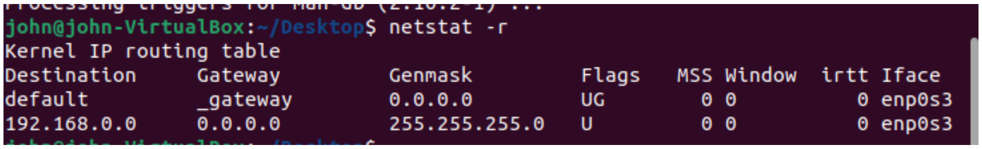
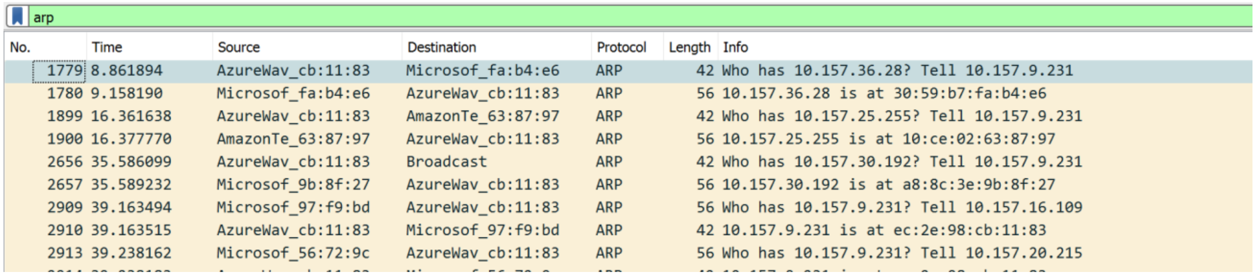
1.1

Ethernet Adapter and Ip address



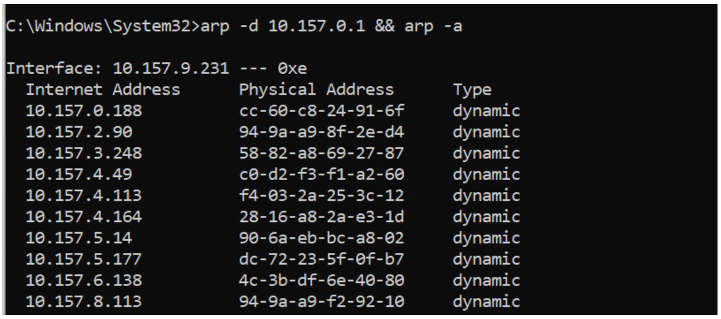
```
Processing triggers for man-db (2.10.2-1) ...
john@john-VirtualBox:~/Desktop$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         _gateway        0.0.0.0         UG        0 0          0 enp0s3
192.168.0.0     0.0.0.0         255.255.255.0   U         0 0          0 enp0s3
john@john-VirtualBox:~/Desktop$
```

Instance

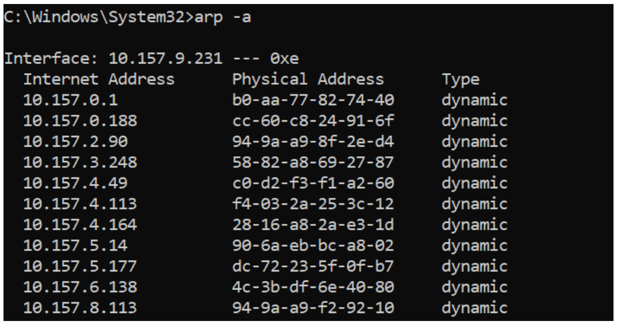


```
No.     Time        Source              Destination         Protocol  Length Info
   1779 8.861894    AzureWav_cb:11:83   Microsof_fa:b4:e6   ARP       42 Who has 10.157.36.28? Tell 10.157.9.231
   1780 9.158190    Microsof_fa:b4:e6   AzureWav_cb:11:83   ARP       56 10.157.36.28 is at 30:59:b7:fa:b4:e6
   1899 16.361638   AzureWav_cb:11:83   AmazonTe_63:87:97   ARP       42 Who has 10.157.25.255? Tell 10.157.9.231
   1900 16.377770   AmazonTe_63:87:97   AzureWav_cb:11:83   ARP       56 10.157.25.255 is at 10:ce:02:63:87:97
   2656 35.586099   AzureWav_cb:11:83   Broadcast           ARP       42 Who has 10.157.30.192? Tell 10.157.9.231
   2657 35.589232   Microsof_9b:8f:27   AzureWav_cb:11:83   ARP       56 10.157.30.192 is at a8:8c:3e:9b:8f:27
   2909 39.163494   Microsof_97:f9:bd   AzureWav_cb:11:83   ARP       56 Who has 10.157.9.231? Tell 10.157.16.109
   2910 39.163515   AzureWav_cb:11:83   Microsof_97:f9:bd   ARP       42 10.157.9.231 is at ec:2e:98:cb:11:83
   2913 39.238162   Microsof_56:72:9c   AzureWav_cb:11:83   ARP       56 Who has 10.157.9.231? Tell 10.157.20.215
```

After deletion

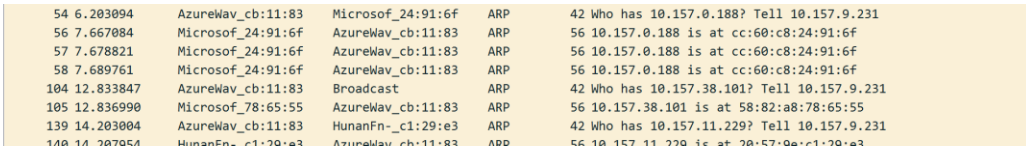


```
C:\Windows\System32>arp -d 10.157.0.1 && arp -a

Interface: 10.157.9.231 --- 0xe
  Internet Address      Physical Address      Type
  10.157.0.188          cc-60-c8-24-91-6f     dynamic
  10.157.2.90           94-9a-a9-8f-2e-d4     dynamic
  10.157.3.248          58-82-a8-69-27-87     dynamic
  10.157.4.49           c0-d2-f3-f1-a2-60     dynamic
  10.157.4.113          f4-03-2a-25-3c-12     dynamic
  10.157.4.164          28-16-a8-2a-e3-1d     dynamic
  10.157.5.14           90-6a-eb-bc-a8-02     dynamic
  10.157.5.177          dc-72-23-5f-0f-b7     dynamic
  10.157.6.138          4c-3b-df-6e-40-80     dynamic
  10.157.8.113          94-9a-a9-f2-92-10     dynamic
```

