3: http://compeng4dn4.mooo.com:50008/photos/351.jpeg

TCP: Tcpdump

A tcpdump packet capture was started and the scan written to a WireShark compatible pcap file for analysis. The tcpdump terminal output is shown in Listing 1. The Ncat terminal output where the group member's names and MAC ID numbers were echoed is shown below in Listing 2.

Listing 1: Tcpdump Packet Capture of Python Echo Server Connection

```
sudo tcpdump -nnvvX -i 1 -S host compeng4dn4.mooo.com -w mooo_echo.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
21 packets captured
88 packets received by filter
0 packets dropped by kernel
```

Listing 2: Ncat Connection to Python Echo Server

```
ncat --crlf compeng4dn4.mooo.com 50007
Wecome to COMPENG 4DN4 Echo Server!
Raeed Hassan
Raeed Hassan
400188200
400188200
Aaron Pinto
Aaron Pinto
400190637
400190637
```

The capture displayed in WireShark is shown below in Figure 4. The first three packets (1-3) contain the three way handshake which establishes the TCP connection. The next two packets are (4-5) shows that the server is letting the host know that it is sending some data in packet 4 (the message "Wecome to COMPENG 4DN4 Echo Server!") while acknowledging the last signal with a [PSH, ACK] signal, and that the host acknowledges this in packet 5.

Packets 6-9 are for the first message sent to the server ("Raeed Hassan") which is echoed back to the host. The host sends the message in packet 6, with a [PSH, ACK] signal as discussed earlier, which is then acknowledged by the server in packet 7. The server then does the same when echoing the message back, with a [PSH, ACK] signal in packet 8, which is acknowledged by the host in packet 9.

For the next three messages, the ACK signal from the server (such as in packet 7) is dropped and instead only 3 packets are sent, a [PSH, ACK] signal from the host to the server when sending the message, a [PSH, ACK] signal from the server to the host when it echoes back the signal, and an ACK signal from the host acknowledging the previous signal. Packets 10-12 are for the next message sent to the server ("400188200") which is echoed back to the host. Packets 13-15 are for the next message sent to the server ("Aaron Pinto") which is echoed back to the host. Packets 16-18 are for the next message sent to the server ("400190637") which is echoed back to the host.

The last three packets (19-21) close the TCP connection. The host sending sends a FIN signal in packet 19 to indicate it wishes to finish the TCP connection, which the server acknowledges in packet 20 and sends back it's own FIN signal. The connection is finished as the host acknowledges the server's signal in packet 21.

