FTP based File Transfer Application using Sockets

Sbonelo Mdluli : 1101772

School of Electrical & Information Engineering, University of the Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa

***Abstract*—This report presents an implementation of a file transfer application using TCP. The development of the application is per RFC959 which presents the standard for file transfer application implementation. The implementation contains multiple commands and a GUI to be used by the client. A multi threaded version of the application is also implemented to allow multiple user to connect to the server. The implementation can be further improved to support bulk transmission and to allow the user to have GUI version of the server. Passive mode can also be improved to support more multimedia file formats. Wireshark is used to verify the functionality of the implemented application.**

I. INTRODUCTION

File transfer protocol (FTP) is a computer network protocol used for transferring files between computers . FTP uses Transfer Control Protocol (TCP). FTP was invented at M.I.T in 1971 before TCP/IP existed. The implementation of FTP is based on a client-server model and specifications are described by Request for Comment (RFC) document. This report presents the implementation of a file transfer applications based FTP. The application is built using python 3.6 run on Ubuntu 20.04.

II. BACKGROUND

FTP ensures data integrity and that files between the client and server are transferred without data loss. A client is defined as a computer that accesses the resources of another computer. A server is an always on host that responds to client requests. The interaction between the client and server is through commands defined by the RFC. To begin a FTP session two separate TCP connections are required; a data connection on port 20 and a command connection on port 21. FTP provides two ways in which a client can connect to a server namely active mode and passive mode.

A. Active Mode

In active mode the client connects to the server on port 21 using any unprivileged port (P >1024) [1]. The server then initiates a data connection between it and the client. After receiving the response the client obtains an unused unprivileged port to use for data connection. The problem with active mode is that most clients are behind a firewall which may not allow TCP connections on privileged ports.

B. Passive Mode

In passive mode the client is the one that initiates a TCP data connection unlike in active mode where its the server that does so. This modification solves the firewall problem in active mode since data connection flows from client to server. This means that data connection from the server uses an unprivileged port instead of port 20 [1].

C. Data formats

Different file formats are transmitted using specific data formats. The most commonly used data formats are ASCII and image type. All data transfers must be terminated with and end-of-file (EOF) [2], [3],

- The ASCII data format is used to transfer text files. The data is converted to 8 bits before and after transmission.
- The image type is used for efficient storage and retrieval of data in binary format. The data is converted into 8 bits before being transmitted.

There are mainly two transmission modes for FTP namely stream and block mode. In stream mode the data is sent as a stream of bytes. This mode allows any file type to be sent. Block mode allows a file to be transmitted in a series of blocks. The last block must indicate the end-of-file.

D. Problems with FTP

FTP was not designed with security in mind, the data is transmitted without encryption it is only encoded. This means that data can be intercepted through a packet analyzer. The protocol fails to handle high volume requests. Another issue with FTP is that is does not produce an error log file or report if there were problems during the transmission of a file. This means that it takes a while for issues to be resolved. An alternative to FTP is HTTP which is a browser based file transfer protocol [4].

III. System overview

The application is accessed through the the terminal. The functionality of the application is based on the RFC 959, the response codes are also derived from this standard. The application was extensively tested on a single computer and error handling was implemented in order for the application not to crash during use.

A. Feature Overview

The basic core features are:

- Users can login after making requesting a connection to the server. The user logs in by providing a user name and password to gain access to the other features of the application.
- sending and downloading data
- Selecting transfer mode and data formats

Auxiliary features include:

- Printing current working directory
- changing working directory
- list directory files
- creating a new directory and deleting a file or directory.
- Multi threading is also used to allow multiple users to connect to the server

These features allow the user to manage their file structure in the server. Some functionality is not implemented.

B. Features not implemented

- All minimal client features were implemented as per the RFC.
- Server features not implemented include ability to interact with standard FTP client.
- The application has only been tested on a single machine.

IV. Command Overview

The client and server were developed simultaneously. In order to a connection between server and client a TCP connection is initiated using sockets. The code below is used to declare in a TCP connection in the server file. This code specifies the that the application is using stream mode to transfer data, the socket is binded to TCP_IP which is "127.0.1.1" by default but can be changed. Accept is used to accept new connections, it returns a socket object and the client address.

In order to connect to the server the user must input the FTP command, if a connection is successfully established a 220 response code is returned to the client.

```
commandSocket =
    socket.socket(socket.AF_INET,
    socket.SOCK_STREAM)
commandSocket.bind((TCP_IP, TCP_PORT))
commandSocket.listen(1)
CONN, clientControlAddress =
    commandSocket.accept()
```

A. loggin: FTP

The login process is needed to verify a users details. User are given default usernames and passwords. The login in details are saves in a text file in the server. Only the administrator who has access to the log file in the server has writes to edit the document. This data is retrieved and stored in a dictionary in the code. This is to ensure that we have unique usernames and we can quickly search for a user in the log file. To login the user inputs the command USER followed by their user name. If the user name is exists in the database 331 is returned to the user whereby they are prompted to input a password using PASS with their password if the password is valid response code 230 is sent to the client telling them that they have successfully logged in. After a successful login the current working directory is set to the users folder. The code is structured in such a way that the user will only have access to the commands after logging in.

The transfer mode is set to active mode by default meaning that the server listens for a connection from a client and port 20 is used for data and port 21 for exchanging commands.

B. Print current working directory: PWD

In order to print the current working directory the user uses PWD. A response code of 257 is sent to the client to indicate the command was executed successfully. The implementation uses python os module. The os pwd command prints the working directory starting from the host machine home. This is an issue in this application because it causes security concerns. Users should not know the full path to their folder because it means that users know how the server is configures and which files. With such information users can backtrack and delete other users content or even the script used to run the server. To deal with this only the path starting from the current user's folder is displayed.

C. Make directory: MKD

This allows the user to create their own sub folders. This folder also uses os module to folder manipulation.

This is done using the MKD command. If the folder already exists response code 251 is returned to the client and if the new folder was successfully created 257 is sent to the client.

D. Change working directory: CWD

This command allows the user to access sub folders within their designated root folder. The command takes the pathname to the destination. Response code 250 is returned indicating that the file action is successful and 553 if it did not complete. The command will not complete if the pathname does not exist.

E. List files: LIST

The LIST command is used to list directory contents. The contents displayed are from the current working directory of the user in the server. A response code of 150 indicated that the command successfully executes.

F. Remove files: DELE

The DELE command is used to delete or remove a file within a directory or This command takes the file that the user wishes to delete as an argument. To delete a file we use os.remove(file name).

G. Remove files: RMD

RMD is used to delete a directory and its contents. The command is very similar to the DELE command. The only difference is that it uses shutil.rmtree(file name) to delete a folder and its contents.

H. Uploading files to server: STOR

In order to upload files the client to server the user uses the STOR command to do. The command takes in the file name with file extension as an argument. The file the user intends on uploading to the server must be in the same directory as the client code. This is because the client code determines if the file exists before attempting to upload it. The response code 150 is used to indicate a successful file upload.

I. Downloading files from server: RETR

RETR is used to download previously stored files in the server. This command accepts the filename with file extension as an argument. A response code of 150 indicates a successful download from the server.

J. Quit connection: QUIT

The user can stop a TCP connection between themselves and the server by using the QUIT command. The command closes the previously opened TCP object that was initiated that was opened using the conn command.

K. Help: HELP

The help command displays available ftp commands to the user. The help command does not have a response code since it only implemented on the client side.

L. System: SYST

This command returns the underlying server operating system.

M. Passive mode: PASV

This command is used to initiate passive mode instead of the default active mode. This command tells the client to listen for a connection from the server.

N. Data port: PORT

This command is used to specify a data port for transmission. The commands takes a 32-bit host address and 16-bit port address. These are concatenated to be h1,h2,h2,h4,p1,p2 where h1 is the high order 8-bit representing the host address

V. Feature implementation

This section provides a detailed code implementation of the implemented commands and the GUI used by the client.

Client side code function implementation is wrapped in try catch blocks in order avoid the application from crashing and for error handling, exceptions are thrown if the client code receives an error. All methods in the client side code start by sending the user command to the server. The methods in the server and client have the same name as the commands used.

The client methods send commands to the server received from the user through the GUI, server then calls the appropriate method. The struct module is used to interpret and send binary data across a network. The pack method is used to transmit the data format and data in a compact form. The unpack method is used to unbudle the transmitted data. Below we discuss some of the code implementation in detail which might be confusing to the reader the other methods are simple to understand and contain documentation.

A. RETR

A file is broken down and sent in parts defined by the buffer size. We use a for loop to send the pieces to the client in a stream of bits. The file is opened using ’rb’ mode on the server side. The module read is used to convert the file contents into a string the file contents are sent as a string this is found between line 219 - 223 in the server code

The file is opened in ’wb’ mode on the client side. This means that the file is opened in binary mode, the mode also ensure that the data remains as is during transmission. The receiving loop on the client is controlled by the file size to be received, this means that the received file size is less that the one in the server because of how the code is structured.

B. STOR

Likewise this command operates in a similar method like RETR, the difference is that operations are reversed. The server accepts the file in ’wb’ and the client opens the file in ’rb’. The file is opened using read and is sent to the server as a string between line 148-143 in client code. The server receives the file in a while loop between line 101 - 105.

C. LIST

To list file we use os.listdir(os.getcwd()) to get the list of files and sub directories in the current directory. Using this we get a file listing as strings. The list is then traversed using a for loop to send the file list to the client. The client receives the number os files in the directory first in order to bound the receiving for loop line 179 - 188 in client code.

D. PORT

Determining the data port is solely determined by the server code. The server converts client IP address into a port number. In order to do this we multiply the 5th octet(sequence of 8bits) and add it with the 6th octet. This is done is line 187 in server code. The new port is then used to unitize the socket object.