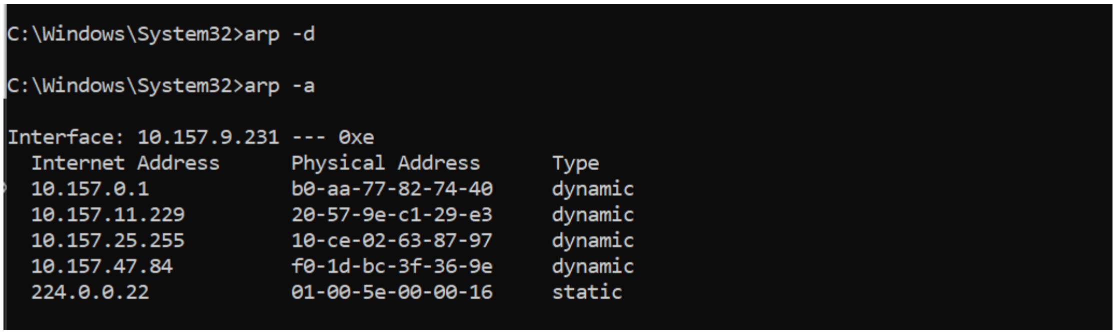


```
C:\Windows\System32>arp -a

Interface: 10.157.9.231 --- 0xe
  Internet Address      Physical Address      Type
  10.157.0.1            b0-aa-77-82-74-40     dynamic
  10.157.0.188          cc-60-c8-24-91-6f     dynamic
  10.157.2.90           94-9a-a9-8f-2e-d4     dynamic
  10.157.3.248          58-82-a8-69-27-87     dynamic
  10.157.4.49           c0-d2-f3-f1-a2-60     dynamic
  10.157.4.113          f4-03-2a-25-3c-12     dynamic
  10.157.4.164          28-16-a8-2a-e3-1d     dynamic
  10.157.5.14           90-6a-eb-bc-a8-02     dynamic
  10.157.5.177          dc-72-23-5f-0f-b7     dynamic
  10.157.6.138          4c-3b-df-6e-40-80     dynamic
  10.157.8.113          94-9a-a9-f2-92-10     dynamic
```

Updated trace

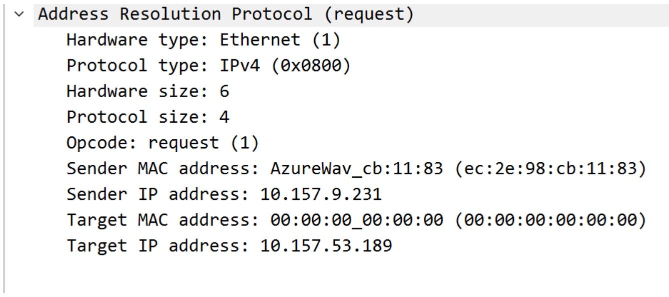


```
     54 6.203094      AzureWav_cb:11:83   Microsof_24:91:6f   ARP      42 Who has 10.157.0.188? Tell 10.157.9.231
     56 7.667084      Microsof_24:91:6f   AzureWav_cb:11:83   ARP      56 10.157.0.188 is at cc:60:c8:24:91:6f
     57 7.678821      Microsof_24:91:6f   AzureWav_cb:11:83   ARP      56 10.157.0.188 is at cc:60:c8:24:91:6f
     58 7.689761      Microsof_24:91:6f   AzureWav_cb:11:83   ARP      56 10.157.0.188 is at cc:60:c8:24:91:6f
    104 12.833847     AzureWav_cb:11:83   Broadcast           ARP      42 Who has 10.157.38.101? Tell 10.157.9.231
    105 12.836990     Microsof_78:65:55   AzureWav_cb:11:83   ARP      56 10.157.38.101 is at 58:82:a8:78:65:55
    139 14.203004     AzureWav_cb:11:83   HunanFn-_c1:29:e3   ARP      42 Who has 10.157.11.229? Tell 10.157.9.231
    140 14.207954     HunanFn-_c1:29:e3   AzureWav_cb:11:83   ARP      56 10.157.11.229 is at 20:57:9e:c1:29:e3
```

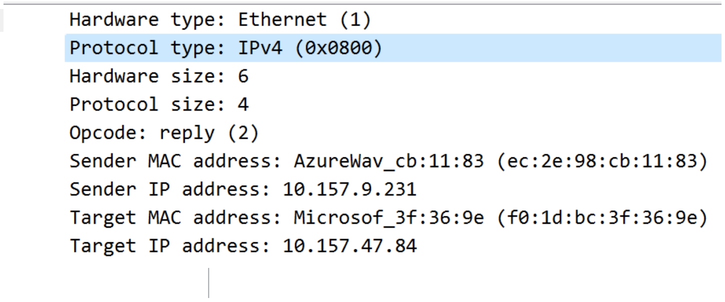


```
C:\Windows\System32>arp -d

C:\Windows\System32>arp -a

Interface: 10.157.9.231 --- 0xe
  Internet Address      Physical Address      Type
  10.157.0.1            b0-aa-77-82-74-40     dynamic
  10.157.11.229         20-57-9e-c1-29-e3     dynamic
  10.157.25.255         10-ce-02-63-87-97     dynamic
  10.157.47.84          f0-1d-bc-3f-36-9e     dynamic
  224.0.0.22            01-00-5e-00-00-16     static
```

Request and reply



```
⌄ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: AzureWav_cb:11:83 (ec:2e:98:cb:11:83)
    Sender IP address: 10.157.9.231
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 10.157.53.189
```



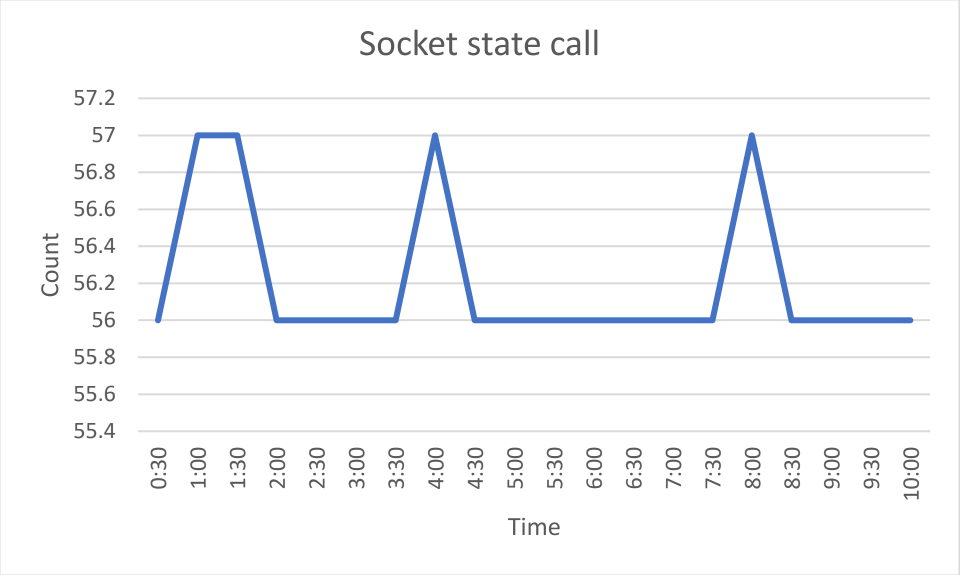
```
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender MAC address: AzureWav_cb:11:83 (ec:2e:98:cb:11:83)
Sender IP address: 10.157.9.231
Target MAC address: Microsof_3f:36:9e (f0:1d:bc:3f:36:9e)
Target IP address: 10.157.47.84
```