Figure 4: Tcpdump Capture of Python Echo Server in WireShark

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.2.55	99.236.34.223	TCP	74	55182 → 50007 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2269558917 TSecr=0 WS=128
2	0.025617	99.236.34.223	192.168.2.55	TCP	74	50007 → 55182 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1452 SACK_PERM TSval=389693670 TSecr=2269558917 WS=128
3	0.025660	192.168.2.55	99.236.34.223	TCP	66	55182 → 50007 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2269558943 TSecr=389693670
4	0.049202	99.236.34.223	192.168.2.55	TCP	103	50007 → 55182 [PSH, ACK] Seq=1 Ack=1 Win=65280 Len=37 TSval=389693696 TSecr=2269558943
5	0.049245	192.168.2.55	99.236.34.223	TCP	66	55182 → 50007 [ACK] Seq=1 Ack=38 Win=64256 Len=0 TSval=2269558966 TSecr=389693696
6	7.351131	192.168.2.55	99.236.34.223	TCP	80	55182 → 50007 [PSH, ACK] Seq=1 Ack=38 Win=64256 Len=14 TSval=2269566268 TSecr=389693696
7	7.375637	99.236.34.223	192.168.2.55	TCP	66	50007 → 55182 [ACK] Seq=38 Ack=15 Win=65280 Len=0 TSval=389701019 TSecr=2269566268
8	7.375638	99.236.34.223	192.168.2.55	TCP	80	50007 → 55182 [PSH, ACK] Seq=38 Ack=15 Win=65280 Len=14 TSval=389701019 TSecr=2269566268
9	7.375676	192.168.2.55	99.236.34.223	TCP	66	55182 → 50007 [ACK] Seq=15 Ack=52 Win=64256 Len=0 TSval=2269566293 TSecr=389701019
10	11.683288	192.168.2.55	99.236.34.223	TCP	77	55182 → 50007 [PSH, ACK] Seq=15 Ack=52 Win=64256 Len=11 TSval=2269570600 TSecr=389701019
11	11.708132	99.236.34.223	192.168.2.55	TCP	77	50007 → 55182 [PSH, ACK] Seq=52 Ack=26 Win=65280 Len=11 TSval=389705352 TSecr=2269570600
12	11.708168	192.168.2.55	99.236.34.223	TCP	66	55182 → 50007 [ACK] Seq=26 Ack=63 Win=64256 Len=0 TSval=2269570625 TSecr=389705352
13	15.369517	192.168.2.55	99.236.34.223	TCP	79	55182 → 50007 [PSH, ACK] Seq=26 Ack=63 Win=64256 Len=13 TSval=2269574286 TSecr=389705352
14	15.395181	99.236.34.223	192.168.2.55	TCP	79	50007 → 55182 [PSH, ACK] Seq=63 Ack=39 Win=65280 Len=13 TSval=389709039 TSecr=2269574286
15	15.395306	192.168.2.55	99.236.34.223	TCP	66	55182 → 50007 [ACK] Seq=39 Ack=76 Win=64256 Len=0 TSval=2269574312 TSecr=389709039
16	19.915362	192.168.2.55	99.236.34.223	TCP	77	55182 → 50007 [PSH, ACK] Seq=39 Ack=76 Win=64256 Len=11 TSval=2269578832 TSecr=389709039
17	19.942417	99.236.34.223	192.168.2.55	TCP	77	50007 → 55182 [PSH, ACK] Seq=76 Ack=50 Win=65280 Len=11 TSval=389713584 TSecr=2269578832
18	19.942472	192.168.2.55	99.236.34.223	TCP	66	55182 → 50007 [ACK] Seq=50 Ack=87 Win=64256 Len=0 TSval=2269578859 TSecr=389713584
19	27.040246	192.168.2.55	99.236.34.223	TCP	66	55182 → 50007 [FIN, ACK] Seq=50 Ack=87 Win=64256 Len=0 TSval=2269585957 TSecr=389713584
20	27.073821	99.236.34.223	192.168.2.55	TCP	66	50007 → 55182 [FIN, ACK] Seq=87 Ack=51 Win=65280 Len=0 TSval=389720710 TSecr=2269585957
21	27.073857	192.168.2.55	99.236.34.223	TCP	66	55182 → 50007 [ACK] Seq=51 Ack=88 Win=64256 Len=0 TSval=2269585991 TSecr=389720710

DNS

The `nslookup` utility was used to do a DNS query on the `compeng4dn4.mooo.com` server. The terminal output of the DNS query is shown in Listing 3.

Listing 3: DNS Query Terminal Output

```
nslookup compeng4dn4.mooo.com
Server:  mynetwork
```

```
Address:  192.168.2.1

Non-authoritative answer:
Name:    compeng4dn4.mooo.com
Address:  99.236.34.223
```

The WireShark capture for the DNS query is shown below in Figure 5. The capture was filtered for DNS queries containing `compeng4dn4.mooo.com`, using the display filter `dns.qry.name contains "compeng4dn4.mooo.com"`.

The DNS query first searches for `compeng4dn4.mooo.com` with the DNS suffix `.home`, performing the query with "A" and "AAAA" records, for mapping hostnames to IPv4 and IPv6 addresses. Both DNS queries return "no such name" indicating that the queries are unsuccessful. The DNS query then queries `compeng4dn4.mooo.com` with no suffix, returning the correct IPv4 address for the A record. The AAAA record does not return any address.

