Assignment - 1
Design Document - Computer Networks

20110054
Dhairya Shah
GitHub Repository: https://github.com/DhairyaShah981/Networks_Assignment1

Contents

Introduction

In this assignment, I have tried to implement a python program using sockets that support the five commands: CWD (prints the current directory of the server), LS (prints the list of files/folders in the server folder), cd <> (changes the directory, and prints the changed directory), DWD <file> (downloads the file from the server to the client directory), UPD<file> (uploads a file from the client directory to the server directory).

ReadMe Instructions

My GitHub repository contains two folders: server and client. The client folder contains utilities.py and client.py. The server folder contains server.py, bruh.txt, and a folder named Images and utilities.py. After cloning the repository, you can open two command prompts and first run the server.py and then run the client.py to test the commands. You will get the outputs and prompts on both the client and server. You may note that I am currently implementing server and client on the same operating system, so it works like two processes communicating with each other. (Inter-Process Communication) Implementing the server on a different system can also work fine if we set the IP address accordingly.

Error Statements

Every assignment is incomplete without solving bugs and errors. While running the commands, you may run into some errors covered in the print statements; for example, KeyboardInterrupt, and ConnectionError are considered socket errors, or if your input is invalid, it prints 'Invalid command.' Specifically, in the cd command, if the path does not exist, it prints 'No such directory exists!'. In dwd and upd if the specified file is unavailable, it prints 'Socket Error,' and you can check the server terminal, which will show 'No such file exist.'

Layers

The system uses a template N layer model for Remote File Sharing (RFS) client and server. The top 3 layers are the File Service, Crypto Service, and the Networking layer:

1) **File Sharing**: To implement the above 5 commands, I am using a few inbuilt python functions from the os library, which, when called, invoke some system calls, which eventually give us the information about the server directory. The file-sharing layer is responsible for calling these Operating Systems APIs (Application Programming Interface), and implementing all the five RFS commands. The file-sharing layer moves the files and messages between the local and remote file systems. The files and messages are transferred from server to client and vice versa using sendall(), recv(1024), send_msg(), recv_msg().

   The protocol design for this layer is as follows. For the cwd command, I used the os.getcwd() function, which gives the current directory, and then sent this as a message to the client. For the ls command, I used os.listdir() function, which gives the list of files/folders in the current directory and then sent this as a string message to the client. For the cd command, I changed the directory using os.chdir(path) and then sent the changed directory as a string message to the client. For the dwd command, I am first searching for the given file in the server directory, reading the whole file as string, sending it to the client as packets of size 1024, receiving the data in the client and writing the data in the new file made in the client folder. For the upd command, I am first searching for the given file in the client directory, reading the whole file as string, sending it to the server as packets of 1024, receiving the data in server and writing the data in the new file made in the server folder. All the functions for the first 3 commands are in the utilities.py, while for the upd and dwd the process is implemented in the server and client itself.

2) **Crypto Service**: Originally, the file transfer was implemented using only the utf-8 encoding, which encodes the characters into bytes. While decoding, the bytes were again converted to characters. As given in the assignment, I have introduced another encryption layer visible in the Wireshark dump. There are three modes of encryption available for the user: Plaintext, Substitute, and Transpose. The user is asked to input an integer from 1, 2, and 3 corresponding to the three encryption layers. Now in both server and client, I have implemented all the 5 commands in 3 ‘if conditions’ (for three encryption layers).
   a) **Plaintext**: In this encryption mode, the text doesn't change. Hence I have not defined a particular encryption function for this and implemented all the commands without encrypting and decrypting.

b) **Substitute**: In this encryption mode, all the characters are offset by 2. While decrypting, the characters are offset by -2. Both the encrypt and decrypt functions are implemented in the utilities.py. In the functions, I loop through the message character by character and offset it by manipulating the ASCII values.

c) **Transpose**: In this encryption mode, the message is simply reversed. There are two interpretations: should I reverse it completely or reverse it word by word? I realized later that reversing it word by word makes more sense in the encryption paradigm. In the function, I loop through the message word by word, reverse them, and append them to the encrypted message. Also, the encrypt and decrypt functions will be the same for the transpose encryption.

3) **Networking Layer**: My program runs on the foundations of the socket library, and that's why we call it a sockets program. When we create a sockets program, we use something that generally is called a sockets library. A sockets library consists of compile-time structures, statically linked support modules, and run-time support modules. In the client.py file, when I create a socket using: socket.socket(socket.AF_INET, socket.SOCK_STREAM), the first argument refers to the IPv4 address, and the second argument refers to the streaming socket, which is the TCP (Transmission Control Protocol). The TCP manages the networking layer whenever we invoke the in-built functions of the socket library. The TCP facilitates communication among the machines using their IP address through sockets. Apart from TCP, I could have used UDP, but in the lectures, I learned that UDP is an unreliable and connectionless protocol. In UDP, there is no function to ensure that data is received in the same order as it was transmitted.

<u>Challenges Faced</u>

The assignment was lengthy, not in terms of writing code but debugging. My friends and I spent hours debugging and managing the border cases. At one point, I felt the border cases are more than the mainstream cases. The first three commands were easy to implement, but I faced issues managing the edge cases of different path names while adding the encryption layer. Dwd and upd functions were also easy to implement at first without caring about the encryption and edge cases. Starting with the simple .txt file, which can only be read as strings by reading it as ‘r’ and writing as ‘w’ instead of ‘rb’ and ‘wb,’ for which it doesn’t work. I had to implement it using strings because my encryption layer only accepts strings.

If the file name consisted of more than 1 word separated by spaces, my logic only took the first word from them, which threw an error at the start. That's why I traversed through the path and appended all the words in my filename. While downloading and uploading, I was prompted when the download was successful from the server, but the terminal didn't print that even after downloading. That prompt was written in the file itself. I solved this by adding an appropriate break statement. Transferring the messages in bytes and strings and encrypting and decrypting them at appropriate times was very cumbersome, and 1 small mistake cost me hours of debugging.

Limitations and Further Improvements

My current program only supports the downloading and uploading .txt files because I am using ‘r’ and ‘w’, which can only read and write as strings. I tried implementing .png and .jpeg files by converting bytes to strings and vice versa, but the files got corrupted during the process. I found an online resource that says we can implement file transfer using a python library named ‘Base64’ which can convert an image to a string and vice versa. I haven't implemented it because of time constraints and some new bugs.

References

1) I used the skeleton code of the server, client, and basic utilities like send_msg(), and recv_msg() from the book suggested by the professor. The skeleton code helped me how to set up an echo server and client on the same system.
2) For file transfer, I referred to this resource.
3) For the in-built os functions, I referred to the python documentation.

Command Prompt Screenshots of Output

Plaintext: cwd

```
Connected to 192.168.56.1:4040
Select any number from 1, 2, and 3: 1
Type message in encrypted mode, enter to send, 'exit' to quit
cwd
Sent message: cwd
Received echo: C:\Users\DELL\cnass\server
Closed connection to server
```

Plaintext: ls

```
Connected to 192.168.56.1:4040
Select any number from 1, 2, and 3: 1
Type message in encrypted mode, enter to send, 'exit' to quit
ls
Sent message: ls
Received echo: ['bruh.txt', 'Images', 'server.py', 'utilities.py', '__pycache__']
Closed connection to server
```

Plaintext: cd Images

```
Connected to 192.168.56.1:4040
Select any number from 1, 2, and 3: 1
Type message in encrypted mode, enter to send, 'exit' to quit
cd Images
Sent message: cd Images
Received echo: C:\Users\DELL\cnass\server\Images
Closed connection to server
```

Plaintext: dwd bruh.txt

```
Connected to 192.168.56.1:4040
Select any number from 1, 2, and 3: 1
Type message in encrypted mode, enter to send, 'exit' to quit
dwd bruh.txt
Download Completed!
Closed connection to server
```

Plaintext: upd bruh.txt

```
Connected to 192.168.56.1:4040
Select any number from 1, 2, and 3: 1
Type message in encrypted mode, enter to send, 'exit' to quit
upd bruh.txt
Upload Completed!
Closed connection to server
```

Substitute: cwd

```
Connected to 192.168.56.1:4040
Select any number from 1, 2, and 3: 2
Type message, enter to send, 'exit' to quit
cwd
Sent message: cwd
Received echo: C:\Users\DELL\cnass\server
Closed connection to server
```

Substitute: ls

```
Connected to 192.168.56.1:4040
Select any number from 1, 2, and 3: 2
Type message, enter to send, 'exit' to quit
ls
Sent message: ls
Received echo: ['bruh.txt', 'Images', 'server.py', 'utilities.py', '__pycache__']
Closed connection to server
```

Substitute: cd Images

```
Connected to 192.168.56.1:4040
Select any number from 1, 2, and 3: 2
Type message, enter to send, 'exit' to quit
cd Images
Sent message: cd Images
Received echo: C:\Users\DELL\cnass\server\Images
Closed connection to server
```

Substitute: dwd bruh.txt

```
Connected to 192.168.56.1:4040
Select any number from 1, 2, and 3: 2
Type message, enter to send, 'exit' to quit
dwd bruh.txt
Download Completed!
Closed connection to server
```

Substitute: upd bruh.txt

```
Connected to 192.168.56.1:4040
Select any number from 1, 2, and 3: 2
Type message, enter to send, 'exit' to quit
upd bruh.txt
Upload Completed!
Closed connection to server
```

Transpose: cwd

```
Connected to 192.168.56.1:4040
Select any number from 1, 2, and 3: 3
Type message in encrypted mode, enter to send, 'exit' to quit
cwd
Sent message: cwd
Received echo: C:\Users\DELL\cnass\server
Closed connection to server
```

Transpose: ls

```
Connected to 192.168.56.1:4040
Select any number from 1, 2, and 3: 3
Type message in encrypted mode, enter to send, 'exit' to quit
ls
Sent message: ls
Received echo: ['bruh.txt', 'Images', 'server.py', 'utilities.py', '__pycache__']
Closed connection to server
```

Transpose: cd Images

```
Connected to 192.168.56.1:4040
Select any number from 1, 2, and 3: 3
Type message in encrypted mode, enter to send, 'exit' to quit
cd Images
Sent message: cd Images
Received echo: C:\Users\DELL\cnass\server\Images
Closed connection to server
```

Transpose: dwd bruh.txt

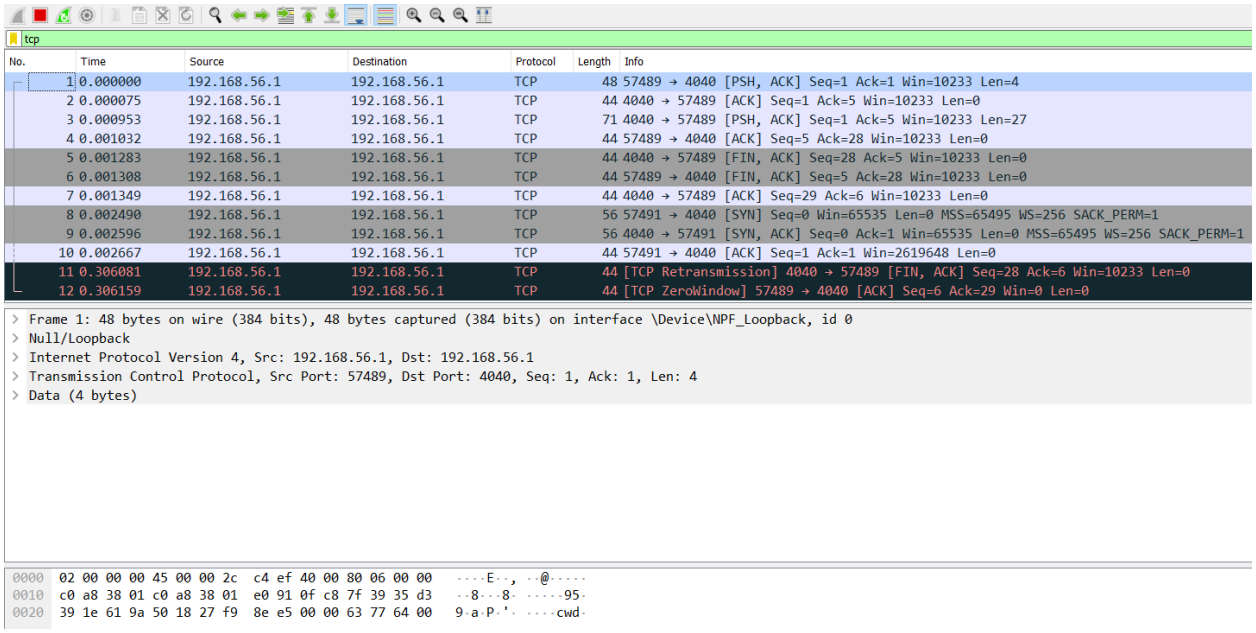
```
Connected to 192.168.56.1:4040
Select any number from 1, 2, and 3: 3
Type message in encrypted mode, enter to send, 'exit' to quit
dwd bruh.txt
Download Completed!
Closed connection to server
```

Transpose: upd bruh.txt

```
Connected to 192.168.56.1:4040
Select any number from 1, 2, and 3: 3
Type message in encrypted mode, enter to send, 'exit' to quit
upd bruh.txt
Upload Completed!
Closed connection to server
```

Wireshark Screenshots of Packets

Plaintext: cwd command



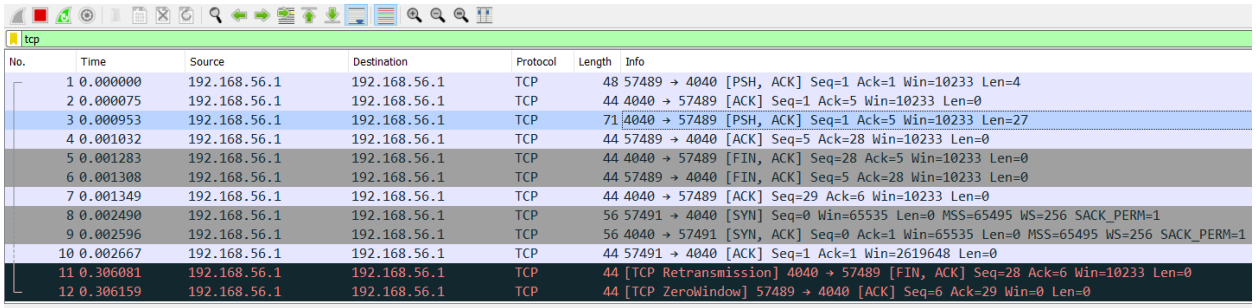
tcp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.56.1	192.168.56.1	TCP	48	57489 → 4040 [PSH, ACK] Seq=1 Ack=1 Win=10233 Len=4
2	0.000075	192.168.56.1	192.168.56.1	TCP	44	4040 → 57489 [ACK] Seq=1 Ack=5 Win=10233 Len=0
3	0.000953	192.168.56.1	192.168.56.1	TCP	71	4040 → 57489 [PSH, ACK] Seq=1 Ack=5 Win=10233 Len=27
4	0.001032	192.168.56.1	192.168.56.1	TCP	44	57489 → 4040 [ACK] Seq=5 Ack=28 Win=10233 Len=0
5	0.001283	192.168.56.1	192.168.56.1	TCP	44	4040 → 57489 [FIN, ACK] Seq=28 Ack=5 Win=10233 Len=0
6	0.001308	192.168.56.1	192.168.56.1	TCP	44	57489 → 4040 [FIN, ACK] Seq=5 Ack=28 Win=10233 Len=0
7	0.001349	192.168.56.1	192.168.56.1	TCP	44	4040 → 57489 [ACK] Seq=29 Ack=6 Win=10233 Len=0
8	0.002490	192.168.56.1	192.168.56.1	TCP	56	57491 → 4040 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
9	0.002596	192.168.56.1	192.168.56.1	TCP	56	4040 → 57491 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
10	0.002667	192.168.56.1	192.168.56.1	TCP	44	57491 → 4040 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
11	0.306081	192.168.56.1	192.168.56.1	TCP	44	[TCP Retransmission] 4040 → 57489 [FIN, ACK] Seq=28 Ack=6 Win=10233 Len=0
12	0.306159	192.168.56.1	192.168.56.1	TCP	44	[TCP ZeroWindow] 57489 → 4040 [ACK] Seq=6 Ack=29 Win=0 Len=0

```
> Frame 1: 48 bytes on wire (384 bits), 48 bytes captured (384 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1
> Transmission Control Protocol, Src Port: 57489, Dst Port: 4040, Seq: 1, Ack: 1, Len: 4
> Data (4 bytes)
```

```
0000  02 00 00 00 45 00 00 2c  c4 ef 40 00 80 06 00 00   ····E··, ··@·····
0010  c0 a8 38 01 c0 a8 38 01  e0 91 0f c8 7f 39 35 d3   ··8···8· ·····95·
0020  39 1e 61 9a 50 18 27 f9  8e e5 00 00 63 77 64 00   9·a·P·'· ····cwd·
```