Details of ARP over Ethernet

1. What opcode is used to indicate a request? What about a reply?
   a. 1 for request, 2 for reply
2. How large is the ARP header for a request? What about for a reply? You will need to research this (hint: some sources define what belongs to the header differently, name which source you base your answer on)
   a. 28 bytes. (https://www.netometer.com/qa/arp.html#:~:text=The%20size%20of%20an%20ARP%20request%20or%20reply%20packet%20is%2028%20bytes.)source
3. What value is carried on a request for the unknown target MAC address?
   a. 00:00:00_00:00:00
4. What Ethernet Type value indicates that ARP is the higher layer protocol?
   a. 0x0806

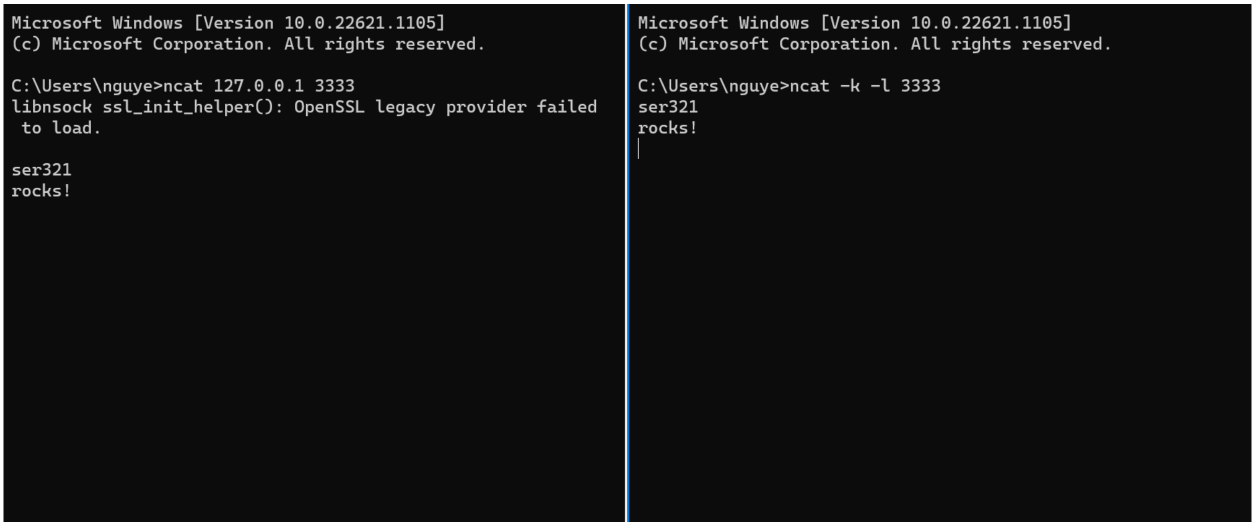
1.2. Understanding TCP network sockets

Manually set a timer and went line by line counting



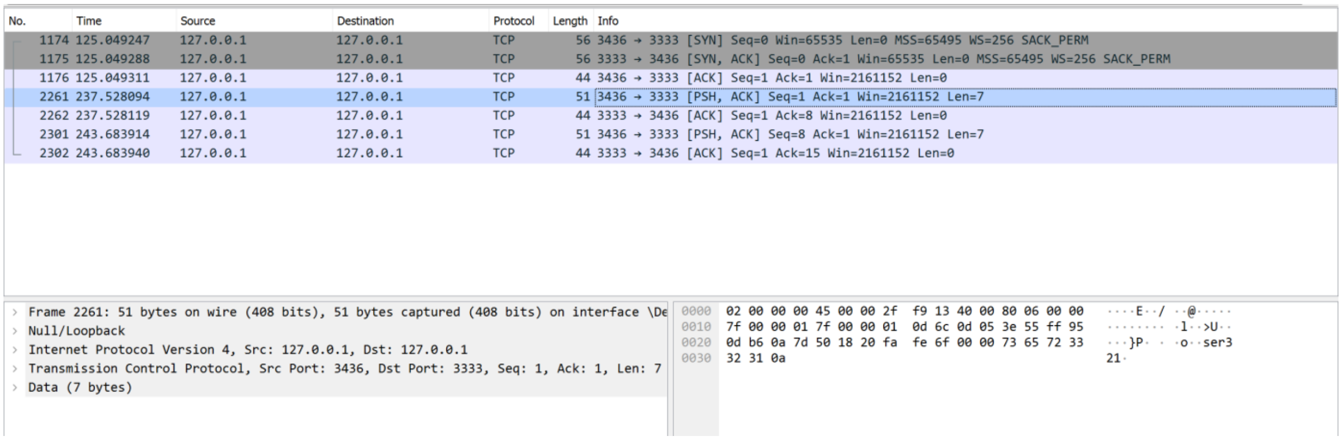
1.3. Sniffing TCP/UDP traffic

Tcp sniffing



```
Microsoft Windows [Version 10.0.22621.1105]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nguye>ncat 127.0.0.1 3333
libnsock ssl_init_helper(): OpenSSL legacy provider failed
 to load.

ser321
rocks!
```

