#Filters

IP Filters

show packets containing the below ip

```
ip.addr == 192.168.1.1
```

show packets not containing the below ip

```
ip.addr != 192.168.1.1
```

show packets containing both below IPs

```
ip.addr == 192.168.1.1 && ip.addr == 192.168.1.2
```

show http packets

```
http
```

show https packets

```
tcp.port == 443
```

show email packets

```
smtp
```

show DNS packets

```
dns
```

filter DNS query types records

DNS A record

```
dns.qry.type == 1
```

DNS TXT record

```
dns.qry.type == 16
```

Filtering for http methods

[GET]

```
http.request.method == "GET"
```

[POST]

```
http.request.method == "POST"
```

show packets between time range

Say you want to find http traffic between 08/12/2021 11:24:00 and 01/12/2021 11:03:00 then the below filter is used

```
http and (frame.time >= "Dec 08, 2021 11:03:00") && (frame.time <= "Dec 08, 2021 11:24:00")
```

Finding domain names in https/ssl packets

First we filter for https/ssl traffic

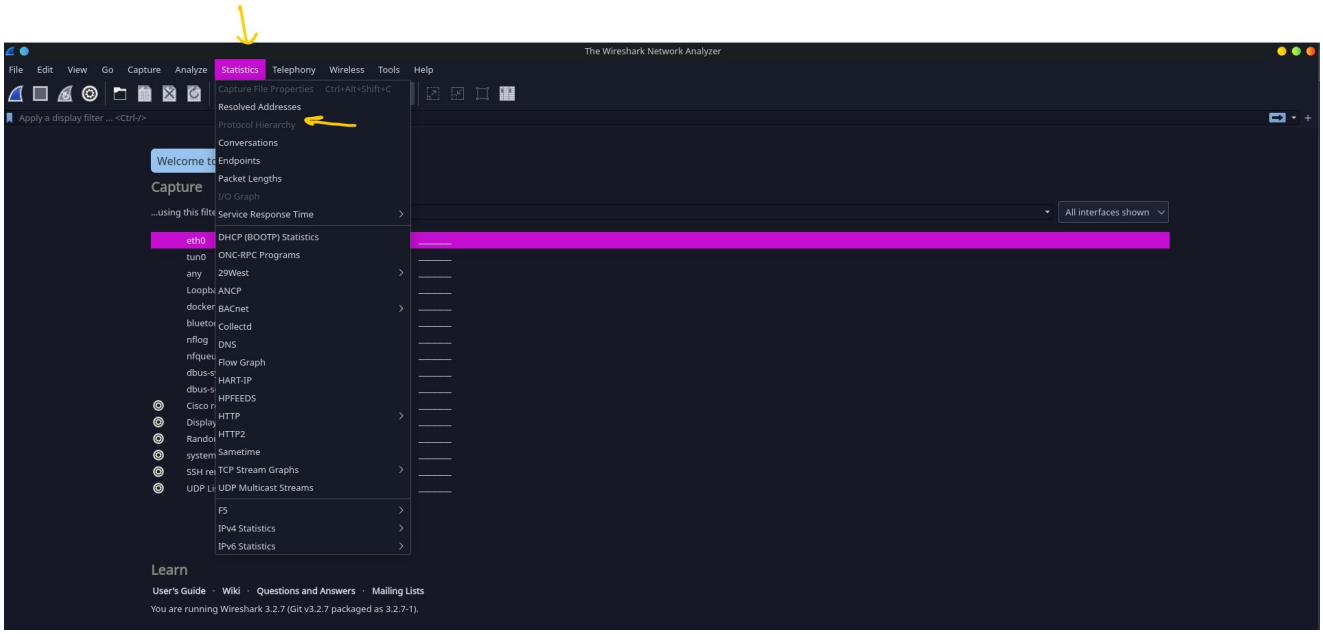
```
tcp.port == 443
```

Then we look for packets where the info section contains [client hello] then follow [TCP stream].