Plaintext: cwd response



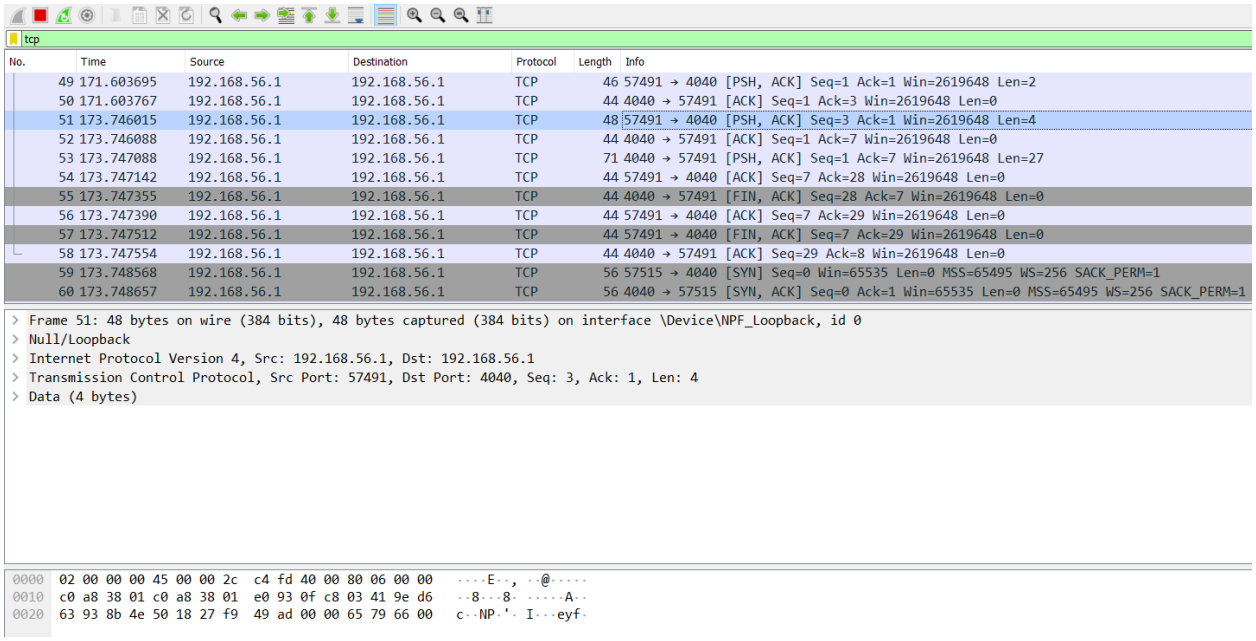
tcp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.56.1	192.168.56.1	TCP	48	57489 → 4040 [PSH, ACK] Seq=1 Ack=1 Win=10233 Len=4
2	0.000075	192.168.56.1	192.168.56.1	TCP	44	4040 → 57489 [ACK] Seq=1 Ack=5 Win=10233 Len=0
3	0.000953	192.168.56.1	192.168.56.1	TCP	71	4040 → 57489 [PSH, ACK] Seq=1 Ack=5 Win=10233 Len=27
4	0.001032	192.168.56.1	192.168.56.1	TCP	44	57489 → 4040 [ACK] Seq=5 Ack=28 Win=10233 Len=0
5	0.001283	192.168.56.1	192.168.56.1	TCP	44	4040 → 57489 [FIN, ACK] Seq=28 Ack=5 Win=10233 Len=0
6	0.001308	192.168.56.1	192.168.56.1	TCP	44	57489 → 4040 [FIN, ACK] Seq=5 Ack=28 Win=10233 Len=0
7	0.001349	192.168.56.1	192.168.56.1	TCP	44	4040 → 57489 [ACK] Seq=29 Ack=6 Win=10233 Len=0
8	0.002490	192.168.56.1	192.168.56.1	TCP	56	57491 → 4040 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
9	0.002596	192.168.56.1	192.168.56.1	TCP	56	4040 → 57491 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
10	0.002667	192.168.56.1	192.168.56.1	TCP	44	57491 → 4040 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
11	0.306081	192.168.56.1	192.168.56.1	TCP	44	[TCP Retransmission] 4040 → 57489 [FIN, ACK] Seq=28 Ack=6 Win=10233 Len=0
12	0.306159	192.168.56.1	192.168.56.1	TCP	44	[TCP ZeroWindow] 57489 → 4040 [ACK] Seq=6 Ack=29 Win=0 Len=0

```
> Frame 3: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1
> Transmission Control Protocol, Src Port: 4040, Dst Port: 57489, Seq: 1, Ack: 5, Len: 27
> Data (27 bytes)
```

```
0000  02 00 00 00 45 00 00 43  c4 f1 40 00 80 06 00 00   ····E··C ··@·····
0010  c0 a8 38 01 c0 a8 38 01  0f c8 e0 91 39 1e 61 9a   ··8···8· ····9·a·
0020  7f 39 35 d7 50 18 27 f9  5f 60 00 00 43 3a 5c 55   ·95·P·'· _`··C:\U
0030  73 65 72 73 5c 44 45 4c  4c 5c 63 6e 61 73 73 5c   sers\DEL L\cnass\
0040  73 65 72 76 65 72 00                              server·
```

Substitute: cwd command



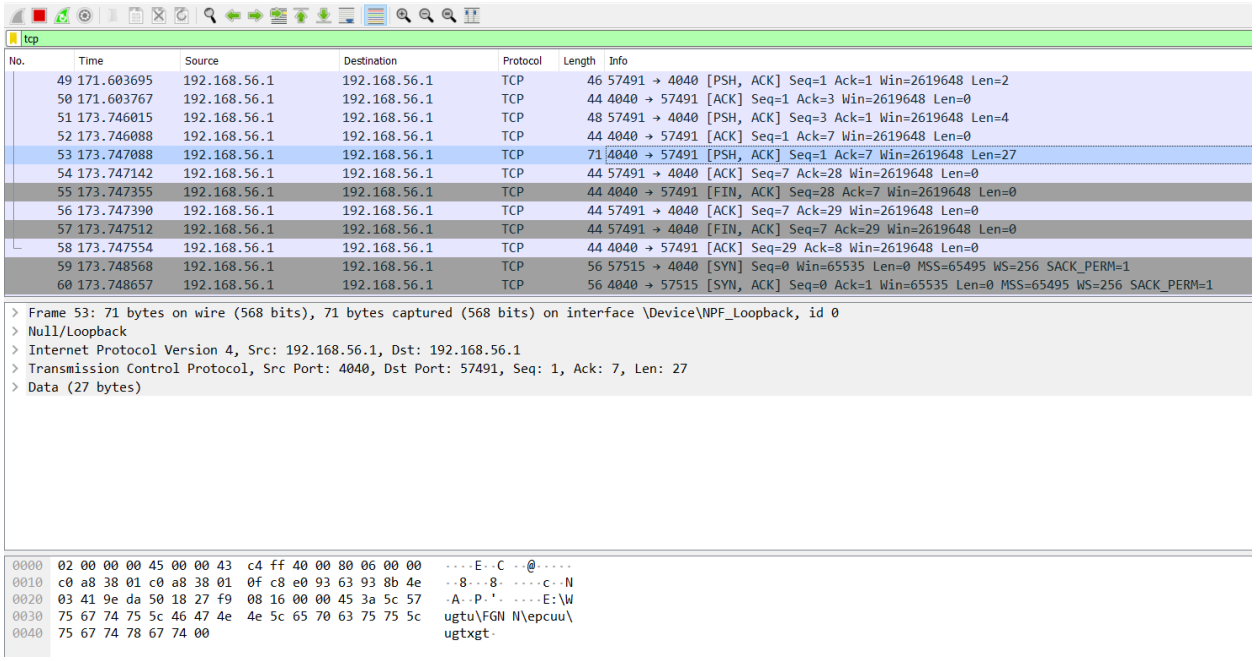
tcp

No.	Time	Source	Destination	Protocol	Length	Info
49	171.603695	192.168.56.1	192.168.56.1	TCP	46	57491 → 4040 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=2
50	171.603767	192.168.56.1	192.168.56.1	TCP	44	4040 → 57491 [ACK] Seq=1 Ack=3 Win=2619648 Len=0
51	173.746015	192.168.56.1	192.168.56.1	TCP	48	57491 → 4040 [PSH, ACK] Seq=3 Ack=1 Win=2619648 Len=4
52	173.746088	192.168.56.1	192.168.56.1	TCP	44	4040 → 57491 [ACK] Seq=1 Ack=7 Win=2619648 Len=0
53	173.747088	192.168.56.1	192.168.56.1	TCP	71	4040 → 57491 [PSH, ACK] Seq=1 Ack=7 Win=2619648 Len=27
54	173.747142	192.168.56.1	192.168.56.1	TCP	44	57491 → 4040 [ACK] Seq=7 Ack=28 Win=2619648 Len=0
55	173.747355	192.168.56.1	192.168.56.1	TCP	44	4040 → 57491 [FIN, ACK] Seq=28 Ack=7 Win=2619648 Len=0
56	173.747390	192.168.56.1	192.168.56.1	TCP	44	57491 → 4040 [ACK] Seq=7 Ack=29 Win=2619648 Len=0
57	173.747512	192.168.56.1	192.168.56.1	TCP	44	57491 → 4040 [FIN, ACK] Seq=7 Ack=29 Win=2619648 Len=0
58	173.747554	192.168.56.1	192.168.56.1	TCP	44	4040 → 57491 [ACK] Seq=29 Ack=8 Win=2619648 Len=0
59	173.748568	192.168.56.1	192.168.56.1	TCP	56	57515 → 4040 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
60	173.748657	192.168.56.1	192.168.56.1	TCP	56	4040 → 57515 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1

```
> Frame 51: 48 bytes on wire (384 bits), 48 bytes captured (384 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1
> Transmission Control Protocol, Src Port: 57491, Dst Port: 4040, Seq: 3, Ack: 1, Len: 4
> Data (4 bytes)
```

```
0000  02 00 00 00 45 00 00 2c  c4 fd 40 00 80 06 00 00   ····E··, ··@·····
0010  c0 a8 38 01 c0 a8 38 01  e0 93 0f c8 03 41 9e d6   ··8···8· ·····A··
0020  63 93 8b 4e 50 18 27 f9  49 ad 00 00 65 79 66 00   c··NP·'· I···eyf·
```

Substitute: cwd response



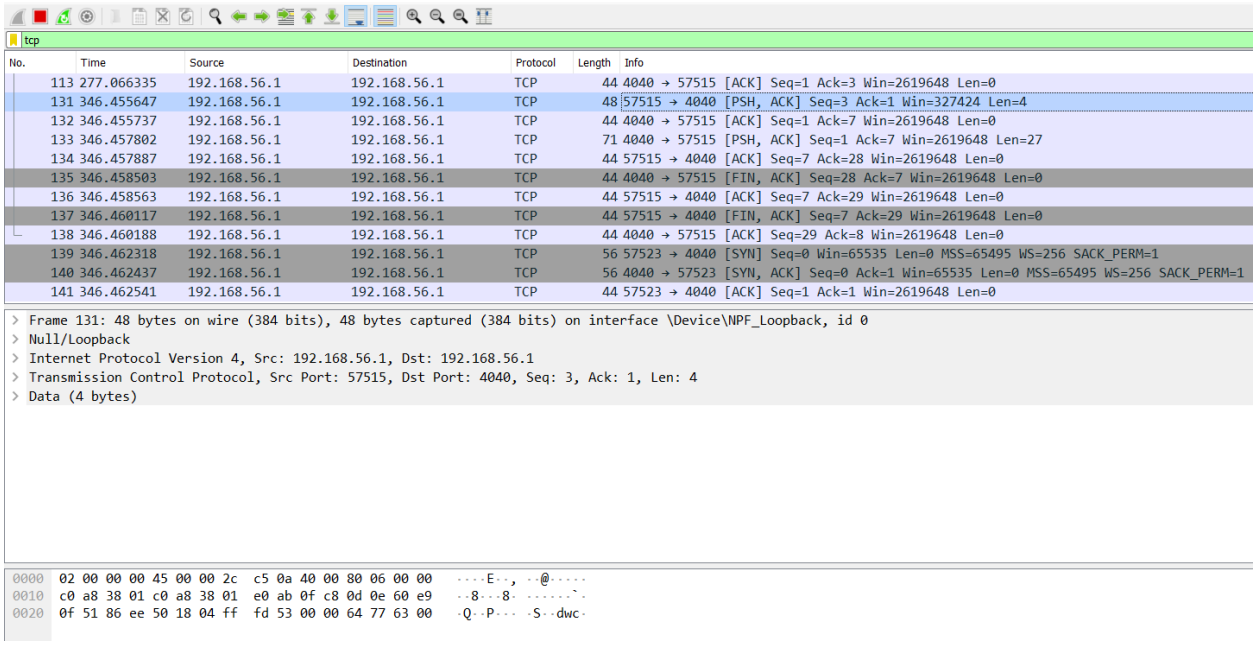
tcp

No.	Time	Source	Destination	Protocol	Length	Info
49	171.603695	192.168.56.1	192.168.56.1	TCP	46	57491 → 4040 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=2
50	171.603767	192.168.56.1	192.168.56.1	TCP	44	4040 → 57491 [ACK] Seq=1 Ack=3 Win=2619648 Len=0
51	173.746015	192.168.56.1	192.168.56.1	TCP	48	57491 → 4040 [PSH, ACK] Seq=3 Ack=1 Win=2619648 Len=4
52	173.746088	192.168.56.1	192.168.56.1	TCP	44	4040 → 57491 [ACK] Seq=1 Ack=7 Win=2619648 Len=0
53	173.747088	192.168.56.1	192.168.56.1	TCP	71	4040 → 57491 [PSH, ACK] Seq=1 Ack=7 Win=2619648 Len=27
54	173.747142	192.168.56.1	192.168.56.1	TCP	44	57491 → 4040 [ACK] Seq=7 Ack=28 Win=2619648 Len=0
55	173.747355	192.168.56.1	192.168.56.1	TCP	44	4040 → 57491 [FIN, ACK] Seq=28 Ack=7 Win=2619648 Len=0
56	173.747390	192.168.56.1	192.168.56.1	TCP	44	57491 → 4040 [ACK] Seq=7 Ack=29 Win=2619648 Len=0
57	173.747512	192.168.56.1	192.168.56.1	TCP	44	57491 → 4040 [FIN, ACK] Seq=7 Ack=29 Win=2619648 Len=0
58	173.747554	192.168.56.1	192.168.56.1	TCP	44	4040 → 57491 [ACK] Seq=29 Ack=8 Win=2619648 Len=0
59	173.748568	192.168.56.1	192.168.56.1	TCP	56	57515 → 4040 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
60	173.748657	192.168.56.1	192.168.56.1	TCP	56	4040 → 57515 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1

```
> Frame 53: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1
> Transmission Control Protocol, Src Port: 4040, Dst Port: 57491, Seq: 1, Ack: 7, Len: 27
> Data (27 bytes)
```

```
0000  02 00 00 00 45 00 00 43  c4 ff 40 00 80 06 00 00   ····E··C ··@·····
0010  c0 a8 38 01 c0 a8 38 01  0f c8 e0 93 63 93 8b 4e   ··8···8· ····c··N
0020  03 41 9e da 50 18 27 f9  08 16 00 00 45 3a 5c 57   ·A··P·'· ····E:\W
0030  75 67 74 75 5c 46 47 4e  4e 5c 65 70 63 75 75 5c   ugtu\FGN N\epcuu\
0040  75 67 74 78 67 74 00                              ugtxgt·
```

Transpose: cwd command



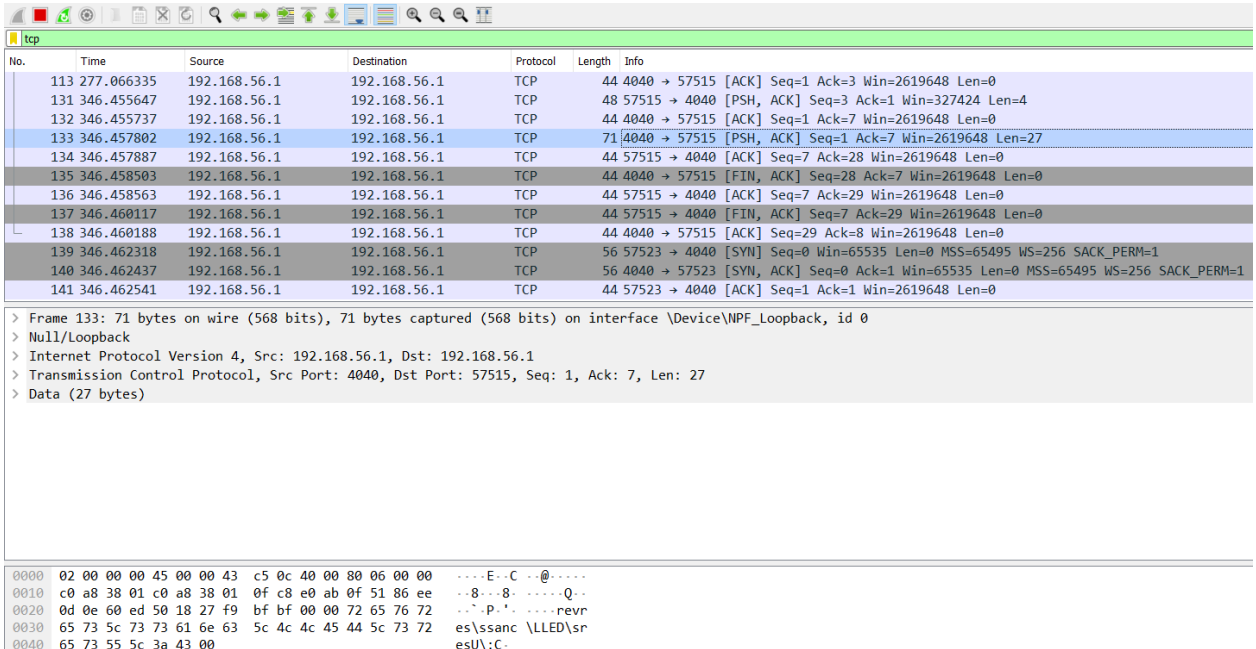
tcp

No.	Time	Source	Destination	Protocol	Length	Info
113	277.066335	192.168.56.1	192.168.56.1	TCP	44	4040 → 57515 [ACK] Seq=1 Ack=3 Win=2619648 Len=0
131	346.455647	192.168.56.1	192.168.56.1	TCP	48	57515 → 4040 [PSH, ACK] Seq=3 Ack=1 Win=327424 Len=4
132	346.455737	192.168.56.1	192.168.56.1	TCP	44	4040 → 57515 [ACK] Seq=1 Ack=7 Win=2619648 Len=0
133	346.457802	192.168.56.1	192.168.56.1	TCP	71	4040 → 57515 [PSH, ACK] Seq=1 Ack=7 Win=2619648 Len=27
134	346.457887	192.168.56.1	192.168.56.1	TCP	44	57515 → 4040 [ACK] Seq=7 Ack=28 Win=2619648 Len=0
135	346.458503	192.168.56.1	192.168.56.1	TCP	44	4040 → 57515 [FIN, ACK] Seq=28 Ack=7 Win=2619648 Len=0
136	346.458563	192.168.56.1	192.168.56.1	TCP	44	57515 → 4040 [ACK] Seq=7 Ack=29 Win=2619648 Len=0
137	346.460117	192.168.56.1	192.168.56.1	TCP	44	57515 → 4040 [FIN, ACK] Seq=7 Ack=29 Win=2619648 Len=0
138	346.460188	192.168.56.1	192.168.56.1	TCP	44	4040 → 57515 [ACK] Seq=29 Ack=8 Win=2619648 Len=0
139	346.462318	192.168.56.1	192.168.56.1	TCP	56	57523 → 4040 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
140	346.462437	192.168.56.1	192.168.56.1	TCP	56	4040 → 57523 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
141	346.462541	192.168.56.1	192.168.56.1	TCP	44	57523 → 4040 [ACK] Seq=1 Ack=1 Win=2619648 Len=0

```
> Frame 131: 48 bytes on wire (384 bits), 48 bytes captured (384 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1
> Transmission Control Protocol, Src Port: 57515, Dst Port: 4040, Seq: 3, Ack: 1, Len: 4
> Data (4 bytes)
```

```
0000  02 00 00 00 45 00 00 2c  c5 0a 40 00 80 06 00 00   ····E··, ··@·····
0010  c0 a8 38 01 c0 a8 38 01  e0 ab 0f c8 0d 0e 60 e9   ··8···8· ·····`·
0020  0f 51 86 ee 50 18 04 ff  fd 53 00 00 64 77 63 00   ·Q··P··· ·S··dwc·
```