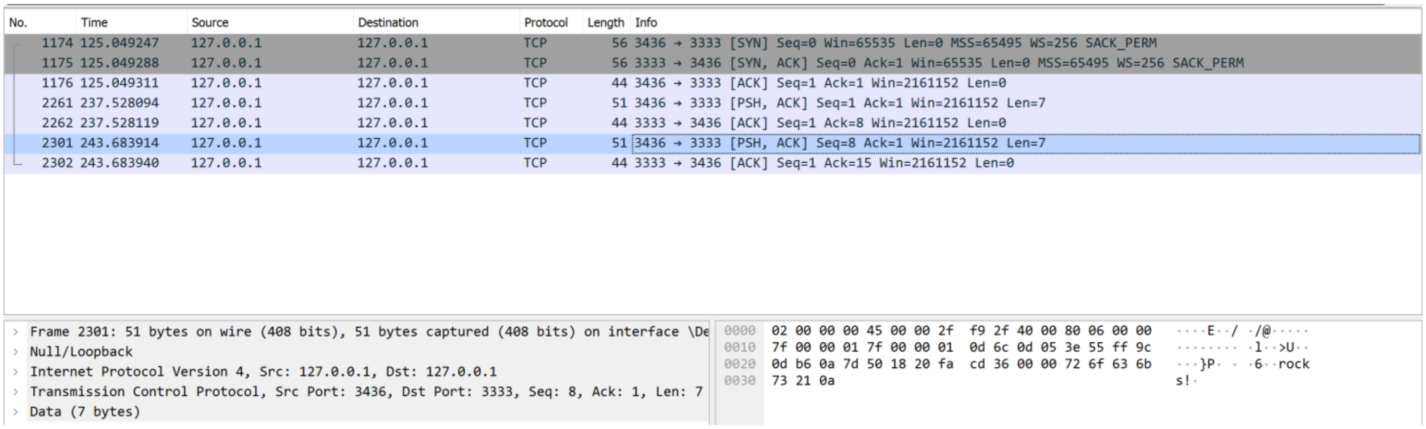
```
Microsoft Windows [Version 10.0.22621.1105]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nguye>ncat -k -l 3333
ser321
rocks!
```



```
No.   Time        Source     Destination  Protocol Length Info
 1174 125.049247  127.0.0.1  127.0.0.1    TCP      56 3436 → 3333 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
 1175 125.049288  127.0.0.1  127.0.0.1    TCP      56 3333 → 3436 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
 1176 125.049311  127.0.0.1  127.0.0.1    TCP      44 3436 → 3333 [ACK] Seq=1 Ack=1 Win=2161152 Len=0
 2261 237.528094  127.0.0.1  127.0.0.1    TCP      51 3436 → 3333 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=7
 2262 237.528119  127.0.0.1  127.0.0.1    TCP      44 3333 → 3436 [ACK] Seq=1 Ack=8 Win=2161152 Len=0
 2301 243.683914  127.0.0.1  127.0.0.1    TCP      51 3436 → 3333 [PSH, ACK] Seq=8 Ack=1 Win=2161152 Len=7
 2302 243.683940  127.0.0.1  127.0.0.1    TCP      44 3333 → 3436 [ACK] Seq=1 Ack=15 Win=2161152 Len=0
```

```
> Frame 2261: 51 bytes on wire (408 bits), 51 bytes captured (408 bits) on interface \De
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 3436, Dst Port: 3333, Seq: 1, Ack: 1, Len: 7
> Data (7 bytes)
```

```
0000  02 00 00 00 45 00 00 2f  f9 13 40 00 80 06 00 00   ····E··/ ··@·····
0010  7f 00 00 01 7f 00 00 01  0d 6c 0d 05 3e 55 ff 95   ········ ·l··>U··
0020  0d b6 0a 7d 50 18 20 fa  fe 6f 00 00 73 65 72 33   ···}P· · ·o··ser3
0030  32 31 0a                                           21·
```



```
No.   Time        Source     Destination  Protocol Length Info
 1174 125.049247  127.0.0.1  127.0.0.1    TCP      56 3436 → 3333 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
 1175 125.049288  127.0.0.1  127.0.0.1    TCP      56 3333 → 3436 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
 1176 125.049311  127.0.0.1  127.0.0.1    TCP      44 3436 → 3333 [ACK] Seq=1 Ack=1 Win=2161152 Len=0
 2261 237.528094  127.0.0.1  127.0.0.1    TCP      51 3436 → 3333 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=7
 2262 237.528119  127.0.0.1  127.0.0.1    TCP      44 3333 → 3436 [ACK] Seq=1 Ack=8 Win=2161152 Len=0
 2301 243.683914  127.0.0.1  127.0.0.1    TCP      51 3436 → 3333 [PSH, ACK] Seq=8 Ack=1 Win=2161152 Len=7
 2302 243.683940  127.0.0.1  127.0.0.1    TCP      44 3333 → 3436 [ACK] Seq=1 Ack=15 Win=2161152 Len=0
```

```
> Frame 2301: 51 bytes on wire (408 bits), 51 bytes captured (408 bits) on interface \De
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 3436, Dst Port: 3333, Seq: 8, Ack: 1, Len: 7
> Data (7 bytes)
```

```
0000  02 00 00 00 45 00 00 2f  f9 2f 40 00 80 06 00 00   ····E··/ ·/@·····
0010  7f 00 00 01 7f 00 00 01  0d 6c 0d 05 3e 55 ff 9c   ········ ·l··>U··
0020  0d b6 0a 7d 50 18 20 fa  cd 36 00 00 72 6f 63 6b   ···}P· · ·6··rock
0030  73 21 0a                                           s!·
```

a) Explain both the commands you used in detail. What did they actually do?
Ncat -k -l 3333 establishes a port server for client to talk to
Ncat 127.0.0.1 3333 talks to the server from client on port 3333

b) How many frames were send back and forth to capture these 2 lines?
51 bytes for both

c) How many packets were sent back and forth to capture only those 2 lines?
3 packets

d) How many packets were needed to capture the whole "process" (starting the communication, ending the communication)?

7 packets

d) How many bytes is the data (only the data) that was send?
7 bytes

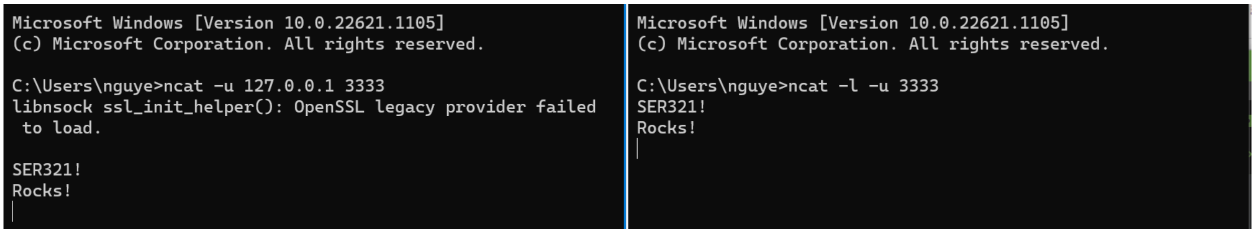
e) How many total bytes went over the wire (back and forth) for the whole process?
14 bytes

g) How much overhead was there. Basically how many bytes was the whole process

compared to the actually data that we did send.

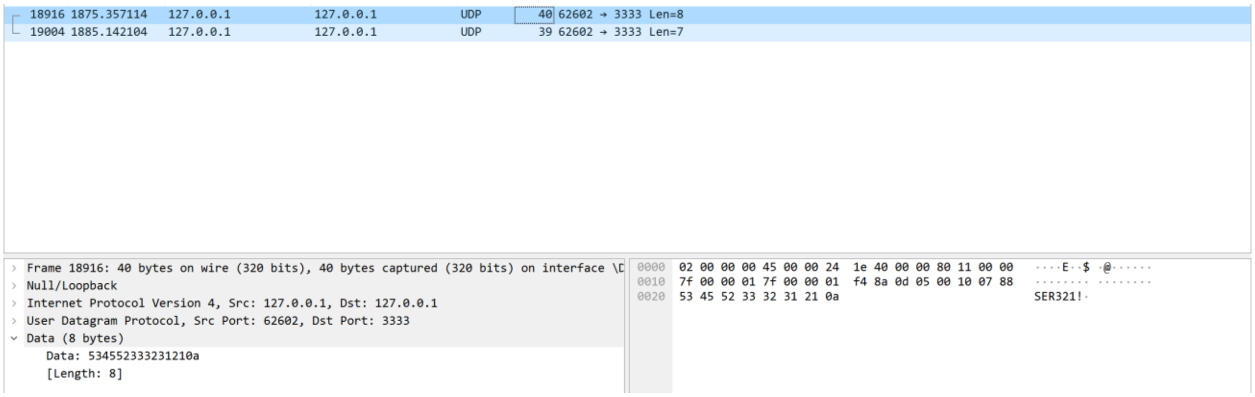
132 bytes

Udp sniffing



```
Microsoft Windows [Version 10.0.22621.1105]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nguye>ncat -u 127.0.0.1 3333
libnsock ssl_init_helper(): OpenSSL legacy provider failed
 to load.

SER321!
Rocks!
```