Data Extraction and Statistics

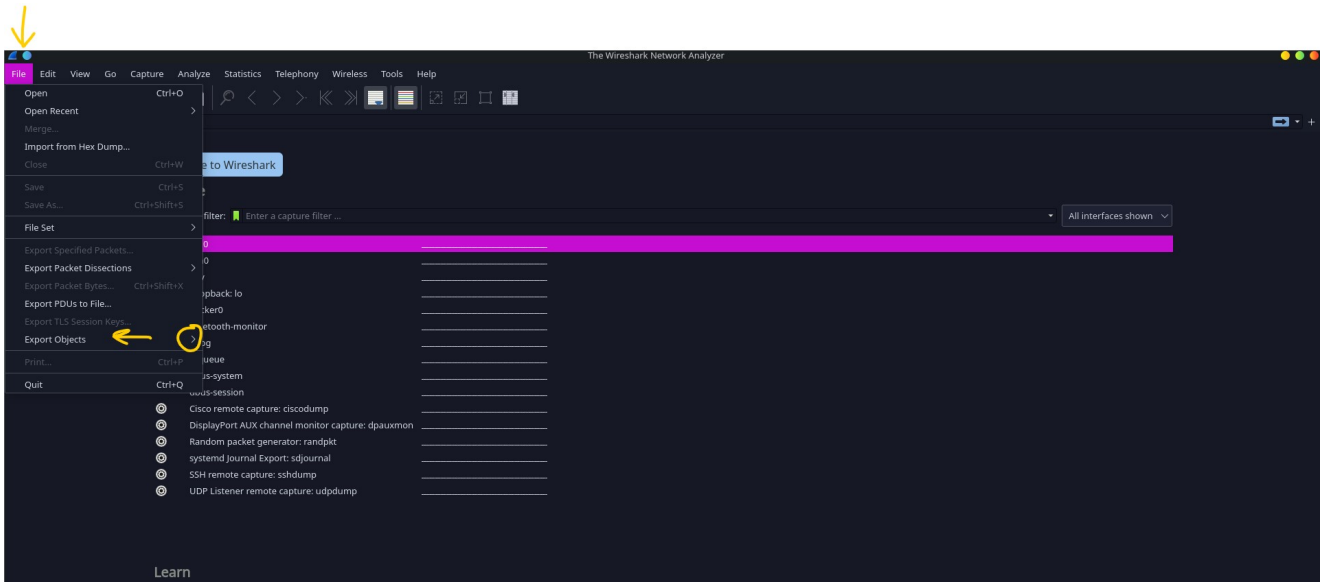
show number of packets for a specific potocal such as http

From wireshark menu >> statistics >> protocol hierarchy >> note the packets field that indicates the number and the corresponding protocol.



Exporting images and files transferred through HTTP

```
Select File -> Export Objects -> HTTP
Selecting specific packet number
Go -> Go to Packet
```



