Q3.

Flask to build a web application
uWSGI as a production server
requests to exercise your server

To install all of these dependencies, use pip:

$ pip install flask uwsgi requests

In a file called server.py, you create a Flask application:

```
# server.py
from flask import Flask

SECRET_MESSAGE = "fluffy tail"
app = Flask(__name__)

@app.route("/")
def get_secret_message():
    return SECRET_MESSAGE
```

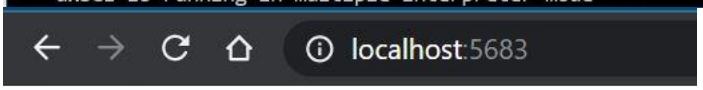
You write a script called client.py that will help them get the secret message:

```
# client.py
import os
import requests

def get_secret_message():
    url = os.environ["SECRET_URL"]
    response = requests.get(url)
    print(f"The secret message is: {response.text}")

if __name__ == "__main__":
    get_secret_message()
```

This code will print out the secret message as long as they have the SECRET_URL environment variable set. In this case, the SECRET_URL is 127.0.0.1:5683.

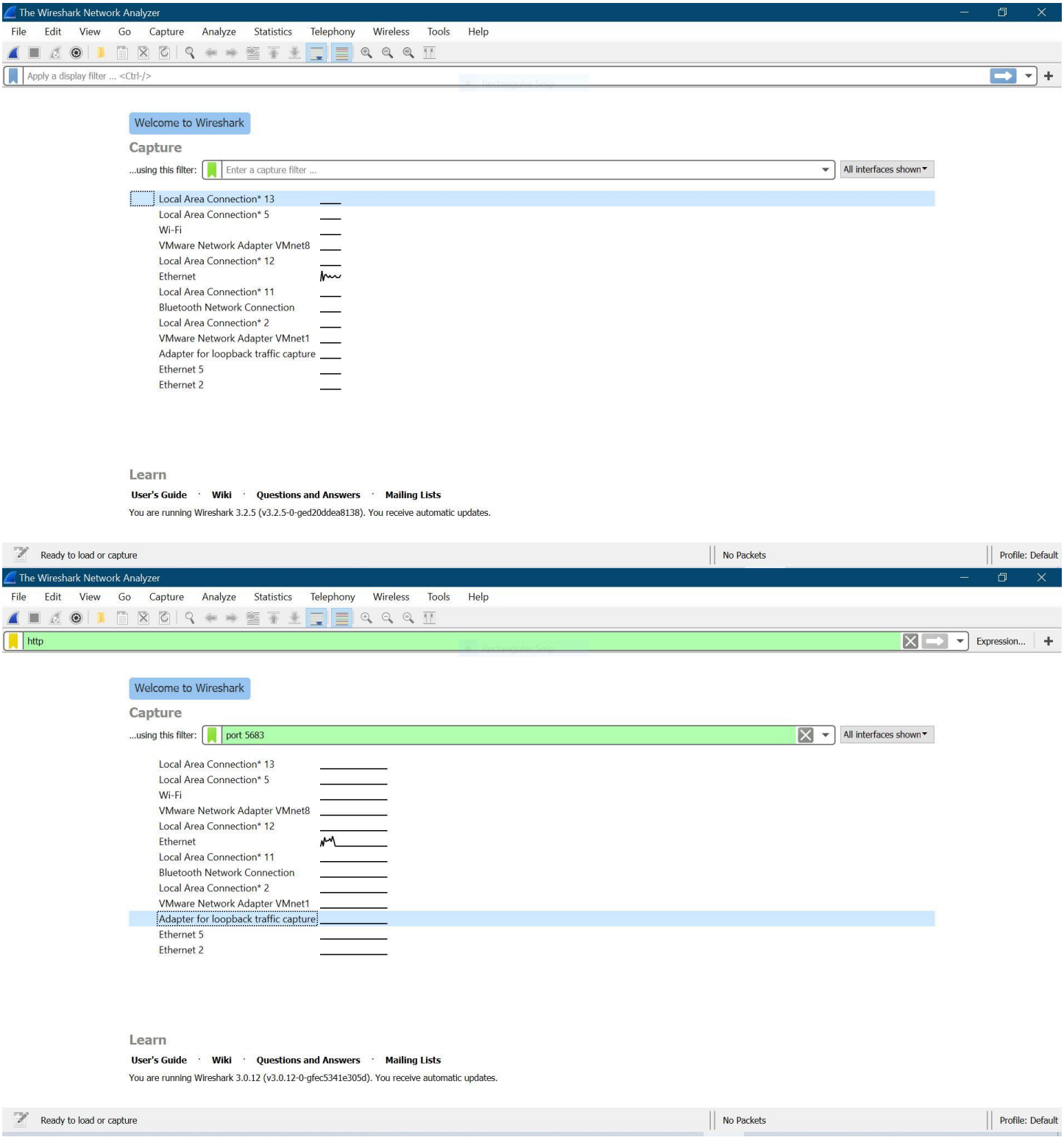
Deploy your application on your secret server and run it:

```
chiu@LAPTOP-EON0OEAL:~$ clear
chiu@LAPTOP-EON0OEAL:~$ uwsgi --http-socket 127.0.0.1:5683 --mount /=server:app
*** Starting uWSGI 2.0.19.1 (64bit) on [Sun Jul 26 22:39:23 2020] ***
compiled with version: 7.5.0 on 21 July 2020 04:34:20
