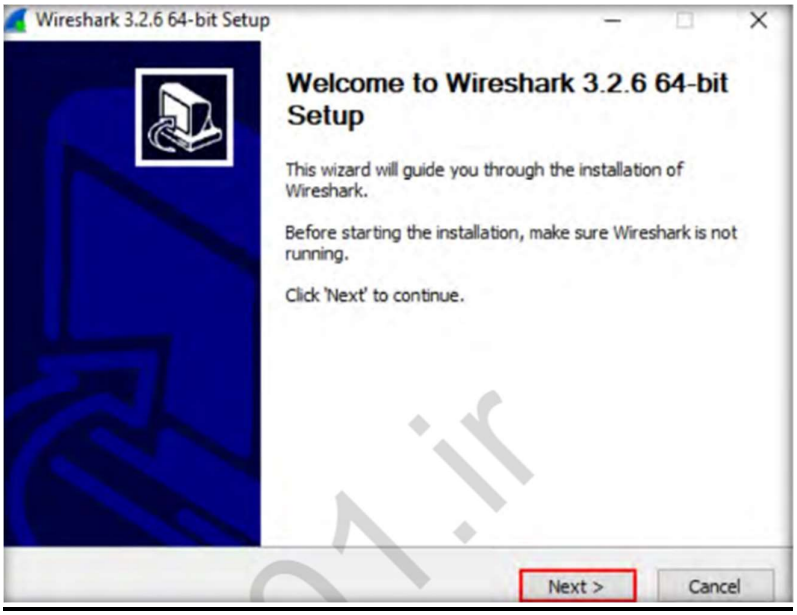
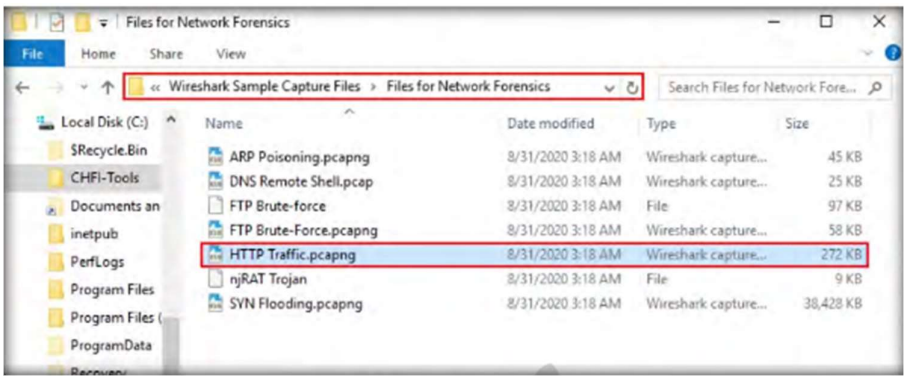
PRACTICAL NO. 05

a. Using Web attack detectiontools [Wireshark]

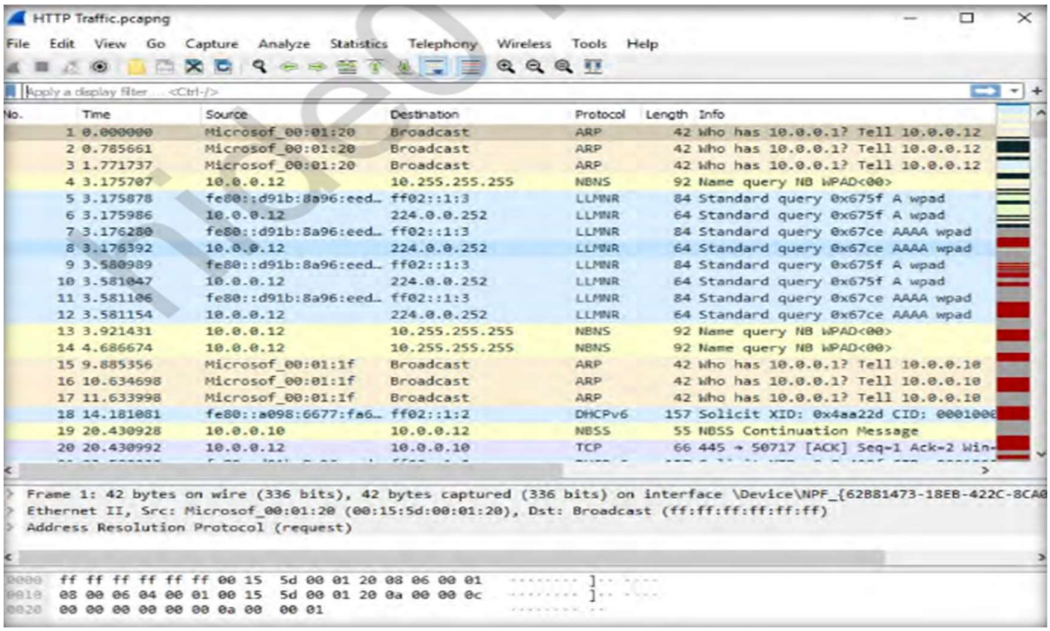
1. Download and install wireshark



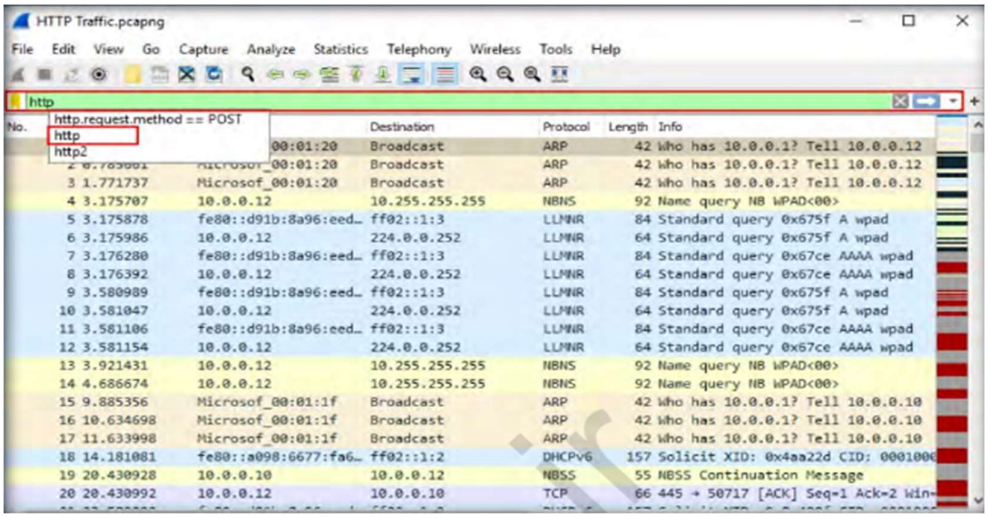
2. After completing the installation, navigate to C:\CHFI-Tools\EvidenceFiles\WiresharkSample Capture Files\Files for Network Forensics and double-click HTTP Traffic.pcapng.

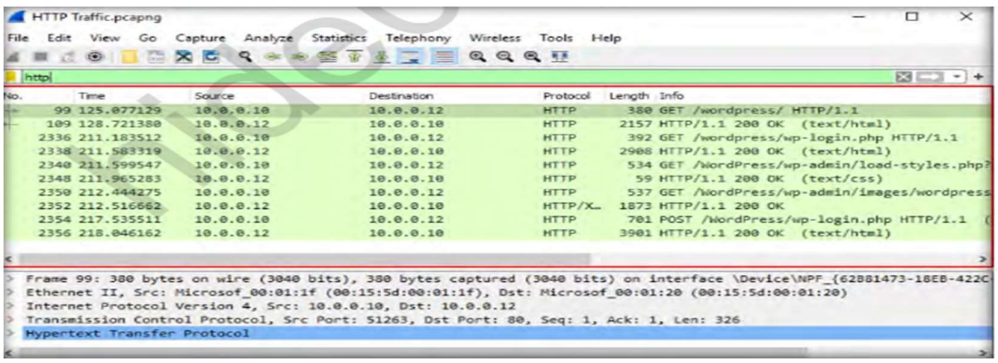


3. The above operation will launch the Wireshark GUI window and display the packets captured in HTTP Traffic.pcapng, as shown in the following screenshot

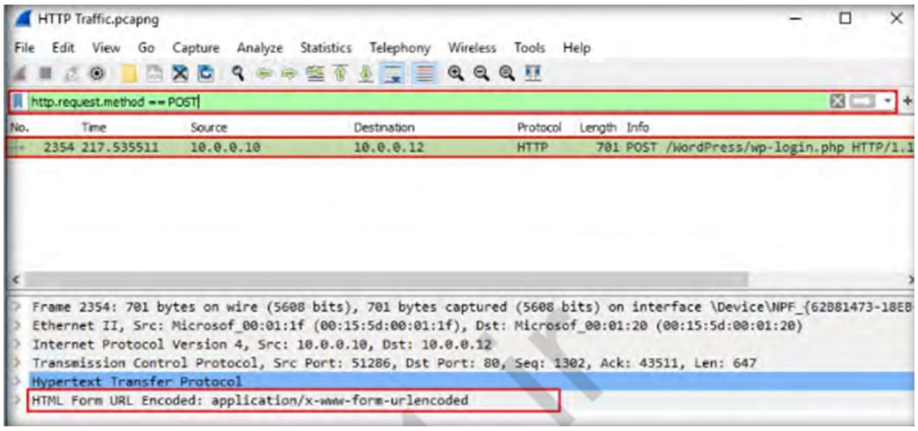


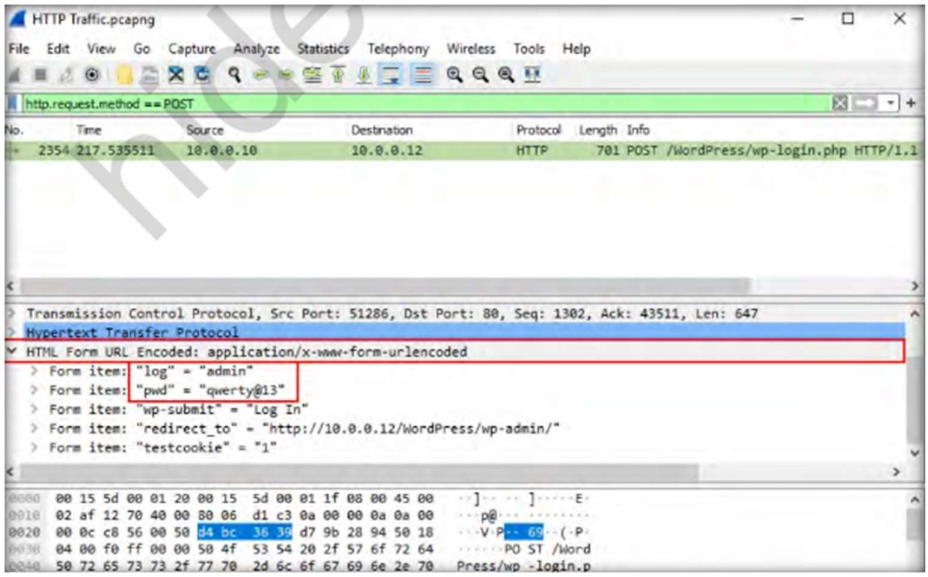
4. We will now apply the http filter so that the application displays results related to traffic generated through http. Type http in the Filter field and press Enter to filter the http traffic, as indicated in the screenshot below:



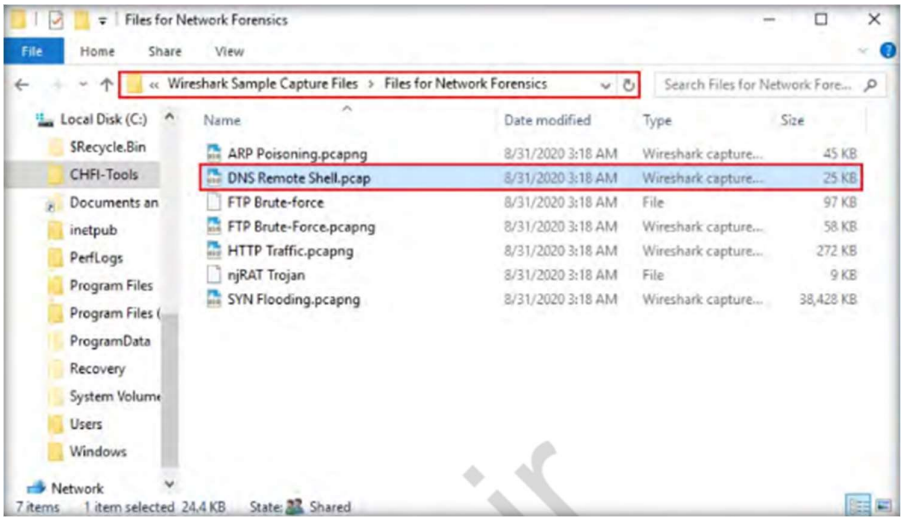


5. After examining the entries displayed by the application in the screenshot above, it is evident that the http traffic is associated with a WordPress website, and it is being transmitted in a plain text format. Generally, user credentials are stored in the POST requests. Therefore, examining packet(s) containing POST request can help an investigator find the user credentials.
6. We will now filter the traffic to obtain results specific only to POST request(s). Now, type http.request.method== POST in the Filter field and press Enter. Wireshark filters the traffic containing POST request(s) and displays them, as shown in the screenshot below. The user credentials stored in this request can be found under the Packet Details pane in the middle of the application window, under the HTML Form URL Encoded node.

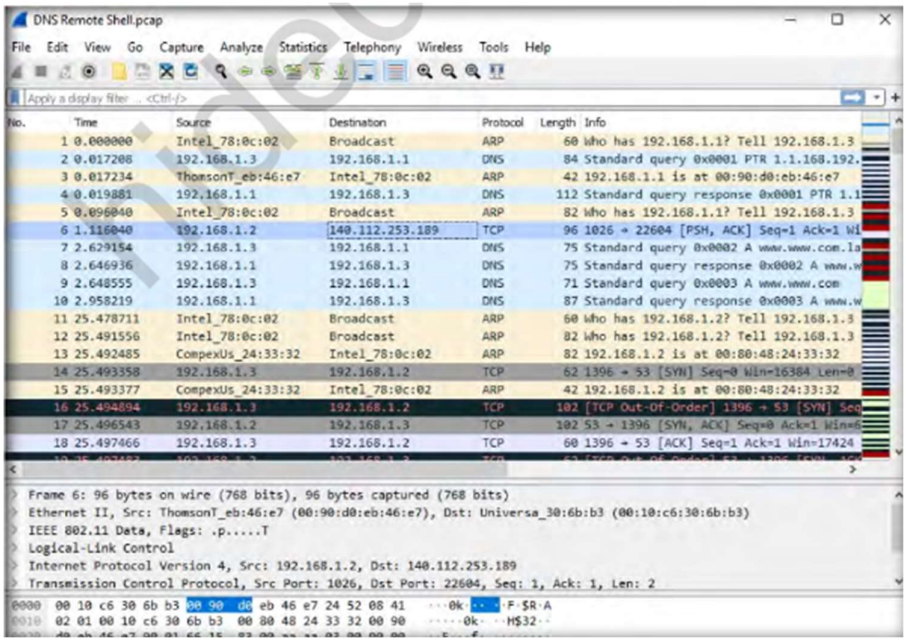




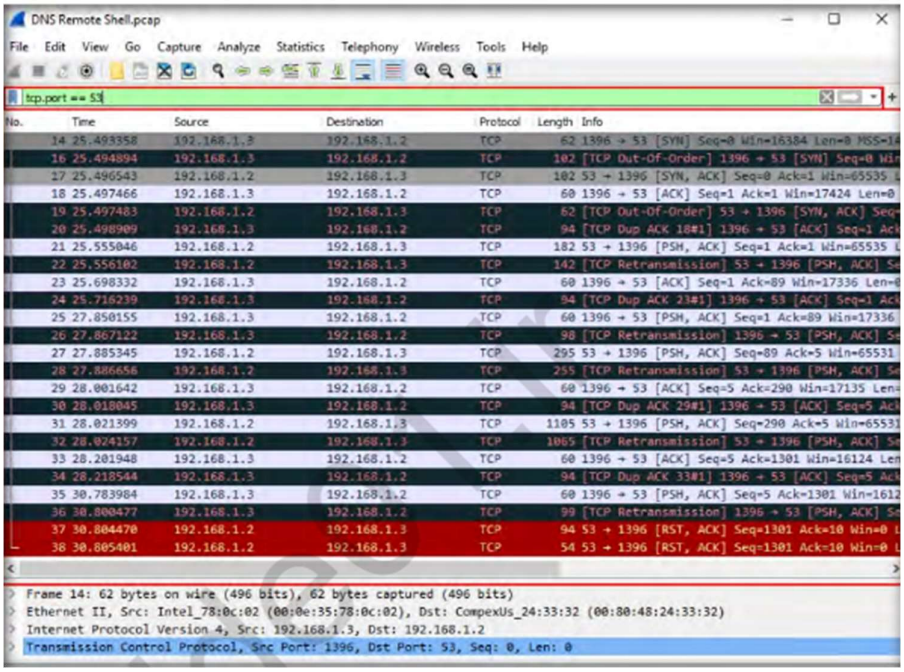
7. Now, navigate to C:\CHFI-Tools\Evidence Files\Wireshark Sample Capture Files\Files for Network Forensics and double-click DNS RemoteShell.pcapng.



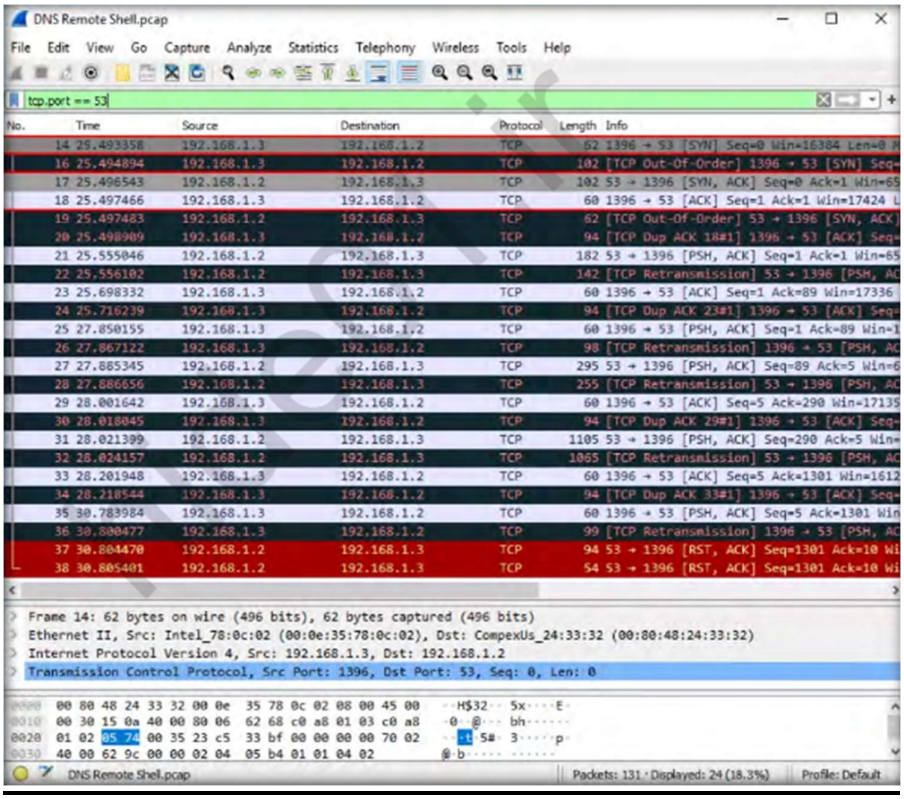
8. The network traffic entries captured in the DNS Remote Shell.pcapng file will now be displayed in the Wireshark GUI window, as shown in the following screenshot