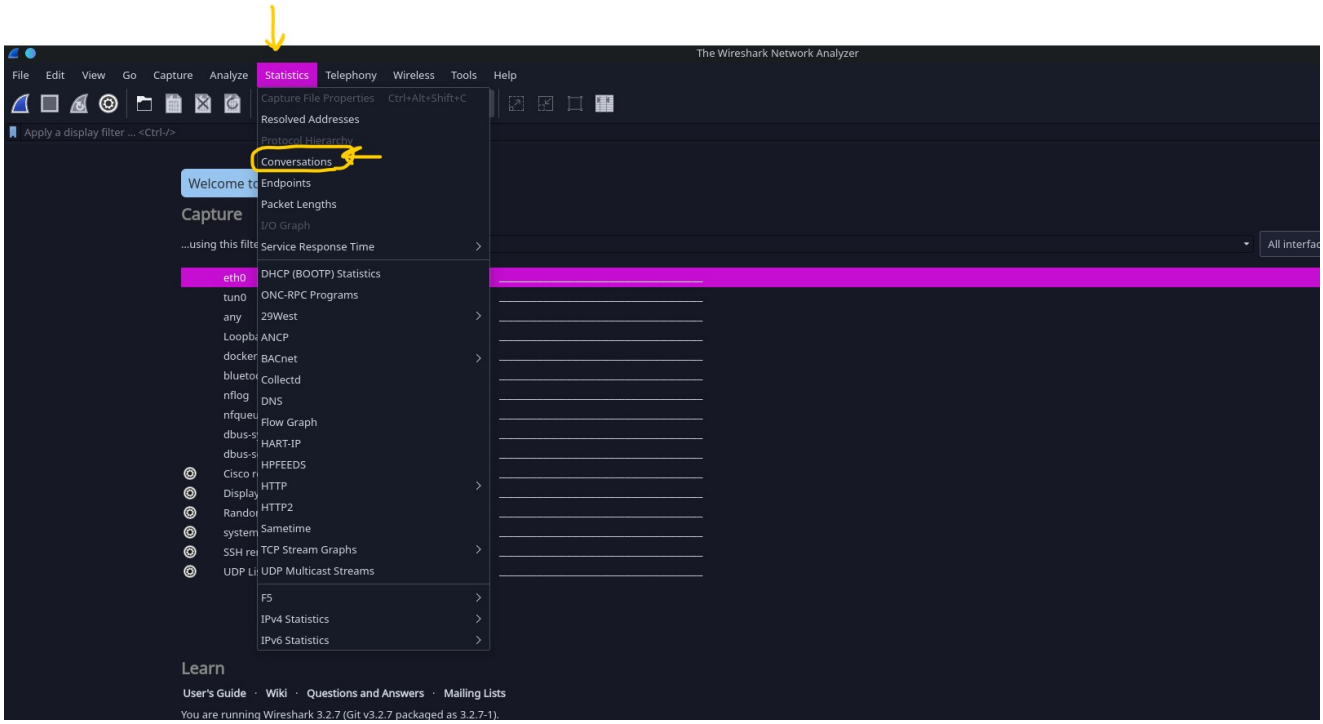
OR

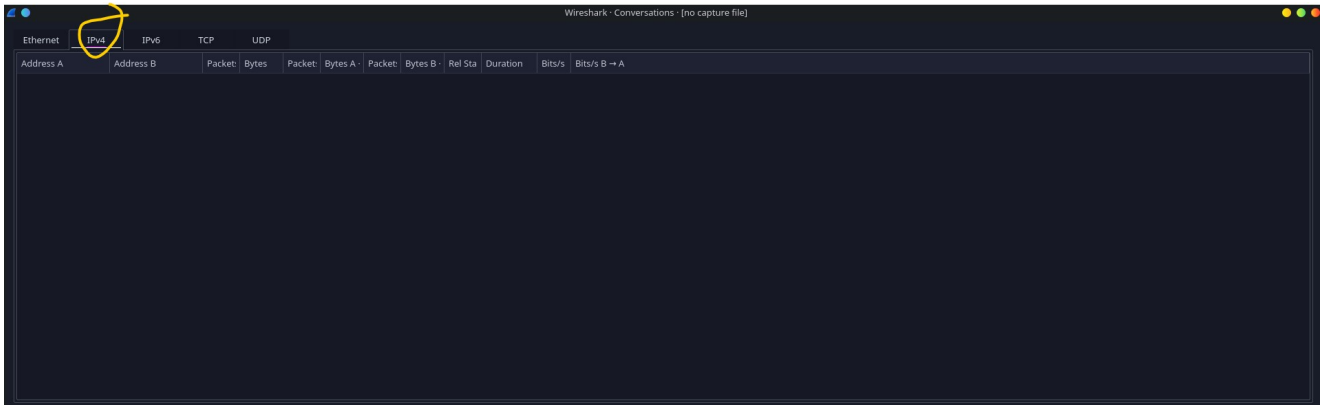
```
Frame.number == [99]
```

Extracting source or destination ip addresses from a pcap

```
tshark -T json -e 'ip.src' -e 'ip.dst' -r
filename.pcap  | grep '\.[0-9]' | sort -u
```

Or you can follow the below order from Wireshark menu
Statistics >> Conversation >> IPV4 field