os: Linux-4.4.0-18362-Microsoft #836-Microsoft Mon May 05 16:04:00 PST 2020
nodename: LAPTOP-EON0OEAL
machine: x86_64
clock source: unix
detected number of CPU cores: 4
current working directory: /home/chiu
detected binary path: /home/chiu/.local/bin/uwsgi
!!! no internal routing support, rebuild with pcre support !!!
*** WARNING: you are running uWSGI without its master process manager ***
your processes number limit is 7823
your memory page size is 4096 bytes
detected max file descriptor number: 1024
lock engine: pthread robust mutexes
thunder lock: disabled (you can enable it with --thunder-lock)
TCP_DEFER_ACCEPT setsockopt(): Protocol not available [core/socket.c line 744]
uwsgi socket 0 bound to TCP address 127.0.0.1:5683 fd 3
Python version: 2.7.17 (default, Apr 15 2020, 17:20:14)  [GCC 7.5.0]
*** Python threads support is disabled. You can enable it with --enable-threads ***
Python main interpreter initialized at 0x7fffe90eadc0
your server socket listen backlog is limited to 100 connections
your mercy for graceful operations on workers is 60 seconds
mapped 72920 bytes (71 KB) for 1 cores
*** Operational MODE: single process ***
mounting server:app on /
WSGI app 0 (mountpoint='/') ready in 0 seconds on interpreter 0x7fffe90eadc0 pid: 68 (default app)
*** uWSGI is running in multiple interpreter mode ***
```

