ws-dcode

Decode websocket with protobuf in wireshark

Prerequisites

The below are the required in your system:

- `Firefox or Chrome`
- `Wireshark`
- `LuaRocks` (Lua package manager)
- `lua-cjson & lua-protobuf` (Lua modules)

Install Dependencies

Install all the required prerequisites

step1: Install Wireshark (GUI + CLI)

- `sudo apt update && sudo apt upgrade -y`
- `sudo apt install -y wireshark`
- `sudo usermod -aG wireshark $USER` (Optional)Allow non-root users to capture packets

step2: Install Firefox

- `sudo apt install -y firefox`

step3: Install Lua and LuaRocks (Lua package manager)

Install the dependency modules manually

step3.1: Install cjson

```
sudo apt install -y build-essential lua5.2 liblua5.2-dev make gcc
mkdir ~/protolib
cd ~/protolib
wget https://github.com/mpx/lua-cjson/archive/refs/tags/2.1.0.tar.gz
tar -xvzf 2.1.0.tar.gz
cd lua-cjson-2.1.0
make LUA_INCLUDE_DIR=/usr/include/lua5.2 LUA_VERSION=5.2
sudo cp cjson.so /usr/lib/x86_64-linux-gnu/lua/5.2/
```

step3.2: Install pb (lua-protobuf)

```
cd ~/protolib
git clone https://github.com/starwing/lua-protobuf.git
cd lua-protobuf
cmake -DLUA_INCLUDE_DIR=/usr/include/lua5.2 -DCMAKE_C_FLAGS="-I/usr/include/lua5.2"
make
sudo cp ./pb.so /usr/lib/x86_64-linux-gnu/lua/5.2/
```

step4: Get Lua websocket decoder plugin

```
cd ~/protolib
git clone https://github.com/Induzio/ws-dcode.git
```

