**Exp No: 1**

**Web and Mobile Security**

**Name-** ANIKET KUMAR

**Branch-** CSE

**Semester-** 5th

**Section-** 20BCS\_WM-703

**Subject-** WMS Lab

**Subject Code-** CSP-338

**UID-** 20BCS5306

**Group-** B

**Aim :** Identify HTTP packet on Wireshark

**Objective :** To analyse http traffic

**Software/Hardware Requirements :** Windows 7 or above

**Tools to be used :**

1. Wireshark packet sniffer and packet capture library
2. Microsoft Word
3. WinZip as necessary

**Introduction** : Wireshark is a network packet analyser. A network packet analyser presents captured packet data in as much detail as possible.

You could think of a network packet analyser as a measuring device for examining what’s happening inside a network cable, just like an electrician uses a voltmeter for examining what’s happening inside an electric cable (but at a higher level, of course).

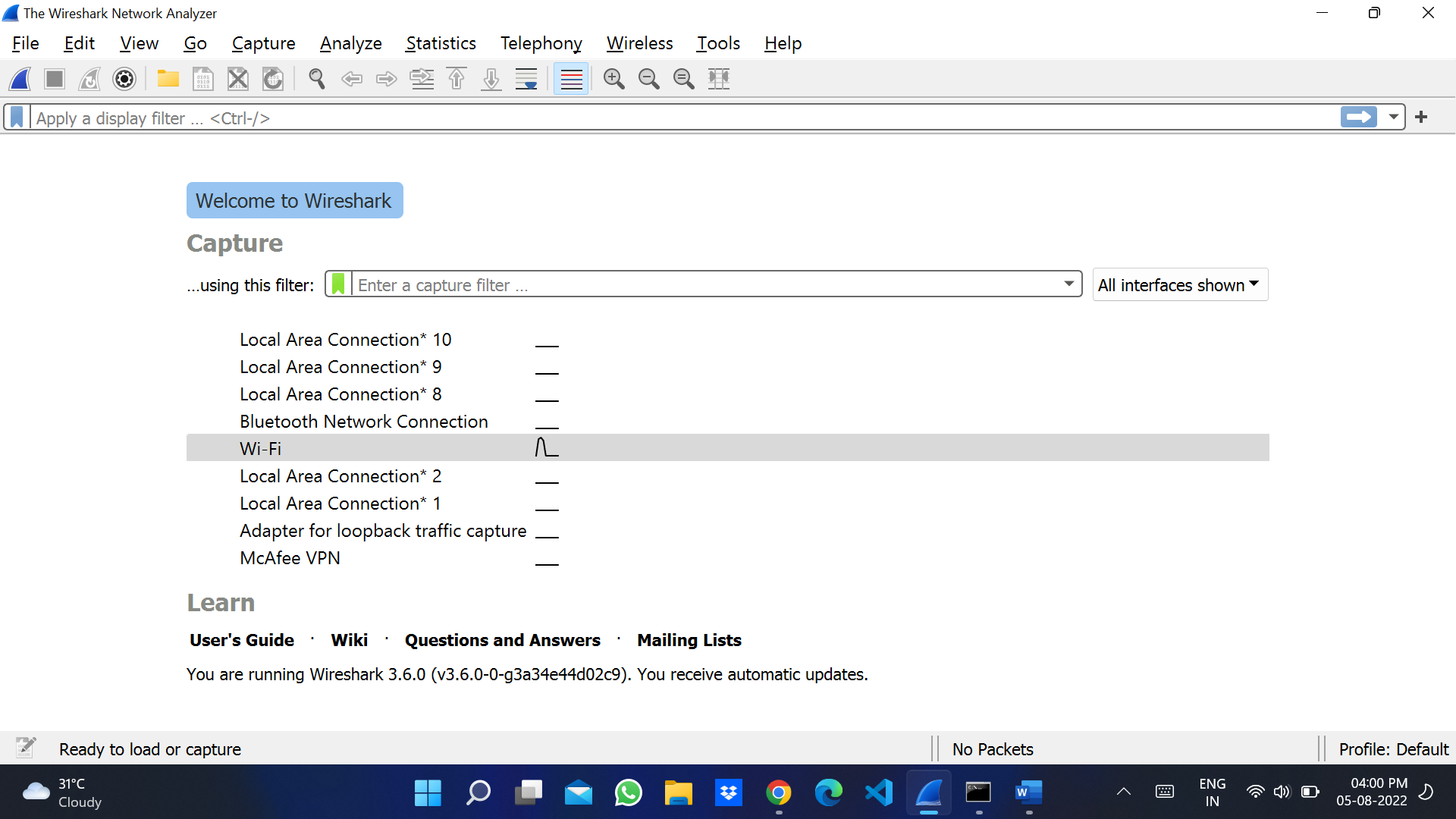
In the past, such tools were either very expensive, proprietary, or both. However, with the advent of Wireshark, that has changed. Wireshark is available for free, is open source, and is one of the best packet analysers available today.