Figure 5: DNS Query WireShark Capture

dns.qry.name contains "compeng4dn4.mooo.com"

No.	Time	Source	Destination	Protocol	Length	Info
202	16.449334	192.168.2.49	192.168.2.1	DNS	85	Standard query 0x0002 A compeng4dn4.mooo.com.home
203	16.454914	192.168.2.1	192.168.2.49	DNS	85	Standard query response 0x0002 No such name A compeng4dn4.mooo.com.home
204	16.455021	192.168.2.49	192.168.2.1	DNS	85	Standard query 0x0003 AAAA compeng4dn4.mooo.com.home
205	16.459511	192.168.2.1	192.168.2.49	DNS	85	Standard query response 0x0003 No such name AAAA compeng4dn4.mooo.com.home
206	16.459602	192.168.2.49	192.168.2.1	DNS	80	Standard query 0x0004 A compeng4dn4.mooo.com
207	16.463819	192.168.2.1	192.168.2.49	DNS	96	Standard query response 0x0004 A compeng4dn4.mooo.com A 99.236.34.223
208	16.465093	192.168.2.49	192.168.2.1	DNS	80	Standard query 0x0005 AAAA compeng4dn4.mooo.com
209	16.469126	192.168.2.1	192.168.2.49	DNS	80	Standard query response 0x0005 AAAA compeng4dn4.mooo.com

Traceroute

The traceroute terminal output is shown below in Listing 4. We can see that the traceroute begins first with a hoop to the local router at 192.168.2.1, then takes 10 hops to route to 24.156.158.102, with all subsequent hops timing out.

Listing 4: Traceroute Terminal Output

```
tracert compeng4dn4.mooo.com

Tracing route to compeng4dn4.mooo.com [99.236.34.223]
over a maximum of 30 hops:

  1     5 ms     5 ms     5 ms  mynetwork [192.168.2.1]
  2    22 ms    15 ms    19 ms  10.11.2.49
  3     *        *        *     Request timed out.
  4    14 ms    21 ms     *     cksnon1673w_lag37.net.bell.ca [142.124.127.44]
  5     9 ms    18 ms    22 ms  cr01-toroonxnhe9-bundle-ether1.net.bell.ca [142.124.127.159]
  6    22 ms    26 ms    24 ms  bx5-torontoxn_ae0.net.bell.ca [64.230.52.229]
  7    29 ms    28 ms    20 ms  rogers_bx5-torontoxn.net.bell.ca [184.150.158.205]
  8    22 ms    20 ms    21 ms  209.148.235.221
  9    23 ms    13 ms    15 ms  3039-dgw01.hstr.rmgt.net.rogers.com [209.148.237.94]
  10    23 ms    33 ms    18 ms  24.156.158.102
  11     *        *        *     Request timed out.
  12     *        *        *     Request timed out.
  13     *        *        *     Request timed out.
  14     *        *        *     Request timed out.
  15     *        *        *     Request timed out.
  16     *        *        *     Request timed out.
  17     *        *        *     Request timed out.
  18     *        *        *     Request timed out.
```

```
 19     *        *        *     Request timed out.
 20     *        *        *     Request timed out.
 21     *        *        *     Request timed out.
 22     *        *        *     Request timed out.
 23     *        *        *     Request timed out.
 24     *        *        *     Request timed out.
 25     *        *        *     Request timed out.
 26     *        *        *     Request timed out.
 27     *        *        *     Request timed out.
 28     *        *        *     Request timed out.
 29     *        *        *     Request timed out.
 30     *        *        *     Request timed out.

Trace complete.
```

The WireShark capture was done with a capture filter of `icmp`. The WireShark capture for traceroute is shown below in Figure 6. We can see that what the traceroute command does is continually send ICMP echo requests to the targeted hostname (99.236.34.223) from 192.168.2.49 (the `localhost`). We can also follow the hops in the traceroute by checking the source destination of the "Time-to-live exceeded" ICMP packets, with the first two hops of the traceroute (192.168.2.1 and 10.11.2.49) appearing in the figure.

