

TABLE OF CONTENTS

Sr. No.	Topic	Date	Page No.	Remark
1.	Performing Foot printing for website Using ‘Internet Archive’ (WayBackMachine)			
2.	Performing Foot printing for website Using ‘Whois.com’ (To Fetch DNS Information).			
3.	Using ‘NS lookup’ to perform foot printing.			
4.	Using ‘ping’ to perform foot printing.			
5.	Using ‘Nmap’ to perform scan for ports.			
6.	Using ‘Nmap’ to perform scan for network.			
7.	Perform the use of IDS (Intrusion Detection Systems) Tool “Snort”			
8.	Network sniffing using Wireshark:			
9.	Using CrypTool:			
10.	Using Traceroute, ping, ipconfig, netstat Command			

FOOT PRINTING AND RECONNAISSANCE

- **Foot printing:**

- It means gathering information about a target system that can be executed to perform cyber-attack.
- Foot printing sometimes it's also called Reconnaissance.
- For this method hackers might use different methods or different tools.
- This is simple method for hackers to know the information about the system and devices or network.
- Types of Footprints:
 - **Active Foot printing:** It means performing foot printing by getting indirect touch with target machine.
 - **Passive Foot printing:** It means collecting information about a system located at remote distance from the attacker.
- These are information gathered from foot printing:
 - Operating System from target machine.
 - IP address.
 - Firewall
 - Network Map
 - Security configurations of the target machine
 - Email ID
 - Password
 - Server Configuration
 - URL's (Uniform Resource Locator)
 - VPN (Virtual Private Network)
- From different resources we do foot printing
 - Search Engine
 - Website
 - Social Engineering
 - DNS
 - Email Tracking
 - Social media
- Advantages of Foot printing:
 - It allows hackers to gather the basic security configurations of target machine.
 - It is best method of vulnerabilities.
 - By using this hacker identify as to which attacker is handier to hack the target system.

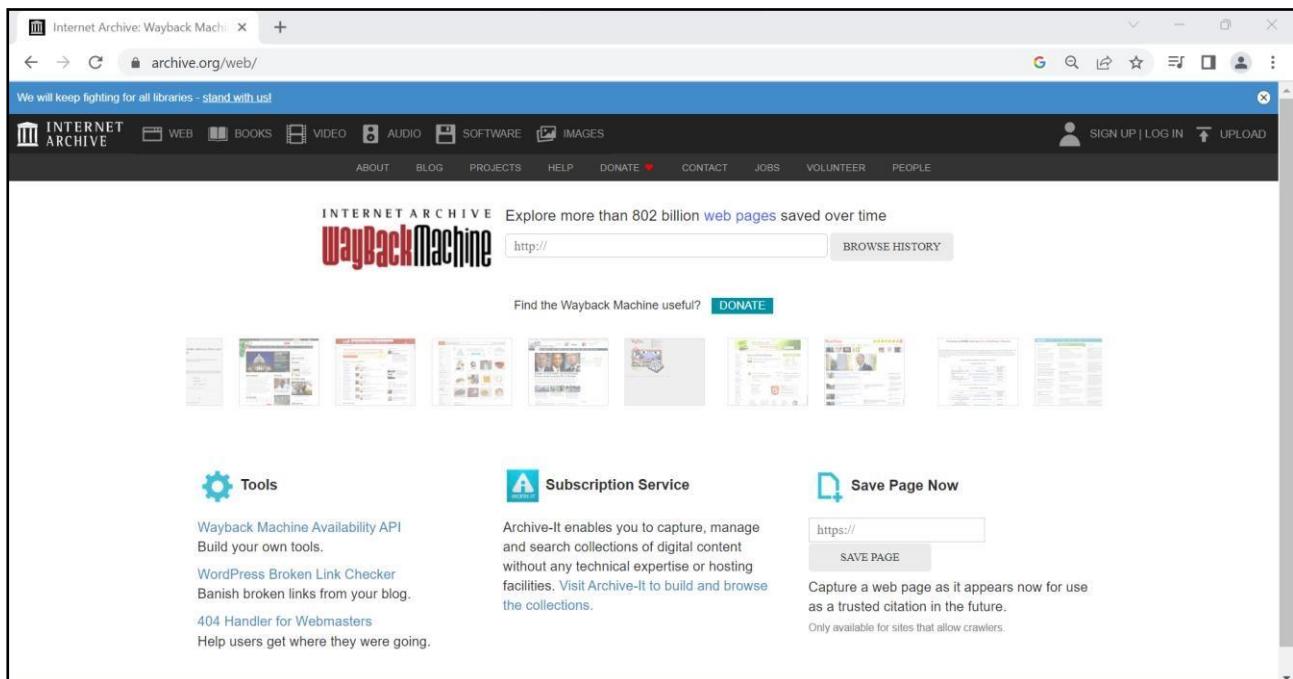
Practical work: Performing Foot printing for website

- ***Website foot printing:*** It is a technique which is used to extract the details related to website.