E. PASV

In order to start a passive mode connection the PASV command initiates a connection from the server to the client in order to avoid firewall issues from the client machine. In order to do this we randomly select unprivileged port between 1024 - 10000. To get the IP address from port we divide the and use modulo operator (divmomde in code) to get a pair of two numbers. Those are then used to represent the server IP address.

F. GUI

The GUI is implemented using PyQt5 which python wrapper for Qt v5. PyQt5 provides a framework for GUI development in python [5] . The GUI was developed after the application was tested in the terminal, including the GUI introduced new complexities. The GUI was designed using Qt designer. In order to link the client implementation and the GUI we import the functions from the client code implementation. This makes the code simpler to understand and maintain. The GUI is adjustable to meet the users needs.

G. Multithreading

Multithreading is used to allow multiple users to simultaneously connect to the same server. The maximum number users of allowed is limited to 5. To close all connections between users a for loop is used to join all threads from the users. The same client code is used for both threaded and serial code.

VI. Implementation Critical Analysis

The GUI is minimal and can be improved to include more functionality such as selecting a file from the clients machine nor just the current directory. The GUI can also be made more robust to include error handling at the moment the GUI freezes when the server receives asynchronously instructions. The instructions should be synchronised for the GUI to not crash. The STOR and RETR command can also be improved to allow the user the user to send or receive multiple file at a time.

The LIST command also facing a similar problem of file listings being out of sync. This results in unpredictable behavior in producing consistent results. The GUI code as also be converted to a more streamline installation process. This can be through converting the code to an executable or appropriate format to supported by the operating system.

The STOR and RETR commands are capable of handling text, images, and multimedia files and pdf files in active mode see figure fig1 - fig3 appendix. However it fails to handle document file such as pdf and word documents and multimedia files in passive see fig 4- fig 19 in appendix.

Another issue occurs when one does traffic monitoring on the application. When a user is changing a directory the full path is exposed when viewed in wireshark. This a security concern because the user can use this knowledge to access other directories in the server. In order to avoid this the server administrator must edit each directory property to only allow a specific access to the appropriate user. Directory traversal should also be limited to the users parent directory. Common issues with TCP are also evident in this application. The data and commands are all sent in plain text, which means passwords and user names can be intercepted. These results can be seen in the appendix.

VII. Conclusion

In this paper we presented an implementation of a file transfer application using sockets. The implementation is done using python. A user simple interface is provided for the user in order for them to have a better experience with the application. A multithreaded version of the server is also implemented. Wireshark is used to perform a packet interception on the application. The application can be further improved and made more robust to support more file formats.

References

[1] S. Site, "Active ftp vs. passive ftp, a definitive explanation." https://slacksite.com/other/ftp.html. Accessed: 2020-06-10.

[2] W3, "File transfer functions." https://www.w3.org/Protocols/rfc959/4_FileTransfer.html. Accessed: 2020-05-30.

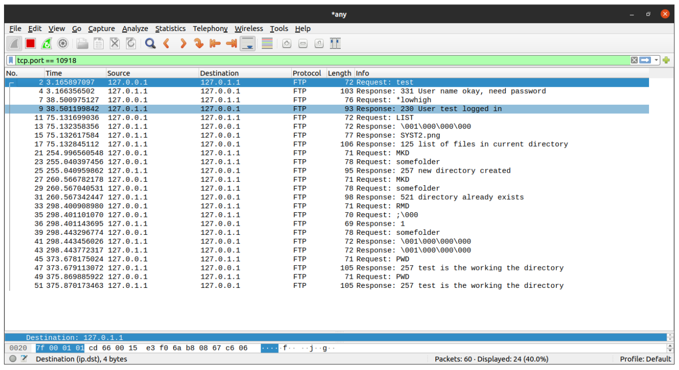
[3] J.Postel and Reynold, "File transfer protocol (ftp)." https://tools.ietf.org/html/rfc959. Accessed: 2020-03-21.

[4] A. Hughes, "The business problem of using file transfer protocol (ftp)." https://www.cleo.com/blog/knowledge-base-file-transfer-protocol-business-problem, note = Accessed: 2020-06-18.

[5] Pypi, "Pyqt5." https://pypi.org/project/PyQt5/. Accessed: 2020-06-20.

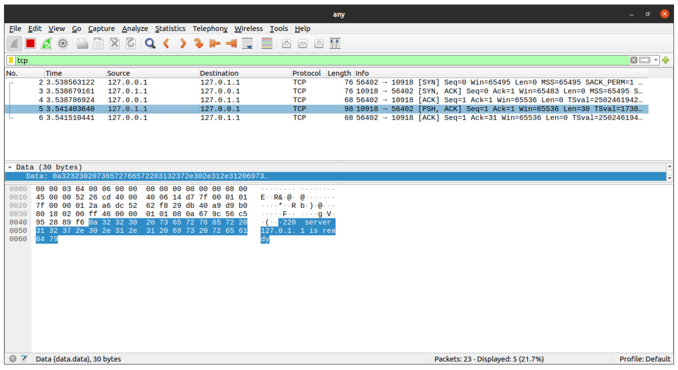
Appendix

The results below are from traffic monitoring the application using wireshark. The screenshots contain the command and server response code.



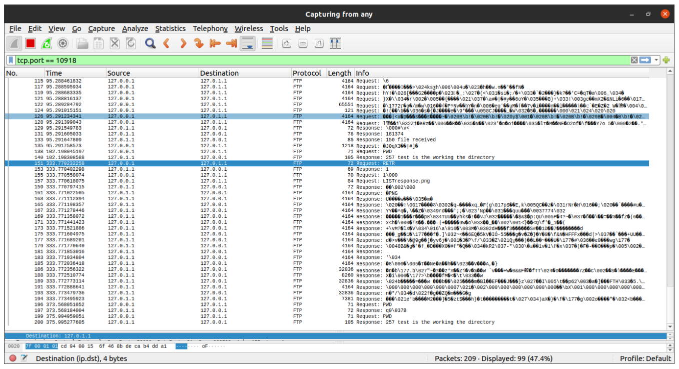
```
tcp.port == 10918
No.  Time          Source     Destination  Protocol Length Info
   2 3.165897097   127.0.0.1  127.0.1.1    FTP       72 Request: test
   4 3.166356502   127.0.1.1  127.0.0.1    FTP      103 Response: 331 User name okay, need password
   7 38.500975127  127.0.0.1  127.0.1.1    FTP       76 Request: *lowhigh
   9 38.501199842  127.0.1.1  127.0.0.1    FTP       93 Response: 230 User test logged in
  11 75.131699036  127.0.0.1  127.0.1.1    FTP       72 Request: LIST
  13 75.132358356  127.0.1.1  127.0.0.1    FTP       72 Response: \001\000\000\000
  15 75.132617584  127.0.1.1  127.0.0.1    FTP       77 Response: SYST2.png
  17 75.132845112  127.0.1.1  127.0.0.1    FTP      106 Response: 125 list of files in current directory
  21 254.996560548 127.0.0.1  127.0.1.1    FTP       71 Request: MKD
  23 255.040397456 127.0.0.1  127.0.1.1    FTP       78 Request: somefolder
  25 255.040959862 127.0.1.1  127.0.0.1    FTP       95 Response: 257 new directory created
  27 260.566782178 127.0.0.1  127.0.1.1    FTP       71 Request: MKD
  29 260.567040531 127.0.0.1  127.0.1.1    FTP       78 Request: somefolder
  31 260.567342447 127.0.1.1  127.0.0.1    FTP       98 Response: 521 directory already exists
  33 298.400908980 127.0.0.1  127.0.1.1    FTP       71 Request: RMD
  35 298.401101070 127.0.0.1  127.0.1.1    FTP       70 Request: ;\000
  36 298.401143695 127.0.1.1  127.0.0.1    FTP       69 Response: 1
  39 298.443296774 127.0.0.1  127.0.1.1    FTP       78 Request: somefolder
  41 298.443456026 127.0.1.1  127.0.0.1    FTP       72 Response: \001\000\000\000
  43 298.443772317 127.0.1.1  127.0.0.1    FTP       72 Response: \001\000\000\000
  45 373.678175024 127.0.0.1  127.0.1.1    FTP       71 Request: PWD
  47 373.679113072 127.0.1.1  127.0.0.1    FTP      105 Response: 257 test is the working the directory
  49 375.869885922 127.0.0.1  127.0.1.1    FTP       71 Request: PWD
  51 375.870173463 127.0.1.1  127.0.0.1    FTP      105 Response: 257 test is the working the directory

  Destination: 127.0.1.1
0020  7f 00 01 01 cd 66 00 15  e3 f0 6a b8 08 67 c6 06   ····f·· ··j··g··
Destination (ip.dst), 4 bytes          Packets: 60 · Displayed: 24 (40.0%)          Profile: Default
```