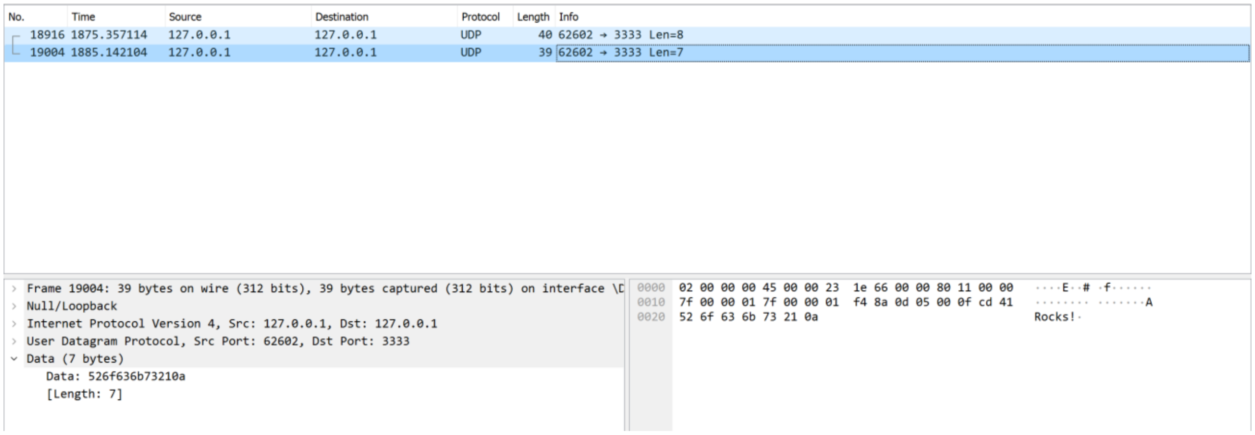
```
Microsoft Windows [Version 10.0.22621.1105]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nguye>ncat -l -u 3333
SER321!
Rocks!
```



```
18916 1875.357114   127.0.0.1    127.0.0.1    UDP   40 62602 → 3333 Len=8
19004 1885.142104   127.0.0.1    127.0.0.1    UDP   39 62602 → 3333 Len=7
```

```
> Frame 18916: 40 bytes on wire (320 bits), 40 bytes captured (320 bits) on interface \D
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> User Datagram Protocol, Src Port: 62602, Dst Port: 3333
v Data (8 bytes)
    Data: 534552333231210a
    [Length: 8]
```

```
0000  02 00 00 00 45 00 00 24  1e 40 00 00 80 11 00 00   ····E··$ ·@······
0010  7f 00 00 01 7f 00 00 01  f4 8a 0d 05 00 10 07 88   ········ ········
0020  53 45 52 33 32 31 21 0a                            SER321!·
```



```
No.   Time          Source       Destination  Protocol  Length  Info
18916 1875.357114   127.0.0.1    127.0.0.1    UDP           40  62602 → 3333 Len=8
19004 1885.142104   127.0.0.1    127.0.0.1    UDP           39  62602 → 3333 Len=7
```

```
> Frame 19004: 39 bytes on wire (312 bits), 39 bytes captured (312 bits) on interface \D
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> User Datagram Protocol, Src Port: 62602, Dst Port: 3333
v Data (7 bytes)
    Data: 526f636b73210a
    [Length: 7]
```

```
0000  02 00 00 00 45 00 00 23  1e 66 00 00 80 11 00 00   ····E··# ·f······
0010  7f 00 00 01 7f 00 00 01  f4 8a 0d 05 00 0f cd 41   ········ ·······A
0020  52 6f 63 6b 73 21 0a                               Rocks!·
```

a) Explain both the commands you used in detail. What did they actually do?

Ncat -l -u 3333 established port 3333 with u flag specifying it is in udp server

Ncat -u 127.0.0.1 3333 u specifying sending to udp server.

b) How many frames were needed to capture those 2 lines?

40 bytes for the first and 39 for the second

c) How many packets were needed to capture those 2 lines?
2 packets

d) How many packets were needed to capture the whole "process" (starting the

communication, ending the communication)?

2 packets

e) How many total bytes went over the wire?

79 total bytes

f) How many bytes is the data (only the data) that was sent?

8 bytes for the first- and 7-bytes fir the second

g) Basically, how many bytes was the whole process compared to the actually data

that we did send.?

15 bytes

h) What is the difference in relative overhead between UDP and TCP and why?

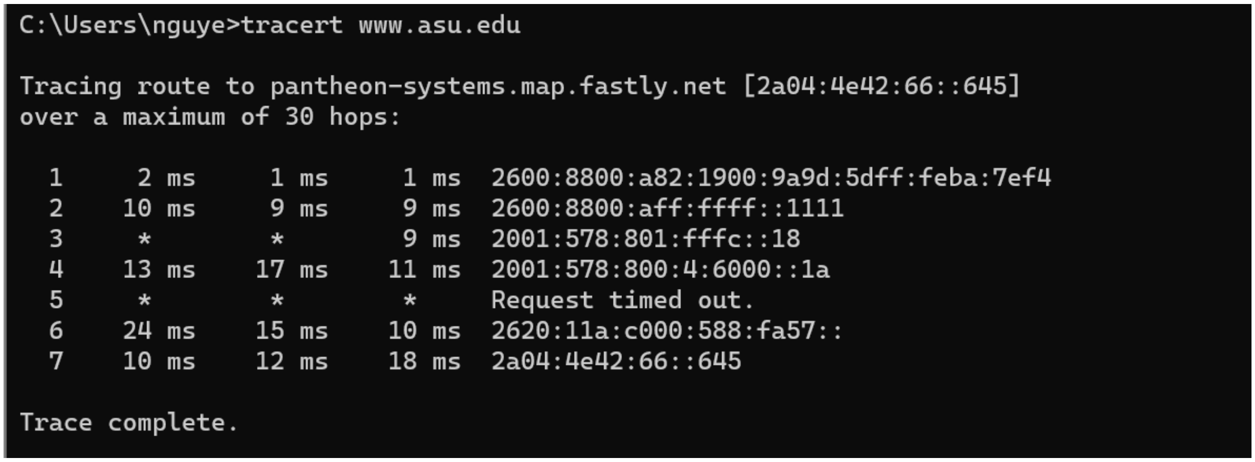
Specifically, what kind of information was exchanged in TCP that was not

exchanged in UDP? Show the relative parts of the packet traces.

For the tcp requires more packets due to checks being made before and after sending data while udp only has 2 packets since its only sending and receive.