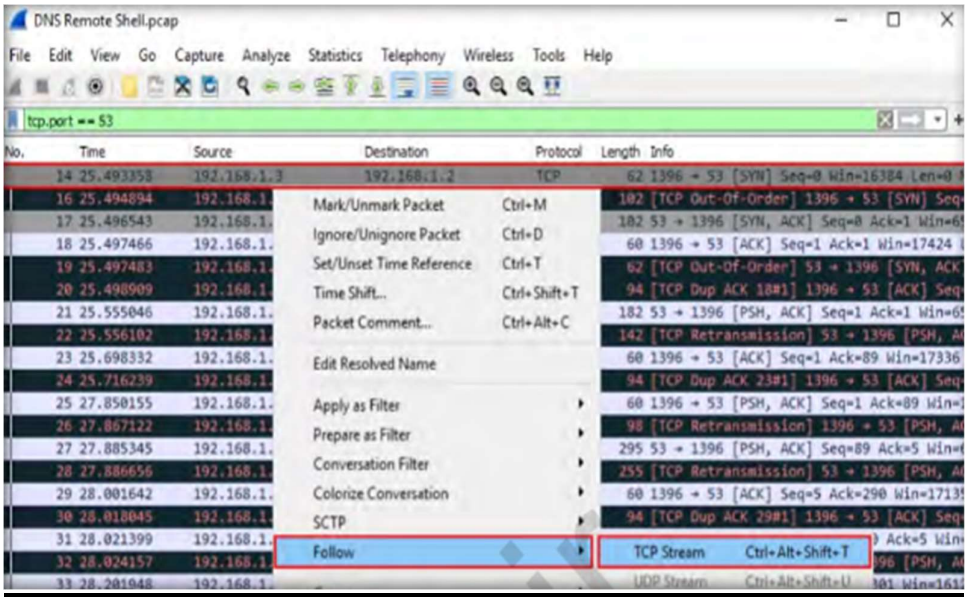
9. Since DNS uses port 53 for communicating with clients, we will be filtering the traffic flowing to and from port 53. To filter traffic flowing on port 53, type the command tcp.port == 53 in the Filter field and press Enter. Wireshark filters the traffic flowing on port 53 and displays it, as shown in the following screenshot

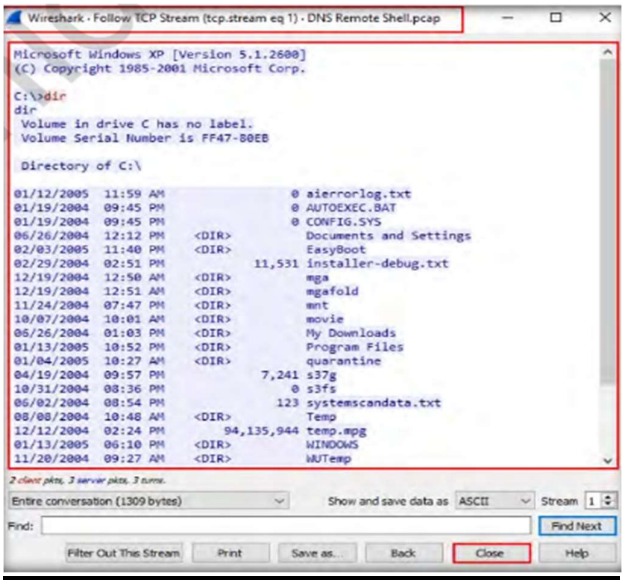


When we examine the first packet from the listed packets, i.e., Packet 14,we can see that Port 1396 on Source IP address (192.168.1.3) is trying to establish a remote connection with Port 53 on the Destination IP address(192.168.1.2) through a [SYN] request. As Port 53 is seen on 192.168.1.2,this IP address represents the DNS server here. When we examine Packet17, we see that the IP address 192.168.1.2, which is the DNS server, is responding with a [SYN, ACK] acknowledgment to IP address 192.168.1.3.Upon examining Packet 18, we see that the IP address 192.168.1.3 is sending an [ACK] acknowledgment to 192.168.1.2, thus establishing a remote connection with the target/victim (i.e., 192.168.1.2). From these observations, we can infer that 192.168.1.3 is the attacker's IP address, which has succeeded in establishing a connection from its Port 1396 with Port 53 on the DNS server, as indicated through Packets 14, 17, and 18 in the screenshot below:

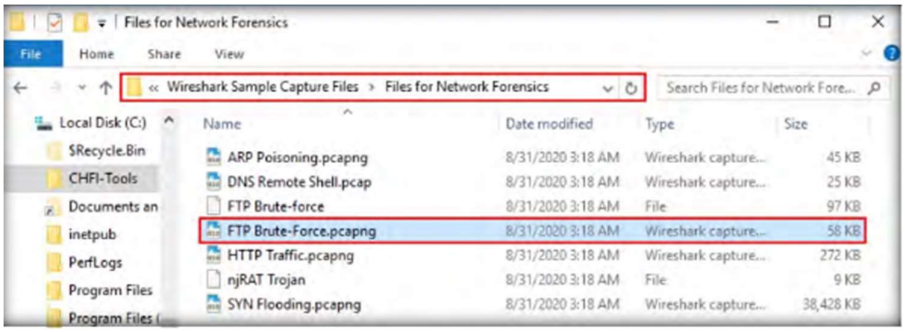


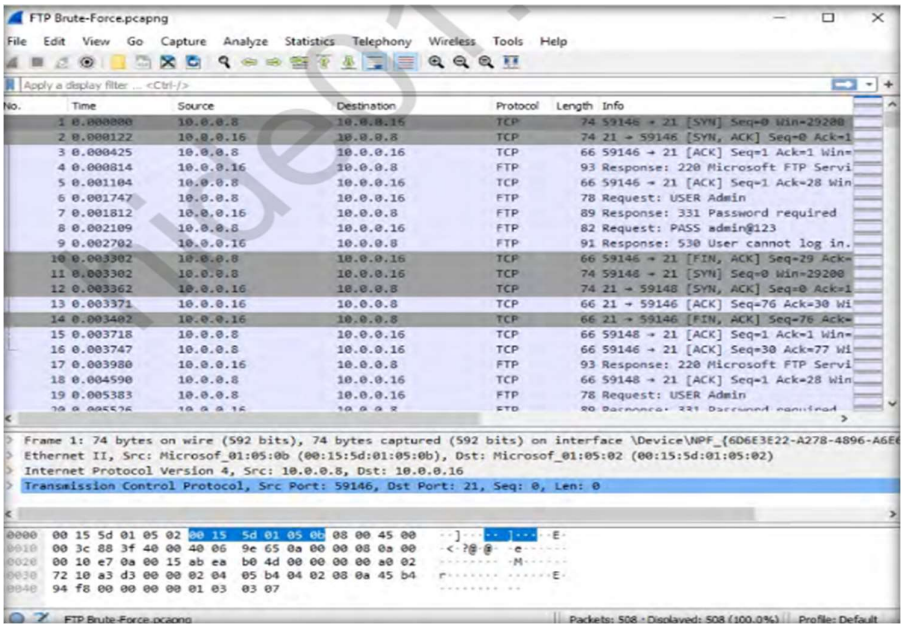
10. Right-click on any one of the packets from 14 to 38 (here, we have right-clicked on packet 14), select Follow from the context menu, and then click TCP Stream from the resultant drop-down list, as shown in the screenshot below:



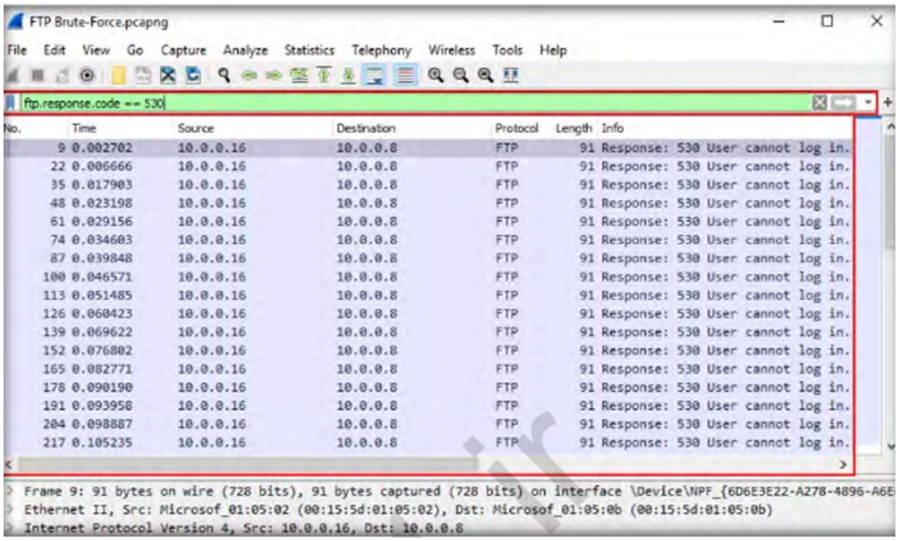


11. Now, we will look for FTP brute-force attempts in the network. Close the current packet capture file in Wireshark.

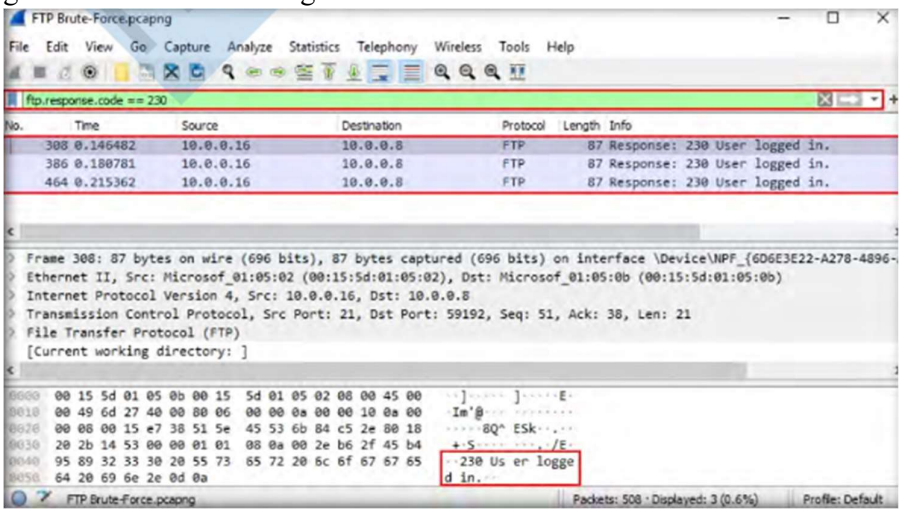




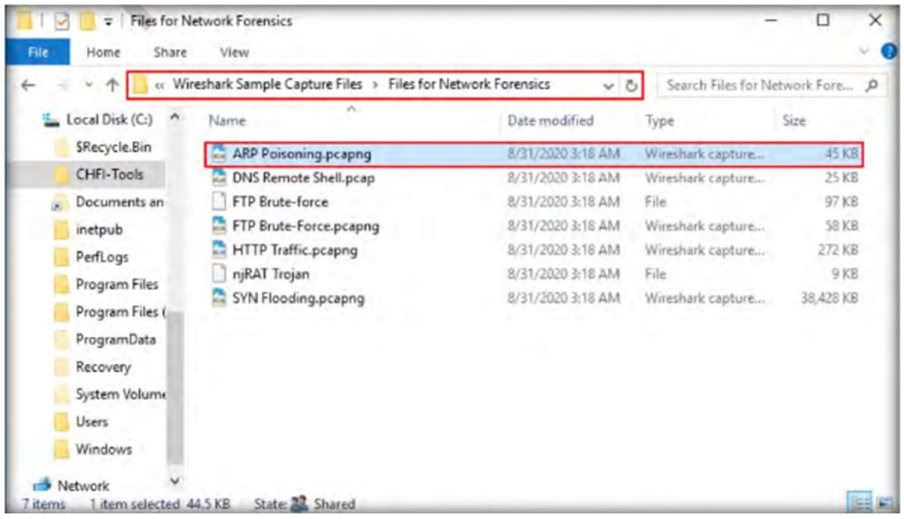
12. Apply the ftp.response.code == 530 filter to monitor all unsuccessful login attempts over FTP. Upon applying the filter, the application will fetch results pertaining to unsuccessful login attempts, as shown in the screenshot below:

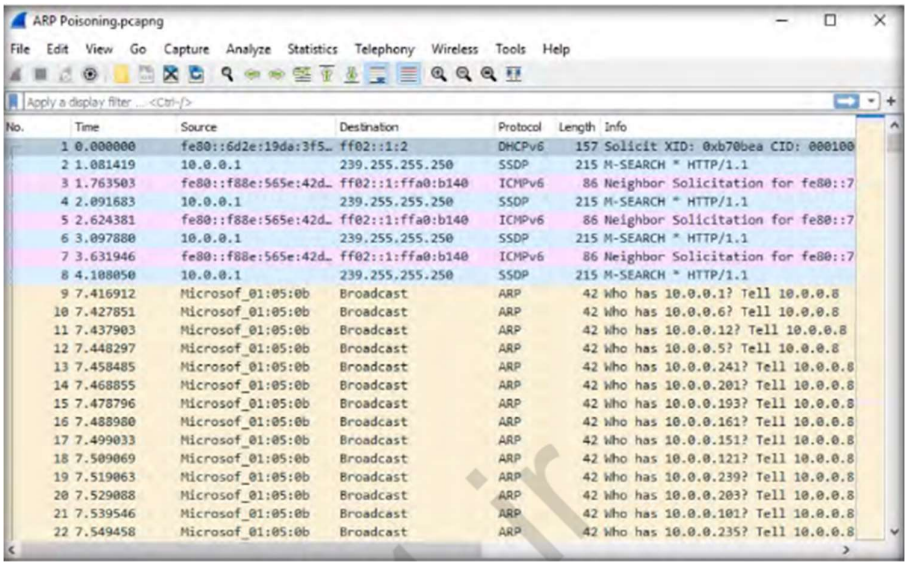


13. The screenshot above shows multiple unsuccessful login attempts madefrom the source IP 10.0.0.16 to the target IP 10.0.0.8, which stronglyindicates a brute-force attack.27. Apply the ftp.response.code == 230 filter to see successful logins on the FTP server. The application will fetch results that show successful logins from the source IP 10.0.0.16 to the target IP 10.0.0.8, as shown in the screenshot below. This indicates the attacker has successfully gained the victim's login credentials

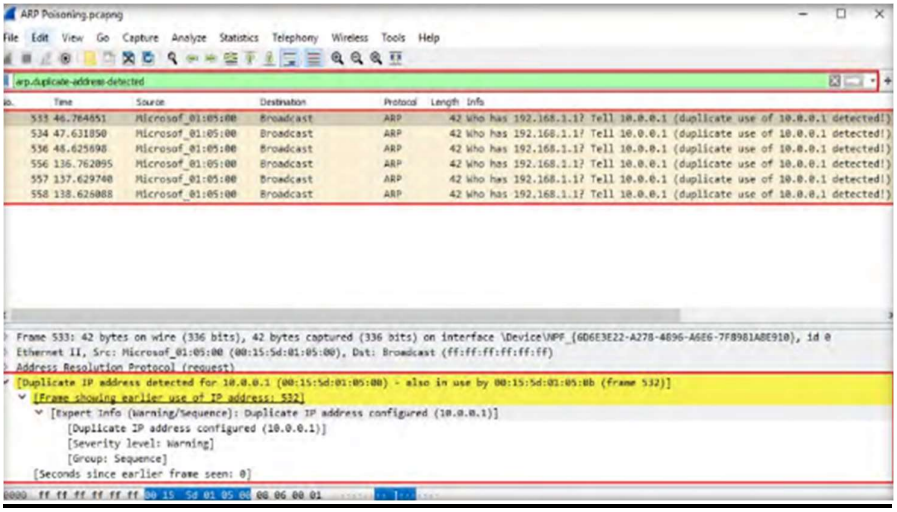


14. Now, we will look for ARP Poisoning attempt in the network.





15. when ARP Poisoning has been attempted, Wireshark will detect duplicate IP addresses on the ARP protocol with the warning message Duplicate IP address detected for <IP address>. Therefore, we need to check if the use of a duplicate IP address has been detected.45. To locate a duplicate IP address in the traffic, apply the arp. duplicate-address-detected filter. Wireshark detects duplicate IP address on the ARP protocol with the warning message Duplicate IP address detected for 10.0.0.1, as shown in the screenshot below. The duplicated IP address can be seen both in the Packet Details pane in the middle of the application window and in its upper pane.



16. In this manner, you can analyze packet capture files that have recorded the traffic on a network as part of a forensic investigation.

b. Using Log & Traffic Capturing & Analysis Tools [Wireshark]

Using Log Capturing and Analysis tools

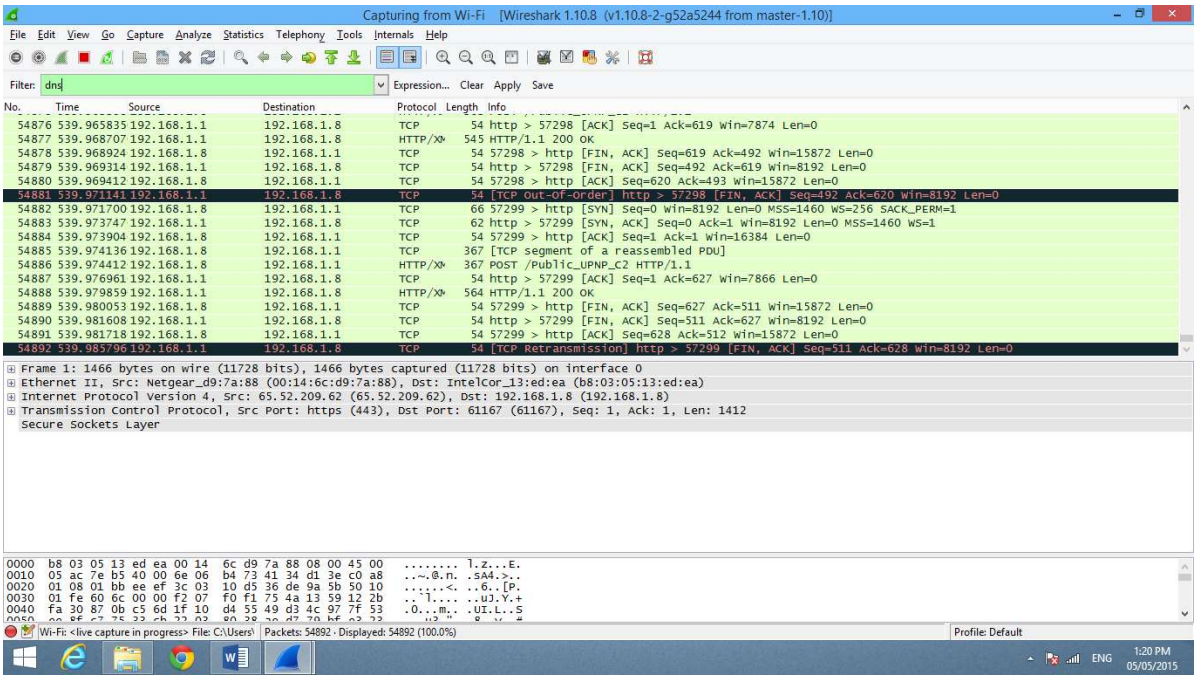
Aim: Exploring Wireshark

Wireshark is a network packet analyzer that intercepts, captures and logs information about packets passing through a network interface. This is useful for analyzing network problems, detecting network intrusions, network misuse, and other security problems, monitor usage and gather statistics, and many other applications.