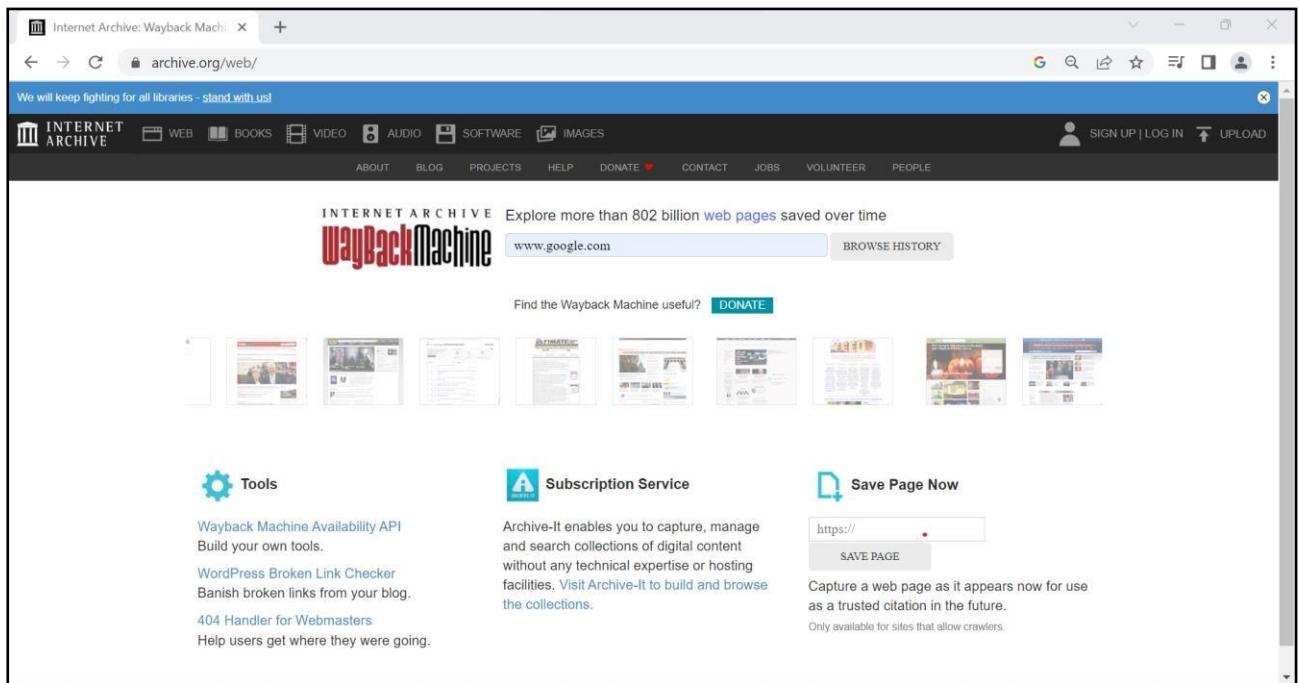
1. Using ‘Internet Archive’ (WayBackMachine) to perform foot printing.

- When hacker or any user wants details of archived website or history of website, they can use <https://archive.org/>
- ‘archive.org’ is the online tool which allows us to archived version of website.
- It refers to the older version of the website which is existed a time before and changed one.
- ‘archive.org’ is the website that collect all snapshots of all the websites at all the regular interval of the time.

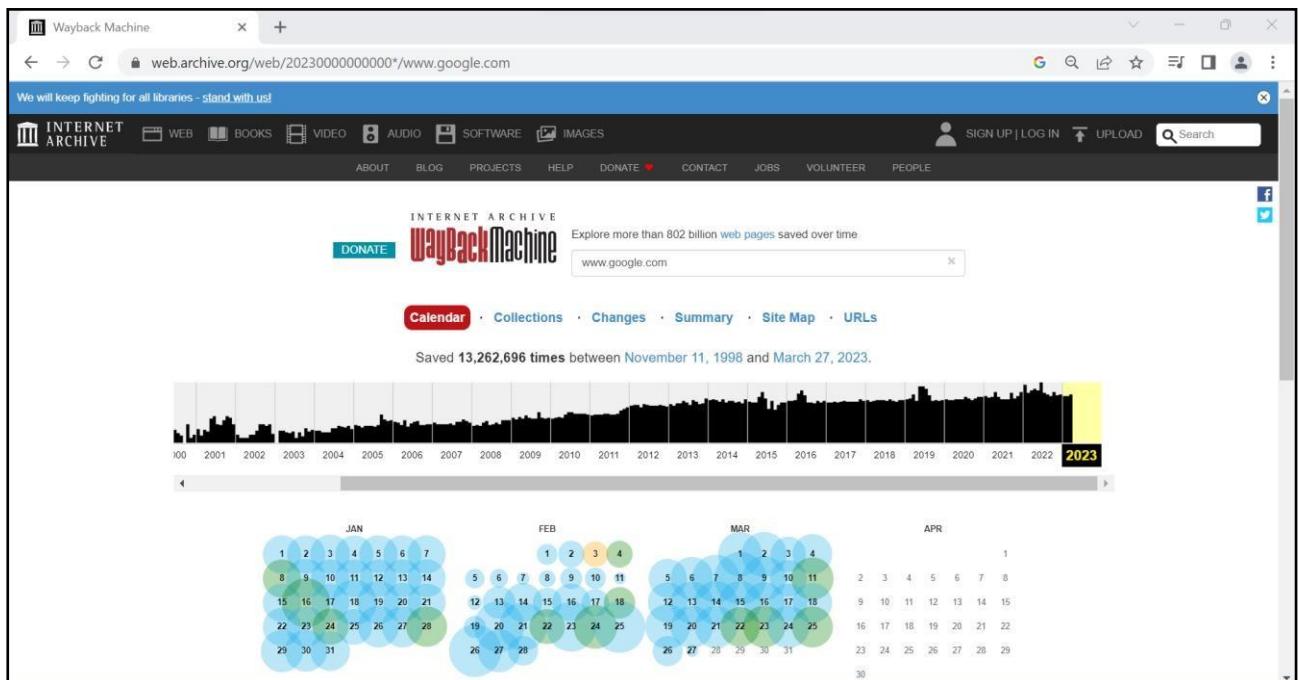
Step 1: Search for ‘archive.org’ on internet.