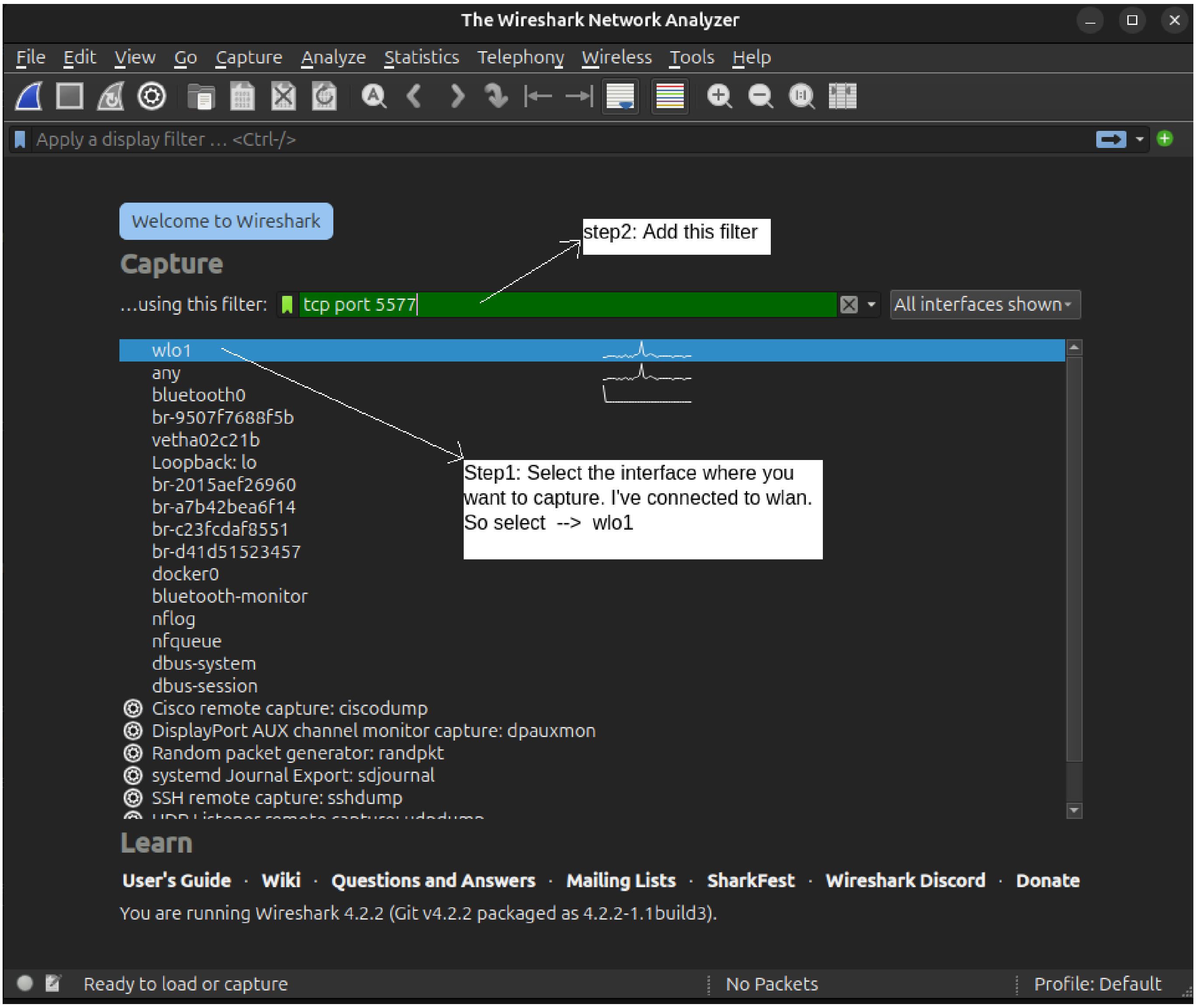
Wireshark setup

step1: open a New terminal

Run wireshark with below cmd:

```
sudo wireshark
```

Now, select wlo1 and in filter



The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Welcome to Wireshark
Capture
All interfaces shown
wlo1
any
bluetooth0
br-9507f7688f5b
vetha02c21b
Loopback: lo
br-2015aef26960
br-a7b42bea6f14
br-c23fcdaf8551
br-d41d51523457
docker0
bluetooth-monitor
nflog
nfqueue
dbus-system
dbus-session
Cisco remote capture: ciscodump
DisplayPort AUX channel monitor capture: dpauxmon
Random packet generator: randpkt
systemd Journal Export: sdjournal
SSH remote capture: sshdump
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists · SharkFest · Wireshark Discord · Donate
Ready to load or capture
No Packets
Profile: Default

…using this filter: tcp port 5577

step2: Add this filter

Step1: Select the interface where you want to capture. I've connected to wlan. So select --> wlo1

You are running Wireshark 4.2.2 (Git v4.2.2 packaged as 4.2.2-1.1build3).

step2: open another terminal

The below commands will capture the ssl keys for decryption

warning: Before running the below cmds please close all the firefox windows opened

```
export SSLKEYLOGFILE=$HOME/protolib/sslkeys.log
firefox &
```

Then visit: https://ysafe.io

once verify if the sslkeys.log is present after visiting by running below cmd in any terminal....