Fig. 1: FTP ACTIVE MODE



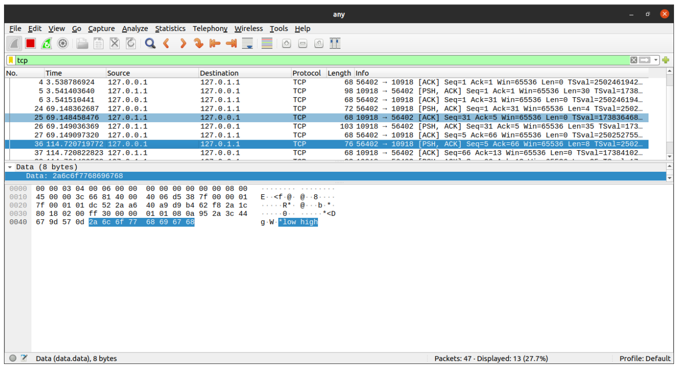
```
tcp
No.  Time         Source     Destination  Protocol Length Info
   2 3.538563122  127.0.0.1  127.0.1.1    TCP       76 56402 → 10918 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 …
   3 3.538679161  127.0.1.1  127.0.0.1    TCP       76 10918 → 56402 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 S…
   4 3.538786924  127.0.0.1  127.0.1.1    TCP       68 56402 → 10918 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2502461942…
   5 3.541403640  127.0.1.1  127.0.0.1    TCP       98 10918 → 56402 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=30 TSval=1738…
   6 3.541510441  127.0.0.1  127.0.1.1    TCP       68 56402 → 10918 [ACK] Seq=1 Ack=31 Win=65536 Len=0 TSval=250246194…

▾ Data (30 bytes)
    Data: 0a3232302073657276657220313237 2e302e312e31206973…
0000  00 00 03 04 00 06 00 00  00 00 00 00 00 00 08 00   ········ ········
0010  45 00 00 52 26 cd 40 00  40 06 14 d7 7f 00 01 01   E··R&·@· @·······
0020  7f 00 00 01 2a a6 dc 52  62 f8 29 db 40 a9 d9 b0   ····*··R b·)·@···
0030  80 18 02 00 ff 46 00 00  01 01 08 0a 67 9c 56 c5   ·····F·· ····g·V·
0040  95 28 89 f6 0a 32 32 30  20 73 65 72 76 65 72 20   ·(···220  server
0050  31 32 37 2e 30 2e 31 2e  31 20 69 73 20 72 65 61   127.0.1. 1 is rea
0060  64 79                                              dy
Data (data.data), 30 bytes          Packets: 23 · Displayed: 5 (21.7%)          Profile: Default
```

Fig. 4: Initiate TCP connection server response



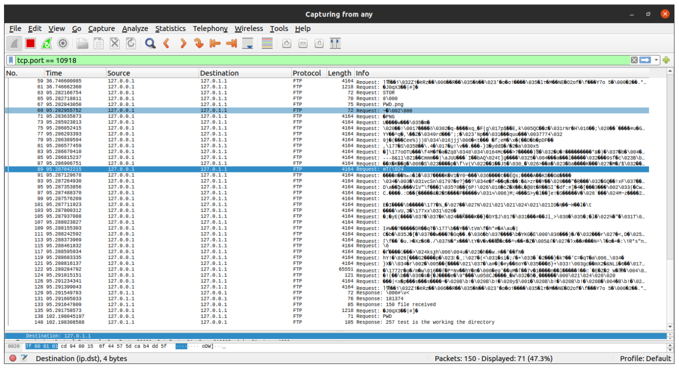
```
tcp.port == 10918
Destination (ip.dst), 4 bytes          Packets: 209 · Displayed: 99 (47.4%)          Profile: Default
```

Fig. 2: FTP ACTIVE MODE download file



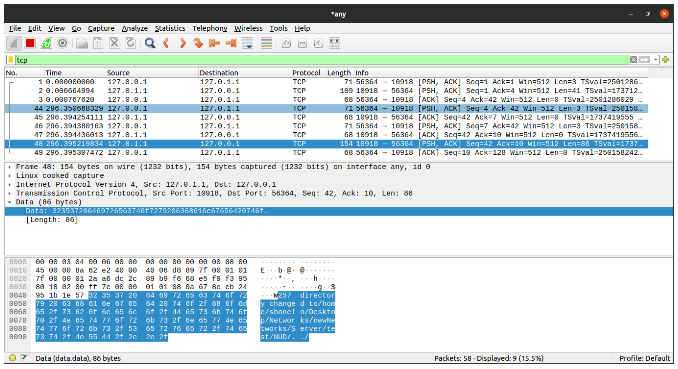
```
tcp
No.  Time           Source     Destination  Protocol Length Info
   4 3.538786924    127.0.0.1  127.0.1.1    TCP       68 56402 → 10918 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2502461942…
   5 3.541403640    127.0.1.1  127.0.0.1    TCP       98 10918 → 56402 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=30 TSval=1738…
   6 3.541510441    127.0.0.1  127.0.1.1    TCP       68 56402 → 10918 [ACK] Seq=1 Ack=31 Win=65536 Len=0 TSval=250246194…
  24 69.148362687   127.0.0.1  127.0.1.1    TCP       72 56402 → 10918 [PSH, ACK] Seq=1 Ack=31 Win=65536 Len=4 TSval=2502…
  25 69.148458476   127.0.1.1  127.0.0.1    TCP       68 10918 → 56402 [ACK] Seq=31 Ack=5 Win=65536 Len=0 TSval=173836468…
  26 69.149036369   127.0.1.1  127.0.0.1    TCP      103 10918 → 56402 [PSH, ACK] Seq=31 Ack=5 Win=65536 Len=35 TSval=173…
  27 69.149097320   127.0.0.1  127.0.1.1    TCP       68 56402 → 10918 [ACK] Seq=5 Ack=66 Win=65536 Len=0 TSval=250252755…
  36 114.720719772  127.0.0.1  127.0.1.1    TCP       76 56402 → 10918 [PSH, ACK] Seq=5 Ack=66 Win=65536 Len=8 TSval=2502…
  37 114.720822823  127.0.1.1  127.0.0.1    TCP       68 10918 → 56402 [ACK] Seq=66 Ack=13 Win=65536 Len=0 TSval=17384102…
▾ Data (8 bytes)
    Data: 2a6c6f7768696768
0000  00 00 03 04 00 06 00 00  00 00 00 00 00 00 08 00   ········ ········
0010  45 00 00 3c 66 81 40 00  40 06 d5 38 7f 00 00 01   E··<f·@· @··8····
0020  7f 00 01 01 dc 52 2a a6  40 a9 d9 b4 62 f8 2a 1c   ·····R*· @···b·*·
0030  80 18 02 00 ff 30 00 00  01 01 08 0a 95 2a 3c 44   ·····0·· ·····*<D
0040  67 9d 57 0d 2a 6c 6f 77  68 69 67 68               g·W·*low high
Data (data.data), 8 bytes          Packets: 47 · Displayed: 13 (27.7%)          Profile: Default
```

Fig. 5: Password interception



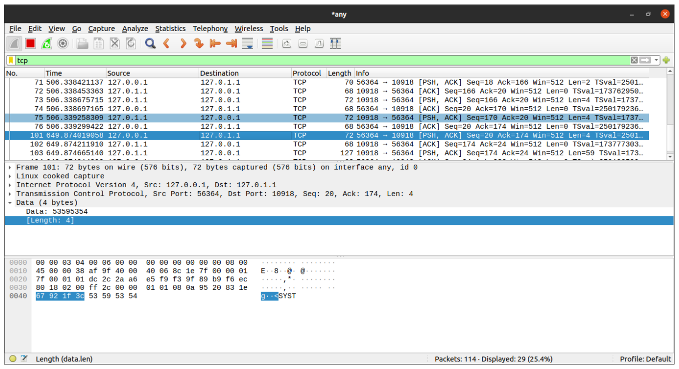
```
tcp.port == 10918
Destination (ip.dst), 4 bytes          Packets: 150 · Displayed: 71 (47.3%)          Profile: Default
```

Fig. 3: FTP ACTIVE MODE stor file



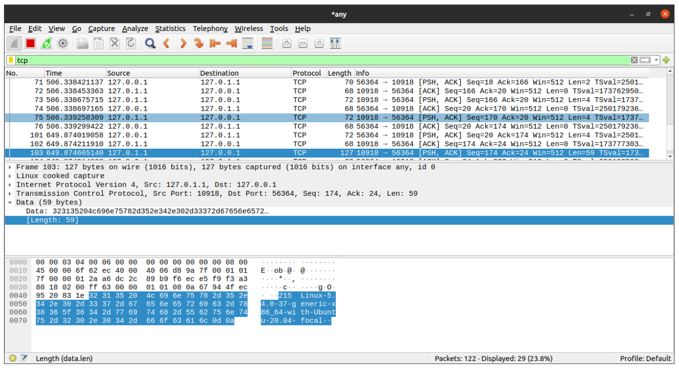
```
tcp
No.  Time           Source     Destination  Protocol Length Info
   1 0.000000000    127.0.0.1  127.0.1.1    TCP       71 56364 → 10918 [PSH, ACK] Seq=1 Ack=1 Win=512 Len=3 TSval=2501286…
   2 0.000664994    127.0.1.1  127.0.0.1    TCP      109 10918 → 56364 [PSH, ACK] Seq=1 Ack=4 Win=512 Len=41 TSval=173712…
   3 0.000767620    127.0.0.1  127.0.1.1    TCP       68 56364 → 10918 [ACK] Seq=4 Ack=42 Win=512 Len=0 TSval=2501286029 …
  44 296.350668329  127.0.0.1  127.0.1.1    TCP       71 56364 → 10918 [PSH, ACK] Seq=4 Ack=42 Win=512 Len=3 TSval=250158…
  45 296.394254111  127.0.1.1  127.0.0.1    TCP       68 10918 → 56364 [ACK] Seq=42 Ack=7 Win=512 Len=0 TSval=1737419555 …
  46 296.394380163  127.0.0.1  127.0.1.1    TCP       71 56364 → 10918 [PSH, ACK] Seq=7 Ack=42 Win=512 Len=3 TSval=250158…
  47 296.394436013  127.0.1.1  127.0.0.1    TCP       68 10918 → 56364 [ACK] Seq=42 Ack=10 Win=512 Len=0 TSval=1737419556…
  48 296.395219834  127.0.1.1  127.0.0.1    TCP      154 10918 → 56364 [PSH, ACK] Seq=42 Ack=10 Win=512 Len=86 TSval=1737…
  49 296.395307472  127.0.0.1  127.0.1.1    TCP       68 56364 → 10918 [ACK] Seq=10 Ack=128 Win=512 Len=0 TSval=250158242…
› Frame 48: 154 bytes on wire (1232 bits), 154 bytes captured (1232 bits) on interface any, id 0
› Linux cooked capture
› Internet Protocol Version 4, Src: 127.0.1.1, Dst: 127.0.0.1
› Transmission Control Protocol, Src Port: 10918, Dst Port: 56364, Seq: 42, Ack: 10, Len: 86
▾ Data (86 bytes)
    Data: 323537206469726563746f7279206368616e67656420746f…
    [Length: 86]
0000  00 00 03 04 00 06 00 00  00 00 00 00 00 00 08 00   ········ ········
0010  45 00 00 8a 62 e2 40 00  40 06 d8 89 7f 00 01 01   E···b·@· @·······
0020  7f 00 00 01 2a a6 dc 2c  89 b9 f6 68 e5 f9 f3 95   ····*··, ···h····
0030  80 18 02 00 ff 7e 00 00  01 01 08 0a 67 8e eb 24   ·····~·· ····g··$
0040  95 1b 1e 57 32 35 37 20  64 69 72 65 63 74 6f 72   ···W257  director
0050  79 20 63 68 61 6e 67 65  64 20 74 6f 2f 68 6f 6d   y change d to/hom
0060  65 2f 73 62 6f 6e 65 6c  6f 2f 44 65 73 6b 74 6f   e/sbonel o/Deskto
0070  70 2f 4e 65 74 77 6f 72  6b 73 2f 6e 65 77 4e 65   p/Networ ks/newNe
0080  74 77 6f 72 6b 73 2f 53  65 72 76 65 72 2f 74 65   tworks/S erver/te
0090  73 74 2f 4e 55 44 2f 2e  2e 2f                     st/NUD/. ./
Data (data.data), 86 bytes          Packets: 58 · Displayed: 9 (15.5%)          Profile: Default
```