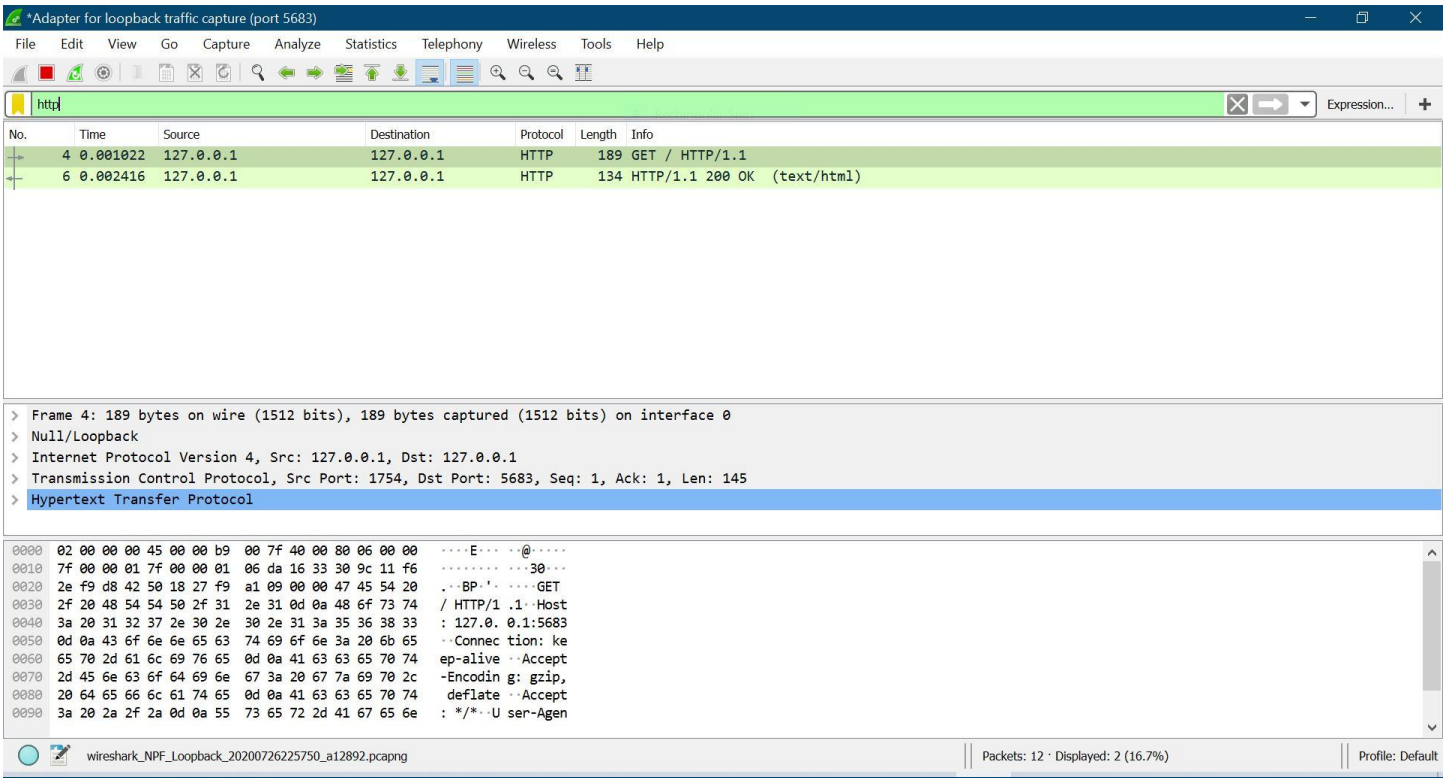


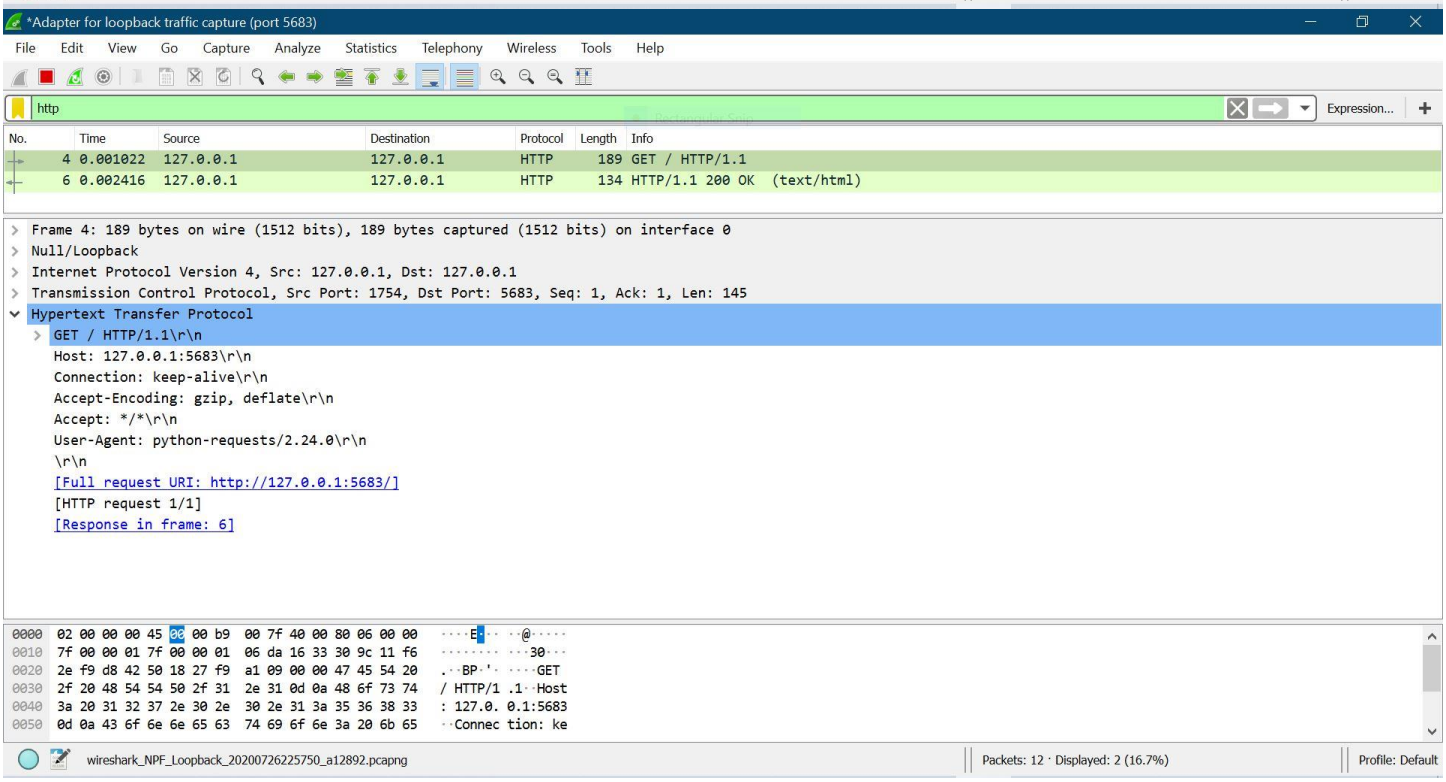

fluffy tail