Transpose: cwd response



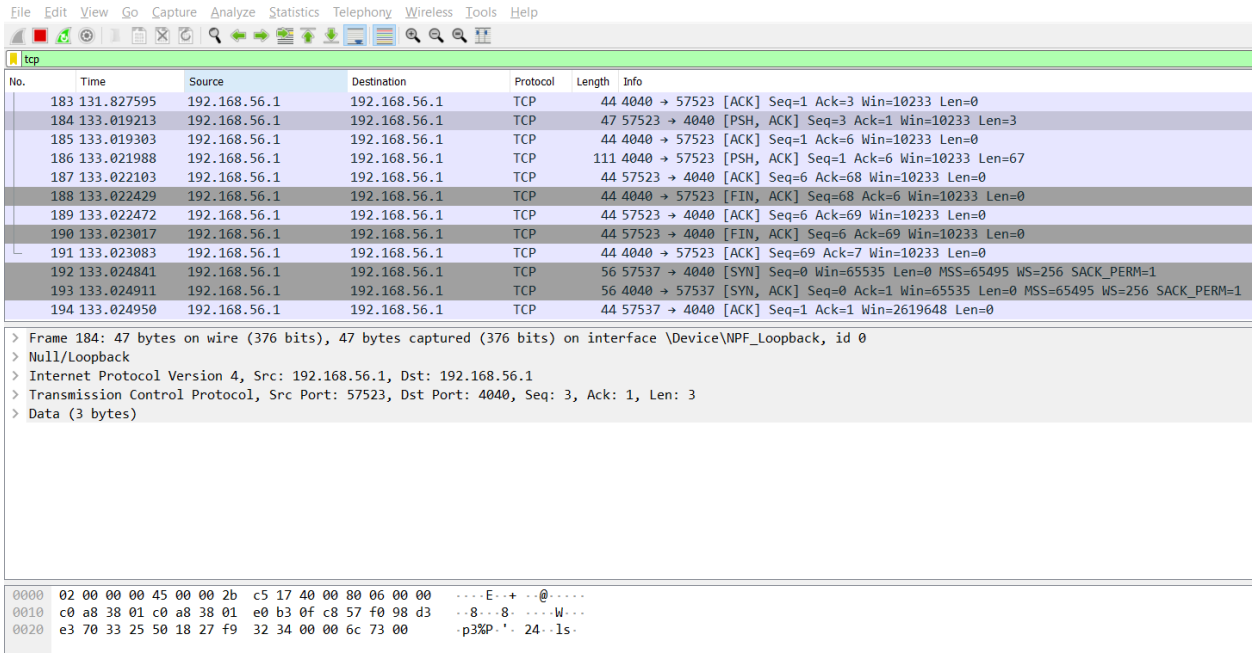
tcp

No.	Time	Source	Destination	Protocol	Length	Info
113	277.066335	192.168.56.1	192.168.56.1	TCP	44	4040 → 57515 [ACK] Seq=1 Ack=3 Win=2619648 Len=0
131	346.455647	192.168.56.1	192.168.56.1	TCP	48	57515 → 4040 [PSH, ACK] Seq=3 Ack=1 Win=327424 Len=4
132	346.455737	192.168.56.1	192.168.56.1	TCP	44	4040 → 57515 [ACK] Seq=1 Ack=7 Win=2619648 Len=0
133	346.457802	192.168.56.1	192.168.56.1	TCP	71	4040 → 57515 [PSH, ACK] Seq=1 Ack=7 Win=2619648 Len=27
134	346.457887	192.168.56.1	192.168.56.1	TCP	44	57515 → 4040 [ACK] Seq=7 Ack=28 Win=2619648 Len=0
135	346.458503	192.168.56.1	192.168.56.1	TCP	44	4040 → 57515 [FIN, ACK] Seq=28 Ack=7 Win=2619648 Len=0
136	346.458563	192.168.56.1	192.168.56.1	TCP	44	57515 → 4040 [ACK] Seq=7 Ack=29 Win=2619648 Len=0
137	346.460117	192.168.56.1	192.168.56.1	TCP	44	57515 → 4040 [FIN, ACK] Seq=7 Ack=29 Win=2619648 Len=0
138	346.460188	192.168.56.1	192.168.56.1	TCP	44	4040 → 57515 [ACK] Seq=29 Ack=8 Win=2619648 Len=0
139	346.462318	192.168.56.1	192.168.56.1	TCP	56	57523 → 4040 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
140	346.462437	192.168.56.1	192.168.56.1	TCP	56	4040 → 57523 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
141	346.462541	192.168.56.1	192.168.56.1	TCP	44	57523 → 4040 [ACK] Seq=1 Ack=1 Win=2619648 Len=0

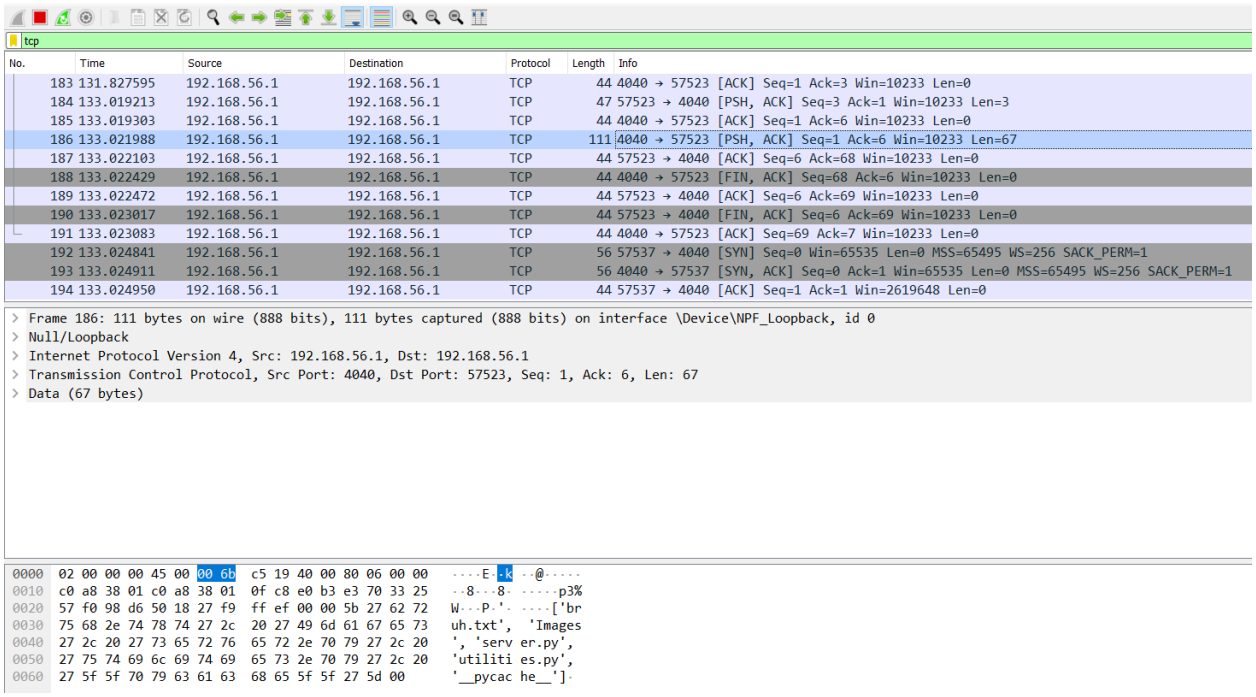
```
> Frame 133: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1
> Transmission Control Protocol, Src Port: 4040, Dst Port: 57515, Seq: 1, Ack: 7, Len: 27
> Data (27 bytes)
```

```
0000  02 00 00 00 45 00 00 43  c5 0c 40 00 80 06 00 00   ····E··C ··@·····
0010  c0 a8 38 01 c0 a8 38 01  0f c8 e0 ab 0f 51 86 ee   ··8···8· ·····Q··
0020  0d 0e 60 ed 50 18 27 f9  bf bf 00 00 72 65 76 72   ··`·P·'· ····revr
0030  65 73 5c 73 73 61 6e 63  5c 4c 4c 45 44 5c 73 72   es\ssanc \LLED\sr
0040  65 73 55 5c 3a 43 00                              esU\:C·
```

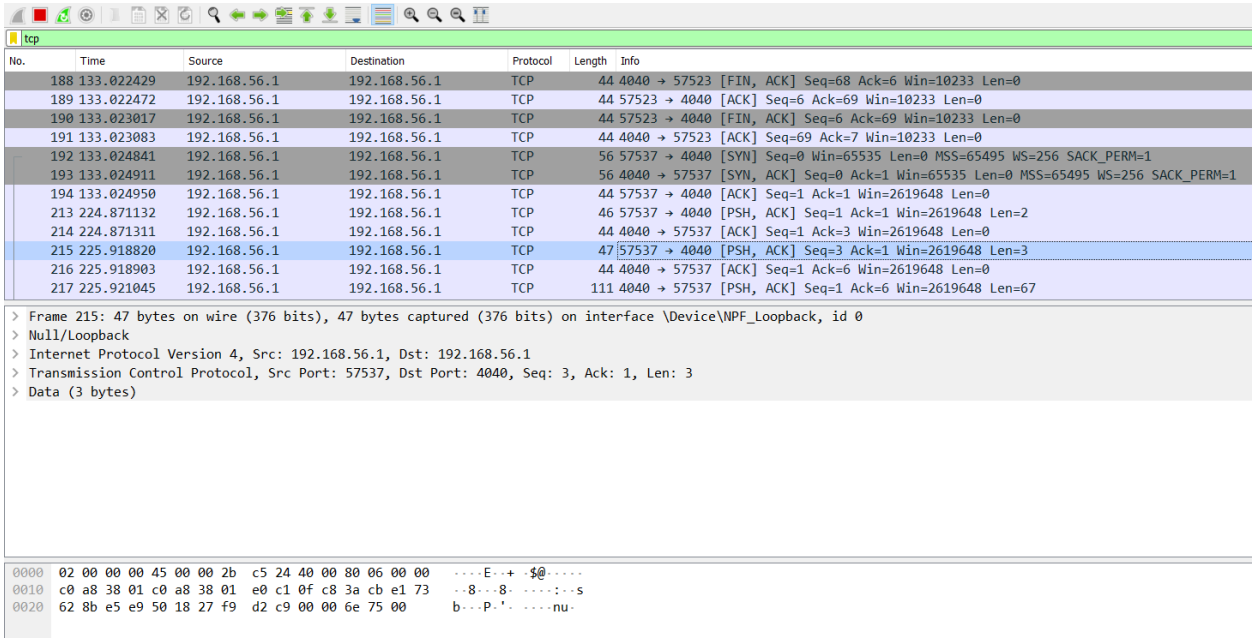
Plaintext: ls command



Plaintext: ls response



Substitute: ls command



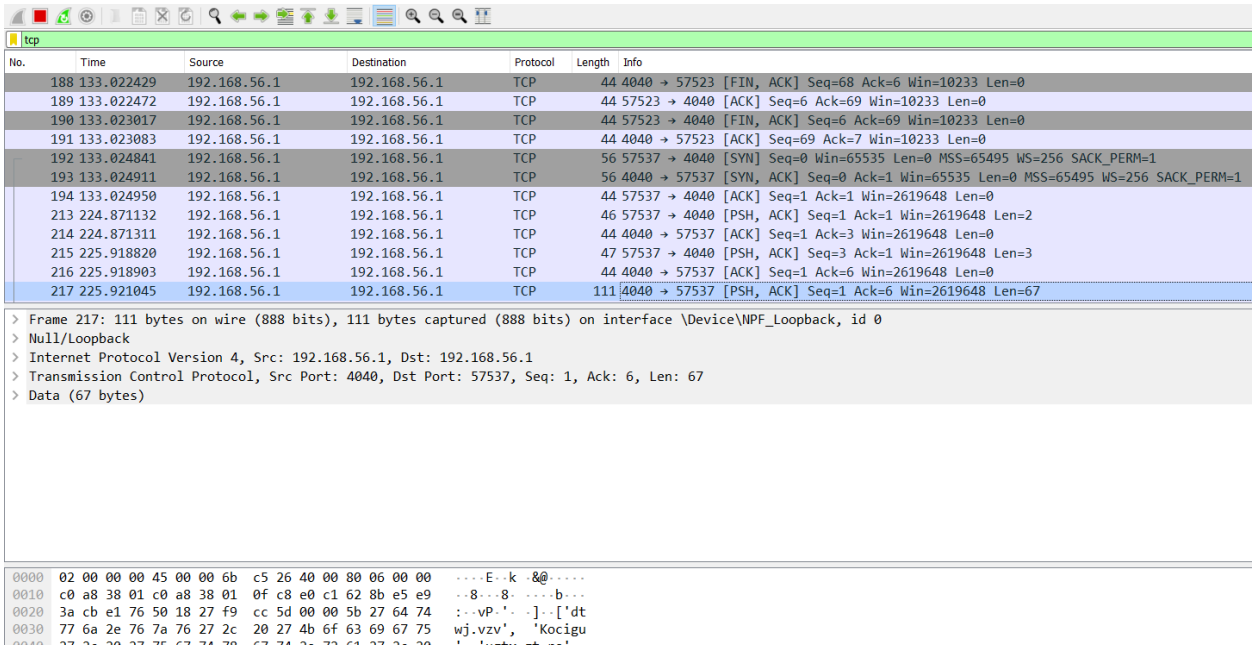
tcp

No.	Time	Source	Destination	Protocol	Length	Info
188	133.022429	192.168.56.1	192.168.56.1	TCP	44	4040 → 57523 [FIN, ACK] Seq=68 Ack=6 Win=10233 Len=0
189	133.022472	192.168.56.1	192.168.56.1	TCP	44	57523 → 4040 [ACK] Seq=6 Ack=69 Win=10233 Len=0
190	133.023017	192.168.56.1	192.168.56.1	TCP	44	57523 → 4040 [FIN, ACK] Seq=6 Ack=69 Win=10233 Len=0
191	133.023083	192.168.56.1	192.168.56.1	TCP	44	4040 → 57523 [ACK] Seq=69 Ack=7 Win=10233 Len=0
192	133.024841	192.168.56.1	192.168.56.1	TCP	56	57537 → 4040 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
193	133.024911	192.168.56.1	192.168.56.1	TCP	56	4040 → 57537 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
194	133.024950	192.168.56.1	192.168.56.1	TCP	44	57537 → 4040 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
213	224.871132	192.168.56.1	192.168.56.1	TCP	46	57537 → 4040 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=2
214	224.871311	192.168.56.1	192.168.56.1	TCP	44	4040 → 57537 [ACK] Seq=1 Ack=3 Win=2619648 Len=0
215	225.918820	192.168.56.1	192.168.56.1	TCP	47	57537 → 4040 [PSH, ACK] Seq=3 Ack=1 Win=2619648 Len=3
216	225.918903	192.168.56.1	192.168.56.1	TCP	44	4040 → 57537 [ACK] Seq=1 Ack=6 Win=2619648 Len=0
217	225.921045	192.168.56.1	192.168.56.1	TCP	111	4040 → 57537 [PSH, ACK] Seq=1 Ack=6 Win=2619648 Len=67

```
> Frame 215: 47 bytes on wire (376 bits), 47 bytes captured (376 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1
> Transmission Control Protocol, Src Port: 57537, Dst Port: 4040, Seq: 3, Ack: 1, Len: 3
> Data (3 bytes)
```

```
0000  02 00 00 00 45 00 00 2b  c5 24 40 00 80 06 00 00   ····E··+ ·$@·····
0010  c0 a8 38 01 c0 a8 38 01  e0 c1 0f c8 3a cb e1 73   ··8···8· ····:··s
0020  62 8b e5 e9 50 18 27 f9  d2 c9 00 00 6e 75 00      b···P·'· ····nu·
```