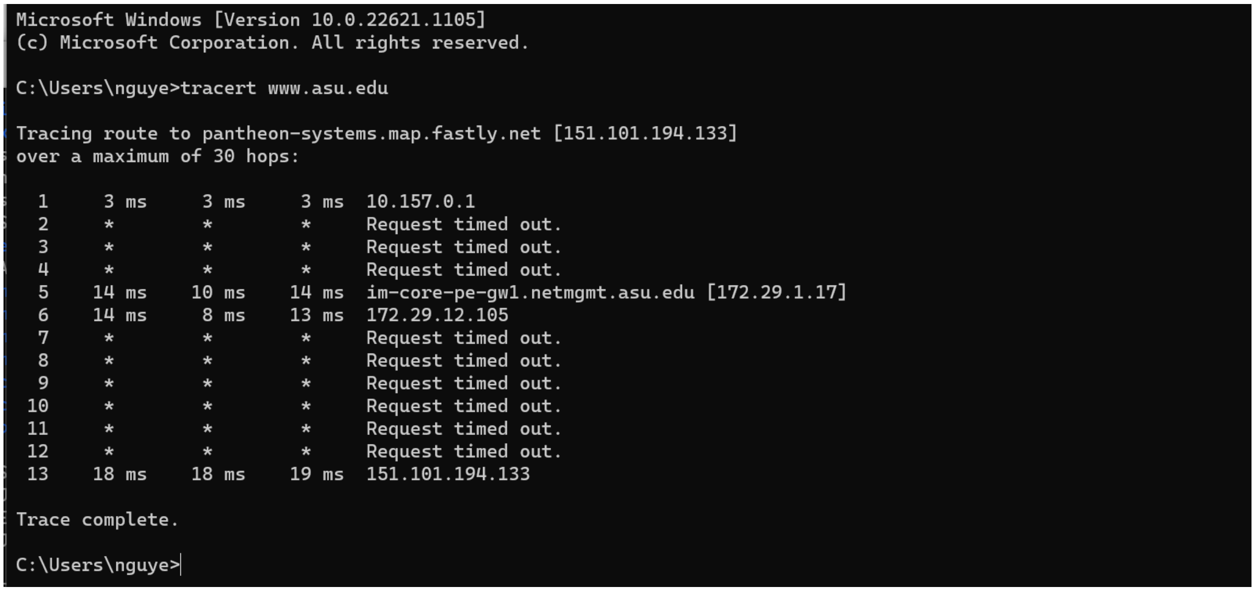
1.4. Internet Protocol (IP) Routing

(Home network)



```
C:\Users\nguye>tracert www.asu.edu

Tracing route to pantheon-systems.map.fastly.net [2a04:4e42:66::645]
over a maximum of 30 hops:

  1     2 ms     1 ms     1 ms  2600:8800:a82:1900:9a9d:5dff:feba:7ef4
  2    10 ms     9 ms     9 ms  2600:8800:aff:ffff::1111
  3     *        *        9 ms  2001:578:801:fffc::18
  4    13 ms    17 ms    11 ms  2001:578:800:4:6000::1a
  5     *        *        *     Request timed out.
  6    24 ms    15 ms    10 ms  2620:11a:c000:588:fa57::
  7    10 ms    12 ms    18 ms  2a04:4e42:66::645

Trace complete.
```

(ASU network)

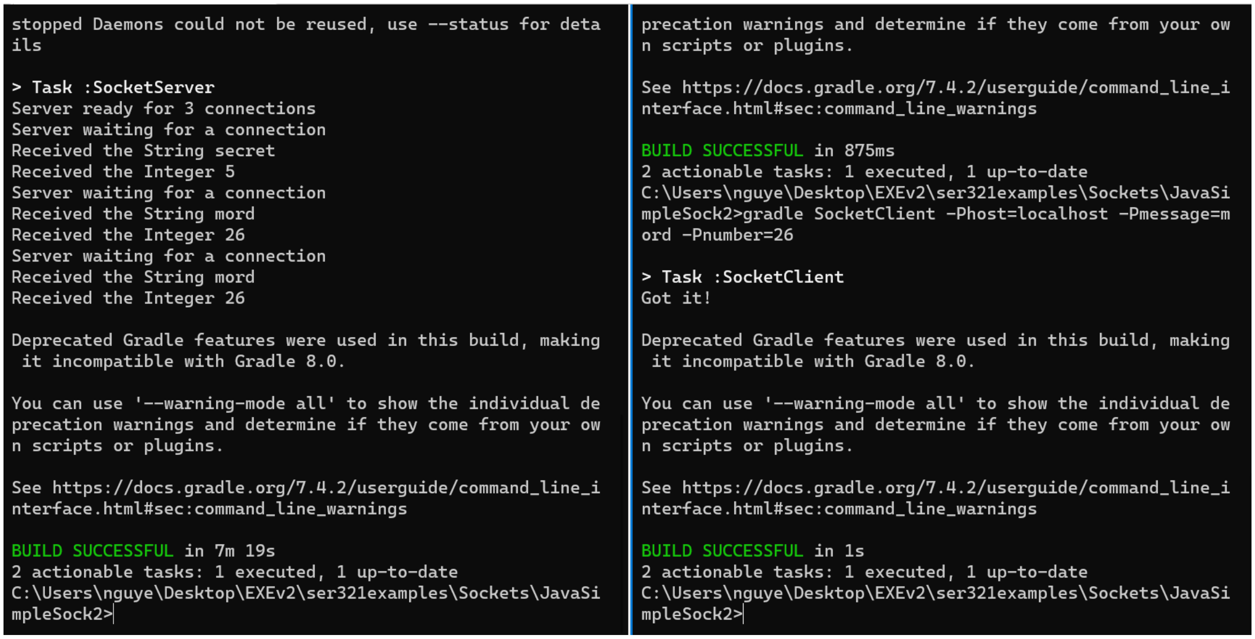


```
Microsoft Windows [Version 10.0.22621.1105]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nguye>tracert www.asu.edu

Tracing route to pantheon-systems.map.fastly.net [151.101.194.133]
over a maximum of 30 hops:

  1     3 ms     3 ms     3 ms  10.157.0.1
  2     *        *        *     Request timed out.
  3     *        *        *     Request timed out.
  4     *        *        *     Request timed out.
  5    14 ms    10 ms    14 ms  im-core-pe-gw1.netmgmt.asu.edu [172.29.1.17]
  6    14 ms     8 ms    13 ms  172.29.12.105
  7     *        *        *     Request timed out.
  8     *        *        *     Request timed out.
  9     *        *        *     Request timed out.
 10     *        *        *     Request timed out.
 11     *        *        *     Request timed out.
 12     *        *        *     Request timed out.
 13    18 ms    18 ms    19 ms  151.101.194.133

Trace complete.

C:\Users\nguye>
```

My home network is faster than asu network because the amount of request time out is few and time finished ms is less than asu.