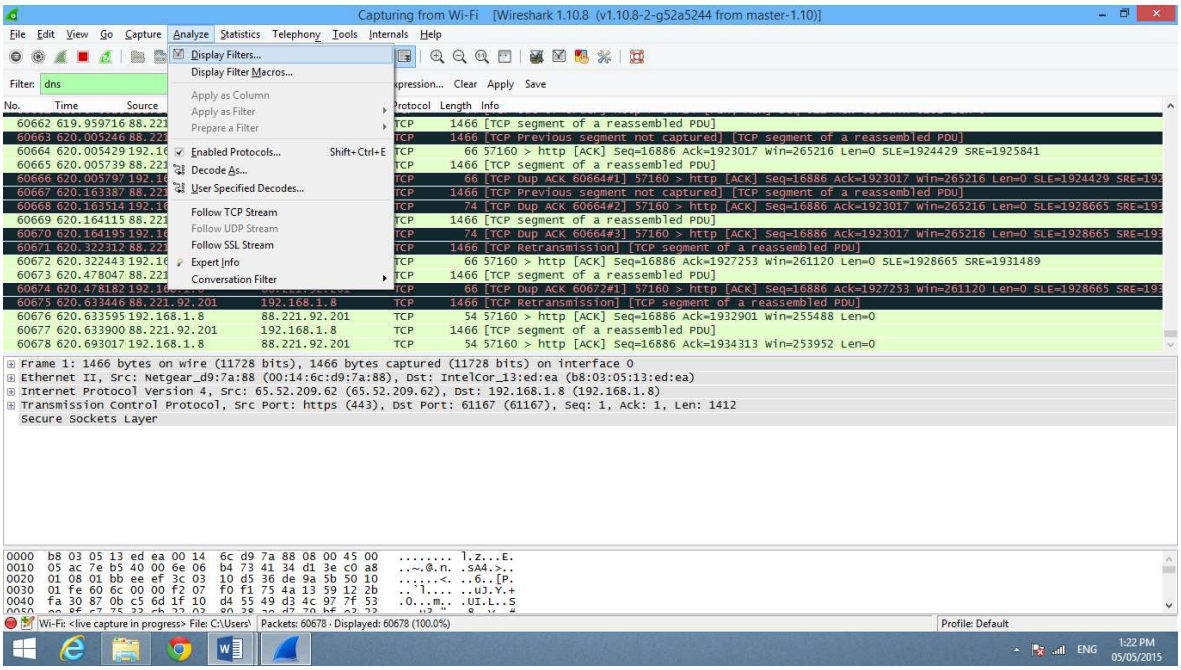
Filtering Packets

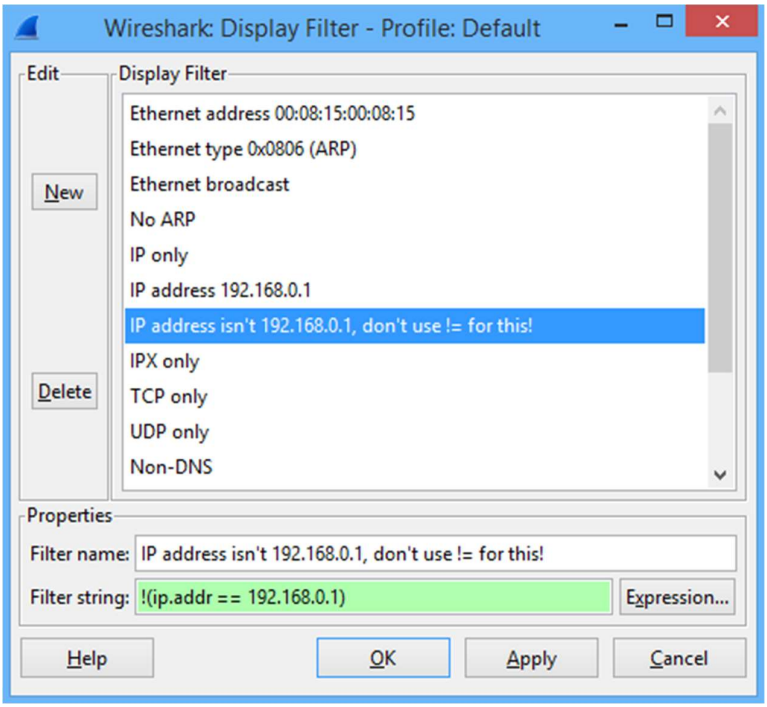
If you're trying to inspect something specific, such as the traffic a program sends when phoning home, it helps to close down all other applications using the network so you can narrow down the traffic. Still, you'll likely have a large amount of packets to sift through. That's where Wireshark's filters come in.

The most basic way to apply a filter is by typing it into the filter box at the top of the window and clicking Apply (or pressing Enter). For example, type "dns" and you'll see only DNS packets. When you start typing, Wireshark will help you autocomplete your filter.

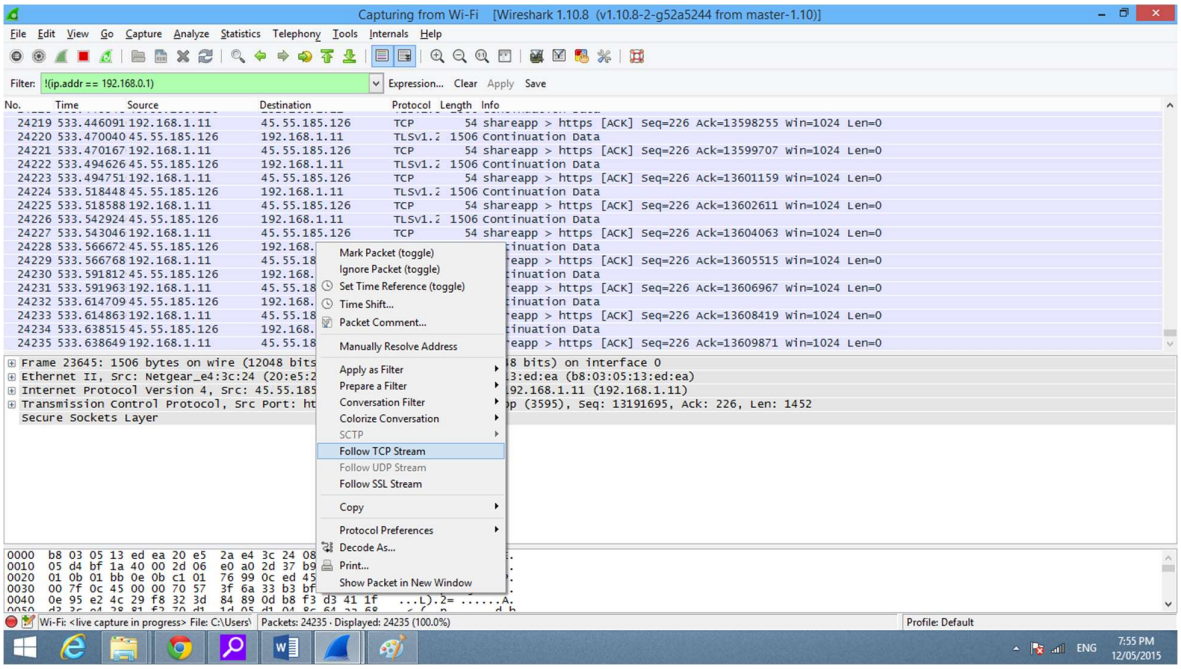


You can also click the Analyze menu and select Display Filters to create a new filter.

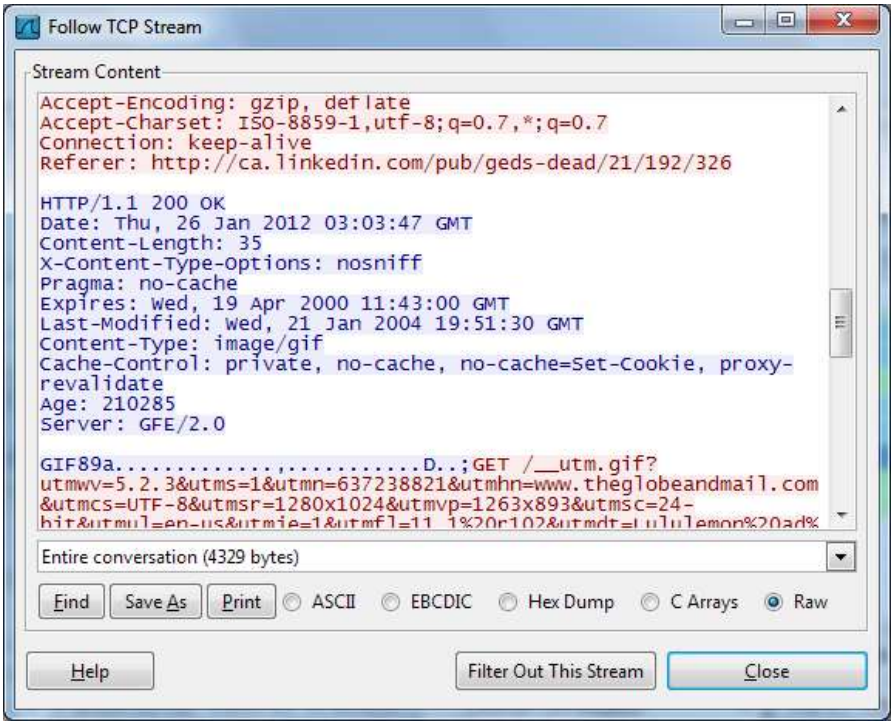




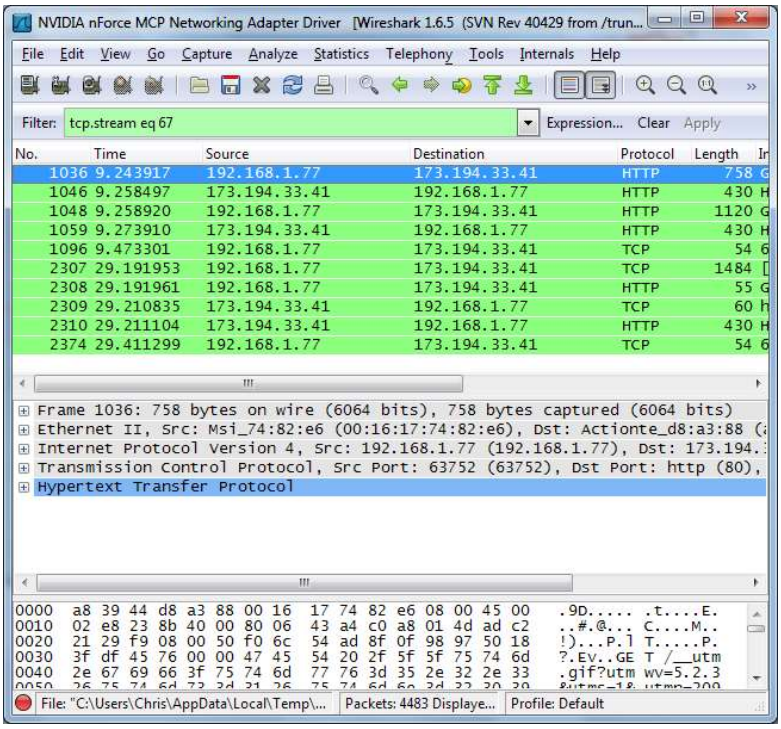
Another interesting thing you can do is right-click a packet and select Follow TCP



You'll see the full conversation between the client and the server.