Substitute: ls response



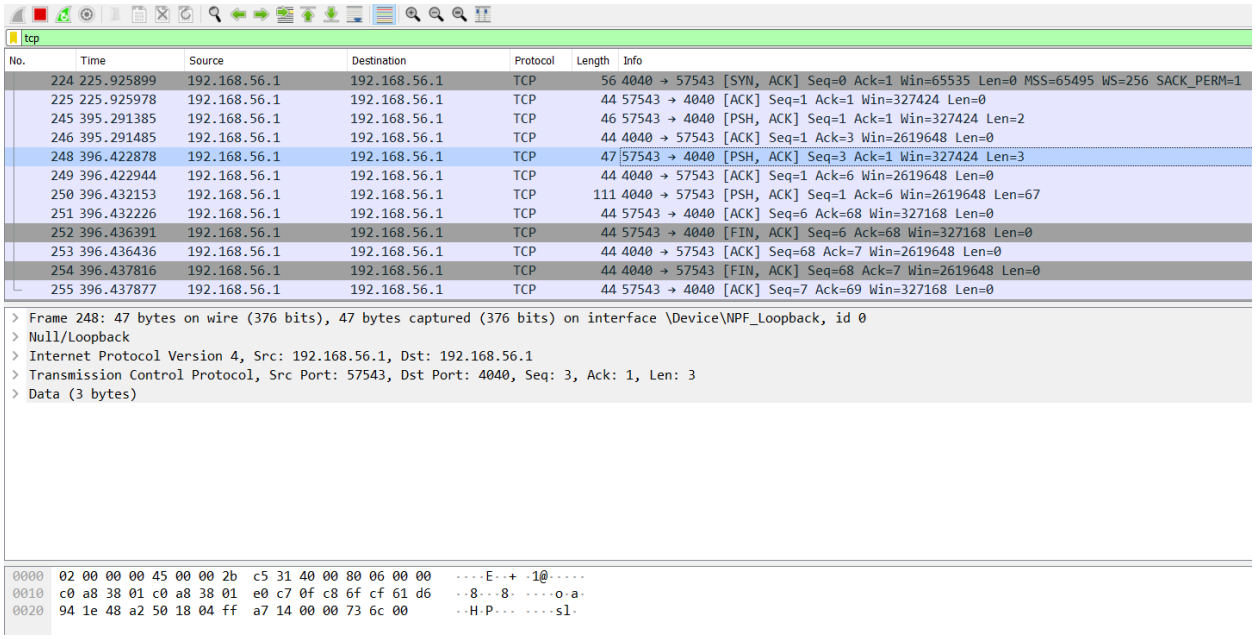
tcp

No.	Time	Source	Destination	Protocol	Length	Info
188	133.022429	192.168.56.1	192.168.56.1	TCP	44	4040 → 57523 [FIN, ACK] Seq=68 Ack=6 Win=10233 Len=0
189	133.022472	192.168.56.1	192.168.56.1	TCP	44	57523 → 4040 [ACK] Seq=6 Ack=69 Win=10233 Len=0
190	133.023017	192.168.56.1	192.168.56.1	TCP	44	57523 → 4040 [FIN, ACK] Seq=6 Ack=69 Win=10233 Len=0
191	133.023083	192.168.56.1	192.168.56.1	TCP	44	4040 → 57523 [ACK] Seq=69 Ack=7 Win=10233 Len=0
192	133.024841	192.168.56.1	192.168.56.1	TCP	56	57537 → 4040 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
193	133.024911	192.168.56.1	192.168.56.1	TCP	56	4040 → 57537 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
194	133.024950	192.168.56.1	192.168.56.1	TCP	44	57537 → 4040 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
213	224.871132	192.168.56.1	192.168.56.1	TCP	46	57537 → 4040 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=2
214	224.871311	192.168.56.1	192.168.56.1	TCP	44	4040 → 57537 [ACK] Seq=1 Ack=3 Win=2619648 Len=0
215	225.918820	192.168.56.1	192.168.56.1	TCP	47	57537 → 4040 [PSH, ACK] Seq=3 Ack=1 Win=2619648 Len=3
216	225.918903	192.168.56.1	192.168.56.1	TCP	44	4040 → 57537 [ACK] Seq=1 Ack=6 Win=2619648 Len=0
217	225.921045	192.168.56.1	192.168.56.1	TCP	111	4040 → 57537 [PSH, ACK] Seq=1 Ack=6 Win=2619648 Len=67

```
> Frame 217: 111 bytes on wire (888 bits), 111 bytes captured (888 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1
> Transmission Control Protocol, Src Port: 4040, Dst Port: 57537, Seq: 1, Ack: 6, Len: 67
> Data (67 bytes)
```

```
0000  02 00 00 00 45 00 00 6b  c5 26 40 00 80 06 00 00   ····E··k ·&@·····
0010  c0 a8 38 01 c0 a8 38 01  0f c8 e0 c1 62 8b e5 e9   ··8···8· ····b···
0020  3a cb e1 76 50 18 27 f9  cc 5d 00 00 5b 27 64 74   :··vP·'· ·]··['dt
0030  77 6a 2e 76 7a 76 27 2c  20 27 4b 6f 63 69 67 75   wj.vzv', 'Kocigu
0040  27 2c 20 27 75 67 74 78  67 74 2e 72 61 27 2c 20   ', 'ugtx gt.ra',
0050  27 77 76 6b 6e 6b 76 6b  67 75 2e 72 61 27 2c 20   'wvknkvk gu.ra',
0060  27 5f 5f 72 61 65 63 65  6a 67 5f 5f 27 5d 00      '__raece jg__']·
```

Transpose: ls command



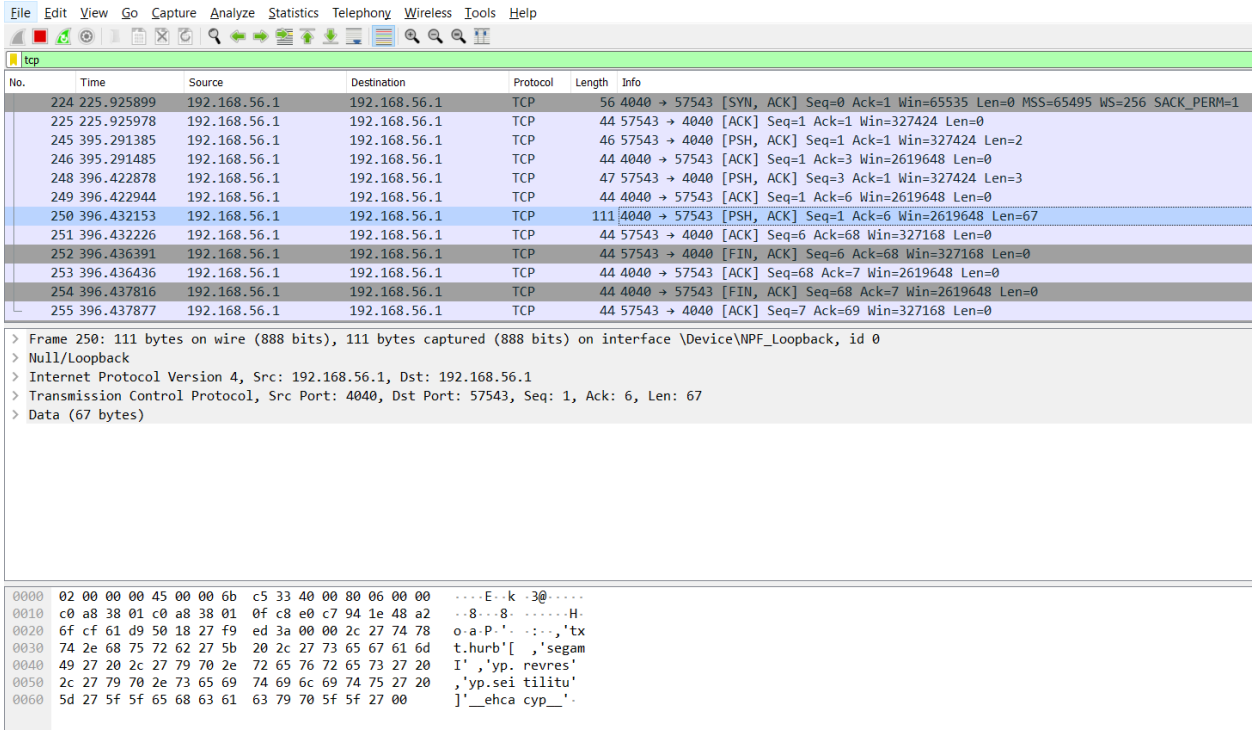
tcp

No.	Time	Source	Destination	Protocol	Length	Info
224	225.925899	192.168.56.1	192.168.56.1	TCP	56	4040 → 57543 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
225	225.925978	192.168.56.1	192.168.56.1	TCP	44	57543 → 4040 [ACK] Seq=1 Ack=1 Win=327424 Len=0
245	395.291385	192.168.56.1	192.168.56.1	TCP	46	57543 → 4040 [PSH, ACK] Seq=1 Ack=1 Win=327424 Len=2
246	395.291485	192.168.56.1	192.168.56.1	TCP	44	4040 → 57543 [ACK] Seq=1 Ack=3 Win=2619648 Len=0
248	396.422878	192.168.56.1	192.168.56.1	TCP	47	57543 → 4040 [PSH, ACK] Seq=3 Ack=1 Win=327424 Len=3
249	396.422944	192.168.56.1	192.168.56.1	TCP	44	4040 → 57543 [ACK] Seq=1 Ack=6 Win=2619648 Len=0
250	396.432153	192.168.56.1	192.168.56.1	TCP	111	4040 → 57543 [PSH, ACK] Seq=1 Ack=6 Win=2619648 Len=67
251	396.432226	192.168.56.1	192.168.56.1	TCP	44	57543 → 4040 [ACK] Seq=6 Ack=68 Win=327168 Len=0
252	396.436391	192.168.56.1	192.168.56.1	TCP	44	57543 → 4040 [FIN, ACK] Seq=6 Ack=68 Win=327168 Len=0
253	396.436436	192.168.56.1	192.168.56.1	TCP	44	4040 → 57543 [ACK] Seq=68 Ack=7 Win=2619648 Len=0
254	396.437816	192.168.56.1	192.168.56.1	TCP	44	4040 → 57543 [FIN, ACK] Seq=68 Ack=7 Win=2619648 Len=0
255	396.437877	192.168.56.1	192.168.56.1	TCP	44	57543 → 4040 [ACK] Seq=7 Ack=69 Win=327168 Len=0

```
> Frame 248: 47 bytes on wire (376 bits), 47 bytes captured (376 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1
> Transmission Control Protocol, Src Port: 57543, Dst Port: 4040, Seq: 3, Ack: 1, Len: 3
> Data (3 bytes)
```

```
0000  02 00 00 00 45 00 00 2b  c5 31 40 00 80 06 00 00   ····E··+ ·1@·····
0010  c0 a8 38 01 c0 a8 38 01  e0 c7 0f c8 6f cf 61 d6   ··8···8· ····o·a·
0020  94 1e 48 a2 50 18 04 ff  a7 14 00 00 73 6c 00      ··H·P··· ····sl·
```

Transpose: ls response



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

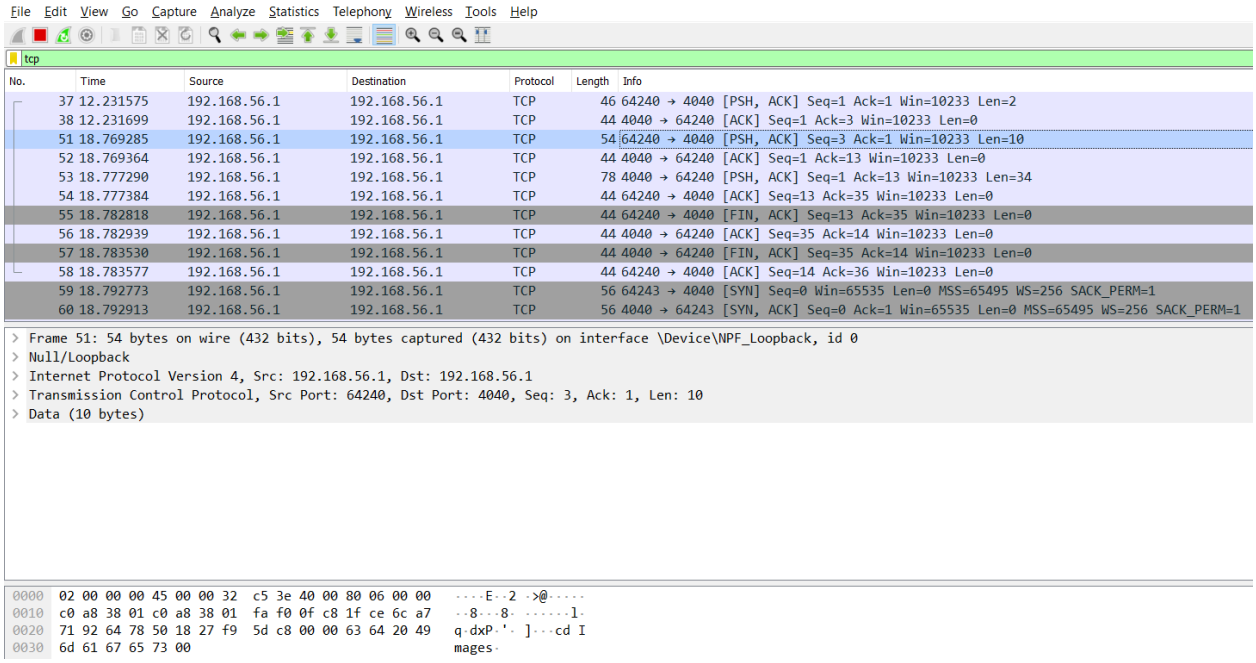
tcp

No.	Time	Source	Destination	Protocol	Length	Info
224	225.925899	192.168.56.1	192.168.56.1	TCP	56	4040 → 57543 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
225	225.925978	192.168.56.1	192.168.56.1	TCP	44	57543 → 4040 [ACK] Seq=1 Ack=1 Win=327424 Len=0
245	395.291385	192.168.56.1	192.168.56.1	TCP	46	57543 → 4040 [PSH, ACK] Seq=1 Ack=1 Win=327424 Len=2
246	395.291485	192.168.56.1	192.168.56.1	TCP	44	4040 → 57543 [ACK] Seq=1 Ack=3 Win=2619648 Len=0
248	396.422878	192.168.56.1	192.168.56.1	TCP	47	57543 → 4040 [PSH, ACK] Seq=3 Ack=1 Win=327424 Len=3
249	396.422944	192.168.56.1	192.168.56.1	TCP	44	4040 → 57543 [ACK] Seq=1 Ack=6 Win=2619648 Len=0
250	396.432153	192.168.56.1	192.168.56.1	TCP	111	4040 → 57543 [PSH, ACK] Seq=1 Ack=6 Win=2619648 Len=67
251	396.432226	192.168.56.1	192.168.56.1	TCP	44	57543 → 4040 [ACK] Seq=6 Ack=68 Win=327168 Len=0
252	396.436391	192.168.56.1	192.168.56.1	TCP	44	57543 → 4040 [FIN, ACK] Seq=6 Ack=68 Win=327168 Len=0
253	396.436436	192.168.56.1	192.168.56.1	TCP	44	4040 → 57543 [ACK] Seq=68 Ack=7 Win=2619648 Len=0
254	396.437816	192.168.56.1	192.168.56.1	TCP	44	4040 → 57543 [FIN, ACK] Seq=68 Ack=7 Win=2619648 Len=0
255	396.437877	192.168.56.1	192.168.56.1	TCP	44	57543 → 4040 [ACK] Seq=7 Ack=69 Win=327168 Len=0

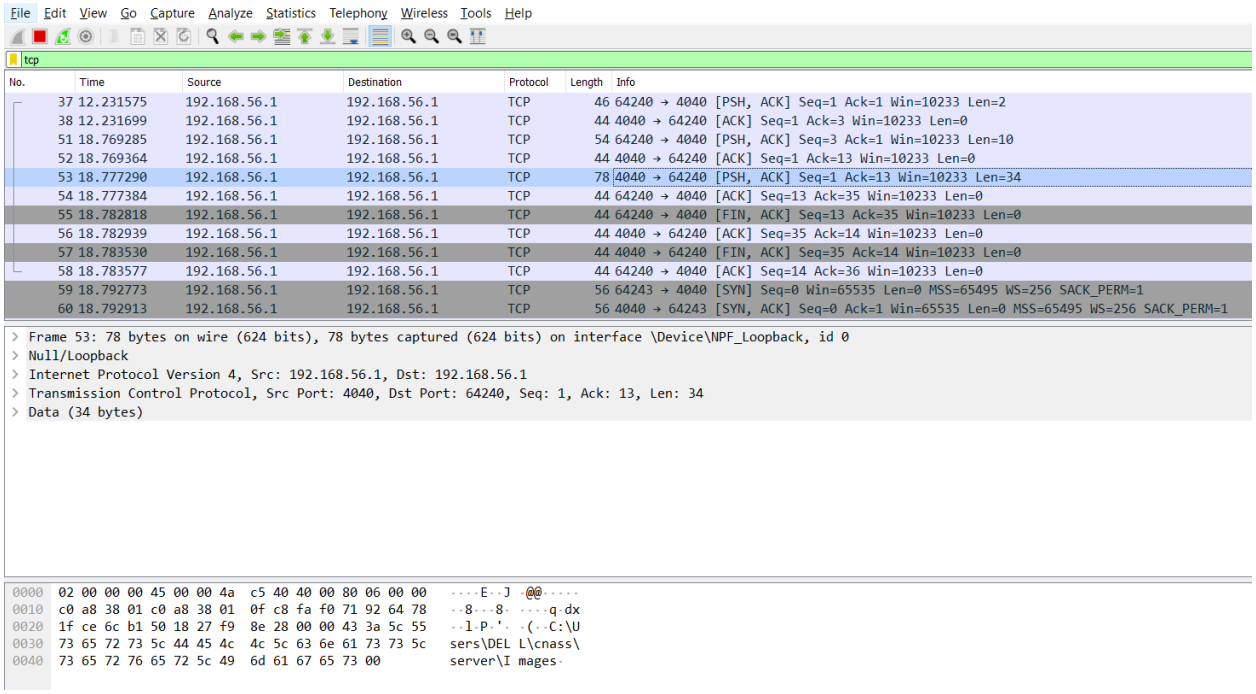
```
> Frame 250: 111 bytes on wire (888 bits), 111 bytes captured (888 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1
> Transmission Control Protocol, Src Port: 4040, Dst Port: 57543, Seq: 1, Ack: 6, Len: 67
> Data (67 bytes)
```

```
0000  02 00 00 00 45 00 00 6b  c5 33 40 00 80 06 00 00   ····E··k ·3@·····
0010  c0 a8 38 01 c0 a8 38 01  0f c8 e0 c7 94 1e 48 a2   ··8···8· ······H·
0020  6f cf 61 d9 50 18 27 f9  ed 3a 00 00 2c 27 74 78   o·a·P·'· ·:··,'tx
0030  74 2e 68 75 72 62 27 5b  20 2c 27 73 65 67 61 6d   t.hurb'[  ,'segam
0040  49 27 20 2c 27 79 70 2e  72 65 76 72 65 73 27 20   I' ,'yp. revres'
0050  2c 27 79 70 2e 73 65 69  74 69 6c 69 74 75 27 20   ,'yp.sei tilitu'
0060  5d 27 5f 5f 65 68 63 61  63 79 70 5f 5f 27 00      ]'__ehca cyp__'·
```