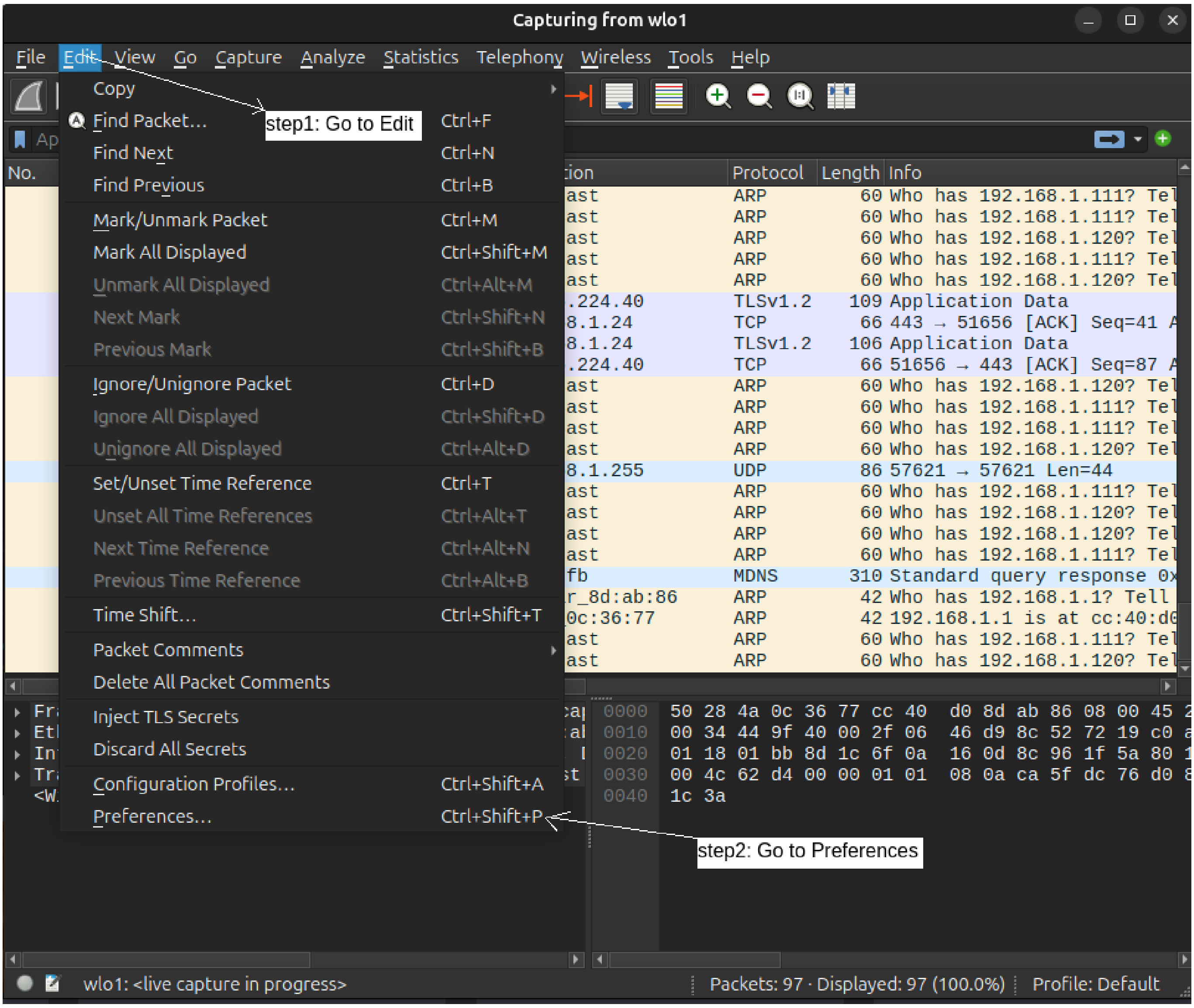
```
ls $HOME/protolib/sslkeys.log
```

step3: Add sslkeys.log in wireshark

Now you have to add the `sslkeys.log` in `Edit > Preferences > Protocols > TLS`

Follow below steps to add:

step3.1: Go to Preferences


Capturing from wlo1
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Wireless
Tools
Help
Copy
Find Packet… Ctrl+F
Find Next Ctrl+N
Find Previous Ctrl+B
Mark/Unmark Packet Ctrl+M
Mark All Displayed Ctrl+Shift+M
Unmark All Displayed Ctrl+Alt+M
Next Mark Ctrl+Shift+N
Previous Mark Ctrl+Shift+B
Ignore/Unignore Packet Ctrl+D
Ignore All Displayed Ctrl+Shift+D
Unignore All Displayed Ctrl+Alt+D
Set/Unset Time Reference Ctrl+T
Unset All Time References Ctrl+Alt+T
Next Time Reference Ctrl+Alt+N
Previous Time Reference Ctrl+Alt+B
Time Shift… Ctrl+Shift+T
Packet Comments
Delete All Packet Comments
Inject TLS Secrets
Discard All Secrets
Configuration Profiles… Ctrl+Shift+A
Preferences… Ctrl+Shift+P
No.
Ap