**Steps/ Methods/ Coding :**

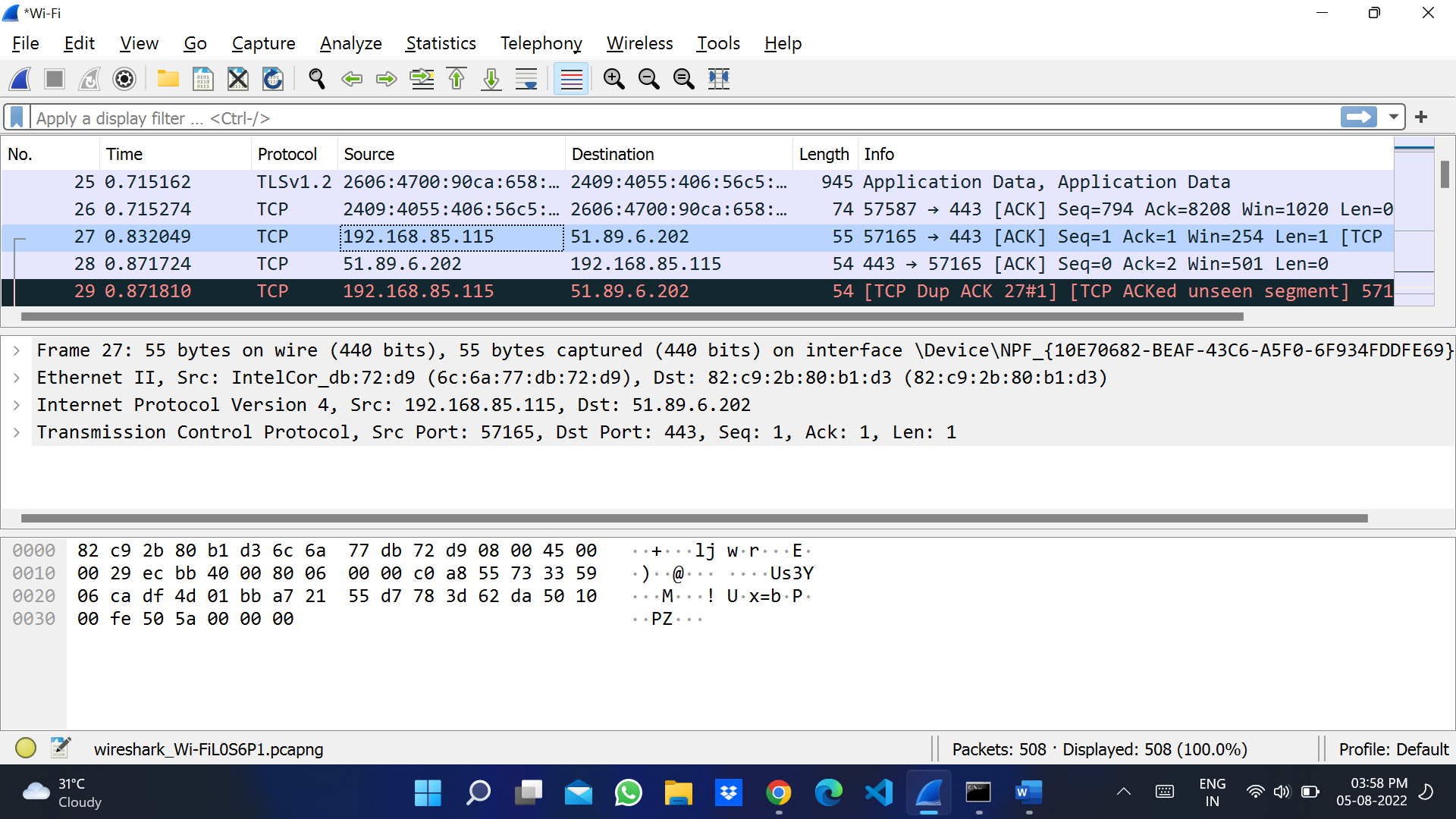
1. Install Wireshark
2. Open your internet browser
3. Clear your browser cache
4. Open Wireshark
5. Click on the network you are connected to
6. Click on **Start** button to capture traffic via this interface.
7. Watchout for your IP address in the list and click on it.
8. Press **Ctrl+E** to stop capturing.
9. After the traffic capture is stopped, please save the captured traffic into a **\*.pcap** format file.