Fig. 6: Change working directory response



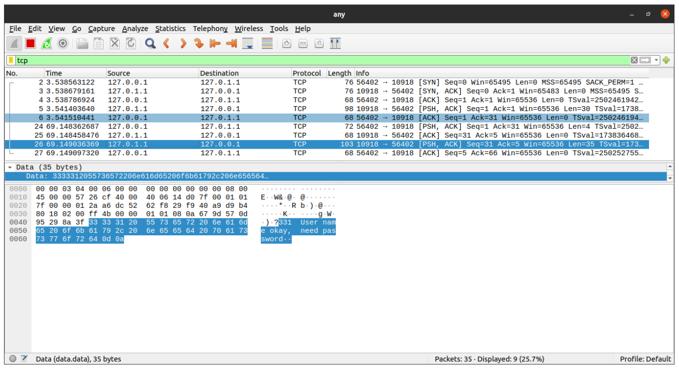
```
*any
tcp
No.  Time           Source      Destination  Protocol Length Info
  71 506.338421137  127.0.0.1   127.0.1.1    TCP      70 56364 → 10918 [PSH, ACK] Seq=18 Ack=166 Win=512 Len=2 TSval=2501…
  72 506.338453363  127.0.1.1   127.0.0.1    TCP      68 10918 → 56364 [ACK] Seq=166 Ack=20 Win=512 Len=0 TSval=173762950…
  73 506.338675715  127.0.1.1   127.0.0.1    TCP      72 10918 → 56364 [PSH, ACK] Seq=166 Ack=20 Win=512 Len=4 TSval=1737…
  74 506.338697165  127.0.0.1   127.0.1.1    TCP      68 56364 → 10918 [ACK] Seq=20 Ack=170 Win=512 Len=0 TSval=250179236…
  75 506.339258309  127.0.1.1   127.0.0.1    TCP      72 10918 → 56364 [PSH, ACK] Seq=170 Ack=20 Win=512 Len=4 TSval=1737…
  76 506.339299422  127.0.0.1   127.0.1.1    TCP      68 56364 → 10918 [ACK] Seq=20 Ack=174 Win=512 Len=0 TSval=250179236…
 101 649.874019058  127.0.0.1   127.0.1.1    TCP      72 56364 → 10918 [PSH, ACK] Seq=20 Ack=174 Win=512 Len=4 TSval=2501…
 102 649.874211910  127.0.1.1   127.0.0.1    TCP      68 10918 → 56364 [ACK] Seq=174 Ack=24 Win=512 Len=0 TSval=173777303…
 103 649.874665140  127.0.1.1   127.0.0.1    TCP     127 10918 → 56364 [PSH, ACK] Seq=174 Ack=24 Win=512 Len=59 TSval=173…

› Frame 101: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface any, id 0
› Linux cooked capture
› Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.1.1
› Transmission Control Protocol, Src Port: 56364, Dst Port: 10918, Seq: 20, Ack: 174, Len: 4
⌄ Data (4 bytes)
    Data: 53595354
    [Length: 4]

0000  00 00 03 04 00 06 00 00  00 00 00 00 00 00 08 00   ········ ········
0010  45 00 00 38 af 9f 40 00  40 06 8c 1e 7f 00 00 01   E··8··@· @·······
0020  7f 00 01 01 dc 2c 2a a6  e5 f9 f3 9f 89 b9 f6 ec   ·····,*· ········
0030  80 18 02 00 ff 2c 00 00  01 01 08 0a 95 20 83 1e   ·····,·· ····· ··
0040  67 92 1f 3c 53 59 53 54                            g··<SYST

Length (data.len)          Packets: 114 · Displayed: 29 (25.4%)          Profile: Default
```

Fig. 7: SYST command



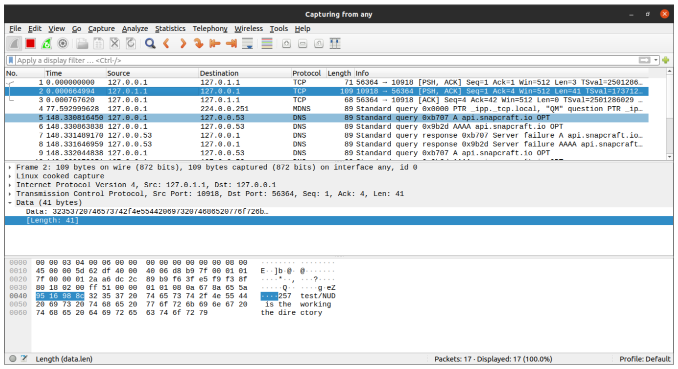
```
*any
tcp
No.  Time           Source      Destination  Protocol Length Info
  71 506.338421137  127.0.0.1   127.0.1.1    TCP      70 56364 → 10918 [PSH, ACK] Seq=18 Ack=166 Win=512 Len=2 TSval=2501…
  72 506.338453363  127.0.1.1   127.0.0.1    TCP      68 10918 → 56364 [ACK] Seq=166 Ack=20 Win=512 Len=0 TSval=173762950…
  73 506.338675715  127.0.1.1   127.0.0.1    TCP      72 10918 → 56364 [PSH, ACK] Seq=166 Ack=20 Win=512 Len=4 TSval=1737…
  74 506.338697165  127.0.0.1   127.0.1.1    TCP      68 56364 → 10918 [ACK] Seq=20 Ack=170 Win=512 Len=0 TSval=250179236…
  75 506.339258309  127.0.1.1   127.0.0.1    TCP      72 10918 → 56364 [PSH, ACK] Seq=170 Ack=20 Win=512 Len=4 TSval=1737…
  76 506.339299422  127.0.0.1   127.0.1.1    TCP      68 56364 → 10918 [ACK] Seq=20 Ack=174 Win=512 Len=0 TSval=250179236…
 101 649.874019058  127.0.0.1   127.0.1.1    TCP      72 56364 → 10918 [PSH, ACK] Seq=20 Ack=174 Win=512 Len=4 TSval=2501…
 102 649.874211910  127.0.1.1   127.0.0.1    TCP      68 10918 → 56364 [ACK] Seq=174 Ack=24 Win=512 Len=0 TSval=173777303…
 103 649.874665140  127.0.1.1   127.0.0.1    TCP     127 10918 → 56364 [PSH, ACK] Seq=174 Ack=24 Win=512 Len=59 TSval=173…

› Frame 103: 127 bytes on wire (1016 bits), 127 bytes captured (1016 bits) on interface any, id 0
› Linux cooked capture
› Internet Protocol Version 4, Src: 127.0.1.1, Dst: 127.0.0.1
› Transmission Control Protocol, Src Port: 10918, Dst Port: 56364, Seq: 174, Ack: 24, Len: 59
⌄ Data (59 bytes)
    Data: 323135204c696e75782d352e342e302d33372d67656e6572…
    [Length: 59]

0000  00 00 03 04 00 06 00 00  00 00 00 00 00 00 08 00   ········ ········
0010  45 00 00 6f 62 ec 40 00  40 06 d8 9a 7f 00 01 01   E··ob·@· @·······
0020  7f 00 00 01 2a a6 dc 2c  89 b9 f6 ec e5 f9 f3 a3   ····*··, ········
0030  80 18 02 00 ff 63 00 00  01 01 08 0a 67 94 4f ec   ·····c·· ····g·O·
0040  95 20 83 1e 32 31 35 20  4c 69 6e 75 78 2d 35 2e   · ··215  Linux-5.
0050  34 2e 30 2d 33 37 2d 67  65 6e 65 72 69 63 2d 78   4.0-37-g eneric-x
0060  38 36 5f 36 34 2d 77 69  74 68 2d 55 62 75 6e 74   86_64-wi th-Ubunt
0070  75 2d 32 30 2e 30 34 2d  66 6f 63 61 6c 0d 0a      u-20.04- focal··

Length (data.len)          Packets: 122 · Displayed: 29 (23.8%)          Profile: Default
```