step1: Go to Edit

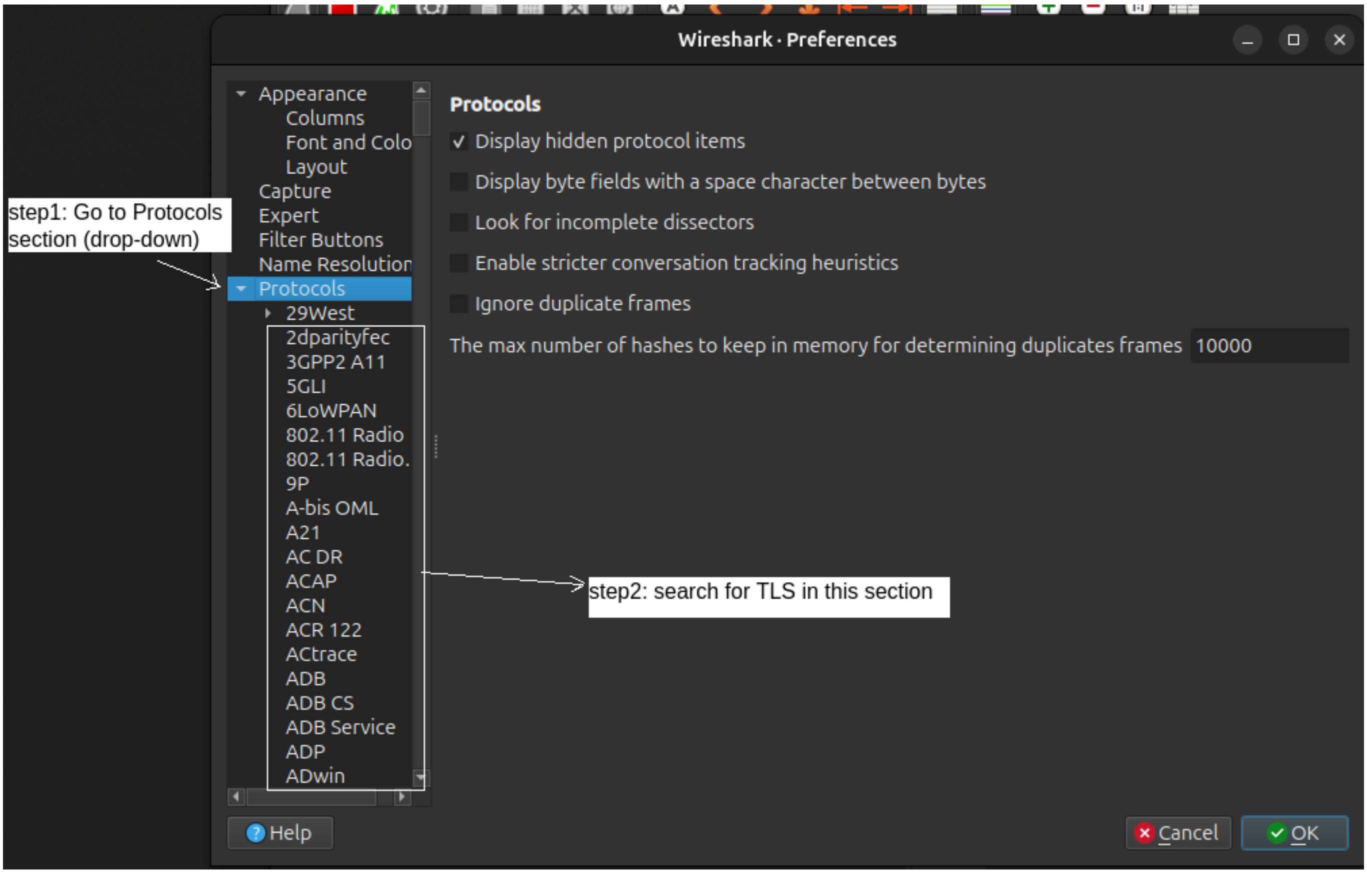
step2: Go to Preferences

```
ion         Protocol Length Info
ast         ARP          60 Who has 192.168.1.111? Tel
ast         ARP          60 Who has 192.168.1.111? Tel
ast         ARP          60 Who has 192.168.1.120? Tel
ast         ARP          60 Who has 192.168.1.111? Tel
ast         ARP          60 Who has 192.168.1.120? Tel
.224.40     TLSv1.2     109 Application Data
8.1.24      TCP          66 443 → 51656 [ACK] Seq=41 A
8.1.24      TLSv1.2     106 Application Data
.224.40     TCP          66 51656 → 443 [ACK] Seq=87 A
ast         ARP          60 Who has 192.168.1.120? Tel
ast         ARP          60 Who has 192.168.1.111? Tel
ast         ARP          60 Who has 192.168.1.111? Tel
ast         ARP          60 Who has 192.168.1.120? Tel
8.1.255     UDP          86 57621 → 57621 Len=44
ast         ARP          60 Who has 192.168.1.111? Tel
ast         ARP          60 Who has 192.168.1.120? Tel
ast         ARP          60 Who has 192.168.1.120? Tel
ast         ARP          60 Who has 192.168.1.111? Tel
fb          MDNS        310 Standard query response 0x
r_8d:ab:86  ARP          42 Who has 192.168.1.1? Tell
0c:36:77    ARP          42 192.168.1.1 is at cc:40:d0
ast         ARP          60 Who has 192.168.1.111? Tel
ast         ARP          60 Who has 192.168.1.120? Tel
```

```
Fr  ca  0000  50 28 4a 0c 36 77 cc 40  d0 8d ab 86 08 00 45 2
Et  al  0010  00 34 44 9f 40 00 2f 06  46 d9 8c 52 72 19 c0 a
In   l  0020  01 18 01 bb 8d 1c 6f 0a  16 0d 8c 96 1f 5a 80 1
Tr  st  0030  00 4c 62 d4 00 00 01 01  08 0a ca 5f dc 76 d0 8
<W.     0040  1c 3a
```