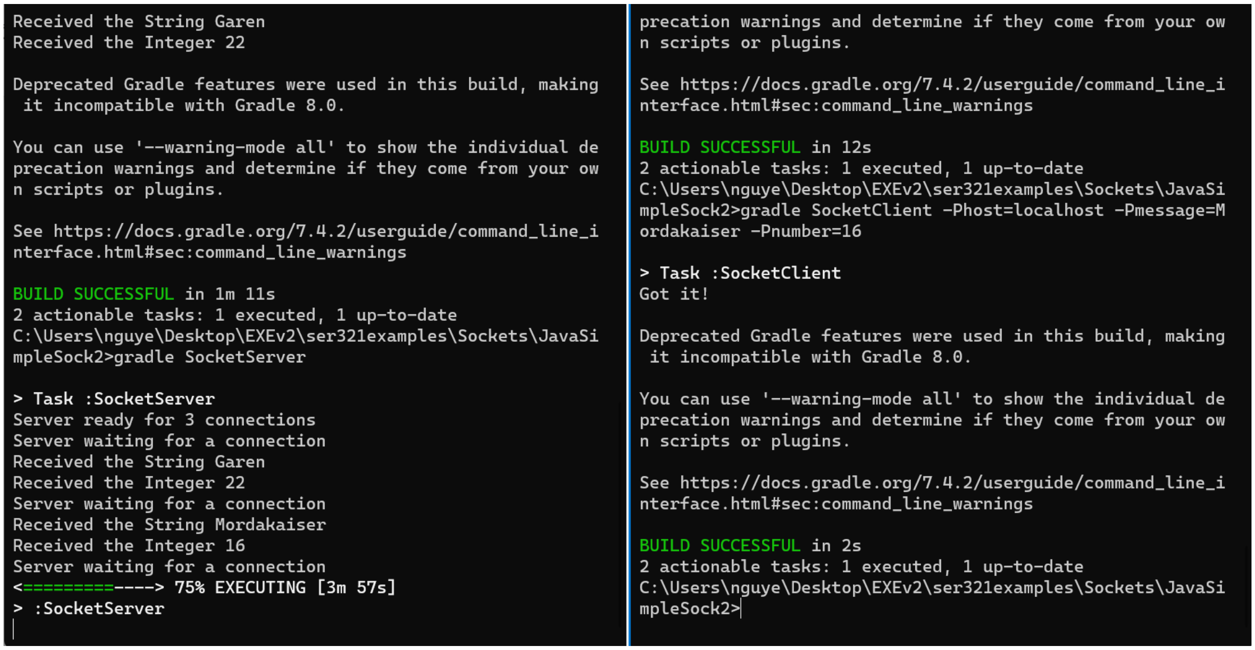
1.5.1. Running things locally

Video: **https://youtu.be/TxMo7_q_QLs**



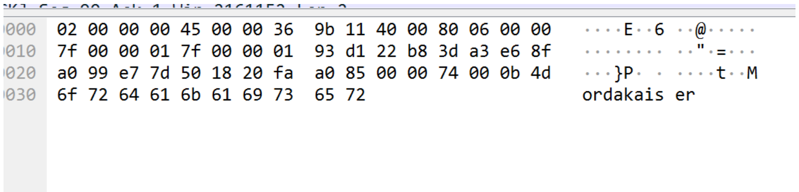
```
stopped Daemons could not be reused, use --status for deta
ils

> Task :SocketServer
Server ready for 3 connections
Server waiting for a connection
Received the String secret
Received the Integer 5
Server waiting for a connection
Received the String mord
Received the Integer 26
Server waiting for a connection
Received the String mord
Received the Integer 26

Deprecated Gradle features were used in this build, making
 it incompatible with Gradle 8.0.

You can use '--warning-mode all' to show the individual de
precation warnings and determine if they come from your ow
n scripts or plugins.

See https://docs.gradle.org/7.4.2/userguide/command_line_i
nterface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 7m 19s
2 actionable tasks: 1 executed, 1 up-to-date
C:\Users\nguye\Desktop\EXEv2\ser321examples\Sockets\JavaSi
mpleSock2>
```

```
precation warnings and determine if they come from your ow
n scripts or plugins.

See https://docs.gradle.org/7.4.2/userguide/command_line_i
nterface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 875ms
2 actionable tasks: 1 executed, 1 up-to-date
C:\Users\nguye\Desktop\EXEv2\ser321examples\Sockets\JavaSi
mpleSock2>gradle SocketClient -Phost=localhost -Pmessage=m
ord -Pnumber=26

> Task :SocketClient
Got it!

Deprecated Gradle features were used in this build, making
 it incompatible with Gradle 8.0.

You can use '--warning-mode all' to show the individual de
precation warnings and determine if they come from your ow
n scripts or plugins.

See https://docs.gradle.org/7.4.2/userguide/command_line_i
nterface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 1s
2 actionable tasks: 1 executed, 1 up-to-date
C:\Users\nguye\Desktop\EXEv2\ser321examples\Sockets\JavaSi
mpleSock2>
```



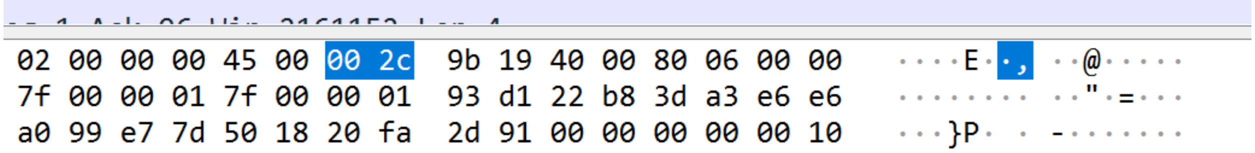
```
Received the String Garen
Received the Integer 22

Deprecated Gradle features were used in this build, making
 it incompatible with Gradle 8.0.

You can use '--warning-mode all' to show the individual de
precation warnings and determine if they come from your ow
n scripts or plugins.

See https://docs.gradle.org/7.4.2/userguide/command_line_i
nterface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 1m 11s
2 actionable tasks: 1 executed, 1 up-to-date
C:\Users\nguye\Desktop\EXEv2\ser321examples\Sockets\JavaSi
mpleSock2>gradle SocketServer

> Task :SocketServer
Server ready for 3 connections
Server waiting for a connection
Received the String Garen
Received the Integer 22
Server waiting for a connection
Received the String Mordakaiser
Received the Integer 16
Server waiting for a connection
<=========----> 75% EXECUTING [3m 57s]
> :SocketServer
```

```
precation warnings and determine if they come from your ow
n scripts or plugins.

See https://docs.gradle.org/7.4.2/userguide/command_line_i
nterface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 12s
2 actionable tasks: 1 executed, 1 up-to-date
C:\Users\nguye\Desktop\EXEv2\ser321examples\Sockets\JavaSi
mpleSock2>gradle SocketClient -Phost=localhost -Pmessage=M
ordakaiser -Pnumber=16

> Task :SocketClient
Got it!

Deprecated Gradle features were used in this build, making
 it incompatible with Gradle 8.0.

You can use '--warning-mode all' to show the individual de
precation warnings and determine if they come from your ow
n scripts or plugins.

See https://docs.gradle.org/7.4.2/userguide/command_line_i
nterface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 2s
2 actionable tasks: 1 executed, 1 up-to-date
C:\Users\nguye\Desktop\EXEv2\ser321examples\Sockets\JavaSi
mpleSock2>
```