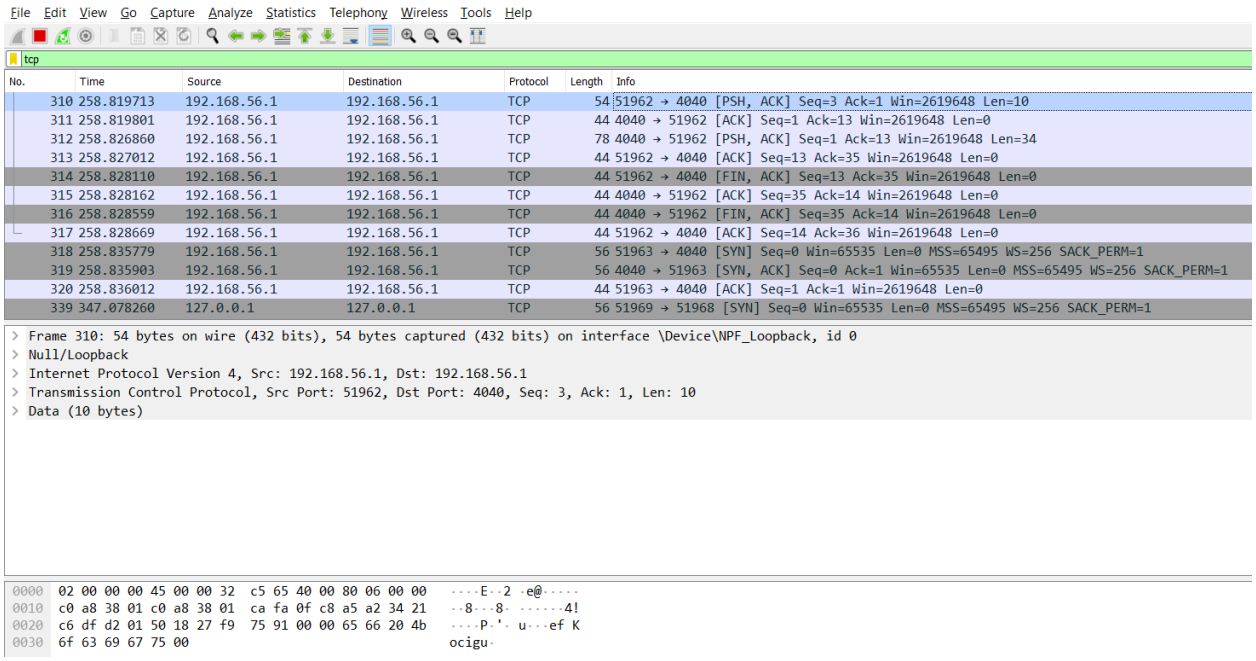
Plaintext: cd Images command



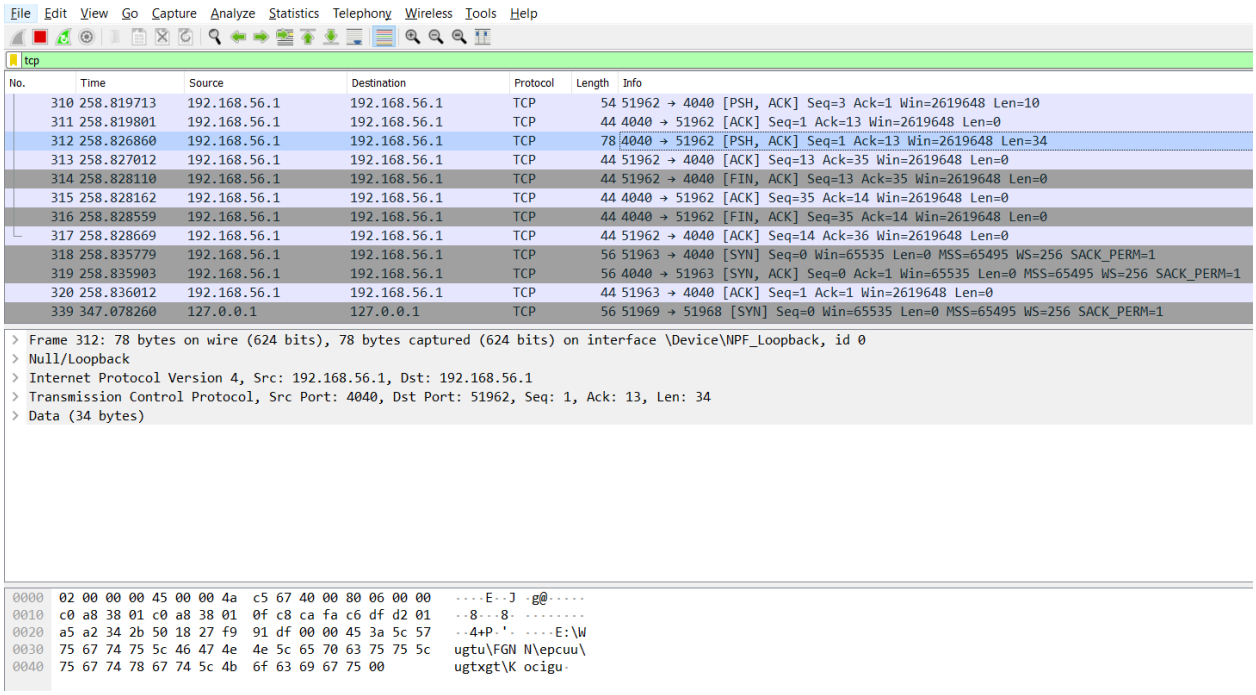
Plaintext: cd Images response



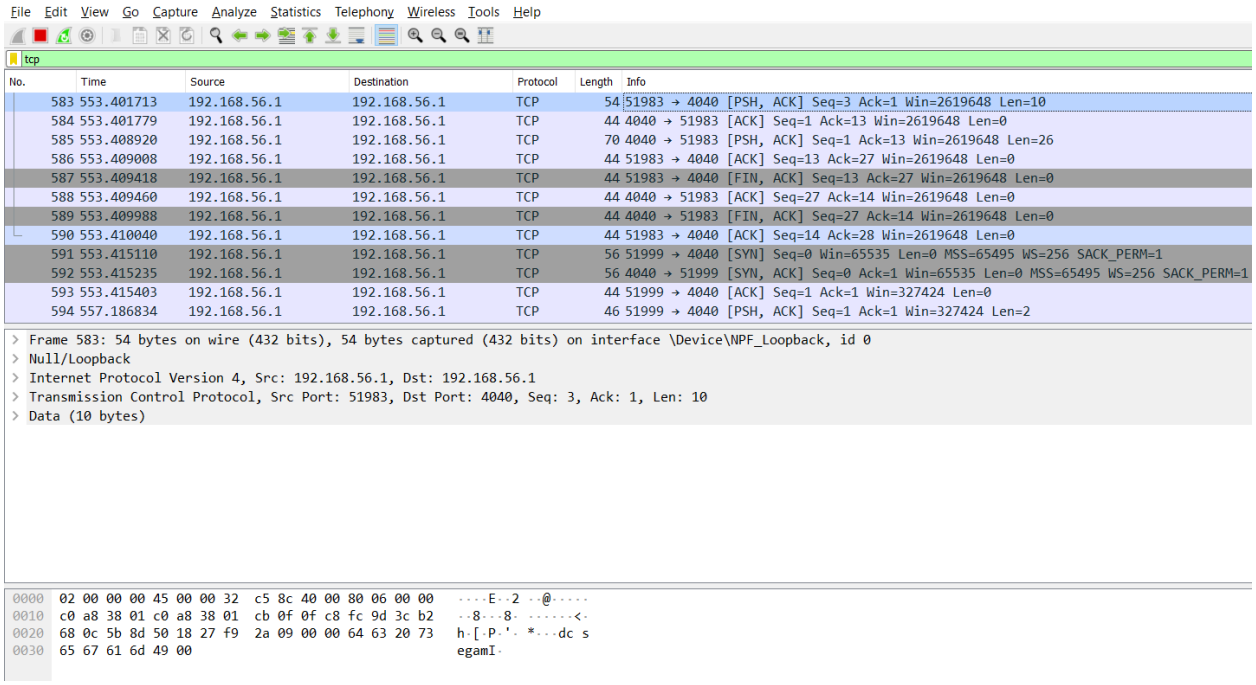
Substitute: cd Images command



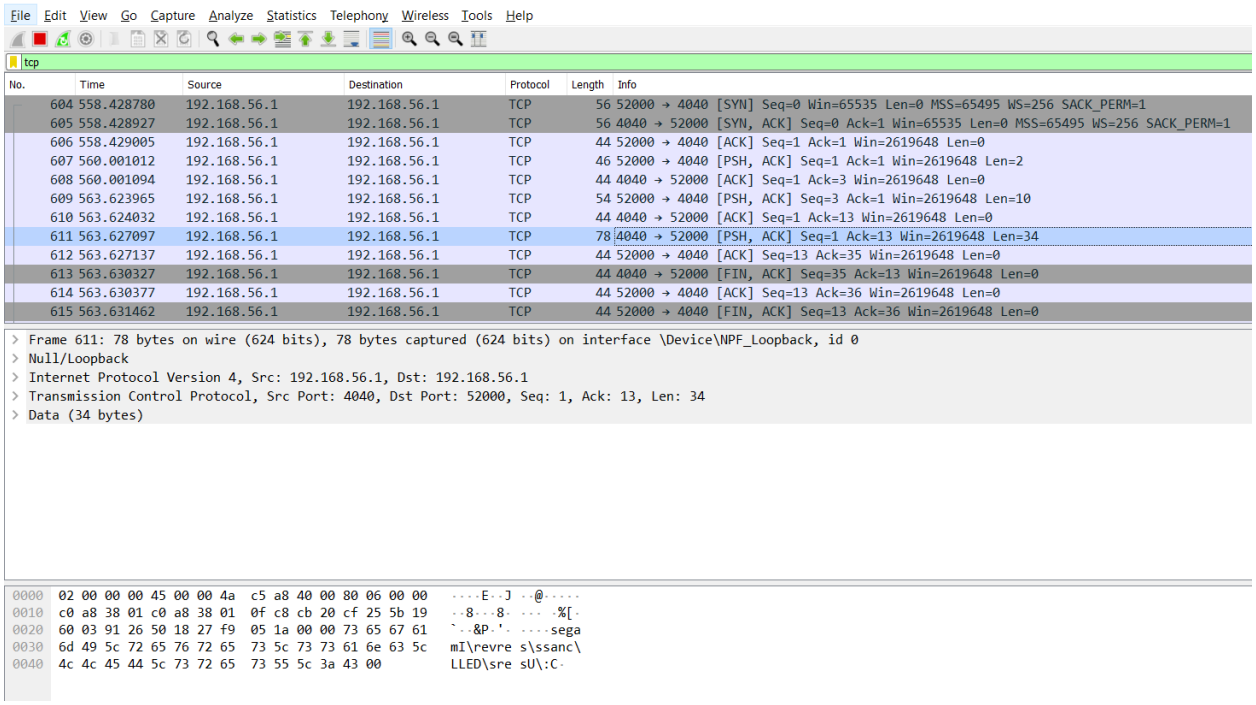
Substitute: cd Images response



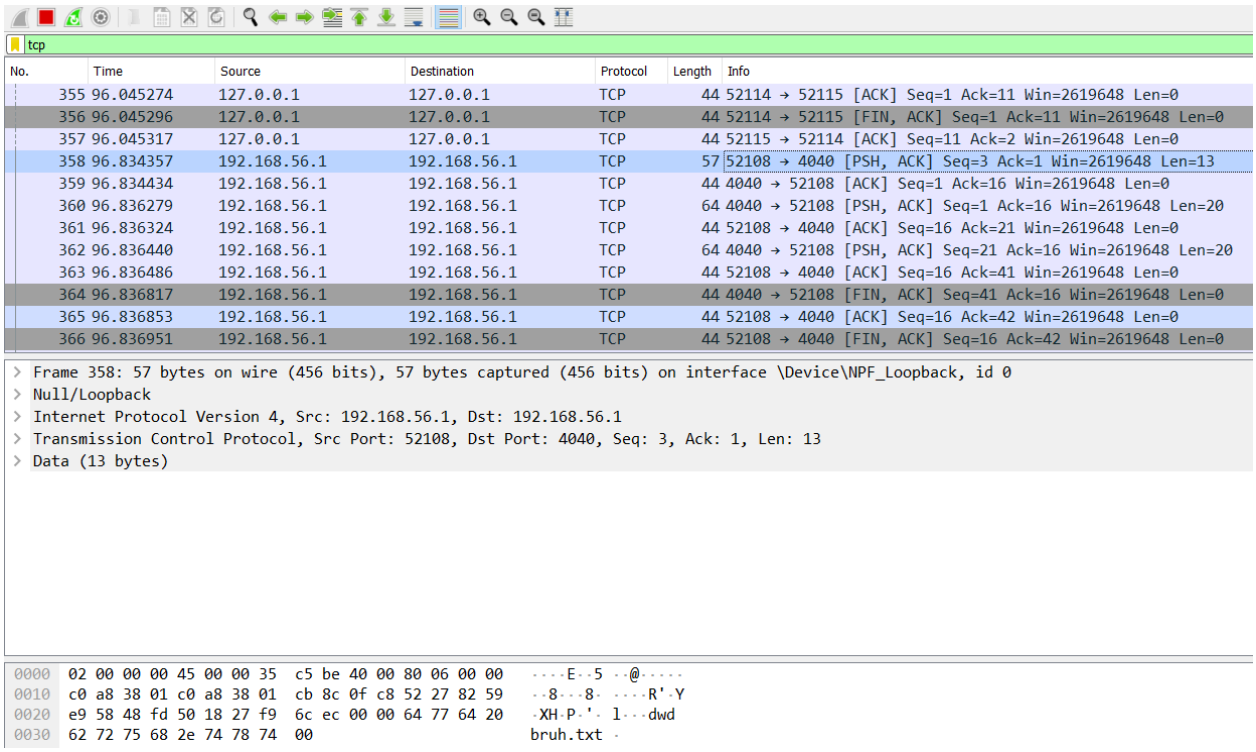
Transpose: cd Images command



Transpose: cd Images response



Plaintext: dwd bruh.txt command



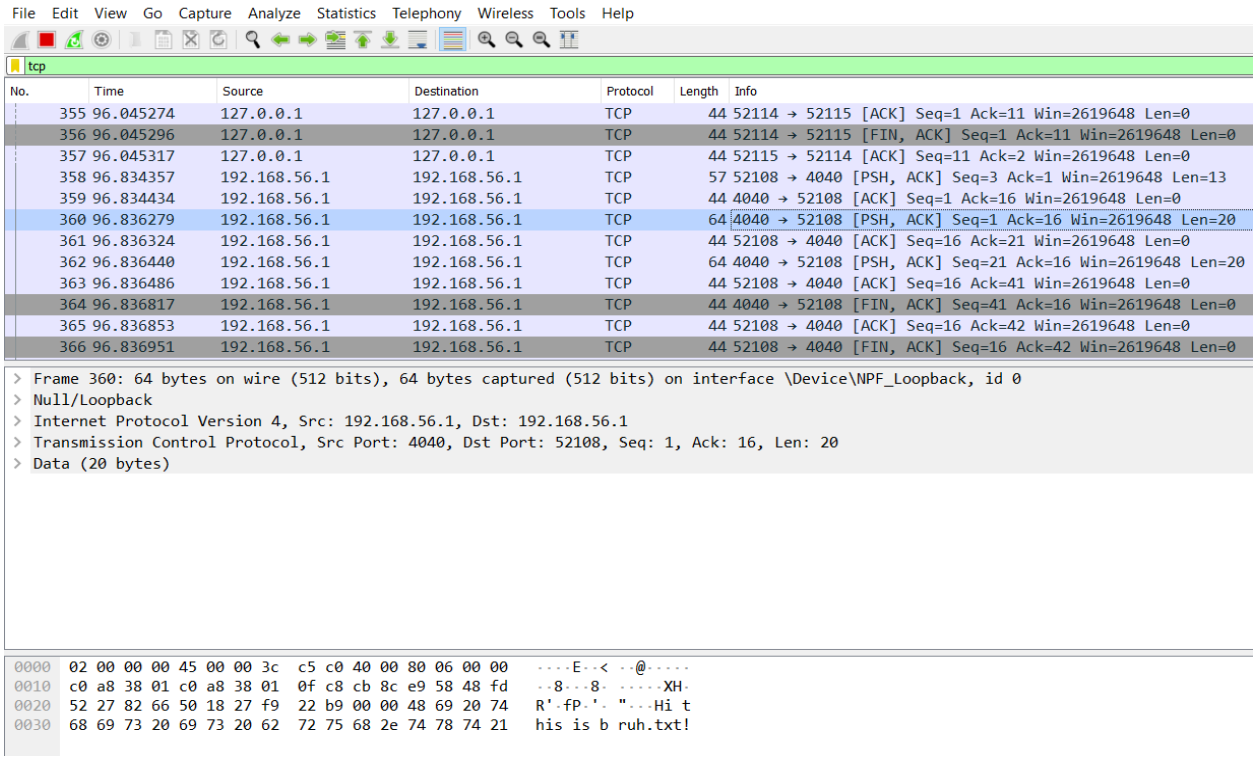
tcp

No.	Time	Source	Destination	Protocol	Length	Info
355	96.045274	127.0.0.1	127.0.0.1	TCP	44	52114 → 52115 [ACK] Seq=1 Ack=11 Win=2619648 Len=0
356	96.045296	127.0.0.1	127.0.0.1	TCP	44	52114 → 52115 [FIN, ACK] Seq=1 Ack=11 Win=2619648 Len=0
357	96.045317	127.0.0.1	127.0.0.1	TCP	44	52115 → 52114 [ACK] Seq=11 Ack=2 Win=2619648 Len=0
358	96.834357	192.168.56.1	192.168.56.1	TCP	57	52108 → 4040 [PSH, ACK] Seq=3 Ack=1 Win=2619648 Len=13
359	96.834434	192.168.56.1	192.168.56.1	TCP	44	4040 → 52108 [ACK] Seq=1 Ack=16 Win=2619648 Len=0
360	96.836279	192.168.56.1	192.168.56.1	TCP	64	4040 → 52108 [PSH, ACK] Seq=1 Ack=16 Win=2619648 Len=20
361	96.836324	192.168.56.1	192.168.56.1	TCP	44	52108 → 4040 [ACK] Seq=16 Ack=21 Win=2619648 Len=0
362	96.836440	192.168.56.1	192.168.56.1	TCP	64	4040 → 52108 [PSH, ACK] Seq=21 Ack=16 Win=2619648 Len=20
363	96.836486	192.168.56.1	192.168.56.1	TCP	44	52108 → 4040 [ACK] Seq=16 Ack=41 Win=2619648 Len=0
364	96.836817	192.168.56.1	192.168.56.1	TCP	44	4040 → 52108 [FIN, ACK] Seq=41 Ack=16 Win=2619648 Len=0
365	96.836853	192.168.56.1	192.168.56.1	TCP	44	52108 → 4040 [ACK] Seq=16 Ack=42 Win=2619648 Len=0
366	96.836951	192.168.56.1	192.168.56.1	TCP	44	52108 → 4040 [FIN, ACK] Seq=16 Ack=42 Win=2619648 Len=0

```
> Frame 358: 57 bytes on wire (456 bits), 57 bytes captured (456 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1
> Transmission Control Protocol, Src Port: 52108, Dst Port: 4040, Seq: 3, Ack: 1, Len: 13
> Data (13 bytes)
```

```
0000  02 00 00 00 45 00 00 35  c5 be 40 00 80 06 00 00   ····E··5 ··@·····
0010  c0 a8 38 01 c0 a8 38 01  cb 8c 0f c8 52 27 82 59   ··8···8· ····R'·Y
0020  e9 58 48 fd 50 18 27 f9  6c ec 00 00 64 77 64 20   ·XH·P·'· l···dwd
0030  62 72 75 68 2e 74 78 74  00                       bruh.txt ·
```

Plaintext: dwd bruh.txt response



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

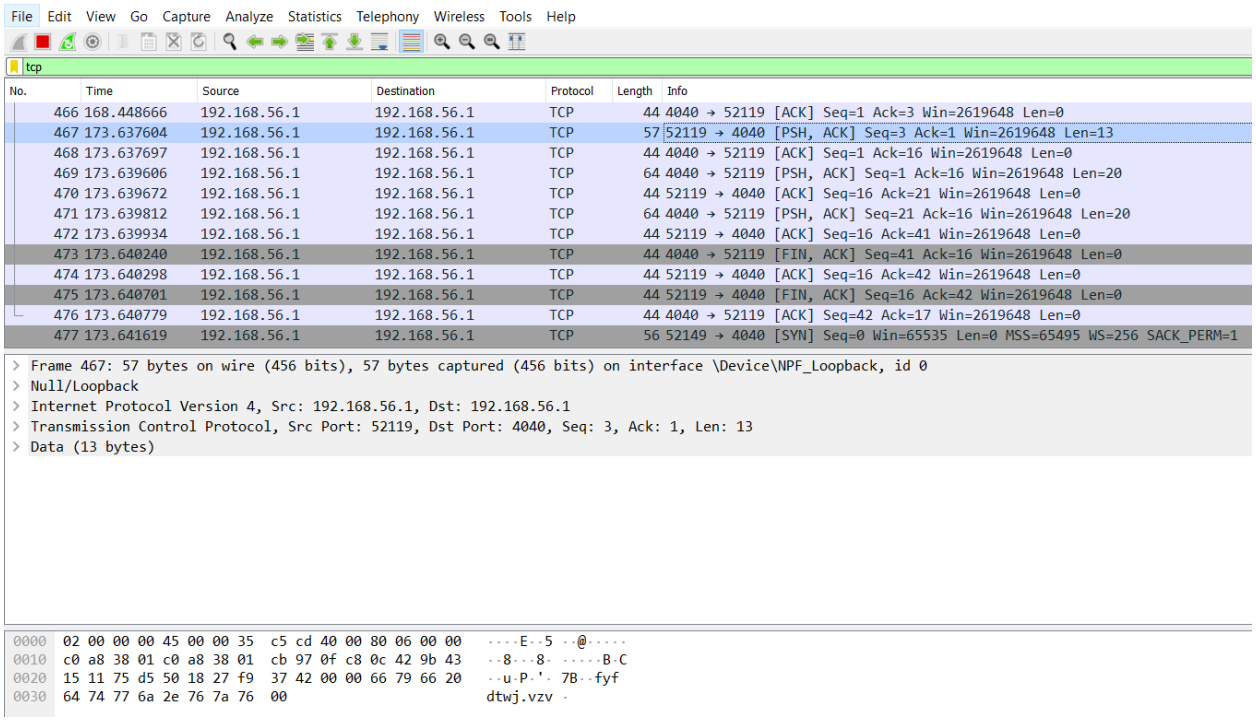
tcp

No.	Time	Source	Destination	Protocol	Length	Info
355	96.045274	127.0.0.1	127.0.0.1	TCP	44	52114 → 52115 [ACK] Seq=1 Ack=11 Win=2619648 Len=0
356	96.045296	127.0.0.1	127.0.0.1	TCP	44	52114 → 52115 [FIN, ACK] Seq=1 Ack=11 Win=2619648 Len=0
357	96.045317	127.0.0.1	127.0.0.1	TCP	44	52115 → 52114 [ACK] Seq=11 Ack=2 Win=2619648 Len=0
358	96.834357	192.168.56.1	192.168.56.1	TCP	57	52108 → 4040 [PSH, ACK] Seq=3 Ack=1 Win=2619648 Len=13
359	96.834434	192.168.56.1	192.168.56.1	TCP	44	4040 → 52108 [ACK] Seq=1 Ack=16 Win=2619648 Len=0
360	96.836279	192.168.56.1	192.168.56.1	TCP	64	4040 → 52108 [PSH, ACK] Seq=1 Ack=16 Win=2619648 Len=20
361	96.836324	192.168.56.1	192.168.56.1	TCP	44	52108 → 4040 [ACK] Seq=16 Ack=21 Win=2619648 Len=0