Fig. 8: SYST server response



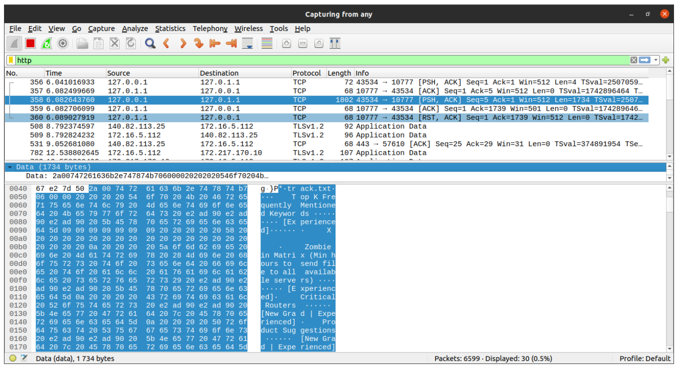
```
any
tcp
No.  Time          Source      Destination  Protocol Length Info
   2 3.538563122   127.0.0.1   127.0.1.1    TCP      76 56402 → 10918 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 …
   3 3.538679161   127.0.1.1   127.0.0.1    TCP      76 10918 → 56402 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 S…
   4 3.538786924   127.0.0.1   127.0.1.1    TCP      68 56402 → 10918 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2502461942…
   5 3.541403640   127.0.1.1   127.0.0.1    TCP      98 10918 → 56402 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=30 TSval=1738…
   6 3.541510441   127.0.0.1   127.0.1.1    TCP      68 56402 → 10918 [ACK] Seq=1 Ack=31 Win=65536 Len=0 TSval=250246194…
  24 69.148362687  127.0.0.1   127.0.1.1    TCP      72 56402 → 10918 [PSH, ACK] Seq=1 Ack=31 Win=65536 Len=4 TSval=2502…
  25 69.148458476  127.0.1.1   127.0.0.1    TCP      68 10918 → 56402 [ACK] Seq=31 Ack=5 Win=65536 Len=0 TSval=173836468…
  26 69.149036369  127.0.1.1   127.0.0.1    TCP     103 10918 → 56402 [PSH, ACK] Seq=31 Ack=5 Win=65536 Len=35 TSval=173…
  27 69.149097320  127.0.0.1   127.0.1.1    TCP      68 56402 → 10918 [ACK] Seq=5 Ack=66 Win=65536 Len=0 TSval=250252755…

⌄ Data (35 bytes)
    Data: 3333312055736572206e616d65206f6b61792c206e656564…

0000  00 00 03 04 00 06 00 00  00 00 00 00 00 00 08 00   ········ ········
0010  45 00 00 57 26 cf 40 00  40 06 14 d0 7f 00 01 01   E··W&·@· @·······
0020  7f 00 00 01 2a a6 dc 52  62 f8 29 f9 40 a9 d9 b4   ····*··R b·)·@···
0030  80 18 02 00 ff 4b 00 00  01 01 08 0a 67 9d 57 0d   ·····K·· ····g·W·
0040  95 29 8a 3f 33 33 31 20  55 73 65 72 20 6e 61 6d   ·)·?331  User nam
0050  65 20 6f 6b 61 79 2c 20  6e 65 65 64 20 70 61 73   e okay,  need pas
0060  73 77 6f 72 64 0d 0a                               sword··

Data (data.data), 35 bytes          Packets: 35 · Displayed: 9 (25.7%)          Profile: Default
```

Fig. 9: user authorisation response



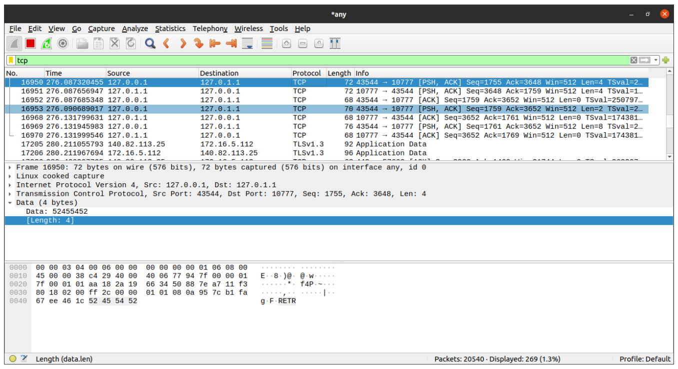
```
Capturing from any
Apply a display filter … <Ctrl-/>
No.  Time           Source      Destination  Protocol Length Info
   1 0.000000000    127.0.0.1   127.0.1.1    TCP      71 56364 → 10918 [PSH, ACK] Seq=1 Ack=1 Win=512 Len=3 TSval=2501286…
   2 0.000664994    127.0.1.1   127.0.0.1    TCP     109 10918 → 56364 [PSH, ACK] Seq=1 Ack=4 Win=512 Len=41 TSval=173712…
   3 0.000767620    127.0.0.1   127.0.1.1    TCP      68 56364 → 10918 [ACK] Seq=4 Ack=42 Win=512 Len=0 TSval=2501286029 …
   4 77.592999628   127.0.0.1   224.0.0.251  MDNS     89 Standard query 0x0000 PTR _ipp._tcp.local, "QM" question PTR _ip…
   5 148.330816450  127.0.0.1   127.0.0.53   DNS      89 Standard query 0xb707 A api.snapcraft.io OPT
   6 148.330863838  127.0.0.1   127.0.0.53   DNS      89 Standard query 0x9b2d AAAA api.snapcraft.io OPT
   7 148.331489170  127.0.0.53  127.0.0.1    DNS      89 Standard query response 0xb707 Server failure A api.snapcraft.io…
   8 148.331646959  127.0.0.53  127.0.0.1    DNS      89 Standard query response 0x9b2d Server failure AAAA api.snapcraft…
   9 148.332044838  127.0.0.1   127.0.0.53   DNS      89 Standard query 0xb707 A api.snapcraft.io OPT

› Frame 2: 109 bytes on wire (872 bits), 109 bytes captured (872 bits) on interface any, id 0
› Linux cooked capture
› Internet Protocol Version 4, Src: 127.0.1.1, Dst: 127.0.0.1
› Transmission Control Protocol, Src Port: 10918, Dst Port: 56364, Seq: 1, Ack: 4, Len: 41
⌄ Data (41 bytes)
    Data: 32353720746573742f4e554420697320746865207776f726b…
    [Length: 41]

0000  00 00 03 04 00 06 00 00  00 00 00 00 00 00 08 00   ········ ········
0010  45 00 00 5d 62 df 40 00  40 06 d8 b9 7f 00 01 01   E··]b·@· @·······
0020  7f 00 00 01 2a a6 dc 2c  89 b9 f6 3f e5 f9 f3 8f   ····*··, ···?····
0030  80 18 02 00 ff 51 00 00  01 01 08 0a 67 8a 65 5a   ·····Q·· ····g·eZ
0040  95 16 98 8c 32 35 37 20  74 65 73 74 2f 4e 55 44   ····257  test/NUD
0050  20 69 73 20 74 68 65 20  77 6f 72 6b 69 6e 67 20    is the  working
0060  74 68 65 20 64 69 72 65  63 74 6f 72 79            the dire ctory

Length (data.len)          Packets: 17 · Displayed: 17 (100.0%)          Profile: Default
```

Fig. 10: SYST



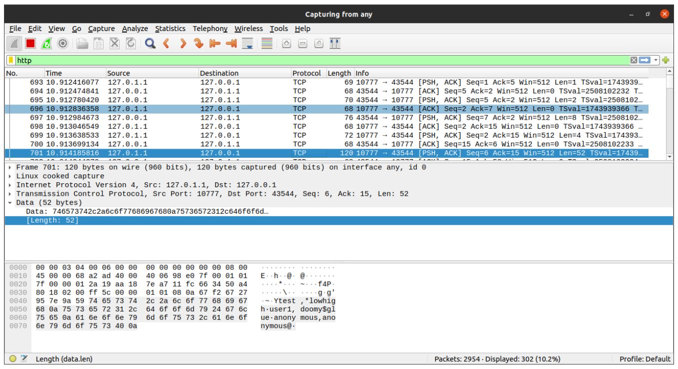
```
Capturing from any
http
No.  Time           Source         Destination    Protocol Length Info
 356 6.041016933    127.0.0.1      127.0.1.1      TCP       72 43534 → 10777 [PSH, ACK] Seq=1 Ack=1 Win=512 Len=4 TSval=2507059…
 357 6.082499669    127.0.1.1      127.0.0.1      TCP       68 10777 → 43534 [ACK] Seq=1 Ack=5 Win=512 Len=0 TSval=1742896464 T…
 358 6.082643760    127.0.0.1      127.0.1.1      TCP     1802 43534 → 10777 [PSH, ACK] Seq=5 Ack=1 Win=512 Len=1734 TSval=2507…
 359 6.082706099    127.0.1.1      127.0.0.1      TCP       68 10777 → 43534 [ACK] Seq=1 Ack=1739 Win=501 Len=0 TSval=174289646…
 360 6.089027919    127.0.1.1      127.0.0.1      TCP       68 10777 → 43534 [RST, ACK] Seq=1 Ack=1739 Win=512 Len=0 TSval=1742…
 508 8.792374597    140.82.113.25  172.16.5.112   TLSv1.2   92 Application Data
 509 8.792824232    172.16.5.112   140.82.113.25  TLSv1.2   96 Application Data
 531 9.052681080    140.82.113.25  172.16.5.112   TCP       68 443 → 57610 [ACK] Seq=25 Ack=29 Win=31 Len=0 TSval=374891954 TSe…
 782 12.538802645   172.16.5.112   172.217.170.10 TLSv1.2  107 Application Data

⌄ Data (1734 bytes)
    Data: 2a00747261636b2e747874b70600002020202020546f70204b…

0040  67 e2 7d 50 2a 00 74 72  61 63 6b 2e 74 78 74 b7   g·}P*·tr ack.txt·
0050  06 00 00 20 20 20 20 54  6f 70 20 4b 20 46 72 65   ···    T op K Fre
0060  71 75 65 6e 74 6c 79 20  4d 65 6e 74 69 6f 6e 65   quently  Mentione
0070  64 20 4b 65 79 77 6f 72  64 73 20 e2 ad 90 e2 ad   d Keywor ds ·····
0080  90 e2 ad 90 20 5b 45 78  70 65 72 69 65 6e 63 65   ···· [Ex perience
0090  64 5d 09 09 09 09 09 09  09 20 20 20 20 20 58 20   d]······ ·     X
00a0  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
00b0  20 20 20 20 0a 20 20 20  20 5a 6f 6d 62 69 65 20       ·     Zombie
00c0  69 6e 20 4d 61 74 72 69  78 20 28 4d 69 6e 20 68   in Matri x (Min h
00d0  6f 75 72 73 20 74 6f 20  73 65 6e 64 20 66 69 6c   ours to  send fil
00e0  65 20 74 6f 20 61 6c 6c  20 61 76 61 69 6c 61 62   e to all  availab
00f0  6c 65 20 73 65 72 76 65  72 73 29 20 e2 ad 90 e2   le serve rs) ····
0100  ad 90 e2 ad 90 20 5b 45  78 70 65 72 69 65 6e 63   ····· [E xperienc
0110  65 64 5d 0a 20 20 20 20  43 72 69 74 69 63 61 6c   ed]·     Critical
0120  20 52 6f 75 74 65 72 73  20 e2 ad 90 e2 ad 90 20    Routers  ······
0130  5b 4e 65 77 20 47 72 61  64 20 7c 20 45 78 70 65   [New Gra d | Expe
0140  72 69 65 6e 63 65 64 5d  0a 20 20 20 20 50 72 6f   rienced] ·    Pro
0150  64 75 63 74 20 53 75 67  67 65 73 74 69 6f 6e 73   duct Sug gestions
0160  20 e2 ad 90 e2 ad 90 20  5b 4e 65 77 20 47 72 61    ······  [New Gra
0170  64 20 7c 20 45 78 70 65  72 69 65 6e 63 65 64 5d   d | Expe rienced]

Data (data), 1 734 bytes          Packets: 6599 · Displayed: 30 (0.5%)          Profile: Default
```