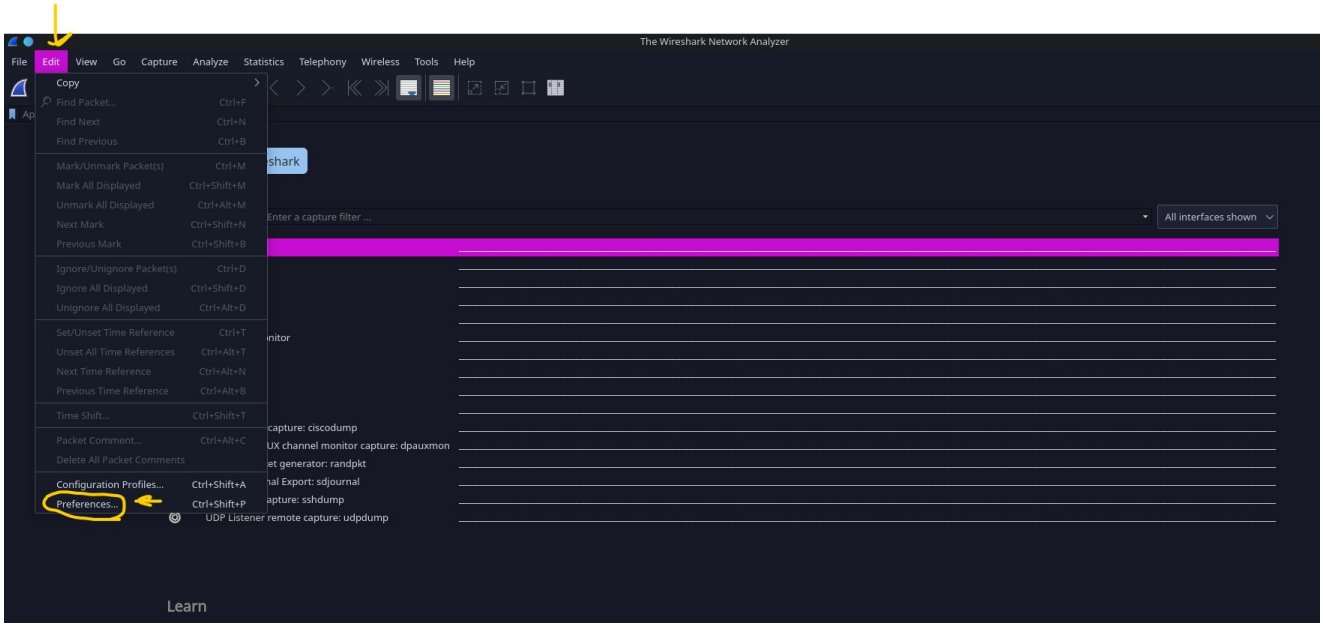


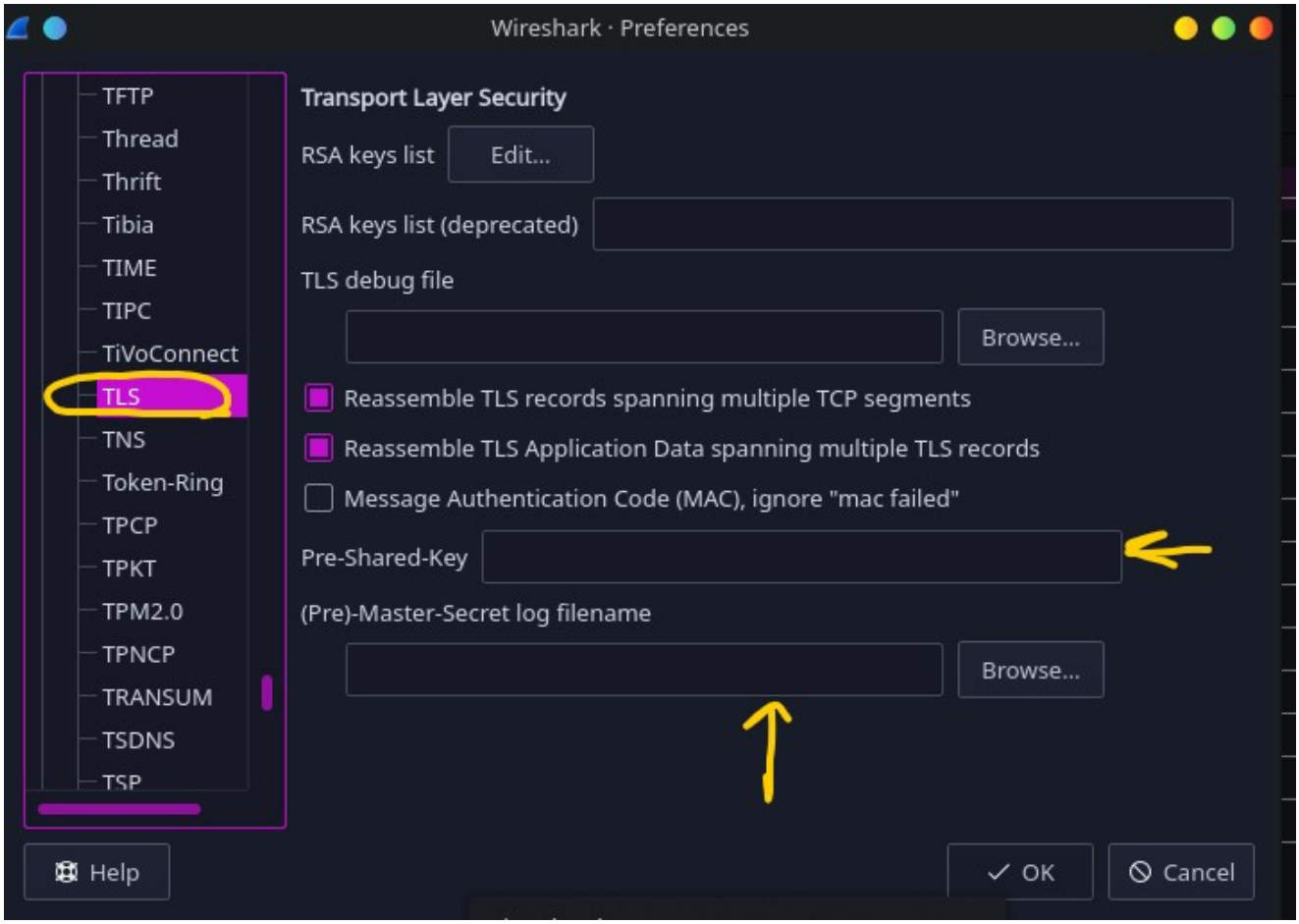


Decrypting SSL traffic

Wireshark can't decrypt encrypted traffic by default so we need to specify the decryption parameters if we got any. Most commonly for [SSL] we will need to provide the [pre-master key or secret].
If you are already inspecting encrypted traffic look for [CLIENT_RANDOM] and see if you can find its

value. Once you save it in a [.txt] file and load it as a pre-shared key as shown below