Figure 6: Traceroute WireShark Capture

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.2.49	99.236.34.223	ICMP	106	Echo (ping) request id=0x0001, seq=236/60416, ttl=1 (no response found!)
2	0.000016	192.168.2.49	99.236.34.223	ICMP	106	Echo (ping) request id=0x0001, seq=236/60416, ttl=1 (no response found!)
3	0.005391	192.168.2.1	192.168.2.49	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
4	0.006001	192.168.2.49	99.236.34.223	ICMP	106	Echo (ping) request id=0x0001, seq=237/60672, ttl=1 (no response found!)
5	0.006013	192.168.2.49	99.236.34.223	ICMP	106	Echo (ping) request id=0x0001, seq=237/60672, ttl=1 (no response found!)
6	0.011327	192.168.2.1	192.168.2.49	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
7	0.011721	192.168.2.49	99.236.34.223	ICMP	106	Echo (ping) request id=0x0001, seq=238/60928, ttl=1 (no response found!)
8	0.011727	192.168.2.49	99.236.34.223	ICMP	106	Echo (ping) request id=0x0001, seq=238/60928, ttl=1 (no response found!)
9	0.017607	192.168.2.1	192.168.2.49	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
10	1.022108	192.168.2.49	99.236.34.223	ICMP	106	Echo (ping) request id=0x0001, seq=239/61184, ttl=2 (no response found!)
11	1.022119	192.168.2.49	99.236.34.223	ICMP	106	Echo (ping) request id=0x0001, seq=239/61184, ttl=2 (no response found!)
12	1.044148	10.11.2.49	192.168.2.49	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
13	1.044623	192.168.2.49	99.236.34.223	ICMP	106	Echo (ping) request id=0x0001, seq=240/61440, ttl=2 (no response found!)
14	1.044629	192.168.2.49	99.236.34.223	ICMP	106	Echo (ping) request id=0x0001, seq=240/61440, ttl=2 (no response found!)
15	1.060458	10.11.2.49	192.168.2.49	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
16	1.061046	192.168.2.49	99.236.34.223	ICMP	106	Echo (ping) request id=0x0001, seq=241/61696, ttl=2 (no response found!)
17	1.061062	192.168.2.49	99.236.34.223	ICMP	106	Echo (ping) request id=0x0001, seq=241/61696, ttl=2 (no response found!)
18	1.080493	10.11.2.49	192.168.2.49	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
19	7.026845	192.168.2.49	99.236.34.223	ICMP	106	Echo (ping) request id=0x0001, seq=242/61952, ttl=3 (no response found!)
20	7.026854	192.168.2.49	99.236.34.223	ICMP	106	Echo (ping) request id=0x0001, seq=242/61952, ttl=3 (no response found!)
21	10.974409	192.168.2.49	99.236.34.223	ICMP	106	Echo (ping) request id=0x0001, seq=243/62208, ttl=3 (no response found!)
22	10.974426	192.168.2.49	99.236.34.223	ICMP	106	Echo (ping) request id=0x0001, seq=243/62208, ttl=3 (no response found!)
23	14.969557	192.168.2.49	99.236.34.223	ICMP	106	Echo (ping) request id=0x0001, seq=244/62464, ttl=3 (no response found!)
24	14.969568	192.168.2.49	99.236.34.223	ICMP	106	Echo (ping) request id=0x0001, seq=244/62464, ttl=3 (no response found!)
25	18.972363	192.168.2.49	99.236.34.223	ICMP	106	Echo (ping) request id=0x0001, seq=245/62720, ttl=4 (no response found!)

Nmap

Standard TCP Connection Scan of `compeng4dn4.mooo.com`

A standard TCP connection scan (using -sT option) of the server `compeng4dn4.mooo.com` over the port range 50000-50009 is performed and shown in Listing 5. The scan showed that ports 50007 and 50008 were open, indicating that there is an application that will accept TCP connections, UDP datagrams, or SCTP associated on these ports. The remainder of the ports reported their states as filtered, indicating Nmap could not determine if the port is open. Nmap states that this usually occurs due to packet filtering preventing Nmap's

probes from reaching the port. The known services corresponding with each port (from the nmap-services database) are also listed, with port 50000 being used for IBM Db2 and port 50002 for Internet/Intranet Input Method Server Framework.