Fig. 11: STRO packet interception



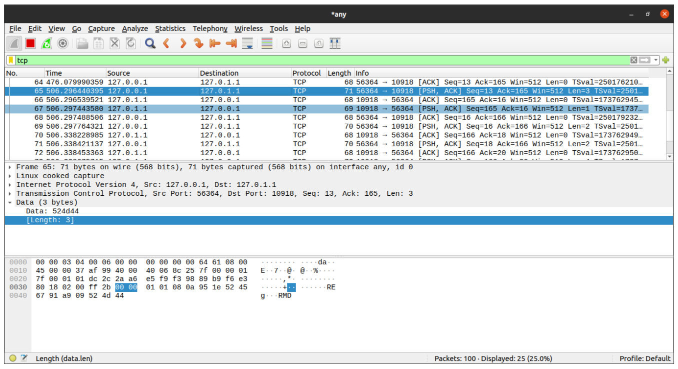
```
*any
tcp
No.    Time            Source         Destination    Protocol Length Info
 16950 276.087320455   127.0.0.1      127.0.1.1      TCP      72 43544 → 10777 [PSH, ACK] Seq=1755 Ack=3648 Win=512 Len=4 TSval=2…
 16951 276.087656947   127.0.1.1      127.0.0.1      TCP      72 10777 → 43544 [PSH, ACK] Seq=3648 Ack=1759 Win=512 Len=4 TSval=1…
 16952 276.087685348   127.0.0.1      127.0.1.1      TCP      68 43544 → 10777 [ACK] Seq=1759 Ack=3652 Win=512 Len=0 TSval=250797…
 16953 276.090689017   127.0.0.1      127.0.1.1      TCP      70 43544 → 10777 [PSH, ACK] Seq=1759 Ack=3652 Win=512 Len=2 TSval=2…
 16968 276.131799631   127.0.1.1      127.0.0.1      TCP      68 10777 → 43544 [ACK] Seq=3652 Ack=1761 Win=512 Len=0 TSval=174381…
 16969 276.131945983   127.0.0.1      127.0.1.1      TCP      76 43544 → 10777 [PSH, ACK] Seq=1761 Ack=3652 Win=512 Len=8 TSval=2…
 16970 276.131999546   127.0.1.1      127.0.0.1      TCP      68 10777 → 43544 [ACK] Seq=3652 Ack=1769 Win=512 Len=0 TSval=174381…
 17205 280.211055793   140.82.113.25  172.16.5.112   TLSv1.3  92 Application Data
 17206 280.211967694   172.16.5.112   140.82.113.25  TLSv1.3  96 Application Data

› Frame 16950: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface any, id 0
› Linux cooked capture
› Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.1.1
› Transmission Control Protocol, Src Port: 43544, Dst Port: 10777, Seq: 1755, Ack: 3648, Len: 4
⌄ Data (4 bytes)
    Data: 52455452
    [Length: 4]

0000  00 00 03 04 00 06 00 00  00 00 00 00 01 06 08 00   ········ ········
0010  45 00 00 38 c4 29 40 00  40 06 77 94 7f 00 00 01   E··8·)@· @·w·····
0020  7f 00 01 01 aa 18 2a 19  66 34 50 88 7e a7 11 f3   ······*· f4P·~···
0030  80 18 02 00 ff 2c 00 00  01 01 08 0a 95 7c b1 fa   ·····,·· ·····|··
0040  67 ee 46 1c 52 45 54 52                            g·F·RETR

Length (data.len)          Packets: 20540 · Displayed: 269 (1.3%)          Profile: Default
```

Fig. 12: RETR packet interception


Capturing from any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http

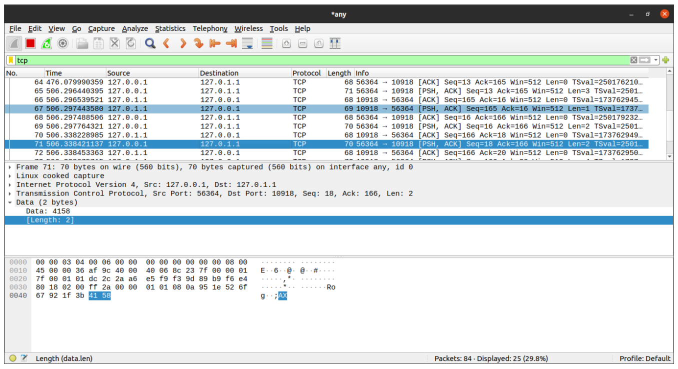
```
No.   Time         Source     Destination Protocol Length Info
  693 10.912416077 127.0.1.1  127.0.0.1   TCP       69 10777 → 43544 [PSH, ACK] Seq=1 Ack=5 Win=512 Len=1 TSval=1743939…
  694 10.912474841 127.0.0.1  127.0.1.1   TCP       68 43544 → 10777 [ACK] Seq=5 Ack=2 Win=512 Len=0 TSval=2508102232 T…
  695 10.912780420 127.0.0.1  127.0.1.1   TCP       70 43544 → 10777 [PSH, ACK] Seq=5 Ack=2 Win=512 Len=2 TSval=2508102…
  696 10.912836358 127.0.1.1  127.0.0.1   TCP       68 10777 → 43544 [ACK] Seq=2 Ack=7 Win=512 Len=0 TSval=1743939366 T…
  697 10.912984673 127.0.0.1  127.0.1.1   TCP       76 43544 → 10777 [PSH, ACK] Seq=7 Ack=2 Win=512 Len=8 TSval=2508102…
  698 10.913046549 127.0.1.1  127.0.0.1   TCP       68 10777 → 43544 [ACK] Seq=2 Ack=15 Win=512 Len=0 TSval=1743939366 …
  699 10.913638533 127.0.1.1  127.0.0.1   TCP       72 10777 → 43544 [PSH, ACK] Seq=2 Ack=15 Win=512 Len=4 TSval=174393…
  700 10.913699134 127.0.0.1  127.0.1.1   TCP       68 43544 → 10777 [ACK] Seq=15 Ack=6 Win=512 Len=0 TSval=2508102233 …
  701 10.914185816 127.0.1.1  127.0.0.1   TCP      120 10777 → 43544 [PSH, ACK] Seq=6 Ack=15 Win=512 Len=52 TSval=17439…

› Frame 701: 120 bytes on wire (960 bits), 120 bytes captured (960 bits) on interface any, id 0
› Linux cooked capture
› Internet Protocol Version 4, Src: 127.0.1.1, Dst: 127.0.0.1
› Transmission Control Protocol, Src Port: 10777, Dst Port: 43544, Seq: 6, Ack: 15, Len: 52
⌄ Data (52 bytes)
    Data: 746573742c2a6c6f77686967680a75736572312c646f6f6d…
    [Length: 52]

0000  00 00 03 04 00 06 00 00  00 00 00 00 00 00 08 00   ········ ········
0010  45 00 00 68 a2 ad 40 00  40 06 98 e0 7f 00 01 01   E··h··@· @·······
0020  7f 00 00 01 2a 19 aa 18  7e a7 11 fc 66 34 50 a4   ····*··· ~···f4P·
0030  80 18 02 00 ff 5c 00 00  01 01 08 0a 67 f2 67 27   ·····\·· ····g·g'
0040  95 7e 9a 59 74 65 73 74  2c 2a 6c 6f 77 68 69 67   ·~·Ytest ,*lowhig
0050  68 0a 75 73 65 72 31 2c  64 6f 6f 6d 79 24 67 6c   h·user1, doomy$gl
0060  75 65 0a 61 6e 6f 6e 79  6d 6f 75 73 2c 61 6e 6f   ue·anony mous,ano
0070  6e 79 6d 6f 75 73 40 0a                            nymous@·

Length (data.len)          Packets: 2954 · Displayed: 302 (10.2%)          Profile: Default
```