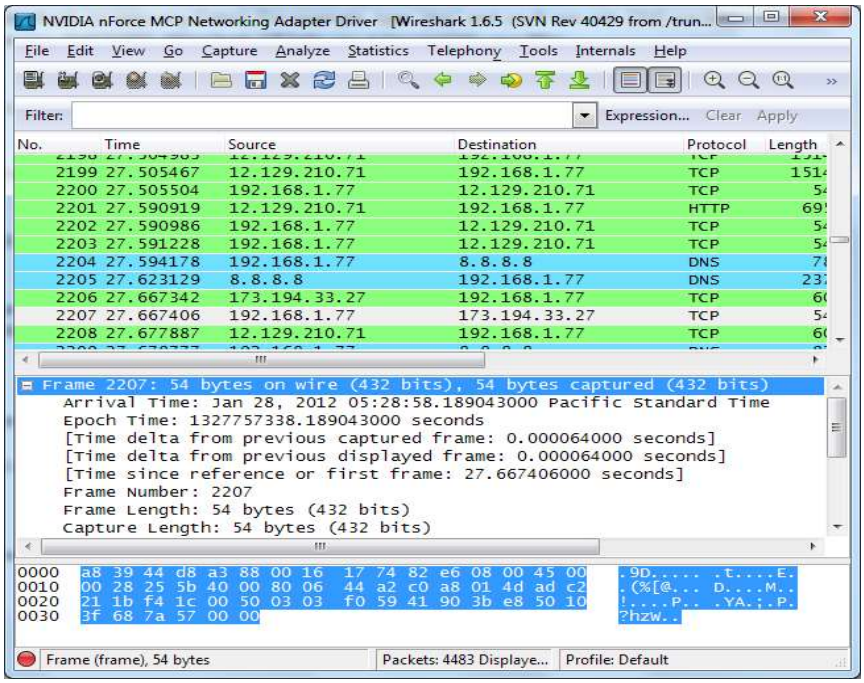


Close the window and you'll find a filter has been applied automatically — Wireshark is showing you the packets that make up the conversation.

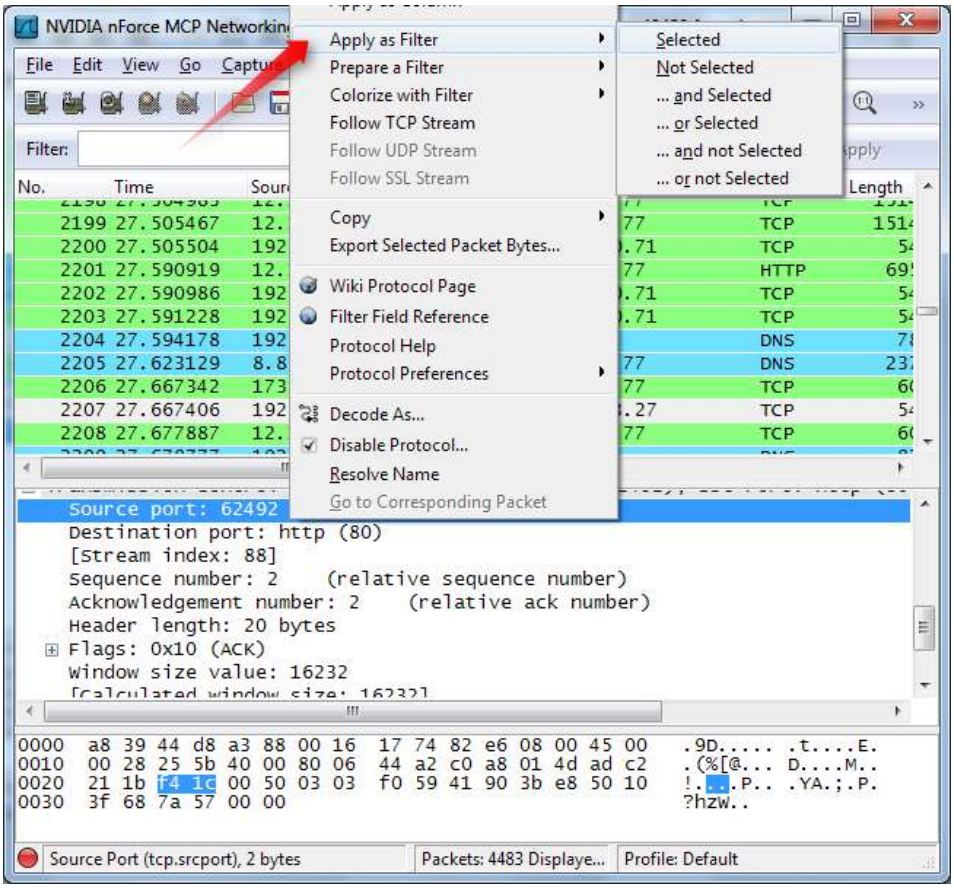


Inspecting Packets

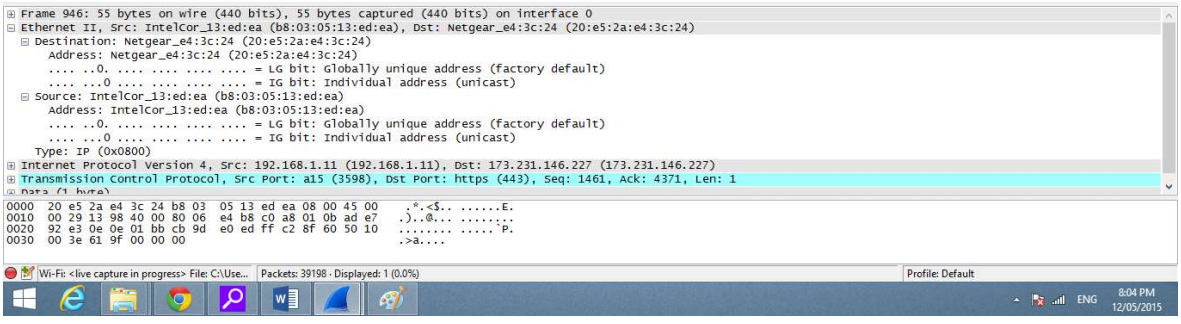
Click a packet to select it and you can dig down to view its details.



You can also create filters from here — just right-click one of the details and use the Apply as Filter submenu to create a filter based on it.

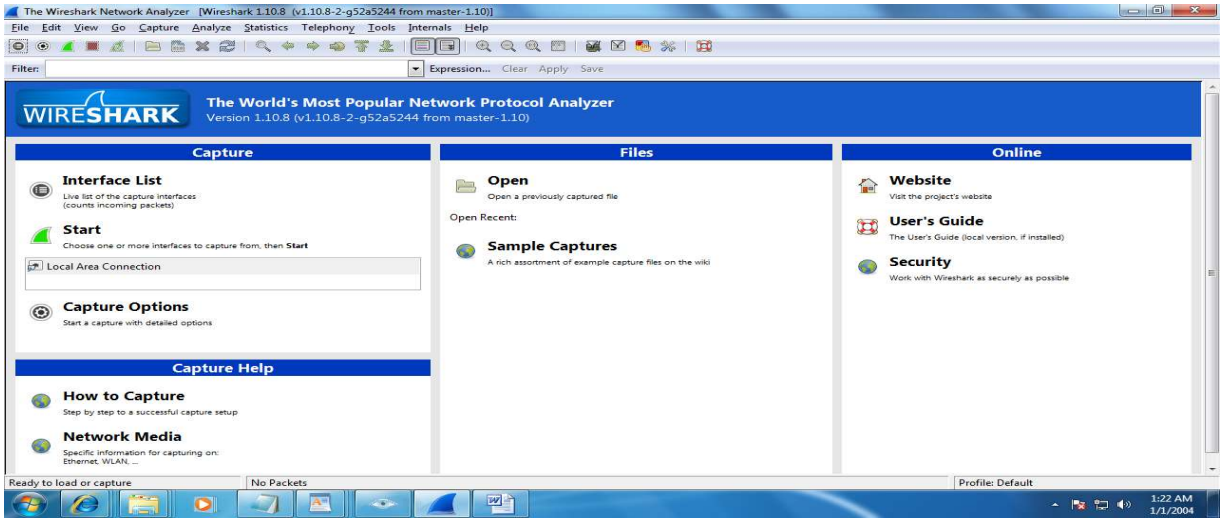

NVIDIA nForce MCP Networking
File Edit View Go Capture
Filter:
No. Time
Apply as Filter
Prepare a Filter
Colorize with Filter
Follow TCP Stream
Follow UDP Stream
Follow SSL Stream
Copy
Export Selected Packet Bytes...
Wiki Protocol Page
Filter Field Reference
Protocol Help
Protocol Preferences
Decode As...
Disable Protocol...
Resolve Name
Go to Corresponding Packet
Selected
Not Selected
... and Selected
... or Selected
... and not Selected
... or not Selected
Length
2199 27.505467 TCP 1514
2200 27.505504 TCP 54
2201 27.590919 HTTP 69
2202 27.590986 TCP 54
2203 27.591228 TCP 54
2204 27.594178 DNS
2205 27.623129 DNS
2206 27.667342 TCP
2207 27.667406 TCP 54
2208 27.677887 TCP
Source port: 62492
Destination port: http (80)
[Stream index: 88]
Sequence number: 2 (relative sequence number)
Acknowledgement number: 2 (relative ack number)
Header length: 20 bytes
Flags: 0x10 (ACK)
Window size value: 16232
0000 a8 39 44 d8 a3 88 00 16 17 74 82 e6 08 00 45 00 .9D..... .t....E.
0010 00 28 25 5b 40 00 80 06 44 a2 c0 a8 01 4d ad c2 .(%[@... D....M..
0020 21 1b f4 1c 00 50 03 03 f0 59 41 90 3b e8 50 10 !....P.. .YA.;.P.
0030 3f 68 7a 57 00 00 ?hzW..
Source Port (tcp.srcport), 2 bytes
Packets: 4483 Displaye...
Profile: Default