362	96.836440	192.168.56.1	192.168.56.1	TCP	64	4040 → 52108 [PSH, ACK] Seq=21 Ack=16 Win=2619648 Len=20
363	96.836486	192.168.56.1	192.168.56.1	TCP	44	52108 → 4040 [ACK] Seq=16 Ack=41 Win=2619648 Len=0
364	96.836817	192.168.56.1	192.168.56.1	TCP	44	4040 → 52108 [FIN, ACK] Seq=41 Ack=16 Win=2619648 Len=0
365	96.836853	192.168.56.1	192.168.56.1	TCP	44	52108 → 4040 [ACK] Seq=16 Ack=42 Win=2619648 Len=0
366	96.836951	192.168.56.1	192.168.56.1	TCP	44	52108 → 4040 [FIN, ACK] Seq=16 Ack=42 Win=2619648 Len=0

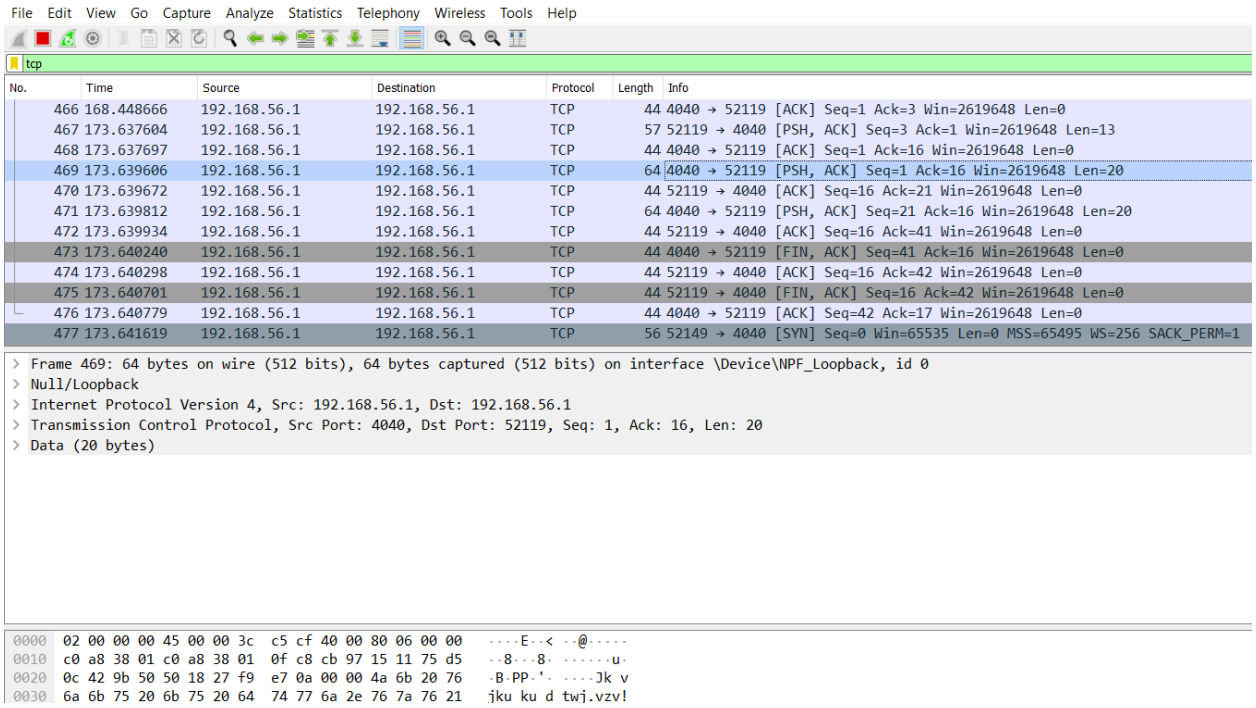
```
> Frame 360: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1
> Transmission Control Protocol, Src Port: 4040, Dst Port: 52108, Seq: 1, Ack: 16, Len: 20
> Data (20 bytes)
```

```
0000  02 00 00 00 45 00 00 3c  c5 c0 40 00 80 06 00 00   ····E··< ··@·····
0010  c0 a8 38 01 c0 a8 38 01  0f c8 cb 8c e9 58 48 fd   ··8···8· ·····XH·
0020  52 27 82 66 50 18 27 f9  22 b9 00 00 48 69 20 74   R'·fP·'· "···Hi t
0030  68 69 73 20 69 73 20 62  72 75 68 2e 74 78 74 21   his is b ruh.txt!
```

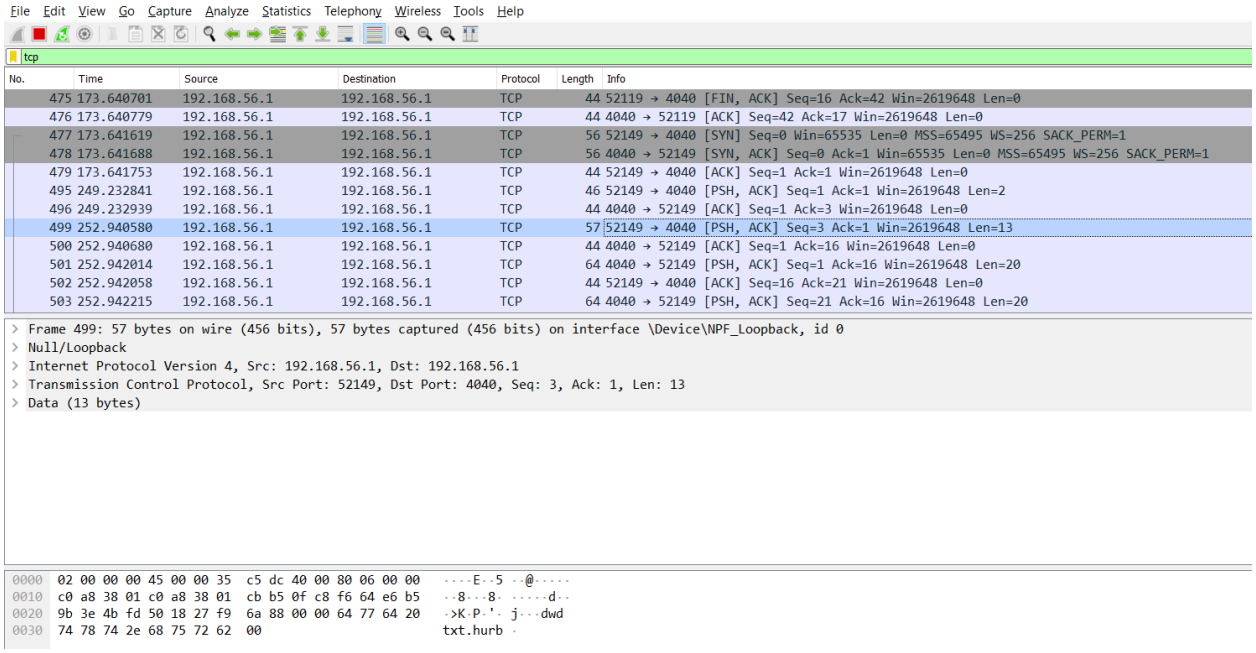
Substitute: dwd bruh.txt command



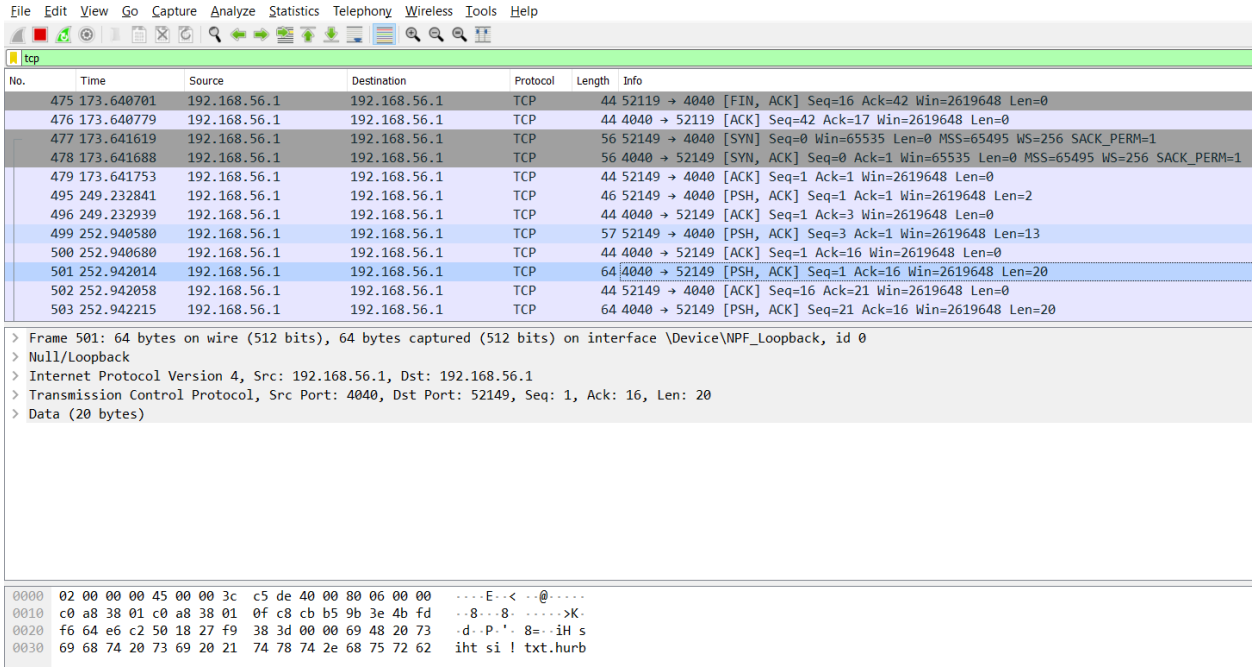
Substitute: dwd bruh.txt response



Transpose: dwd bruh.txt command



Transpose: dwd bruh.txt response



Plaintext: upd bruh.txt command

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp

No.	Time	Source	Destination	Protocol	Length	Info
13	3.575794	192.168.56.1	192.168.56.1	TCP	46	52153 → 4040 [PSH, ACK] Seq=1 Ack=1 Win=1279 Len=2
14	3.575899	192.168.56.1	192.168.56.1	TCP	44	4040 → 52153 [ACK] Seq=1 Ack=3 Win=10233 Len=0
15	7.962544	192.168.56.1	192.168.56.1	TCP	57	52153 → 4040 [PSH, ACK] Seq=3 Ack=1 Win=1279 Len=13
16	7.962645	192.168.56.1	192.168.56.1	TCP	44	4040 → 52153 [ACK] Seq=1 Ack=16 Win=10233 Len=0
17	7.963532	192.168.56.1	192.168.56.1	TCP	64	52153 → 4040 [PSH, ACK] Seq=16 Ack=1 Win=1279 Len=20
18	7.963570	192.168.56.1	192.168.56.1	TCP	44	4040 → 52153 [ACK] Seq=1 Ack=36 Win=10233 Len=0
19	7.965185	192.168.56.1	192.168.56.1	TCP	62	4040 → 52153 [PSH, ACK] Seq=1 Ack=36 Win=10233 Len=18
20	7.965246	192.168.56.1	192.168.56.1	TCP	44	52153 → 4040 [ACK] Seq=36 Ack=19 Win=1279 Len=0
21	7.965590	192.168.56.1	192.168.56.1	TCP	44	4040 → 52153 [FIN, ACK] Seq=19 Ack=36 Win=10233 Len=0
22	7.965652	192.168.56.1	192.168.56.1	TCP	44	52153 → 4040 [FIN, ACK] Seq=36 Ack=19 Win=1279 Len=0
23	7.965662	192.168.56.1	192.168.56.1	TCP	44	52153 → 4040 [ACK] Seq=37 Ack=20 Win=1279 Len=0
24	7.965839	192.168.56.1	192.168.56.1	TCP	44	4040 → 52153 [ACK] Seq=20 Ack=37 Win=10233 Len=0

```
> Frame 15: 57 bytes on wire (456 bits), 57 bytes captured (456 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1
> Transmission Control Protocol, Src Port: 52153, Dst Port: 4040, Seq: 3, Ack: 1, Len: 13
> Data (13 bytes)
```

```
0000  02 00 00 00 45 00 00 35  c5 eb 40 00 80 06 00 00   ····E··5 ··@·····
0010  c0 a8 38 01 c0 a8 38 01  cb b9 0f c8 6f a9 73 2d   ··8···8· ····o·s-
0020  30 b0 b9 84 50 18 04 ff  b8 8b 00 00 75 70 64 20   0···P··· ····upd
0030  62 72 75 68 2e 74 78 74  00                       bruh.txt ·
```