Listing 5: Nmap Standard TCP Connection Scan Over Port Range 50000-50009

```
nmap compeng4dn4.mooo.com -Pn -sT -p 50000-50009
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-11 22:50 Eastern Standard Time
Nmap scan report for compeng4dn4.mooo.com (99.236.34.223)
Host is up (0.030s latency).
rDNS record for 99.236.34.223: cpe382c4a5bff48-cm00fc8db8cbb0.cpe.net.cable.rogers.com

PORT      STATE    SERVICE
50000/tcp filtered ibm-db2
50001/tcp filtered unknown
50002/tcp filtered iiimsf
50003/tcp filtered unknown
50004/tcp filtered unknown
50005/tcp filtered unknown
50006/tcp filtered unknown
50007/tcp open     unknown
50008/tcp open     unknown
50009/tcp filtered unknown

Nmap done: 1 IP address (1 host up) scanned in 3.36 seconds
```

TCP SYN Scan of `compeng4dn4.mooo.com`

A TCP SYN scan (using -sS option) of the server `compeng4dn4.mooo.com` over the port range 50000-50009 is performed and shown in Listing 6. The scan returns the same information, reporting the same state and services for each of the ports scanned.

Listing 6: Nmap TCP SYN Scan Over Port Range 50000-50009

```
nmap compeng4dn4.mooo.com -Pn -sS -p 50000-50009
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-11 22:51 Eastern Standard Time
Nmap scan report for compeng4dn4.mooo.com (99.236.34.223)
Host is up (0.032s latency).
rDNS record for 99.236.34.223: cpe382c4a5bff48-cm00fc8db8cbb0.cpe.net.cable.rogers.com

PORT      STATE    SERVICE
50000/tcp filtered ibm-db2
50001/tcp filtered unknown
50002/tcp filtered iiimsf
50003/tcp filtered unknown
50004/tcp filtered unknown
50005/tcp filtered unknown
50006/tcp filtered unknown
50007/tcp open     unknown
50008/tcp open     unknown
50009/tcp filtered unknown

Nmap done: 1 IP address (1 host up) scanned in 1.50 seconds
```

TCP Scans of Open Port 50007

A standard TCP connection scan (using -sT option) and a TCP SYN scan (using -sS option) of the server `compeng4dn4.mooo.com` on the open port 50007 are performed and shown in Listing 7. For both scans, the Nmap scan reported the same information. The scans reported the same information on port 50007 as the earlier scans on the larger port range.

Listing 7: Nmap Scans on Port 50007

```
sudo nmap compeng4dn4.mooo.com -Pn -sT -p 50007
Starting Nmap 7.80 ( https://nmap.org ) at 2023-02-12 21:32 EST
Nmap scan report for compeng4dn4.mooo.com (99.236.34.223)
Host is up (0.031s latency).
rDNS record for 99.236.34.223: cpe382c4a5bff48-cm00fc8db8cbb0.cpe.net.cable.rogers.com

PORT      STATE SERVICE
50007/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.15 seconds

sudo nmap compeng4dn4.mooo.com -Pn -sS -p 50007
Starting Nmap 7.80 ( https://nmap.org ) at 2023-02-12 21:32 EST
Nmap scan report for compeng4dn4.mooo.com (99.236.34.223)
Host is up (0.024s latency).
rDNS record for 99.236.34.223: cpe382c4a5bff48-cm00fc8db8cbb0.cpe.net.cable.rogers.com

PORT      STATE SERVICE
50007/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds
```

Both scans were captured with tcpdump, writing the scan to a WireShark compatible pcap file for analysis. The tcpdump packet capture command of the scans is shown in Listing 8.

Listing 8: Tcpdump Packet Capture of Scans on Port 50007

```
sudo tcpdump -nnvvX -i 1 -S host compeng4dn4.mooo.com -w 50007.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
7 packets captured
13 packets received by filter
0 packets dropped by kernel
```

The capture displayed in WireShark is shown below in Figure 7. The first four packets (1-4) are from the standard TCP connection scan on port 50007. The last three packets (5-7) are from the TCP SYN scan on port 50007.