Fig. 13: SYST


*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp

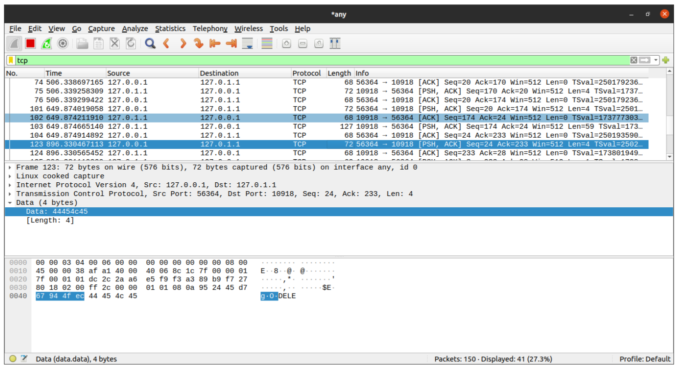
```
No.   Time          Source     Destination Protocol Length Info
   64 476.079990359 127.0.0.1  127.0.1.1   TCP       68 56364 → 10918 [ACK] Seq=13 Ack=165 Win=512 Len=0 TSval=250176210…
   65 506.296440395 127.0.0.1  127.0.1.1   TCP       71 56364 → 10918 [PSH, ACK] Seq=13 Ack=165 Win=512 Len=3 TSval=2501…
   66 506.296539521 127.0.1.1  127.0.0.1   TCP       68 10918 → 56364 [ACK] Seq=165 Ack=16 Win=512 Len=0 TSval=173762945…
   67 506.297443580 127.0.1.1  127.0.0.1   TCP       69 10918 → 56364 [PSH, ACK] Seq=165 Ack=16 Win=512 Len=1 TSval=1737…
   68 506.297488506 127.0.0.1  127.0.1.1   TCP       68 56364 → 10918 [ACK] Seq=16 Ack=166 Win=512 Len=0 TSval=250179232…
   69 506.297764321 127.0.0.1  127.0.1.1   TCP       70 56364 → 10918 [PSH, ACK] Seq=16 Ack=166 Win=512 Len=2 TSval=2501…
   70 506.338228985 127.0.1.1  127.0.0.1   TCP       68 10918 → 56364 [ACK] Seq=166 Ack=18 Win=512 Len=0 TSval=173762949…
   71 506.338421137 127.0.0.1  127.0.1.1   TCP       70 56364 → 10918 [PSH, ACK] Seq=18 Ack=166 Win=512 Len=2 TSval=2501…
   72 506.338453363 127.0.1.1  127.0.0.1   TCP       68 10918 → 56364 [ACK] Seq=166 Ack=20 Win=512 Len=0 TSval=173762950…

› Frame 65: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface any, id 0
› Linux cooked capture
› Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.1.1
› Transmission Control Protocol, Src Port: 56364, Dst Port: 10918, Seq: 13, Ack: 165, Len: 3
⌄ Data (3 bytes)
    Data: 524d44
    [Length: 3]

0000  00 00 03 04 00 06 00 00  00 00 00 00 64 61 08 00   ········ ····da··
0010  45 00 00 37 af 99 40 00  40 06 8c 25 7f 00 00 01   E··7··@· @··%····
0020  7f 00 01 01 dc 2c 2a a6  e5 f9 f3 98 89 b9 f6 e3   ·····,*· ········
0030  80 18 02 00 ff 2b 00 00  01 01 08 0a 95 1e 52 45   ·····+·· ······RE
0040  67 91 a9 09 52 4d 44                               g···RMD

Length (data.len)          Packets: 100 · Displayed: 25 (25.0%)          Profile: Default
```

Fig. 14: Remove directory command


*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp

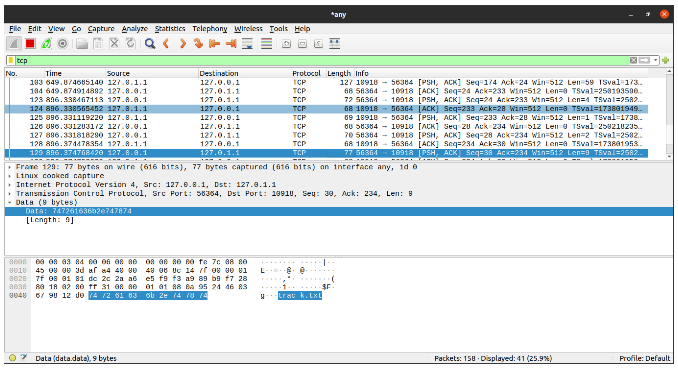
```
No.   Time          Source     Destination Protocol Length Info
   64 476.079990359 127.0.0.1  127.0.1.1   TCP       68 56364 → 10918 [ACK] Seq=13 Ack=165 Win=512 Len=0 TSval=250176210…
   65 506.296440395 127.0.0.1  127.0.1.1   TCP       71 56364 → 10918 [PSH, ACK] Seq=13 Ack=165 Win=512 Len=3 TSval=2501…
   66 506.296539521 127.0.1.1  127.0.0.1   TCP       68 10918 → 56364 [ACK] Seq=165 Ack=16 Win=512 Len=0 TSval=173762945…
   67 506.297443580 127.0.1.1  127.0.0.1   TCP       69 10918 → 56364 [PSH, ACK] Seq=165 Ack=16 Win=512 Len=1 TSval=1737…
   68 506.297488506 127.0.0.1  127.0.1.1   TCP       68 56364 → 10918 [ACK] Seq=16 Ack=166 Win=512 Len=0 TSval=250179232…
   69 506.297764321 127.0.0.1  127.0.1.1   TCP       70 56364 → 10918 [PSH, ACK] Seq=16 Ack=166 Win=512 Len=2 TSval=2501…
   70 506.338228985 127.0.1.1  127.0.0.1   TCP       68 10918 → 56364 [ACK] Seq=166 Ack=18 Win=512 Len=0 TSval=173762949…
   71 506.338421137 127.0.0.1  127.0.1.1   TCP       70 56364 → 10918 [PSH, ACK] Seq=18 Ack=166 Win=512 Len=2 TSval=2501…
   72 506.338453363 127.0.1.1  127.0.0.1   TCP       68 10918 → 56364 [ACK] Seq=166 Ack=20 Win=512 Len=0 TSval=173762950…

› Frame 71: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface any, id 0
› Linux cooked capture
› Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.1.1
› Transmission Control Protocol, Src Port: 56364, Dst Port: 10918, Seq: 18, Ack: 166, Len: 2
⌄ Data (2 bytes)
    Data: 4158
    [Length: 2]

0000  00 00 03 04 00 06 00 00  00 00 00 00 00 00 08 00   ········ ········
0010  45 00 00 36 af 9c 40 00  40 06 8c 23 7f 00 00 01   E··6··@· @··#····
0020  7f 00 01 01 dc 2c 2a a6  e5 f9 f3 9d 89 b9 f6 e4   ·····,*· ········
0030  80 18 02 00 ff 2a 00 00  01 01 08 0a 95 1e 52 6f   ·····*·· ······Ro
0040  67 92 1f 3b 41 58                                  g··;AX

Length (data.len)          Packets: 84 · Displayed: 25 (29.8%)          Profile: Default
```

Fig. 15: RMD response code


*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp

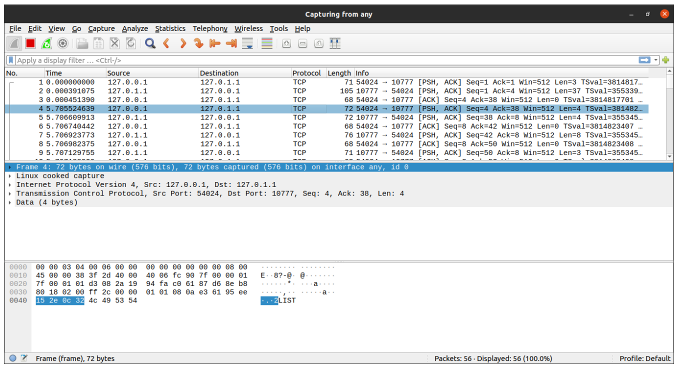
```
No.   Time          Source     Destination Protocol Length Info
   74 506.338697165 127.0.0.1  127.0.1.1   TCP       68 56364 → 10918 [ACK] Seq=20 Ack=170 Win=512 Len=0 TSval=250179236…
   75 506.339258309 127.0.1.1  127.0.0.1   TCP       72 10918 → 56364 [PSH, ACK] Seq=170 Ack=20 Win=512 Len=4 TSval=1737…
   76 506.339299422 127.0.0.1  127.0.1.1   TCP       68 56364 → 10918 [ACK] Seq=20 Ack=174 Win=512 Len=0 TSval=250179236…
  101 649.874019058 127.0.0.1  127.0.1.1   TCP       72 56364 → 10918 [PSH, ACK] Seq=20 Ack=174 Win=512 Len=4 TSval=2501…
  102 649.874211910 127.0.1.1  127.0.0.1   TCP       68 10918 → 56364 [ACK] Seq=174 Ack=24 Win=512 Len=0 TSval=173777303…
  103 649.874665140 127.0.1.1  127.0.0.1   TCP      127 10918 → 56364 [PSH, ACK] Seq=174 Ack=24 Win=512 Len=59 TSval=173…
  104 649.874914892 127.0.0.1  127.0.1.1   TCP       68 56364 → 10918 [ACK] Seq=24 Ack=233 Win=512 Len=0 TSval=250193590…
  123 896.330467113 127.0.0.1  127.0.1.1   TCP       72 56364 → 10918 [PSH, ACK] Seq=24 Ack=233 Win=512 Len=4 TSval=2502…
  124 896.330565452 127.0.1.1  127.0.0.1   TCP       68 10918 → 56364 [ACK] Seq=233 Ack=28 Win=512 Len=0 TSval=173801949…

› Frame 123: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface any, id 0
› Linux cooked capture
› Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.1.1
› Transmission Control Protocol, Src Port: 56364, Dst Port: 10918, Seq: 24, Ack: 233, Len: 4
⌄ Data (4 bytes)
    Data: 44454c45
    [Length: 4]

0000  00 00 03 04 00 06 00 00  00 00 00 00 00 00 08 00   ········ ········
0010  45 00 00 38 af a1 40 00  40 06 8c 1c 7f 00 00 01   E··8··@· @·······
0020  7f 00 01 01 dc 2c 2a a6  e5 f9 f3 a3 89 b9 f7 27   ·····,*· ·······'
0030  80 18 02 00 ff 2c 00 00  01 01 08 0a 95 24 45 d7   ·····,·· ·····$E·
0040  67 94 4f ec 44 45 4c 45                            g·O·DELE

Data (data.data), 4 bytes          Packets: 150 · Displayed: 41 (27.3%)          Profile: Default
```