**Output Screenshots :**

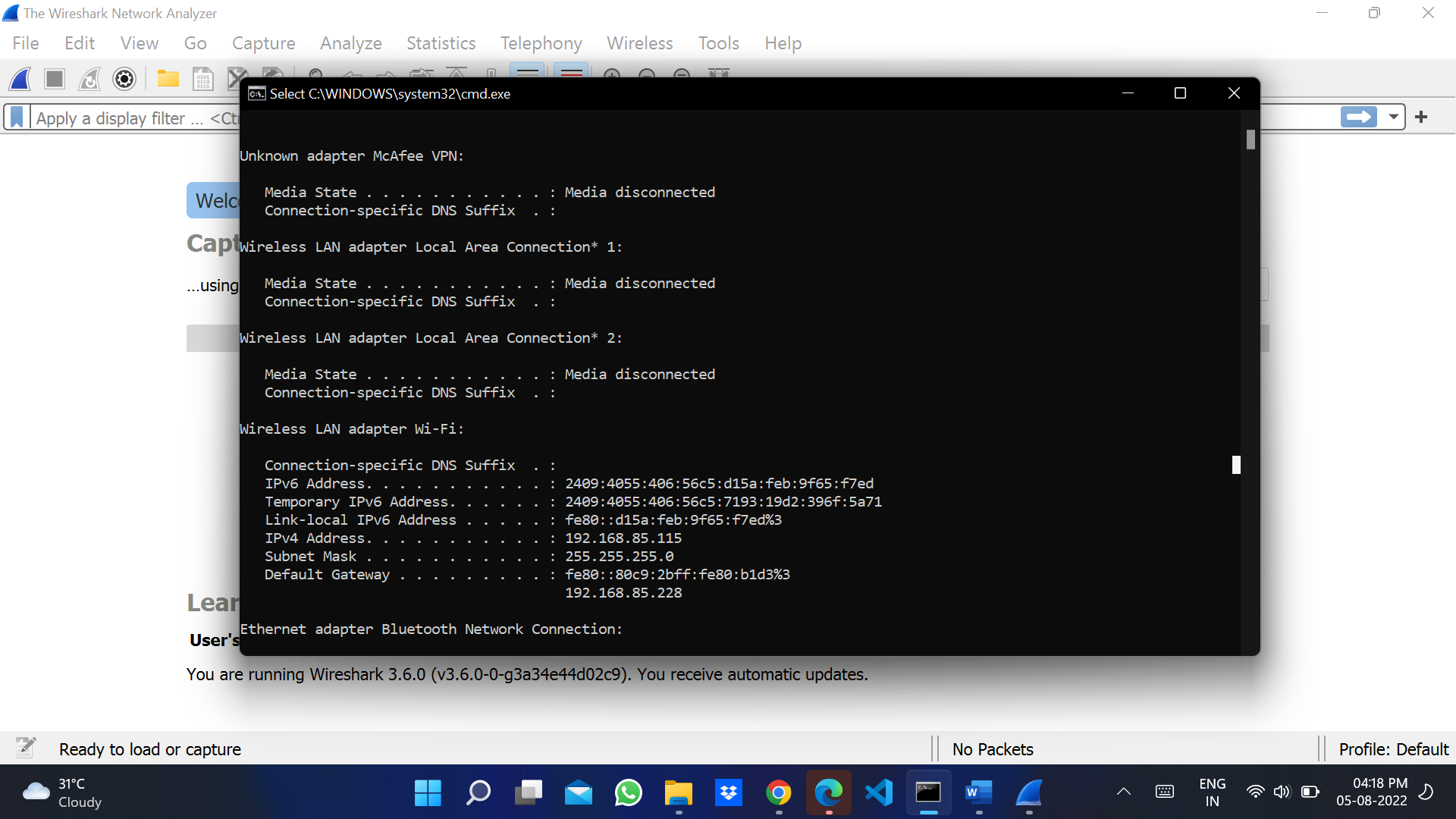
**Home Page :**



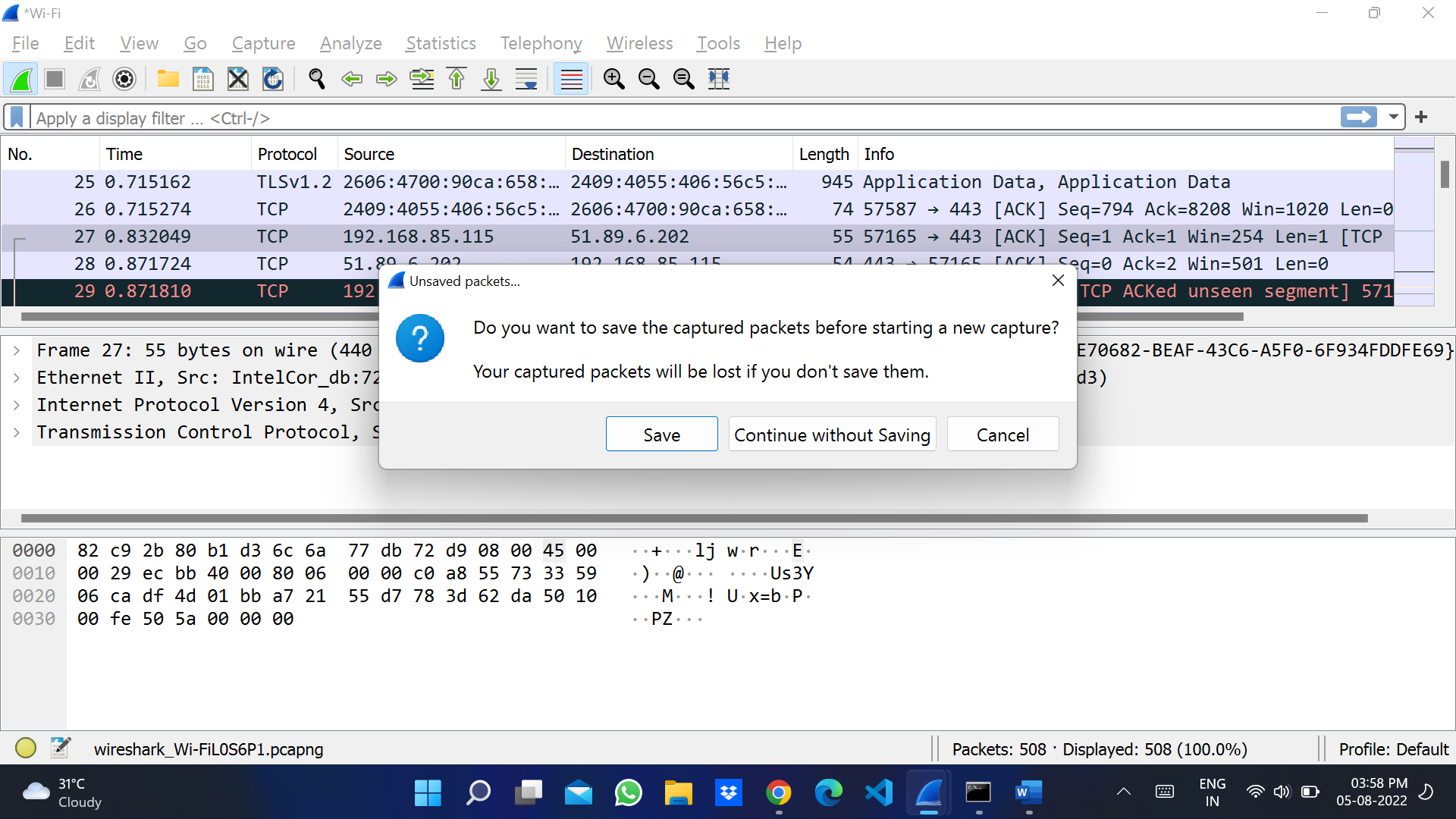
**Start Capturing :**

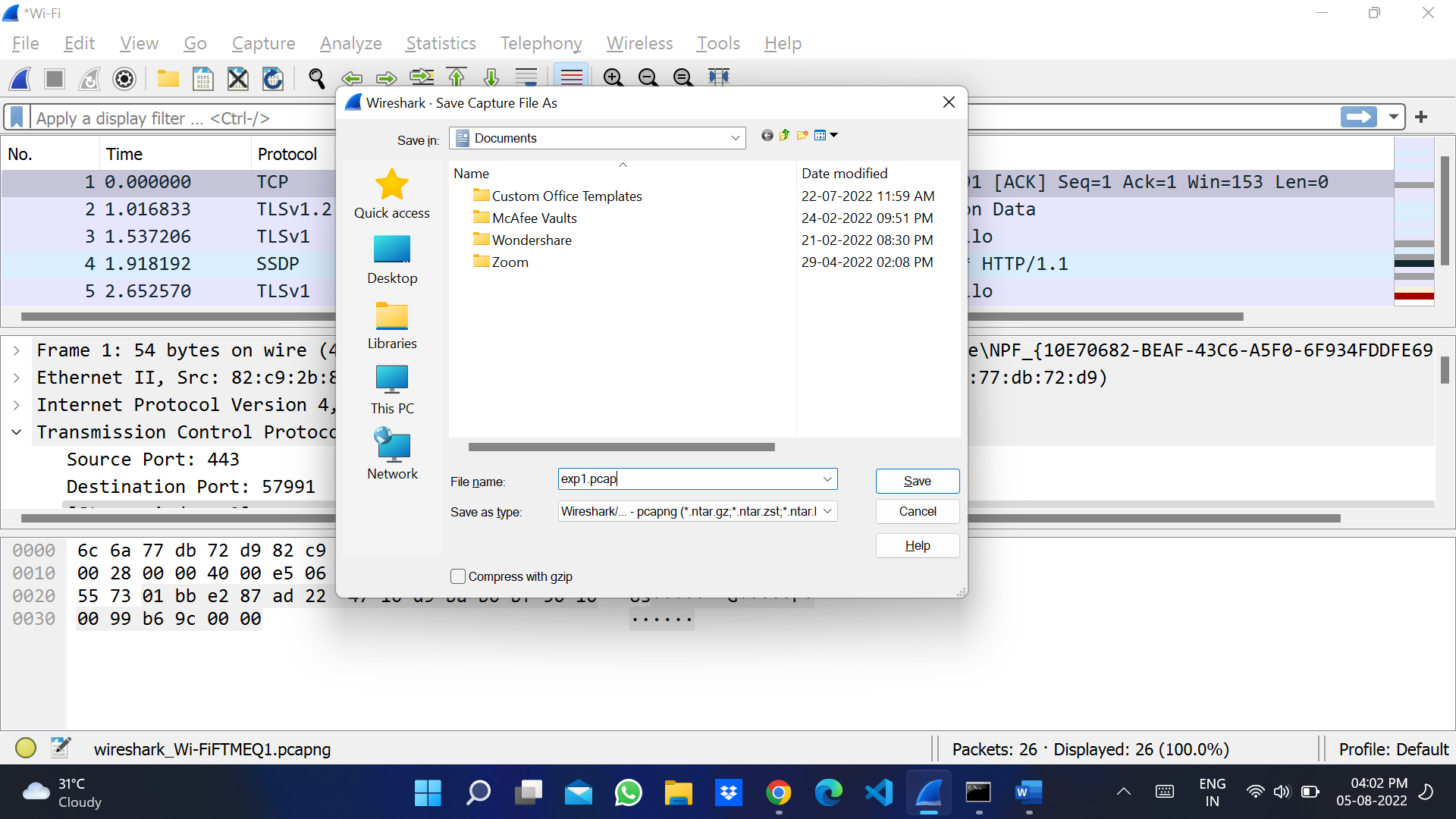


**See ipconfig on cmd :**



**After you stop capturing save the file :**



**Save with .pcap extension :**

**Learning Outcomes :**

Identify requests (from client) and response packets. Find http version, response code/ phrase, requested file (including size). Observe single small file (e.g., simple html file request/ response behaviour for a file that has already been received. Observe how a larger file is sent in multiple segments. Observe multi-file (e.g., web page with image) request/ response behaviour for a page that needs authentication.