wlo1: <live capture in progress> Packets: 97 · Displayed: 97 (100.0%) Profile: Default

step3.2: Go to protocols



step1: Go to Protocols section (drop-down)

step2: search for TLS in this section

Protocols

☑ Display hidden protocol items

□ Display byte fields with a space character between bytes

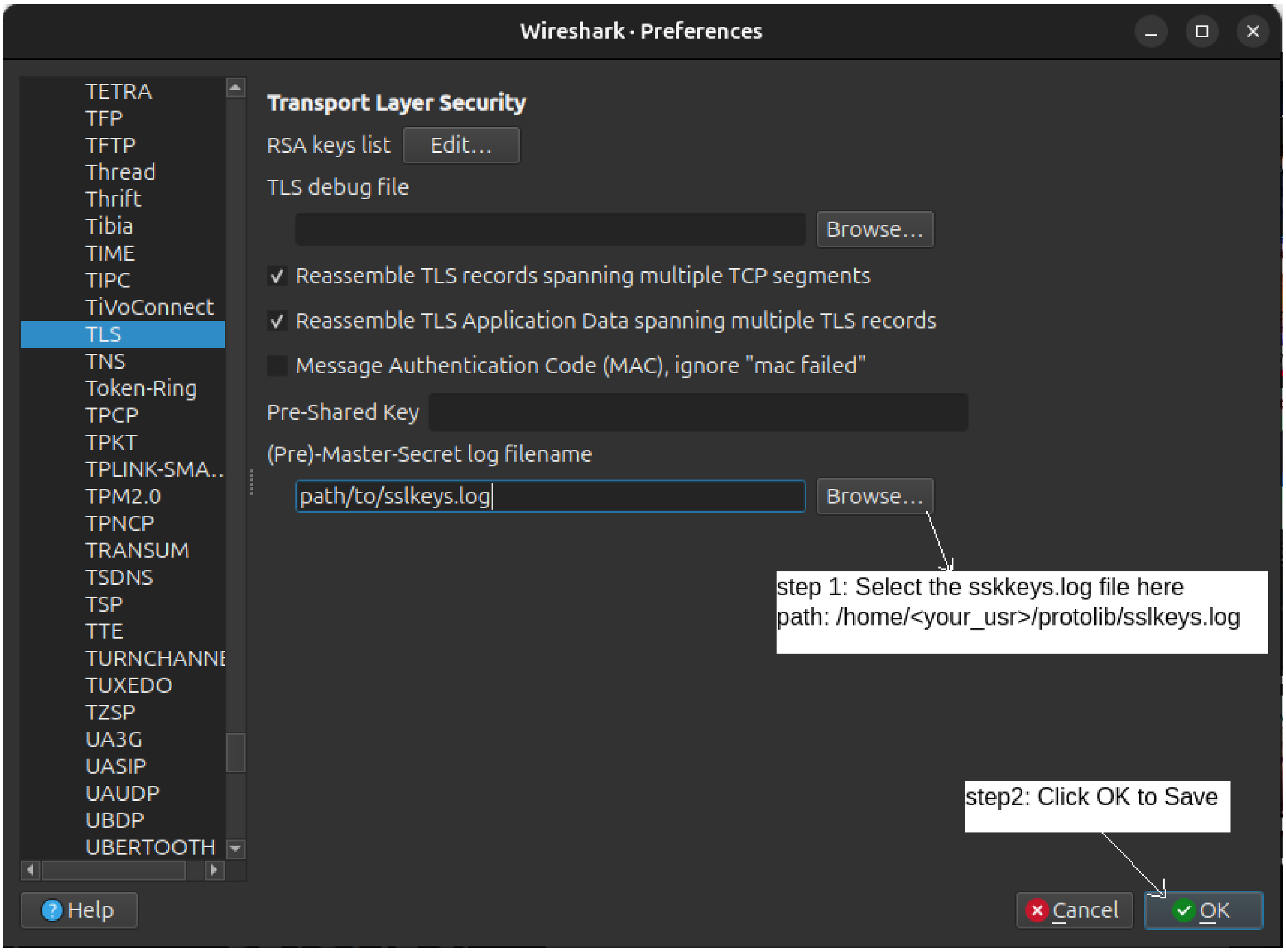
□ Look for incomplete dissectors

□ Enable stricter conversation tracking heuristics

□ Ignore duplicate frames

The max number of hashes to keep in memory for determining duplicates frames 10000

then set (Pre)-Master-Secret log filename:



Transport Layer Security

RSA keys list

TLS debug file

☑ Reassemble TLS records spanning multiple TCP segments

☑ Reassemble TLS Application Data spanning multiple TLS records

□ Message Authentication Code (MAC), ignore "mac failed"

Pre-Shared Key

(Pre)-Master-Secret log filename

path/to/sslkeys.log

step 1: Select the sskkeys.log file here
path: /home/<your_usr>/protolib/sslkeys.log

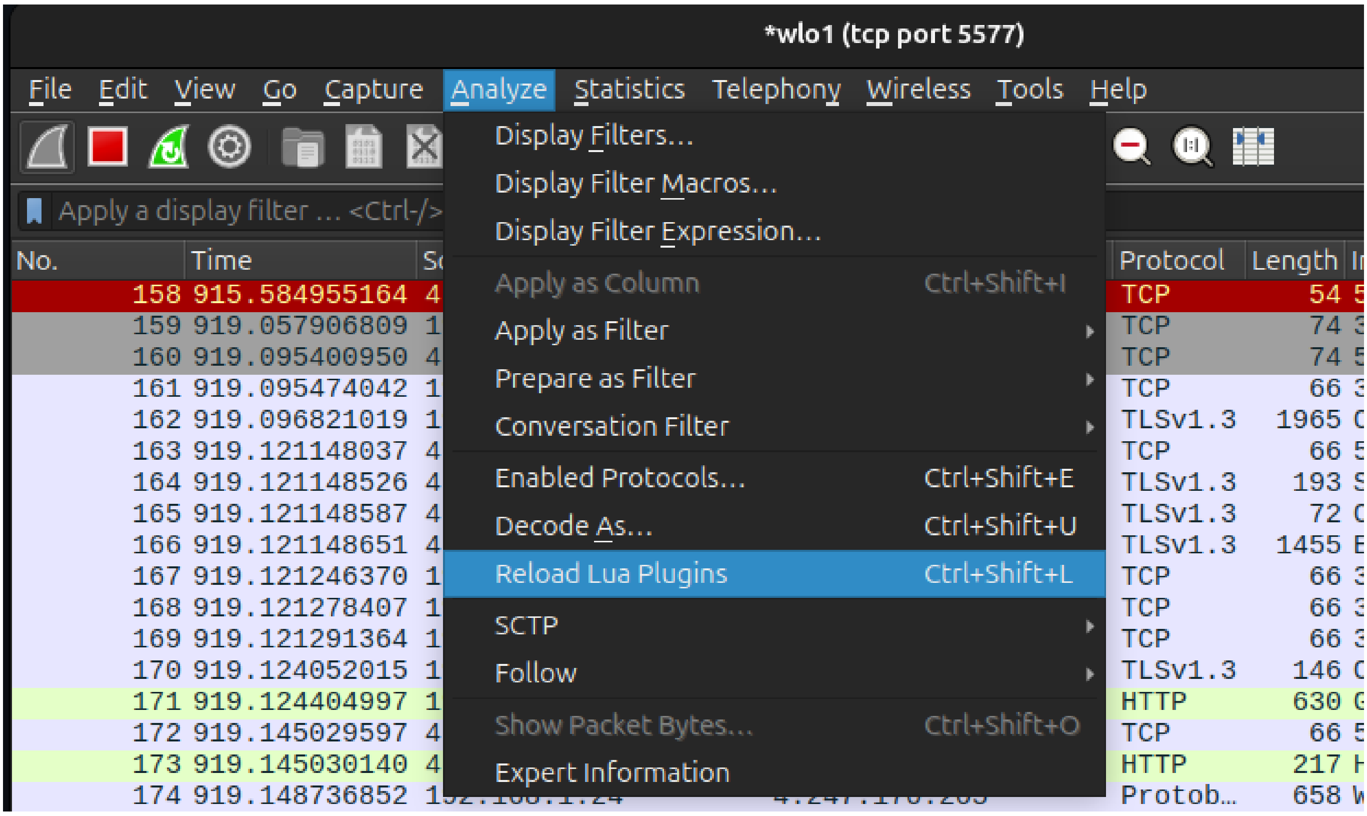
step2: Click OK to Save

step4: Add Protobuf decoding plugin in wireshark

Now run below cmd in any terminal to add the plugin....

```
sudo cp ~/protolib/ws-dcode/proto_dissector.lua /usr/lib/x86_64-linux-gnu/wireshark/plugins
sudo cp ~/protolib/ws-dcode/proto_bundle.desc /usr/lib/x86_64-linux-gnu/wireshark/plugins
```