Fig. 16: SYST


*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp

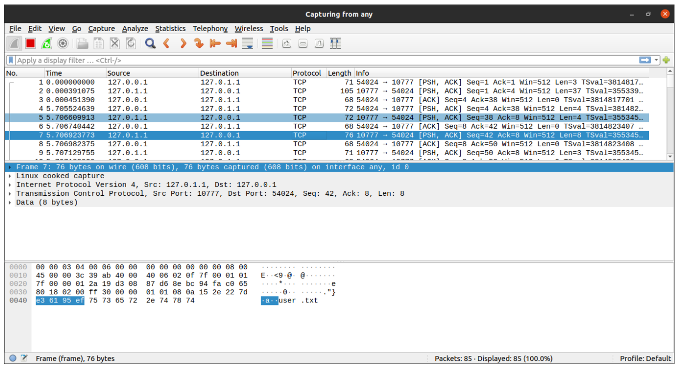
```
No.   Time          Source     Destination Protocol Length Info
  103 649.874665140 127.0.1.1  127.0.0.1   TCP      127 10918 → 56364 [PSH, ACK] Seq=174 Ack=24 Win=512 Len=59 TSval=173…
  104 649.874914892 127.0.0.1  127.0.1.1   TCP       68 56364 → 10918 [ACK] Seq=24 Ack=233 Win=512 Len=0 TSval=250193590…
  123 896.330467113 127.0.0.1  127.0.1.1   TCP       72 56364 → 10918 [PSH, ACK] Seq=24 Ack=233 Win=512 Len=4 TSval=2502…
  124 896.330565452 127.0.1.1  127.0.0.1   TCP       68 10918 → 56364 [ACK] Seq=233 Ack=28 Win=512 Len=0 TSval=173801949…
  125 896.331119220 127.0.1.1  127.0.0.1   TCP       69 10918 → 56364 [PSH, ACK] Seq=233 Ack=28 Win=512 Len=1 TSval=1738…
  126 896.331283172 127.0.0.1  127.0.1.1   TCP       68 56364 → 10918 [ACK] Seq=28 Ack=234 Win=512 Len=0 TSval=250218235…
  127 896.331818290 127.0.0.1  127.0.1.1   TCP       70 56364 → 10918 [PSH, ACK] Seq=28 Ack=234 Win=512 Len=2 TSval=2502…
  128 896.374478354 127.0.1.1  127.0.0.1   TCP       68 10918 → 56364 [ACK] Seq=234 Ack=30 Win=512 Len=0 TSval=173801953…
  129 896.374768420 127.0.0.1  127.0.1.1   TCP       77 56364 → 10918 [PSH, ACK] Seq=30 Ack=234 Win=512 Len=9 TSval=2502…

› Frame 129: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface any, id 0
› Linux cooked capture
› Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.1.1
› Transmission Control Protocol, Src Port: 56364, Dst Port: 10918, Seq: 30, Ack: 234, Len: 9
⌄ Data (9 bytes)
    Data: 747261636b2e747874
    [Length: 9]

0000  00 00 03 04 00 06 00 00  00 00 00 00 fe 7c 08 00   ········ ·····|··
0010  45 00 00 3d af a4 40 00  40 06 8c 14 7f 00 00 01   E··=··@· @·······
0020  7f 00 01 01 dc 2c 2a a6  e5 f9 f3 a9 89 b9 f7 28   ·····,*· ·······(
0030  80 18 02 00 ff 31 00 00  01 01 08 0a 95 24 46 03   ·····1·· ·····$F·
0040  67 98 12 d0 74 72 61 63  6b 2e 74 78 74            g···trac k.txt

Data (data.data), 9 bytes          Packets: 158 · Displayed: 41 (25.9%)          Profile: Default
```

Fig. 17: File to be deleted


Capturing from any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.   Time         Source     Destination Protocol Length Info
    1 0.000000000  127.0.0.1  127.0.1.1   TCP       71 54024 → 10777 [PSH, ACK] Seq=1 Ack=1 Win=512 Len=3 TSval=3814817…
    2 0.000391075  127.0.1.1  127.0.0.1   TCP      105 10777 → 54024 [PSH, ACK] Seq=1 Ack=4 Win=512 Len=37 TSval=355339…
    3 0.000451390  127.0.0.1  127.0.1.1   TCP       68 54024 → 10777 [ACK] Seq=4 Ack=38 Win=512 Len=0 TSval=3814817701 …
    4 5.705524639  127.0.0.1  127.0.1.1   TCP       72 54024 → 10777 [PSH, ACK] Seq=4 Ack=38 Win=512 Len=4 TSval=381482…
    5 5.706609913  127.0.1.1  127.0.0.1   TCP       72 10777 → 54024 [PSH, ACK] Seq=38 Ack=8 Win=512 Len=4 TSval=355345…
    6 5.706740442  127.0.0.1  127.0.1.1   TCP       68 54024 → 10777 [ACK] Seq=8 Ack=42 Win=512 Len=0 TSval=3814823407 …
    7 5.706923773  127.0.1.1  127.0.0.1   TCP       76 10777 → 54024 [PSH, ACK] Seq=42 Ack=8 Win=512 Len=8 TSval=355345…
    8 5.706982375  127.0.0.1  127.0.1.1   TCP       68 54024 → 10777 [ACK] Seq=8 Ack=50 Win=512 Len=0 TSval=3814823408 …
    9 5.707129755  127.0.1.1  127.0.0.1   TCP       71 10777 → 54024 [PSH, ACK] Seq=50 Ack=8 Win=512 Len=3 TSval=355345…

› Frame 4: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface any, id 0
› Linux cooked capture
› Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.1.1
› Transmission Control Protocol, Src Port: 54024, Dst Port: 10777, Seq: 4, Ack: 38, Len: 4
› Data (4 bytes)

0000  00 00 03 04 00 06 00 00  00 00 00 00 00 00 08 00   ········ ········
0010  45 00 00 38 3f 2d 40 00  40 06 fc 90 7f 00 00 01   E··8?-@· @·······
0020  7f 00 01 01 d3 08 2a 19  94 fa c0 61 87 d6 8e b8   ······*· ···a····
0030  80 18 02 00 ff 2c 00 00  01 01 08 0a e3 61 95 ee   ·····,·· ·····a··
0040  15 2e 0c 32 4c 49 53 54                            ·.·2LIST

Frame (frame), 72 bytes          Packets: 56 · Displayed: 56 (100.0%)          Profile: Default
```

Fig. 18: LIST operation



```
No.  Time         Source      Destination  Protocol  Length  Info
   1 0.000000000  127.0.0.1   127.0.1.1    TCP        71 54024 → 10777 [PSH, ACK] Seq=1 Ack=1 Win=512 Len=3 TSval=3814817…
   2 0.000391075  127.0.1.1   127.0.0.1    TCP       105 10777 → 54024 [PSH, ACK] Seq=1 Ack=4 Win=512 Len=37 TSval=355339…
   3 0.000451390  127.0.0.1   127.0.1.1    TCP        68 54024 → 10777 [ACK] Seq=4 Ack=38 Win=512 Len=0 TSval=3814817701 …
   4 5.705524639  127.0.0.1   127.0.1.1    TCP        72 54024 → 10777 [PSH, ACK] Seq=4 Ack=38 Win=512 Len=4 TSval=381482…
   5 5.706609913  127.0.1.1   127.0.0.1    TCP        72 10777 → 54024 [PSH, ACK] Seq=38 Ack=8 Win=512 Len=4 TSval=355345…
   6 5.706740442  127.0.0.1   127.0.1.1    TCP        68 54024 → 10777 [ACK] Seq=8 Ack=42 Win=512 Len=0 TSval=3814823407 …
   7 5.706923773  127.0.1.1   127.0.0.1    TCP        76 10777 → 54024 [PSH, ACK] Seq=42 Ack=8 Win=512 Len=8 TSval=355345…
   8 5.706982375  127.0.0.1   127.0.1.1    TCP        68 54024 → 10777 [ACK] Seq=8 Ack=50 Win=512 Len=0 TSval=3814823408 …
   9 5.707129755  127.0.1.1   127.0.0.1    TCP        71 10777 → 54024 [PSH, ACK] Seq=50 Ack=8 Win=512 Len=3 TSval=355345…

› Frame 7: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface any, id 0
› Linux cooked capture
› Internet Protocol Version 4, Src: 127.0.1.1, Dst: 127.0.0.1
› Transmission Control Protocol, Src Port: 10777, Dst Port: 54024, Seq: 42, Ack: 8, Len: 8
› Data (8 bytes)

0000  00 00 03 04 00 06 00 00  00 00 00 00 00 00 08 00   ········ ········
0010  45 00 00 3c 39 ab 40 00  40 06 02 0f 7f 00 01 01   E··<9·@· @·······
0020  7f 00 00 01 2a 19 d3 08  87 d6 8e bc 94 fa c0 65   ····*··· ·······e
0030  80 18 02 00 ff 30 00 00  01 01 08 0a 15 2e 22 7d   ·····0·· ·····."}
0040  e3 61 95 ef 75 73 65 72  2e 74 78 74               ·a··user .txt

Frame (frame), 76 bytes          Packets: 85 · Displayed: 85 (100.0%)          Profile: Default
```

Fig. 19: LIST server response