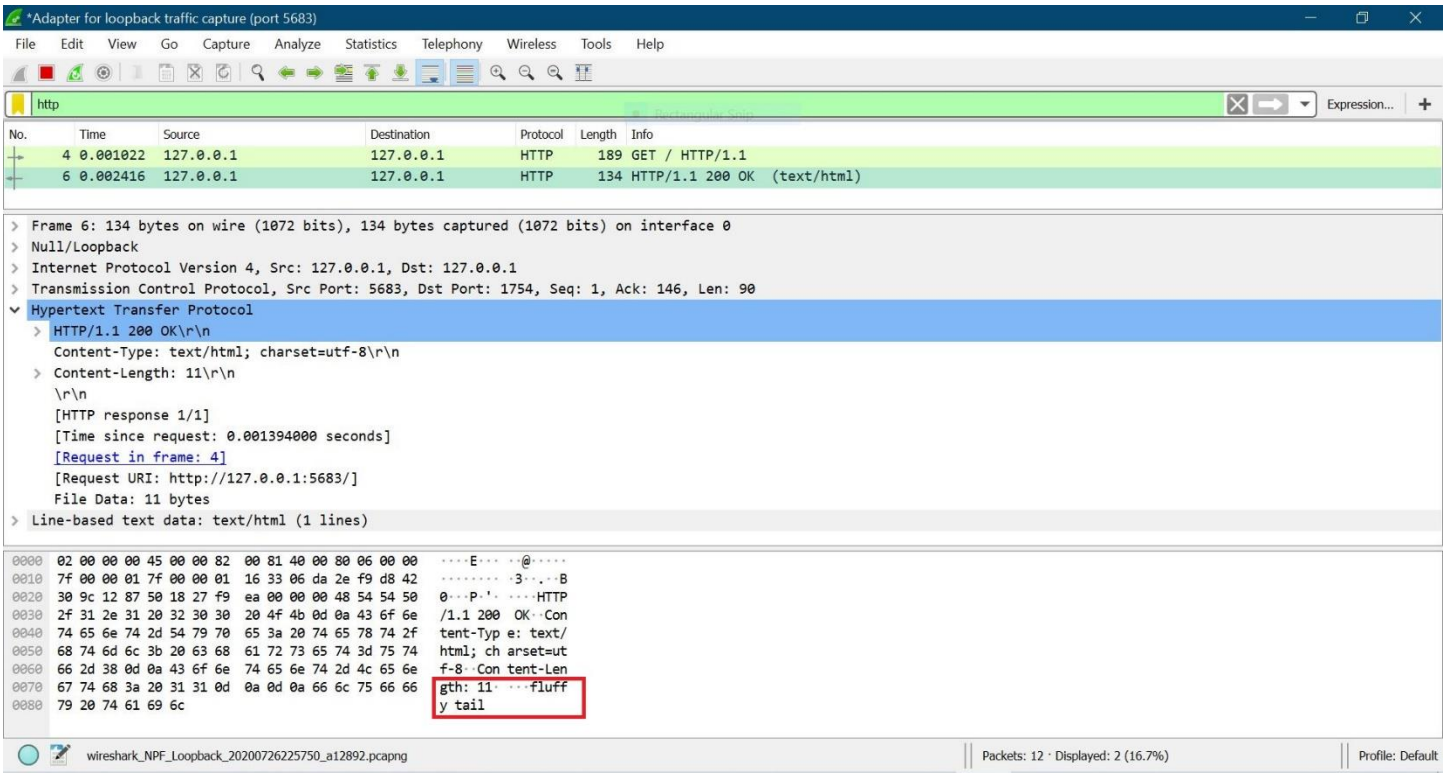
Type port 5683 in the capture filter and http in the display filter on Wireshark:




When you expand the Hypertext Transfer Protocol layer, you can see all the information that makes up an HTTP Request:


*Adapter for loopback traffic capture (port 5683)
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
Expression...
No. Time Source Destination Protocol Length Info
4 0.001022 127.0.0.1 127.0.0.1 HTTP 189 GET / HTTP/1.1
6 0.002416 127.0.0.1 127.0.0.1 HTTP 134 HTTP/1.1 200 OK (text/html)
Frame 4: 189 bytes on wire (1512 bits), 189 bytes captured (1512 bits) on interface 0
Null/Loopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 1754, Dst Port: 5683, Seq: 1, Ack: 1, Len: 145
Hypertext Transfer Protocol
0000 02 00 00 00 45 00 00 b9 00 7f 40 00 80 06 00 00 ····E··· ··@·····
0010 7f 00 00 01 7f 00 00 01 06 da 16 33 30 9c 11 f6 ········ ···30···
0020 2e f9 d8 42 50 18 27 f9 a1 09 00 00 47 45 54 20 .··BP·'· ····GET
0030 2f 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 / HTTP/1 .1··Host
0040 3a 20 31 32 37 2e 30 2e 30 2e 31 3a 35 36 38 33 : 127.0. 0.1:5683
0050 0d 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 ··Connec tion: ke
0060 65 70 2d 61 6c 69 76 65 0d 0a 41 63 63 65 70 74 ep-alive ··Accept
0070 2d 45 6e 63 6f 64 69 6e 67 3a 20 67 7a 69 70 2c -Encodin g: gzip,
0080 20 64 65 66 6c 61 74 65 0d 0a 41 63 63 65 70 74 deflate ··Accept
0090 3a 20 2a 2f 2a 0d 0a 55 73 65 72 2d 41 67 65 6e : */*··U ser-Agen
wireshark_NPF_Loopback_20200726225750_a12892.pcapng
Packets: 12 · Displayed: 2 (16.7%)
Profile: Default


*Adapter for loopback traffic capture (port 5683)
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
Expression...
No. Time Source Destination Protocol Length Info
4 0.001022 127.0.0.1 127.0.0.1 HTTP 189 GET / HTTP/1.1
6 0.002416 127.0.0.1 127.0.0.1 HTTP 134 HTTP/1.1 200 OK (text/html)
Frame 4: 189 bytes on wire (1512 bits), 189 bytes captured (1512 bits) on interface 0
Null/Loopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 1754, Dst Port: 5683, Seq: 1, Ack: 1, Len: 145
Hypertext Transfer Protocol
GET / HTTP/1.1\r\n
Host: 127.0.0.1:5683\r\n
Connection: keep-alive\r\n
Accept-Encoding: gzip, deflate\r\n
Accept: */*\r\n
User-Agent: python-requests/2.24.0\r\n
\r\n
[Full request URI: http://127.0.0.1:5683/]
[HTTP request 1/1]
[Response in frame: 6]
0000 02 00 00 00 45 00 00 b9 00 7f 40 00 80 06 00 00 ····E··· ··@·····
0010 7f 00 00 01 7f 00 00 01 06 da 16 33 30 9c 11 f6 ········ ···30···
0020 2e f9 d8 42 50 18 27 f9 a1 09 00 00 47 45 54 20 .··BP·'· ····GET
0030 2f 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 / HTTP/1 .1··Host
0040 3a 20 31 32 37 2e 30 2e 30 2e 31 3a 35 36 38 33 : 127.0. 0.1:5683
0050 0d 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 ··Connec tion: ke
wireshark_NPF_Loopback_20200726225750_a12892.pcapng
Packets: 12 · Displayed: 2 (16.7%)
Profile: Default




If you look carefully at the hex dump, then you'll see the secret message in plain text!