Capturing from Wi-Fi [Wireshark 1.10.8 (v1.10.8-2-g52a5244 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: frame.time_relative == 130.606445000
Expression... Clear Apply Save
No. Time Source Destination Protocol Length Info
946 130.606445 192.168.1.11 173.231.146.227 TCP 55 [TCP Keep-Alive] a15 > https [ACK] Seq=1461 Ack=4371 Win=15872 Len=1


Frame 946: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface 0
Ethernet II, Src: IntelCor_13:ed:ea (b8:03:05:13:ed:ea), Dst: Netgear_e4:3c:24 (20:e5:2a:e4:3c:24)
Destination: Netgear_e4:3c:24 (20:e5:2a:e4:3c:24)
Address: Netgear_e4:3c:24 (20:e5:2a:e4:3c:24)
.... ..0. .... = LG bit: Globally unique address (factory default)
....0 .... = IG bit: Individual address (unicast)
Source: IntelCor_13:ed:ea (b8:03:05:13:ed:ea)
Address: IntelCor_13:ed:ea (b8:03:05:13:ed:ea)
.... ..0. .... = LG bit: Globally unique address (factory default)
....0 .... = IG bit: Individual address (unicast)
Type: IP (0x0800)
Internet Protocol Version 4, Src: 192.168.1.11 (192.168.1.11), Dst: 173.231.146.227 (173.231.146.227)
Transmission Control Protocol, Src Port: a15 (3598), Dst Port: https (443), Seq: 1461, Ack: 4371, Len: 1
0000 20 e5 2a e4 3c 24 b8 03 05 13 ed ea 08 00 45 00 .*.<$.. ......E.
0010 00 29 13 98 40 00 80 06 e4 b8 c0 a8 01 0b ad e7 .)..@... ....
0020 92 e3 0e 0e 01 bb cb 9d e0 ed ff c2 8f 60 50 10 ....`P.
0030 00 3e 61 9f 00 00 00 .>a....
Wi-Fi: <live capture in progress> File: C:\Use...
Packets: 39198 · Displayed: 1 (0.0%)
Profile: Default
ENG
8:04 PM
12/05/2015

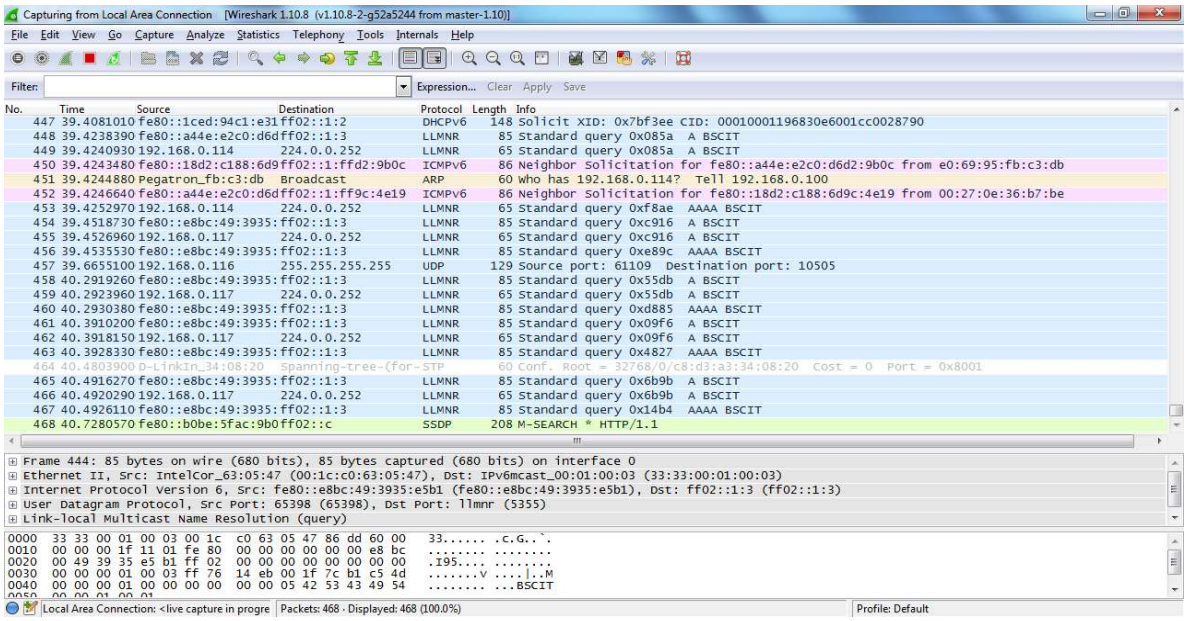
Using Traffic Capturing and Analysis tools

Aim: Exploring Wireshark

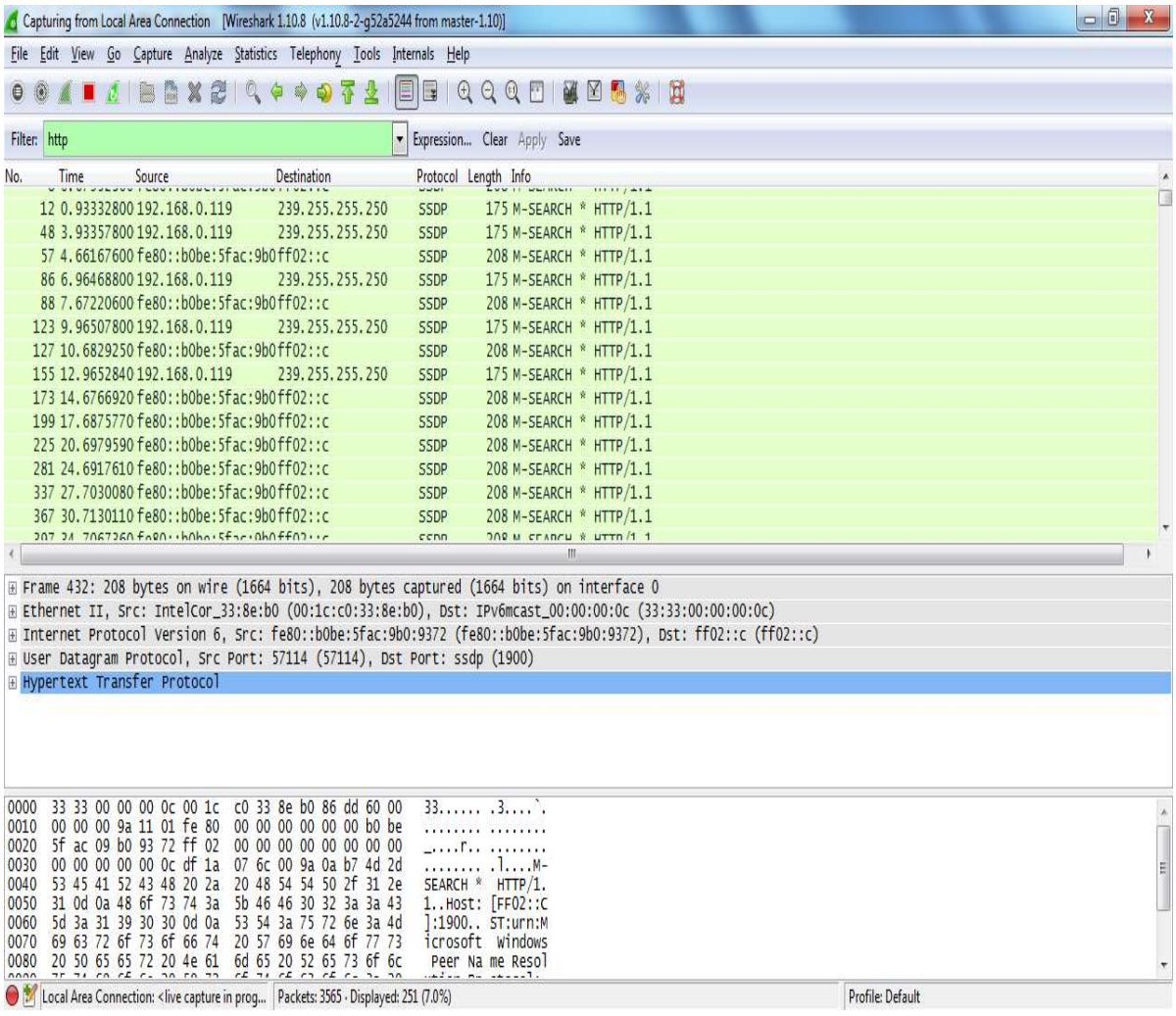


Step 2: On menu bar select Capture. Select interfaces.