Now, reload the lua plugin in wireshark:

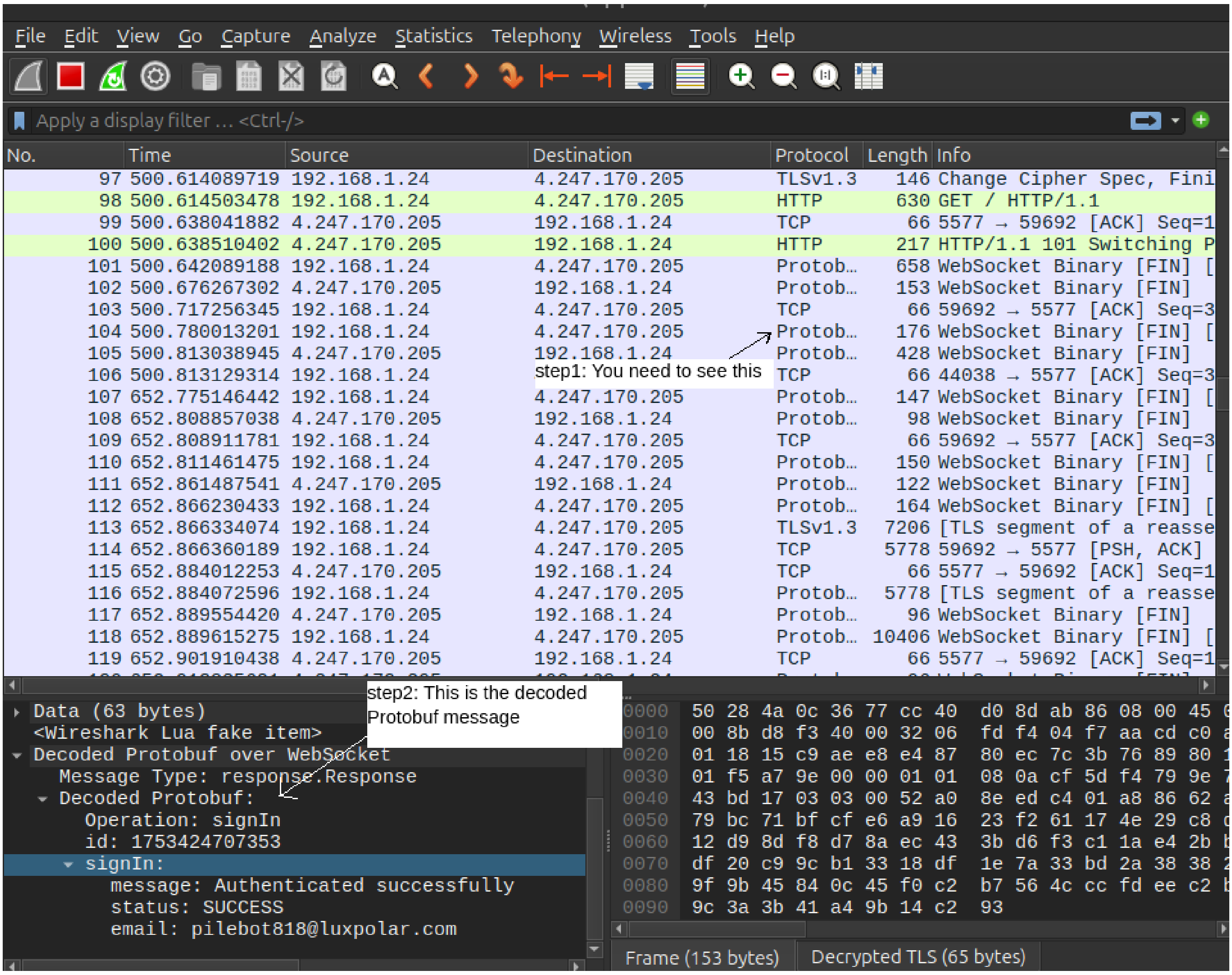


No.	Time	Protocol	Length
158	915.584955164	TCP	54
159	919.057906809	TCP	74
160	919.095400950	TCP	74
161	919.095474042	TCP	66
162	919.096821019	TLSv1.3	1965
163	919.121148037	TCP	66
164	919.121148526	TLSv1.3	193
165	919.121148587	TLSv1.3	72
166	919.121148651	TLSv1.3	1455
167	919.121246370	TCP	66
168	919.121278407	TCP	66
169	919.121291364	TCP	66
170	919.124052015	TLSv1.3	146
171	919.124404997	HTTP	630
172	919.145029597	TCP	66
173	919.145030140	HTTP	217
174	919.148736852	Protob…	658