Plaintext: upd bruh.txt response

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

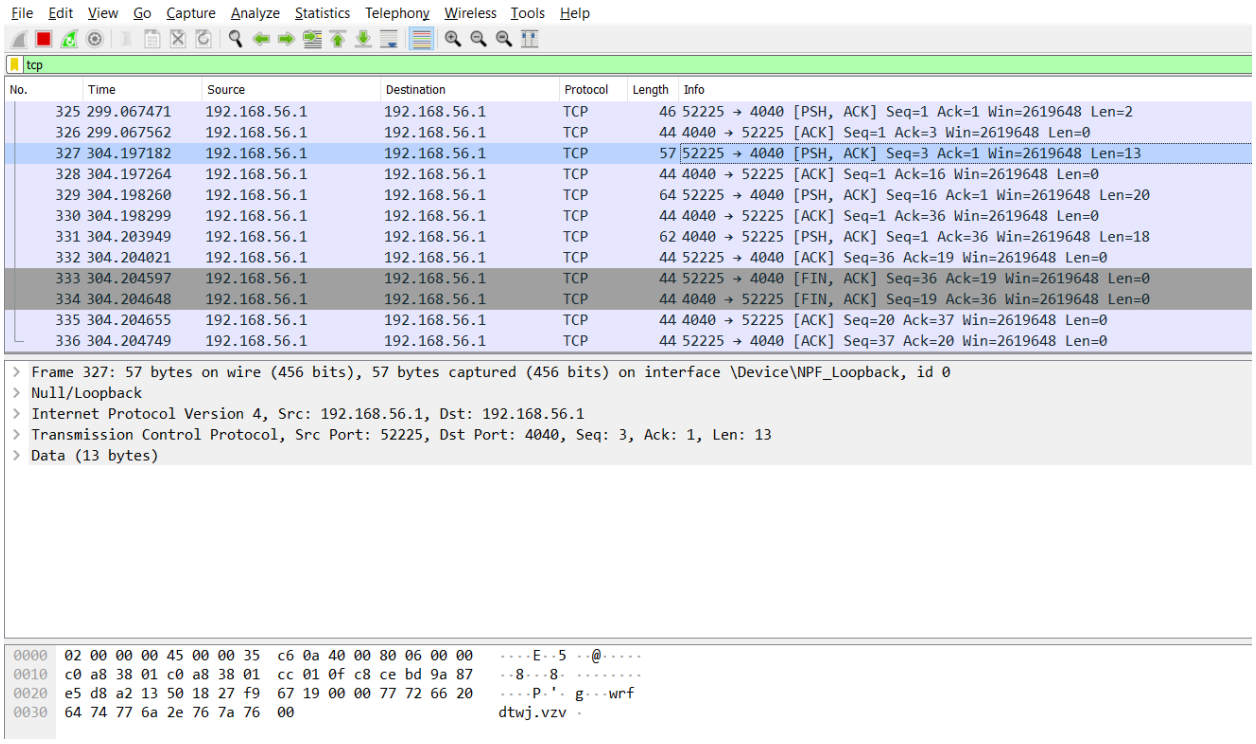
tcp

No.	Time	Source	Destination	Protocol	Length	Info
13	3.575794	192.168.56.1	192.168.56.1	TCP	46	52153 → 4040 [PSH, ACK] Seq=1 Ack=1 Win=1279 Len=2
14	3.575899	192.168.56.1	192.168.56.1	TCP	44	4040 → 52153 [ACK] Seq=1 Ack=3 Win=10233 Len=0
15	7.962544	192.168.56.1	192.168.56.1	TCP	57	52153 → 4040 [PSH, ACK] Seq=3 Ack=1 Win=1279 Len=13
16	7.962645	192.168.56.1	192.168.56.1	TCP	44	4040 → 52153 [ACK] Seq=1 Ack=16 Win=10233 Len=0
17	7.963532	192.168.56.1	192.168.56.1	TCP	64	52153 → 4040 [PSH, ACK] Seq=16 Ack=1 Win=1279 Len=20
18	7.963570	192.168.56.1	192.168.56.1	TCP	44	4040 → 52153 [ACK] Seq=1 Ack=36 Win=10233 Len=0
19	7.965185	192.168.56.1	192.168.56.1	TCP	62	4040 → 52153 [PSH, ACK] Seq=1 Ack=36 Win=10233 Len=18
20	7.965246	192.168.56.1	192.168.56.1	TCP	44	52153 → 4040 [ACK] Seq=36 Ack=19 Win=1279 Len=0
21	7.965590	192.168.56.1	192.168.56.1	TCP	44	4040 → 52153 [FIN, ACK] Seq=19 Ack=36 Win=10233 Len=0
22	7.965652	192.168.56.1	192.168.56.1	TCP	44	52153 → 4040 [FIN, ACK] Seq=36 Ack=19 Win=1279 Len=0
23	7.965662	192.168.56.1	192.168.56.1	TCP	44	52153 → 4040 [ACK] Seq=37 Ack=20 Win=1279 Len=0
24	7.965839	192.168.56.1	192.168.56.1	TCP	44	4040 → 52153 [ACK] Seq=20 Ack=37 Win=10233 Len=0

```
> Frame 17: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1
> Transmission Control Protocol, Src Port: 52153, Dst Port: 4040, Seq: 16, Ack: 1, Len: 20
> Data (20 bytes)
```

```
0000  02 00 00 00 45 00 00 3c  c5 ed 40 00 80 06 00 00   ····E··< ··@·····
0010  c0 a8 38 01 c0 a8 38 01  cb b9 0f c8 6f a9 73 3a   ··8···8· ····o·s:
0020  30 b0 b9 84 50 18 04 ff  7f 51 00 00 48 69 20 74   0···P··· ·Q··Hi t
0030  68 69 73 20 69 73 20 62  72 75 68 2e 74 78 74 21   his is b ruh.txt!
```

Substitute: upd bruh.txt command



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

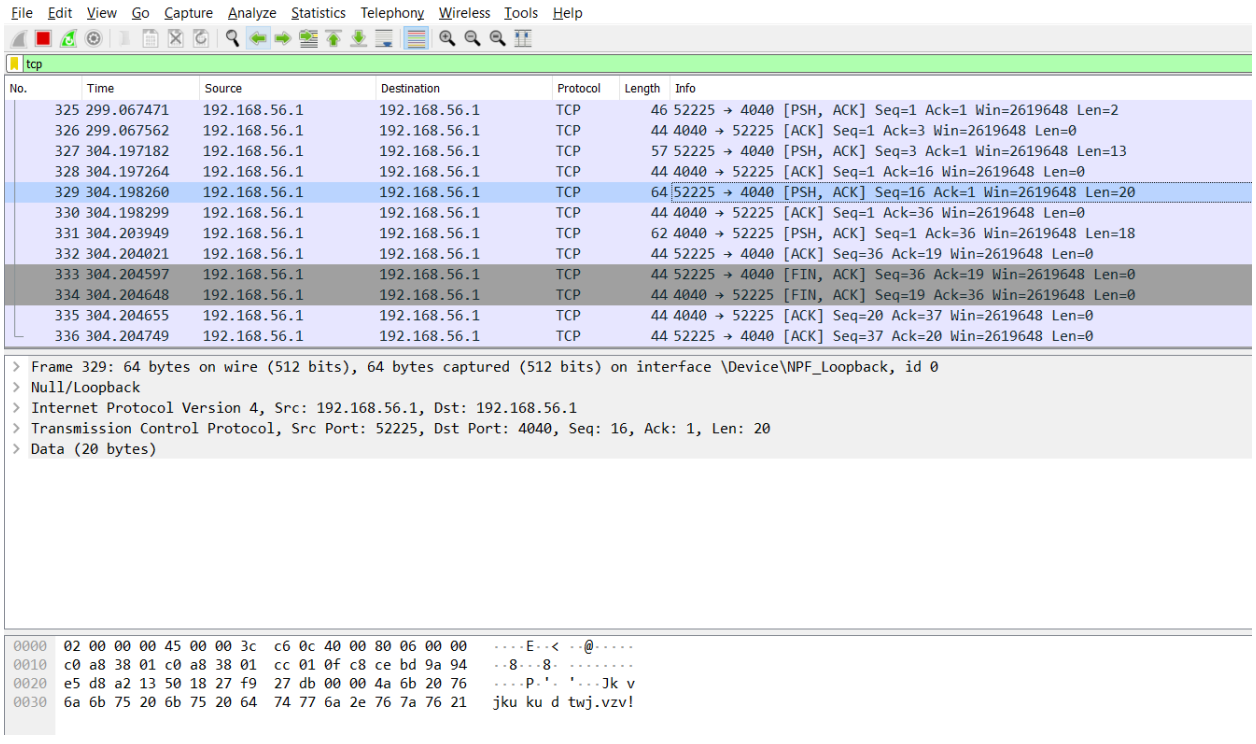
tcp

No.	Time	Source	Destination	Protocol	Length	Info
325	299.067471	192.168.56.1	192.168.56.1	TCP	46	52225 → 4040 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=2
326	299.067562	192.168.56.1	192.168.56.1	TCP	44	4040 → 52225 [ACK] Seq=1 Ack=3 Win=2619648 Len=0
327	304.197182	192.168.56.1	192.168.56.1	TCP	57	52225 → 4040 [PSH, ACK] Seq=3 Ack=1 Win=2619648 Len=13
328	304.197264	192.168.56.1	192.168.56.1	TCP	44	4040 → 52225 [ACK] Seq=1 Ack=16 Win=2619648 Len=0
329	304.198260	192.168.56.1	192.168.56.1	TCP	64	52225 → 4040 [PSH, ACK] Seq=16 Ack=1 Win=2619648 Len=20
330	304.198299	192.168.56.1	192.168.56.1	TCP	44	4040 → 52225 [ACK] Seq=1 Ack=36 Win=2619648 Len=0
331	304.203949	192.168.56.1	192.168.56.1	TCP	62	4040 → 52225 [PSH, ACK] Seq=1 Ack=36 Win=2619648 Len=18
332	304.204021	192.168.56.1	192.168.56.1	TCP	44	52225 → 4040 [ACK] Seq=36 Ack=19 Win=2619648 Len=0
333	304.204597	192.168.56.1	192.168.56.1	TCP	44	52225 → 4040 [FIN, ACK] Seq=36 Ack=19 Win=2619648 Len=0
334	304.204648	192.168.56.1	192.168.56.1	TCP	44	4040 → 52225 [FIN, ACK] Seq=19 Ack=36 Win=2619648 Len=0
335	304.204655	192.168.56.1	192.168.56.1	TCP	44	4040 → 52225 [ACK] Seq=20 Ack=37 Win=2619648 Len=0
336	304.204749	192.168.56.1	192.168.56.1	TCP	44	52225 → 4040 [ACK] Seq=37 Ack=20 Win=2619648 Len=0

```
> Frame 327: 57 bytes on wire (456 bits), 57 bytes captured (456 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1
> Transmission Control Protocol, Src Port: 52225, Dst Port: 4040, Seq: 3, Ack: 1, Len: 13
> Data (13 bytes)
```

```
0000  02 00 00 00 45 00 00 35  c6 0a 40 00 80 06 00 00   ····E··5 ··@·····
0010  c0 a8 38 01 c0 a8 38 01  cc 01 0f c8 ce bd 9a 87   ··8···8· ········
0020  e5 d8 a2 13 50 18 27 f9  67 19 00 00 77 72 66 20   ····P·'· g···wrf
0030  64 74 77 6a 2e 76 7a 76  00                       dtwj.vzv ·
```

Substitute: upd bruh.txt response



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp

No.	Time	Source	Destination	Protocol	Length	Info
325	299.067471	192.168.56.1	192.168.56.1	TCP	46	52225 → 4040 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=2
326	299.067562	192.168.56.1	192.168.56.1	TCP	44	4040 → 52225 [ACK] Seq=1 Ack=3 Win=2619648 Len=0
327	304.197182	192.168.56.1	192.168.56.1	TCP	57	52225 → 4040 [PSH, ACK] Seq=3 Ack=1 Win=2619648 Len=13
328	304.197264	192.168.56.1	192.168.56.1	TCP	44	4040 → 52225 [ACK] Seq=1 Ack=16 Win=2619648 Len=0
329	304.198260	192.168.56.1	192.168.56.1	TCP	64	52225 → 4040 [PSH, ACK] Seq=16 Ack=1 Win=2619648 Len=20
330	304.198299	192.168.56.1	192.168.56.1	TCP	44	4040 → 52225 [ACK] Seq=1 Ack=36 Win=2619648 Len=0
331	304.203949	192.168.56.1	192.168.56.1	TCP	62	4040 → 52225 [PSH, ACK] Seq=1 Ack=36 Win=2619648 Len=18
332	304.204021	192.168.56.1	192.168.56.1	TCP	44	52225 → 4040 [ACK] Seq=36 Ack=19 Win=2619648 Len=0
333	304.204597	192.168.56.1	192.168.56.1	TCP	44	52225 → 4040 [FIN, ACK] Seq=36 Ack=19 Win=2619648 Len=0
334	304.204648	192.168.56.1	192.168.56.1	TCP	44	4040 → 52225 [FIN, ACK] Seq=19 Ack=36 Win=2619648 Len=0
335	304.204655	192.168.56.1	192.168.56.1	TCP	44	4040 → 52225 [ACK] Seq=20 Ack=37 Win=2619648 Len=0
336	304.204749	192.168.56.1	192.168.56.1	TCP	44	52225 → 4040 [ACK] Seq=37 Ack=20 Win=2619648 Len=0

```
> Frame 329: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1
> Transmission Control Protocol, Src Port: 52225, Dst Port: 4040, Seq: 16, Ack: 1, Len: 20
> Data (20 bytes)
```

```
0000  02 00 00 00 45 00 00 3c  c6 0c 40 00 80 06 00 00   ····E··< ··@·····
0010  c0 a8 38 01 c0 a8 38 01  cc 01 0f c8 ce bd 9a 94   ··8···8· ········
0020  e5 d8 a2 13 50 18 27 f9  27 db 00 00 4a 6b 20 76   ····P·'· '···Jk v
0030  6a 6b 75 20 6b 75 20 64  74 77 6a 2e 76 7a 76 21   jku ku d twj.vzv!
```

Transpose: upd bruh.txt command

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp

No.	Time	Source	Destination	Protocol	Length	Info
395	368.362856	127.0.0.1	127.0.0.1	TCP	44	63427 → 63428 [ACK] Seq=1 Ack=11 Win=2619648 Len=0
396	368.362965	127.0.0.1	127.0.0.1	TCP	44	63427 → 63428 [FIN, ACK] Seq=1 Ack=11 Win=2619648 Len=0
397	368.363019	127.0.0.1	127.0.0.1	TCP	44	63428 → 63427 [ACK] Seq=11 Ack=2 Win=2619648 Len=0
412	404.402151	192.168.56.1	192.168.56.1	TCP	46	63410 → 4040 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=2
413	404.402242	192.168.56.1	192.168.56.1	TCP	44	4040 → 63410 [ACK] Seq=1 Ack=3 Win=2619648 Len=0
418	407.397286	192.168.56.1	192.168.56.1	TCP	57	63410 → 4040 [PSH, ACK] Seq=3 Ack=1 Win=2619648 Len=13
419	407.397375	192.168.56.1	192.168.56.1	TCP	44	4040 → 63410 [ACK] Seq=1 Ack=16 Win=2619648 Len=0
420	407.398300	192.168.56.1	192.168.56.1	TCP	64	63410 → 4040 [PSH, ACK] Seq=16 Ack=1 Win=2619648 Len=20
421	407.398374	192.168.56.1	192.168.56.1	TCP	44	4040 → 63410 [ACK] Seq=1 Ack=36 Win=2619648 Len=0
422	407.400995	192.168.56.1	192.168.56.1	TCP	62	4040 → 63410 [PSH, ACK] Seq=1 Ack=36 Win=2619648 Len=18
423	407.401057	192.168.56.1	192.168.56.1	TCP	44	63410 → 4040 [ACK] Seq=36 Ack=19 Win=2619648 Len=0
424	407.402776	192.168.56.1	192.168.56.1	TCP	44	4040 → 63410 [FIN, ACK] Seq=19 Ack=36 Win=2619648 Len=0

```
> Frame 418: 57 bytes on wire (456 bits), 57 bytes captured (456 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1
> Transmission Control Protocol, Src Port: 63410, Dst Port: 4040, Seq: 3, Ack: 1, Len: 13
> Data (13 bytes)
```

```
0000  02 00 00 00 45 00 00 35  c6 19 40 00 80 06 00 00   ····E··5 ··@·····
0010  c0 a8 38 01 c0 a8 38 01  f7 b2 0f c8 57 55 98 8c   ··8···8· ····WU··
0020  5c 2c 52 40 50 18 27 f9  53 9a 00 00 64 70 75 20   \,R@P·'· S···dpu
0030  74 78 74 2e 68 75 72 62  00                       txt.hurb ·
```