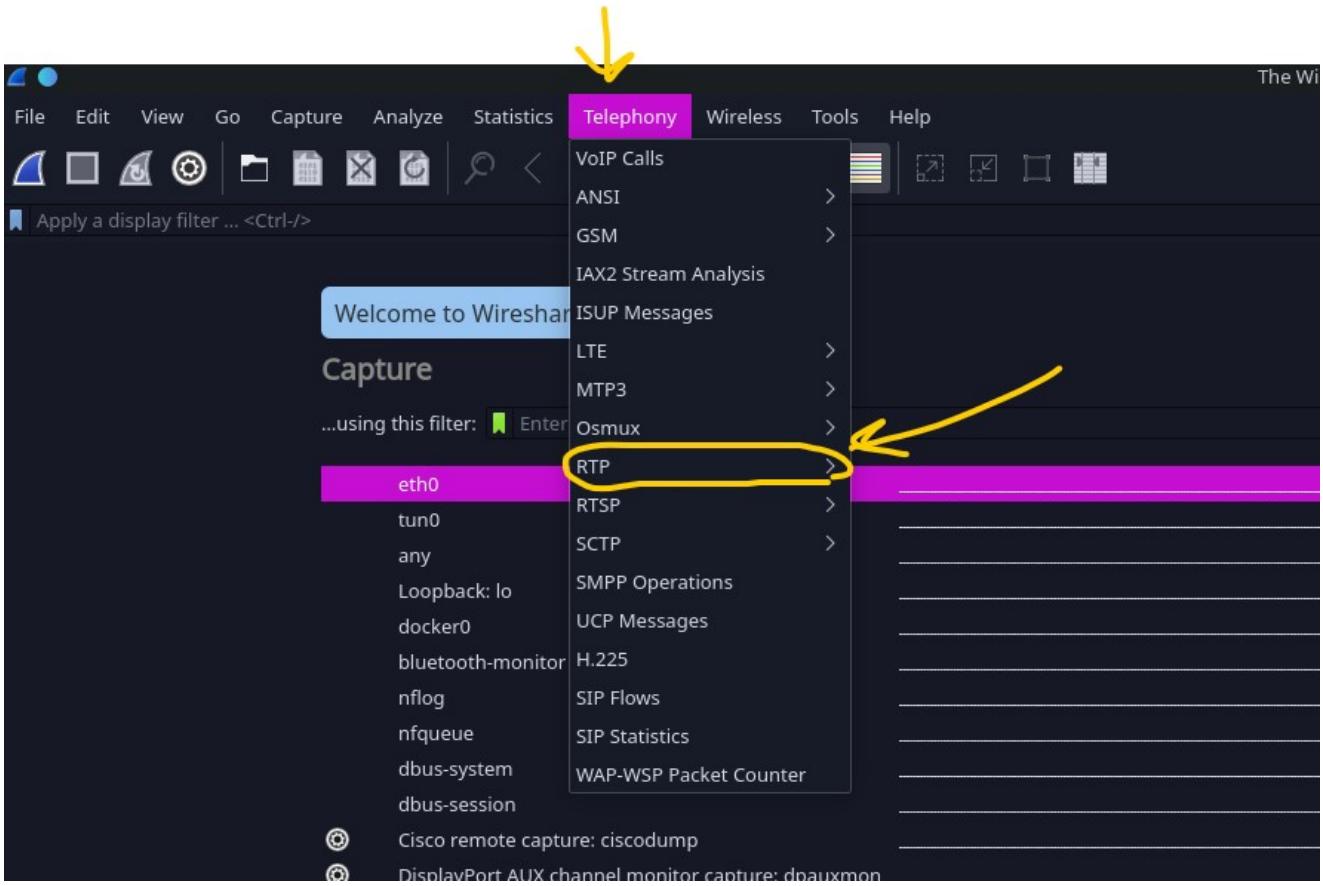


Extracting RTP Audio Files

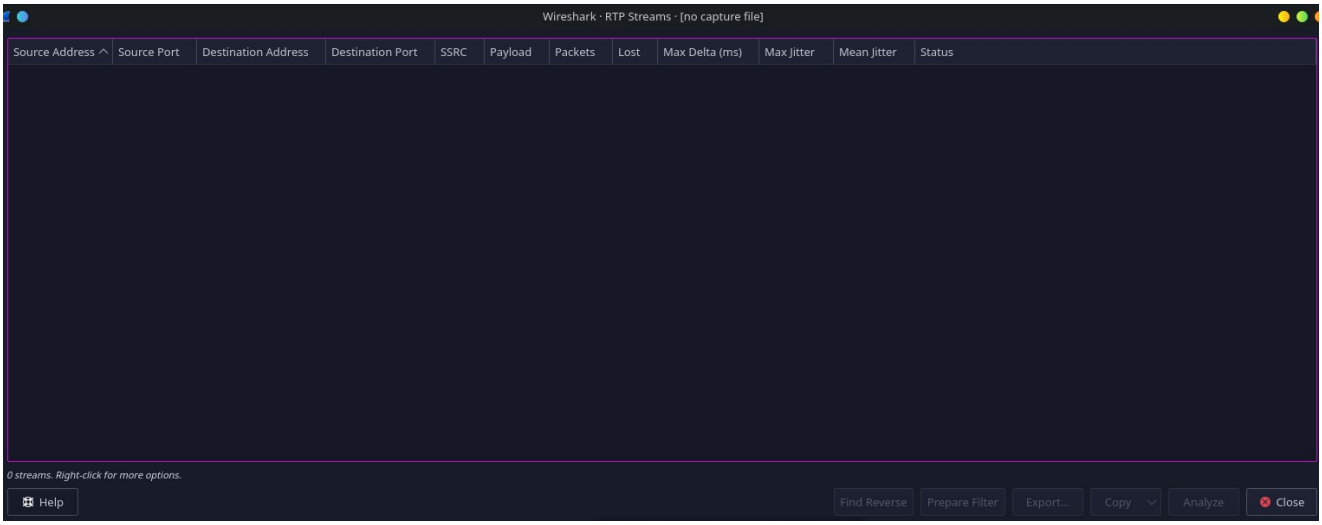
RTP communications normally operate over UDP port [1313] or [51393]. We can create a filter to display packets with these ports

```
udp.port == 1313 or udp.port == 51393
```

After displaying RTP packets, we can right click on the packet and select [Decode as RTP]
Then from the menu we follow as below



Select [RTP Streams] will bring up the below menu



Then you can select the stream and select [Analyze] you will be able to play the Audio