🥹 **Now your wireshark is setup completely....**

step5: Refresh the firefox to see the decoded protobuf packets

When you refresh again then you'll see the decoded packets in the wireshark as protobuf like below:



```
No.   Time           Source          Destination     Protocol  Length Info
   97 500.614089719 192.168.1.24    4.247.170.205   TLSv1.3     146 Change Cipher Spec, Fini
   98 500.614503478 192.168.1.24    4.247.170.205   HTTP        630 GET / HTTP/1.1
   99 500.638041882 4.247.170.205   192.168.1.24    TCP          66 5577 → 59692 [ACK] Seq=1
  100 500.638510402 4.247.170.205   192.168.1.24    HTTP        217 HTTP/1.1 101 Switching P
  101 500.642089188 192.168.1.24    4.247.170.205   Protob…     658 WebSocket Binary [FIN] [
  102 500.676267302 4.247.170.205   192.168.1.24    Protob…     153 WebSocket Binary [FIN]
  103 500.717256345 192.168.1.24    4.247.170.205   TCP          66 59692 → 5577 [ACK] Seq=3
  104 500.780013201 192.168.1.24    4.247.170.205   Protob…     176 WebSocket Binary [FIN] [
  105 500.813038945 4.247.170.205   192.168.1.24    Protob…     428 WebSocket Binary [FIN]
  106 500.813129314 192.168.1.24                    TCP          66 44038 → 5577 [ACK] Seq=3
  107 652.775146442 192.168.1.24    4.247.170.205   Protob…     147 WebSocket Binary [FIN] [
  108 652.808857038 4.247.170.205   192.168.1.24    Protob…      98 WebSocket Binary [FIN]
  109 652.808911781 192.168.1.24    4.247.170.205   TCP          66 59692 → 5577 [ACK] Seq=3
  110 652.811461475 192.168.1.24    4.247.170.205   Protob…     150 WebSocket Binary [FIN] [
  111 652.861487541 4.247.170.205   192.168.1.24    Protob…     122 WebSocket Binary [FIN]
  112 652.866230433 192.168.1.24    4.247.170.205   Protob…     164 WebSocket Binary [FIN] [
  113 652.866334074 192.168.1.24    4.247.170.205   TLSv1.3    7206 [TLS segment of a reasse
  114 652.866360189 192.168.1.24    4.247.170.205   TCP        5778 59692 → 5577 [PSH, ACK]
  115 652.884012253 4.247.170.205   192.168.1.24    TCP          66 5577 → 59692 [ACK] Seq=1
  116 652.884072596 192.168.1.24    4.247.170.205   Protob…    5778 [TLS segment of a reasse
  117 652.889554420 4.247.170.205   192.168.1.24    Protob…      96 WebSocket Binary [FIN]
  118 652.889615275 192.168.1.24    4.247.170.205   Protob…   10406 WebSocket Binary [FIN] [
  119 652.901910438 4.247.170.205   192.168.1.24    TCP          66 5577 → 59692 [ACK] Seq=1
```

step1: You need to see this

step2: This is the decoded Protobuf message

```
▸ Data (63 bytes)
  <Wireshark Lua fake item>
▾ Decoded Protobuf over WebSocket
    Message Type: response.Response
  ▾ Decoded Protobuf:
      Operation: signIn
      id: 1753424707353
    ▾ signIn:
        message: Authenticated successfully
        status: SUCCESS
        email: pilebot818@luxpolar.com
```

```
0000  50 28 4a 0c 36 77 cc 40  d0 8d ab 86 08 00 45 
0010  00 8b d8 f3 40 00 32 06  fd f4 04 f7 aa cd c0 
0020  01 18 15 c9 ae e8 e4 87  80 ec 7c 3b 76 89 80 
0030  01 f5 a7 9e 00 00 01 01  08 0a cf 5d f4 79 9e 
0040  43 bd 17 03 03 00 52 a0  8e ed c4 01 a8 86 62 
0050  79 bc 71 bf cf e6 a9 16  23 f2 61 17 4e 29 c8 
0060  12 d9 8d f8 d7 8a ec 43  3b d6 f3 c1 1a e4 2b 
0070  df 20 c9 9c b1 33 18 df  1e 7a 33 bd 2a 38 38 
0080  9f 9b 45 84 0c 45 f0 c2  b7 56 4c cc fd ee c2 
0090  9c 3a 3b 41 a4 9b 14 c2  93
```

Hurray 🌟!!! Now your wireshark is setup to decode the packets to protobuf....