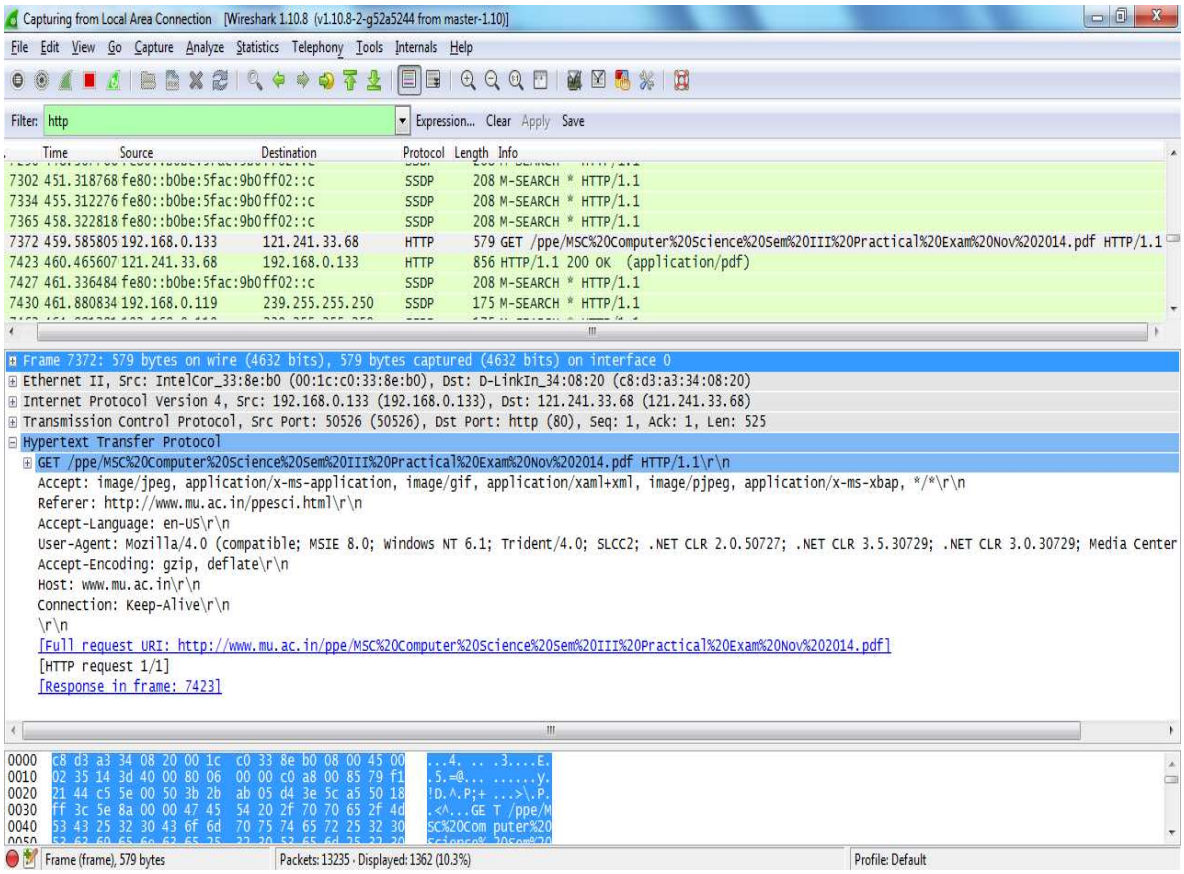
Step 3: Select Once you click on start, then Wireshark starts to capture the packets on that interface.



Step 4: Filter packets with HTTP protocol.

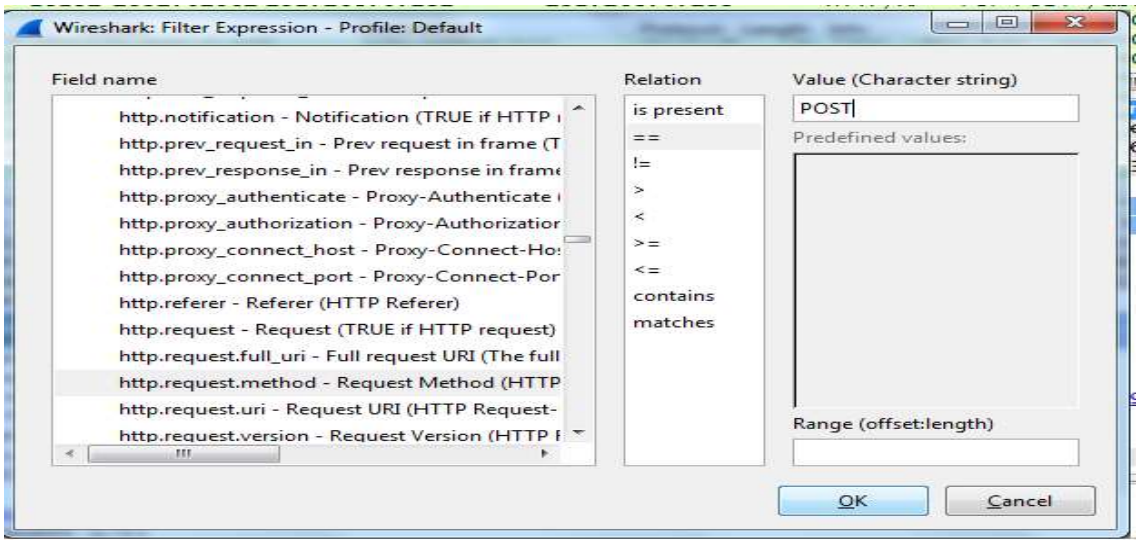


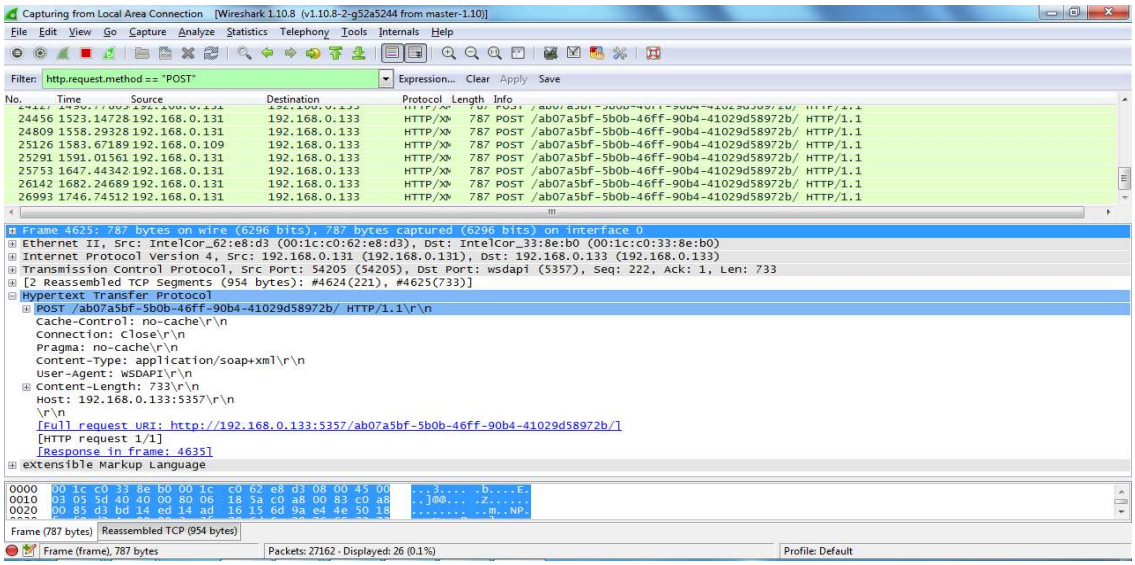
Step 5: A file with only text: http://www.mu.ac.in/ppe/LIBRARY%20SCIENCE-(SEM.I)-SH-2014.pdf



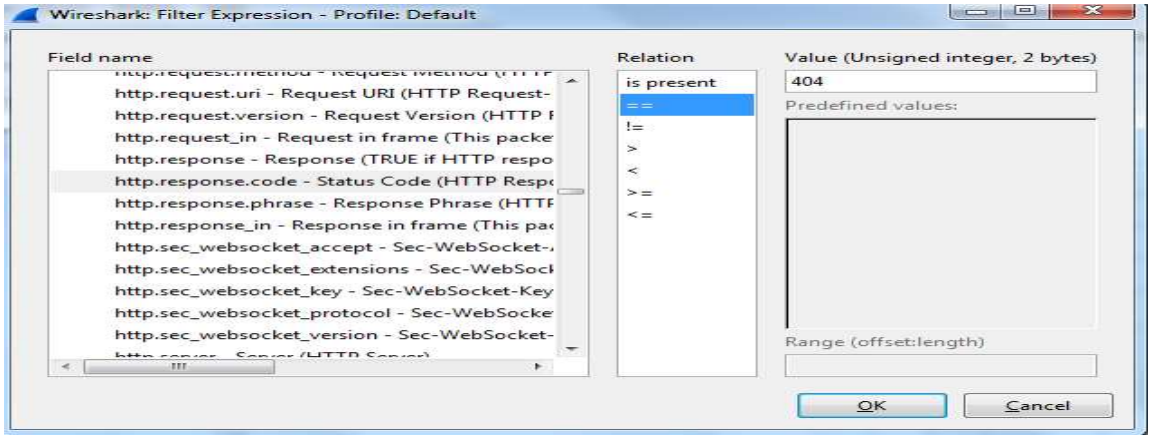
Step 6: Applying different filters using expressions.

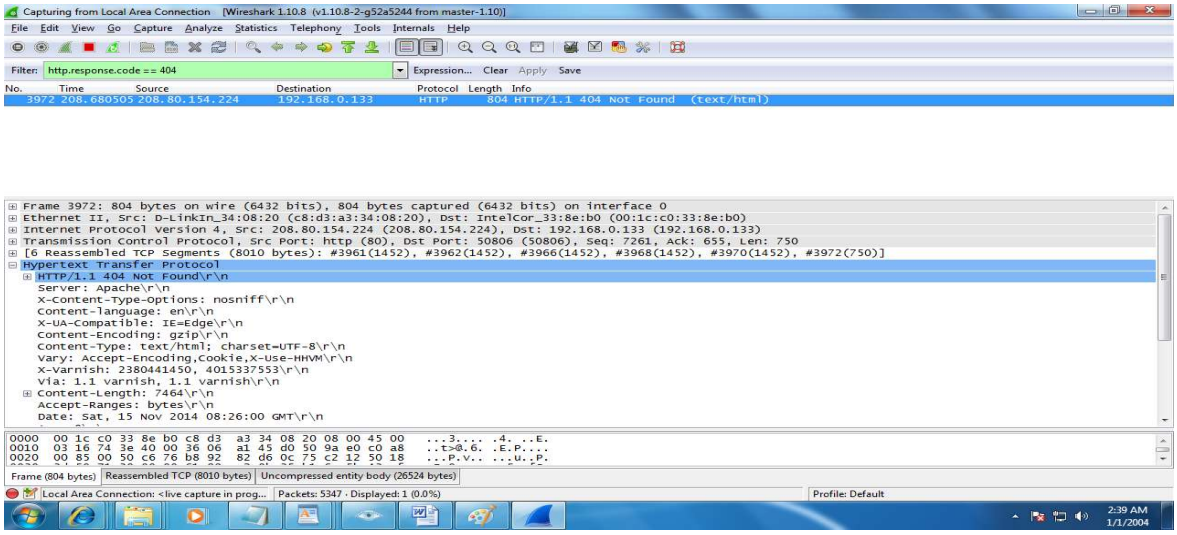
1) Filtering HTTP POST request





2) Filtering 404 not found error





3) Filtering using HTTP Content Length