The first four packets shows the expected behaviour of a TCP connection scan on an open port. The host (192.168.2.55) sends a SYN signal to the destination (port 50007 on 99.236.34.223). The destination lets the host know that the port is open by returning a SYN/ACK signal. The host then establishes the connection and sends an ACK signal. Finally, the host will kill the connection with a RST signal.

The last three packets shows the expected behaviour of a TCP SYN scan on an open port. The first two packets are the same as before, however the host sends a RST signal on the third packet to let the destination know it will not be establishing a connection.

Figure 7: Tcpdump Capture of Port 50007 in WireShark

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.2.55	99.236.34.223	TCP	74	46748 → 50007 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2265637838 TSecr=0 WS=128
2	0.030798	99.236.34.223	192.168.2.55	TCP	74	50007 → 46748 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1452 SACK_PERM TSval=385772596 TSecr=2265637838 WS=128
3	0.030899	192.168.2.55	99.236.34.223	TCP	66	46748 → 50007 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2265637869 TSecr=385772596
4	0.030960	192.168.2.55	99.236.34.223	TCP	66	46748 → 50007 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2265637869 TSecr=385772596
5	17.701279	192.168.2.55	99.236.34.223	TCP	58	62988 → 50007 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
6	17.725210	99.236.34.223	192.168.2.55	TCP	60	50007 → 62988 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1452
7	17.725278	192.168.2.55	99.236.34.223	TCP	54	62988 → 50007 [RST] Seq=1 Win=0 Len=0

TCP Scans of Closed Port 50009

The filtered ports were assumed to be closed, and the scans were performed on the port 50009. A standard TCP connection scan (using -sT option) and a TCP SYN scan (using -sS option) of the server `compeng4dn4.mooo.com` on the closed port 50009 are performed and shown in Listing 9. For both scans, the Nmap scan reported the same information. The scans reported the same information on port 50009 as the earlier scans on the larger port range.

Listing 9: Nmap Scans on Port 50009

```
sudo nmap compeng4dn4.mooo.com -Pn -sT -p 50009
Starting Nmap 7.80 ( https://nmap.org ) at 2023-02-12 21:31 EST
Nmap scan report for compeng4dn4.mooo.com (99.236.34.223)
Host is up.
rDNS record for 99.236.34.223: cpe382c4a5bff48-cm00fc8db8cbb0.cpe.net.cable.rogers.com

PORT      STATE    SERVICE
50009/tcp filtered unknown

Nmap done: 1 IP address (1 host up) scanned in 2.12 seconds

sudo nmap compeng4dn4.mooo.com -Pn -sS -p 50009
Starting Nmap 7.80 ( https://nmap.org ) at 2023-02-12 21:32 EST
Nmap scan report for compeng4dn4.mooo.com (99.236.34.223)
Host is up.
rDNS record for 99.236.34.223: cpe382c4a5bff48-cm00fc8db8cbb0.cpe.net.cable.rogers.com

PORT      STATE    SERVICE
50009/tcp filtered unknown

Nmap done: 1 IP address (1 host up) scanned in 2.18 seconds
```

Both scans were captured with tcpdump, writing the scan to a WireShark compatible pcap file for analysis. The tcpdump packet capture command of the scans is shown in Listing 10.

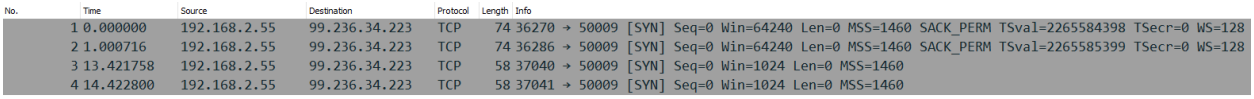
Listing 10: Tcpdump Packet Capture of Scans on Port 50009

```
sudo tcpdump -nnvvX -i 1 -S host compeng4dn4.mooo.com -w 50009.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
4 packets captured
10 packets received by filter
0 packets dropped by kernel
```

The capture displayed in WireShark is shown below in Figure 8. The first two packets (1-2) are from the standard TCP connection scan on port 50009. The last two packets (3-4) are from the TCP SYN scan on port 50009.