Transpose: upd bruh.txt response

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp

No.	Time	Source	Destination	Protocol	Length	Info
395	368.362856	127.0.0.1	127.0.0.1	TCP	44	63427 → 63428 [ACK] Seq=1 Ack=11 Win=2619648 Len=0
396	368.362965	127.0.0.1	127.0.0.1	TCP	44	63427 → 63428 [FIN, ACK] Seq=1 Ack=11 Win=2619648 Len=0
397	368.363019	127.0.0.1	127.0.0.1	TCP	44	63428 → 63427 [ACK] Seq=11 Ack=2 Win=2619648 Len=0
412	404.402151	192.168.56.1	192.168.56.1	TCP	46	63410 → 4040 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=2
413	404.402242	192.168.56.1	192.168.56.1	TCP	44	4040 → 63410 [ACK] Seq=1 Ack=3 Win=2619648 Len=0
418	407.397286	192.168.56.1	192.168.56.1	TCP	57	63410 → 4040 [PSH, ACK] Seq=3 Ack=1 Win=2619648 Len=13
419	407.397375	192.168.56.1	192.168.56.1	TCP	44	4040 → 63410 [ACK] Seq=1 Ack=16 Win=2619648 Len=0
420	407.398300	192.168.56.1	192.168.56.1	TCP	64	63410 → 4040 [PSH, ACK] Seq=16 Ack=1 Win=2619648 Len=20
421	407.398374	192.168.56.1	192.168.56.1	TCP	44	4040 → 63410 [ACK] Seq=1 Ack=36 Win=2619648 Len=0
422	407.400995	192.168.56.1	192.168.56.1	TCP	62	4040 → 63410 [PSH, ACK] Seq=1 Ack=36 Win=2619648 Len=18
423	407.401057	192.168.56.1	192.168.56.1	TCP	44	63410 → 4040 [ACK] Seq=36 Ack=19 Win=2619648 Len=0
424	407.402776	192.168.56.1	192.168.56.1	TCP	44	4040 → 63410 [FIN, ACK] Seq=19 Ack=36 Win=2619648 Len=0

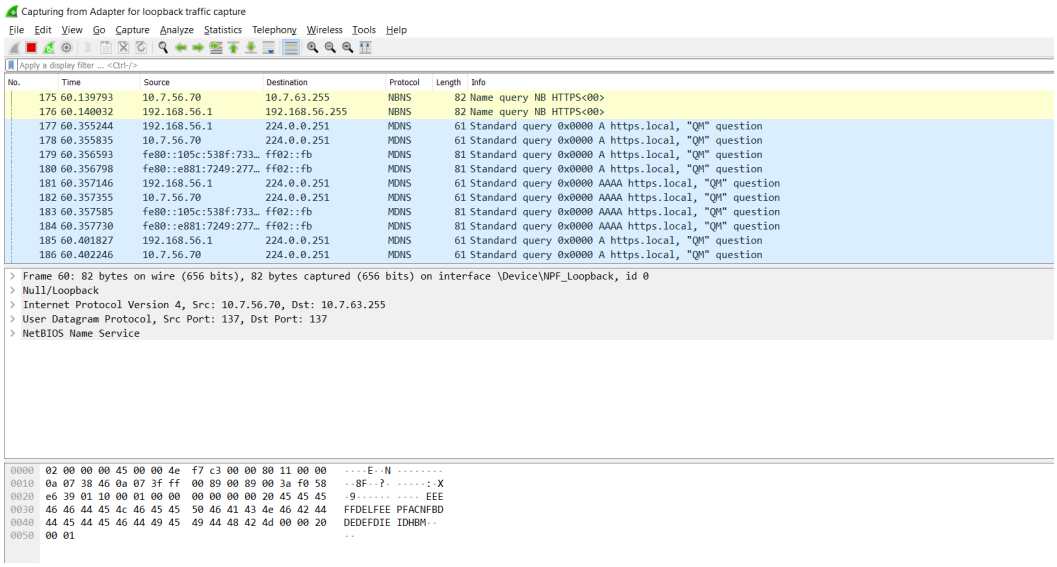
```
> Frame 420: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1
> Transmission Control Protocol, Src Port: 63410, Dst Port: 4040, Seq: 16, Ack: 1, Len: 20
> Data (20 bytes)
```

```
0000  02 00 00 00 45 00 00 3c  c6 1b 40 00 80 06 00 00   ····E··< ··@·····
0010  c0 a8 38 01 c0 a8 38 01  f7 b2 0f c8 57 55 98 99   ··8···8· ····WU··
0020  5c 2c 52 40 50 18 27 f9  32 48 00 00 69 48 20 73   \,R@P·'· 2H··iH s
0030  69 68 74 20 73 69 20 21  74 78 74 2e 68 75 72 62   iht si ! txt.hurb
```

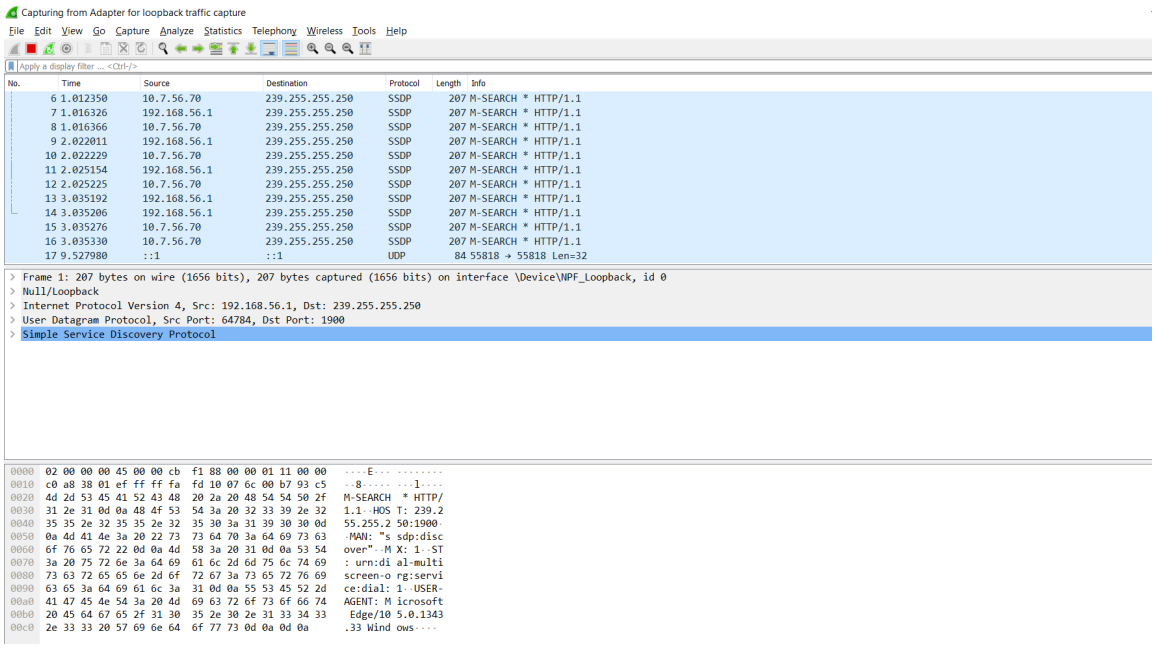
Part 3: Network Tools

5) a) List at-least 5 different network protocols that we have not discussed so far in the classroom and describe in 1-2 sentences the operation/usage of protocol and its layer of operation and indicate the associated RFC number if any

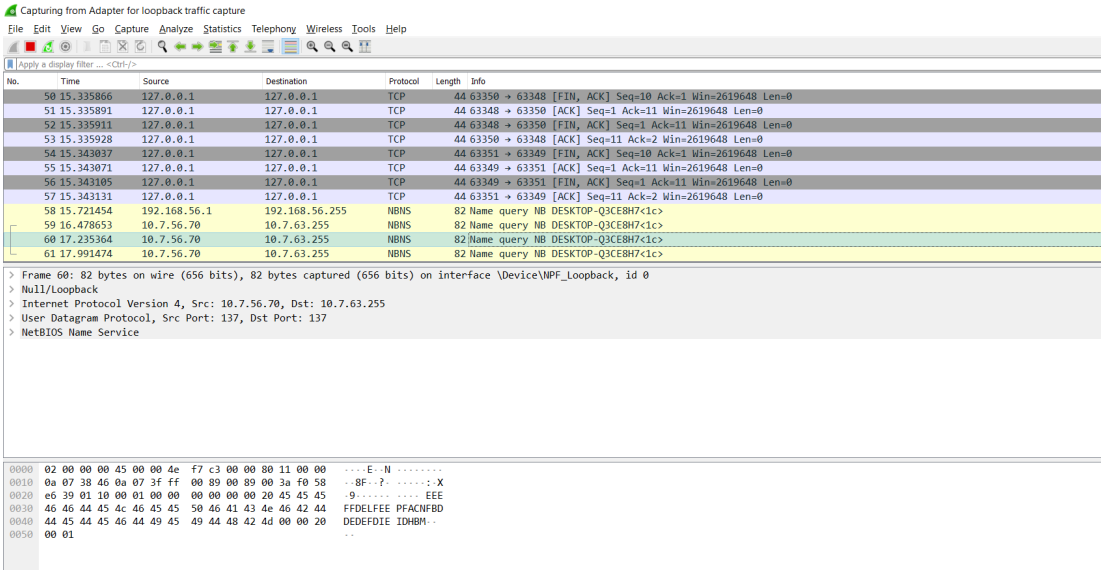
a) MDNS: Multicast DNS (mDNS) protocol resolves hostnames to IP addresses within small networks that do not include a local name server. Its RFC number is 6762.



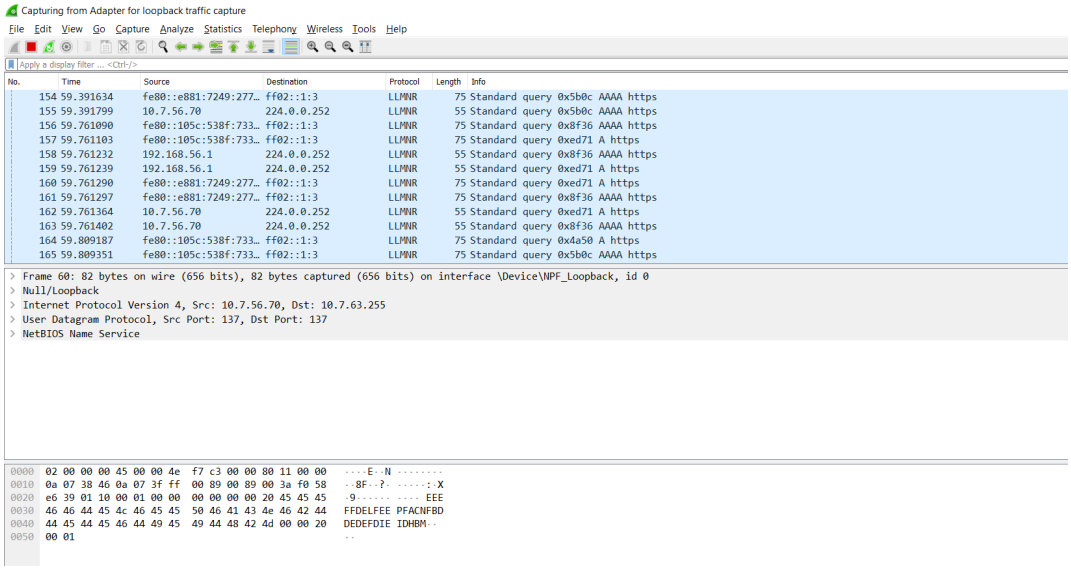
b) SSDP: The Simple Service Discovery Protocol (SSDP) is a network protocol based on the Internet protocol suite for advertisement and discovery of network services and presence information.



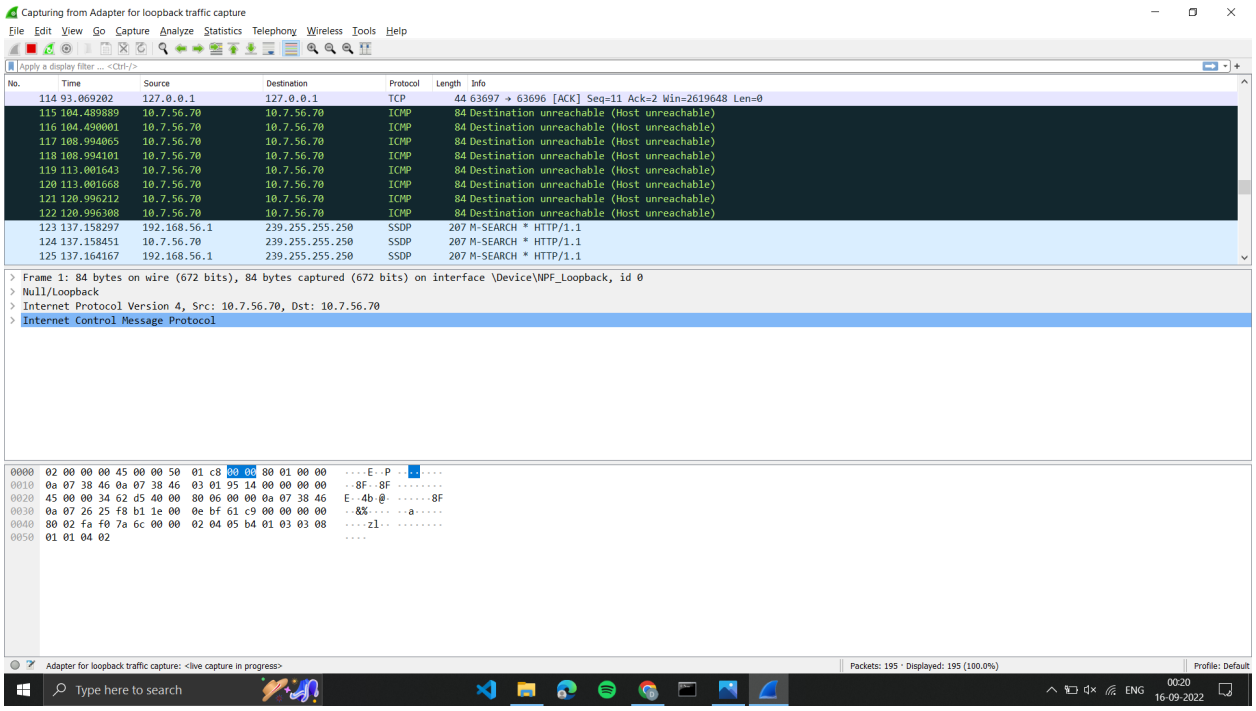
c) NBNS: NBNS stands for NetBIO Name Service, which is a protocol for name resolution. NBNS performs the same function as LLMNR, but using UDP broadcast packets instead of multi cast packets. Its RFC number is 1001.



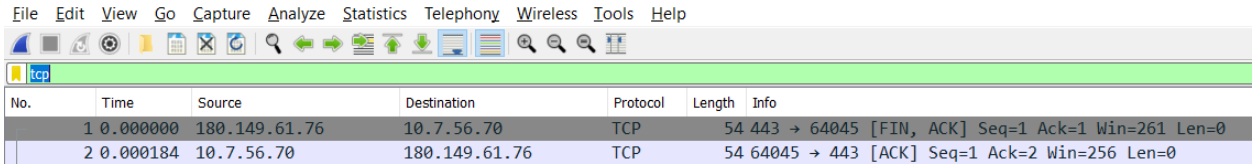
d) LLMNR: The Link-Local Multicast Name Resolution (LLMNR) is a protocol based on the Domain Name System (DNS) packet format that allows both IPv4 and IPv6 hosts to perform name resolution for hosts on the same local link. Its RFC number is 4795.



e) ICMP: The Internet Control Message Protocol (ICMP) is a network layer protocol used by network devices to diagnose network communication issues. ICMP is mainly used to determine whether or not data is reaching its intended destination in a timely manner. Its RFC number is 792.



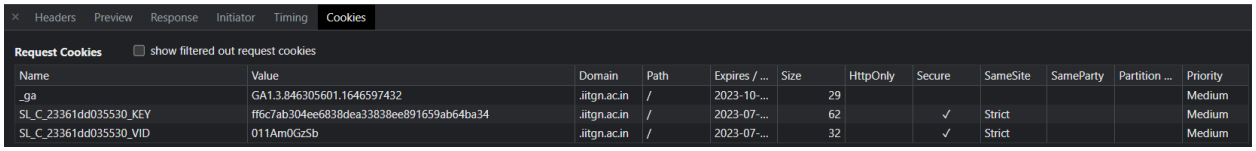
5) b) Identify any one connection and try to estimate the RTT of that connection.



To find RTT I filtered all the TCP connections and in the view menu, and in the Time Display Format, I selected the seconds since the previously displayed bracket. Therefore in the screenshot it is clear that the RTT is 0.000184 seconds.

5) c) List the cookies and identify the characteristics of the cookies setup when you visit ims.iitgn.ac.in and also when you login to the student portal.

Ims in normal Google Chrome: Found 3 cookies out of which 2 are secure. None of the cookies are http only.



Ims through logging in Icognito tab: Found 1 cookie in the incognito tab which is not secure. The cookie is http only.

Ims through logging in normal Google Chrome: Found 4 cookies in the normal browser after signing in out of which 2 are secure. 1 cookie is http only.