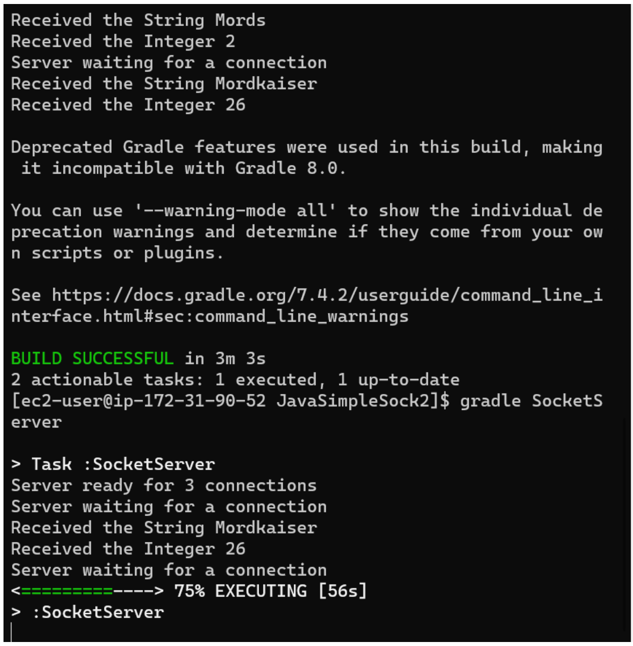
```
000  02 00 00 00 45 00 00 36  9b 11 40 00 80 06 00 00   ····E··6 ··@·····
010  7f 00 00 01 7f 00 00 01  93 d1 22 b8 3d a3 e6 8f   ········ ··"·=···
020  a0 99 e7 7d 50 18 20 fa  a0 85 00 00 74 00 0b 4d   ···}P· · ····t··M
030  6f 72 64 61 6b 61 69 73  65 72                     ordakais er
```

Number is shown as hex value 16 decimal->10 Hex

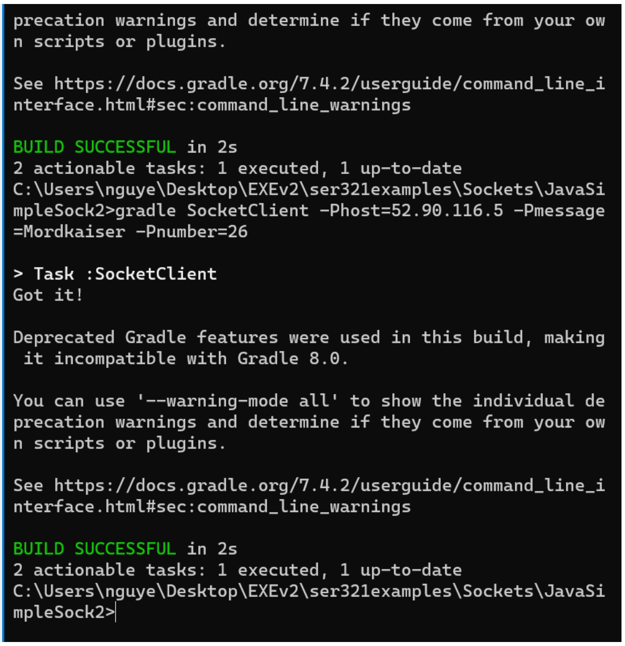


```
02 00 00 00 45 00 00 2c  9b 19 40 00 80 06 00 00   ····E··, ··@·····
7f 00 00 01 7f 00 00 01  93 d1 22 b8 3d a3 e6 e6   ········ ··"·=···
a0 99 e7 7d 50 18 20 fa  2d 91 00 00 00 00 00 10   ···}P· · -·······
```

1.5.2. Server on AWS



```
Received the String Mords
Received the Integer 2
Server waiting for a connection
Received the String Mordkaiser
Received the Integer 26

Deprecated Gradle features were used in this build, making
 it incompatible with Gradle 8.0.

You can use '--warning-mode all' to show the individual de
precation warnings and determine if they come from your ow
n scripts or plugins.

See https://docs.gradle.org/7.4.2/userguide/command_line_i
nterface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 3m 3s
2 actionable tasks: 1 executed, 1 up-to-date
[ec2-user@ip-172-31-90-52 JavaSimpleSock2]$ gradle SocketS
erver

> Task :SocketServer
Server ready for 3 connections
Server waiting for a connection
Received the String Mordkaiser
Received the Integer 26
Server waiting for a connection
<=========----> 75% EXECUTING [56s]
> :SocketServer
```



```
precation warnings and determine if they come from your ow
n scripts or plugins.

See https://docs.gradle.org/7.4.2/userguide/command_line_i
nterface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 2s
2 actionable tasks: 1 executed, 1 up-to-date
C:\Users\nguye\Desktop\EXEv2\ser321examples\Sockets\JavaSi
mpleSock2>gradle SocketClient -Phost=52.90.116.5 -Pmessage
=Mordkaiser -Pnumber=26

> Task :SocketClient
Got it!

Deprecated Gradle features were used in this build, making
 it incompatible with Gradle 8.0.

You can use '--warning-mode all' to show the individual de
precation warnings and determine if they come from your ow
n scripts or plugins.

See https://docs.gradle.org/7.4.2/userguide/command_line_i
nterface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 2s
2 actionable tasks: 1 executed, 1 up-to-date
C:\Users\nguye\Desktop\EXEv2\ser321examples\Sockets\JavaSi
mpleSock2>
```

Changes made when running aws server and client was the ip address specified. Instead of -Phost=localhost it would have been the ip address ec2 had running. Also the port number that was set on aws security being port 8888. The source code was already set to port 8888 so no changes had to be made there.

1.5.3. Client on AWS (2.5 points)

Consider the case you want to run your server locally (so on your home computer) and your client on AWS (you do not have to do this but you can try). Does this work without issues? Can you do it in the same way as in 1.5.2? Why or why not? What is different?

This would still work although the port number would have to change to the ip address of your home computer and the port running would still be the same. It would be the same as 1.5.2.

1.5.4. Client on AWS 2 (3 points)

In this context also explain how the differences in local IP addresses, how your router

plays into all of this. Why can you easily reach your server on AWS with a client running

in your local network but not as easily go the other direction? And what can you do to

reach your server in your local network if you want to reach it from outside your network

(you do not have to do that)? What is the "issue" if you want to run your server locally

and reach it from the "outside world"?

Because the ip on your router is private rather than aws ip network is public. Your router can easily direct traffic to aws server but is a bit harder going the other way from aws client to local server due to your router being private. You will need to port forward your Ip to specifying which port and network to use or run. Issue here would be a security risk due to your information being open to the outside world.