The first two packets and last two packets both show the expected behaviour of a Nmap scan on a filtered port. The host (192.168.2.55) sends a SYN signal to the destination (port 50009 on 99.236.34.223) but does not receive a reply. After waiting some time (around 1 second in this capture), the host sends another SYN signal to try again. If the timeout period is again exceeded, Nmap will give up and mark the port as filtered.

Figure 8: Tcpdump Capture of Port 50009 in WireShark



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.2.55	99.236.34.223	TCP	74	36270 → 50009 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2265584398 TSecr=0 WS=128
2	1.000716	192.168.2.55	99.236.34.223	TCP	74	36286 → 50009 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2265585399 TSecr=0 WS=128
3	13.421758	192.168.2.55	99.236.34.223	TCP	58	37040 → 50009 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4	14.422800	192.168.2.55	99.236.34.223	TCP	58	37041 → 50009 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

Standard TCP Connection Scan of `localhost`

A standard TCP connection scan (using -sT option) of `localhost` was performed and shown in Listing 11. The scan reports all found open ports, and also states that there are 993 tcp ports that were filtered due to no-response. The services corresponding to each open port is also listed. The purposes of each service on these open ports are discussed in Table 1.

Listing 11: Nmap Standard TCP Connection Scan of `localhost`

```
nmap localhost -Pn -sT
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-11 23:39 Eastern Standard Time
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00015s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 993 filtered tcp ports (no-response)
PORT     STATE SERVICE
135/tcp  open  msrpc
445/tcp  open  microsoft-ds
2179/tcp open  vmrdp
5357/tcp open  wsdapi
9010/tcp open  sdr
9080/tcp open  glrpc
9100/tcp open  jetdirect

Nmap done: 1 IP address (1 host up) scanned in 44.45 seconds
```

Table 1: Services on Open Ports of `localhost`

PORT	SERVICE	PURPOSE
135/tcp	msrpc	Microsoft Remote Procedure Call
445/tcp	microsoft-ds	Server Message Block (SMB)
2179/tcp	vmrdp	Microsoft RDP for virtual machines
5357/tcp	wsdapi	Microsoft Network Discovery
9010/tcp	sdr	Secure Data Replicator Protocol
9080/tcp	glrpc	Groove Collaboration software GLRPC
9100/tcp	jetdirect	HP JetDirect (for HP LaserJet printers)

TCP Scan of Port 8000 on `localhost`

A standard TCP connection scan (using -sT option) of port 8000 on `localhost` was performed and shown in Listing 12. The scan reports that the port is filtered (likely due to no response as mentioned in the previous report). The service that typically corresponds to this port is `http-alt`, an alternative HTTP port.

Listing 12: Nmap Standard TCP Connection Scan of Port 8000 on `localhost`

```
nmap localhost -Pn -sT -p 8000
```

```
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-11 23:43 Eastern Standard Time
Nmap scan report for localhost (127.0.0.1)
Host is up.
Other addresses for localhost (not scanned): ::1

PORT     STATE    SERVICE
8000/tcp filtered http-alt

Nmap done: 1 IP address (1 host up) scanned in 2.11 seconds
```

TCP Scan of Port 8000 on `localhost` with HTTP Server

A standard TCP connection scan (using -sT option) of port 8000 on `localhost` was performed while the HTTP server was running, and the scan is shown in Listing 13. The scan reports that the port is now open. When the web service is accessed through a browser, a page that serves the contents of the directory where the server was launched is shown. The webpage is shown in Figure 9.

Listing 13: Nmap Standard TCP Connection Scan of Port 8000 on `localhost` with HTTP Server

```
nmap localhost -Pn -sT -p 8000
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-11 23:43 Eastern Standard Time
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0010s latency).
Other addresses for localhost (not scanned): ::1

PORT     STATE SERVICE
8000/tcp open  http-alt

Nmap done: 1 IP address (1 host up) scanned in 0.09 seconds
```

Figure 9: `localhost:8000`