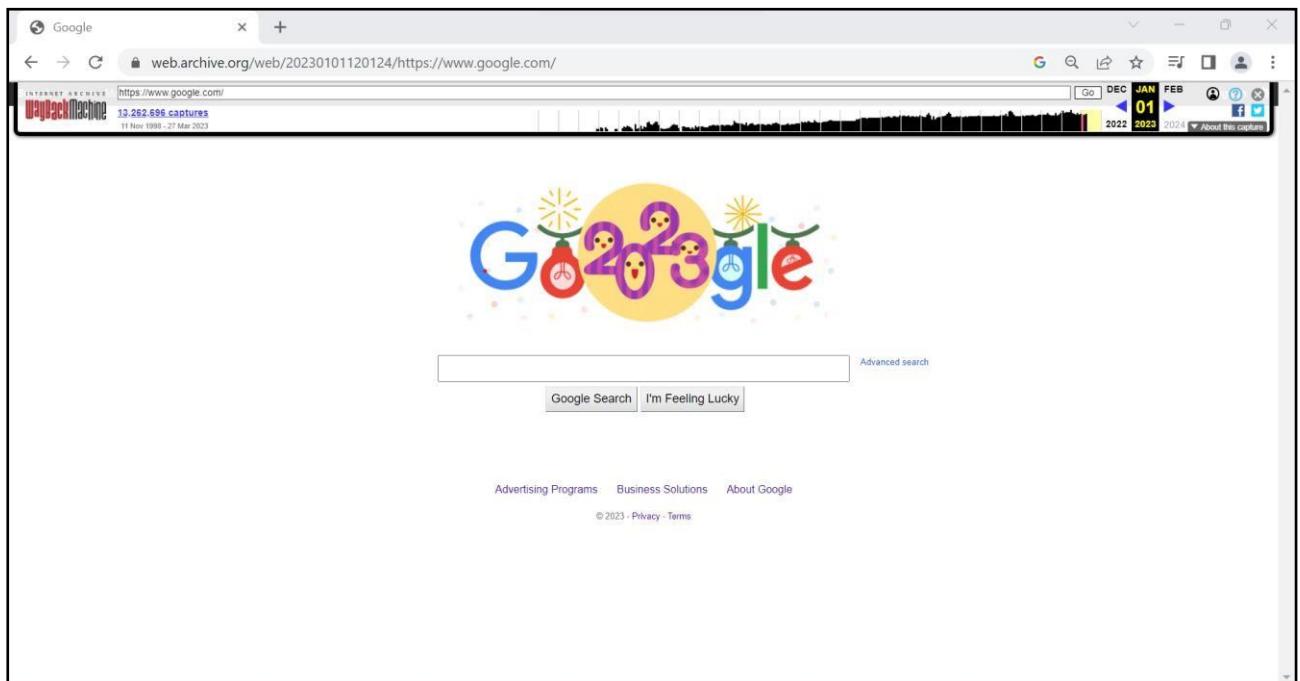
Step 2: Enter the ‘URL’ or ‘Domain Name’ to be searched. Here we are getting information regarding ‘www.google.com’.



Step 3: After hitting the ‘search button’ it shows the calendar representing the snapshots data. Hover or select any date and then select the time to get the view of that website at the selected date and time. Here we are searching for ‘1st January 2023’.



Step 4: Following screen shows the website of google on 1st of January 2023.



2. Using 'Whois.com' to perform foot printing (To Fetch DNS Information).

- DNS means Domain Name System. It is a system which allows us to convert Computer IP address into human readable domain name.
- A 'Whois' domain lookup allows you to trace the ownership and tenure of a domain name.
- The 'Whois' database contains details such as the registration date of the domain name, when it expires, ownership and contact information, nameserver information of the domain, the registrar via which the domain was purchased, etc.

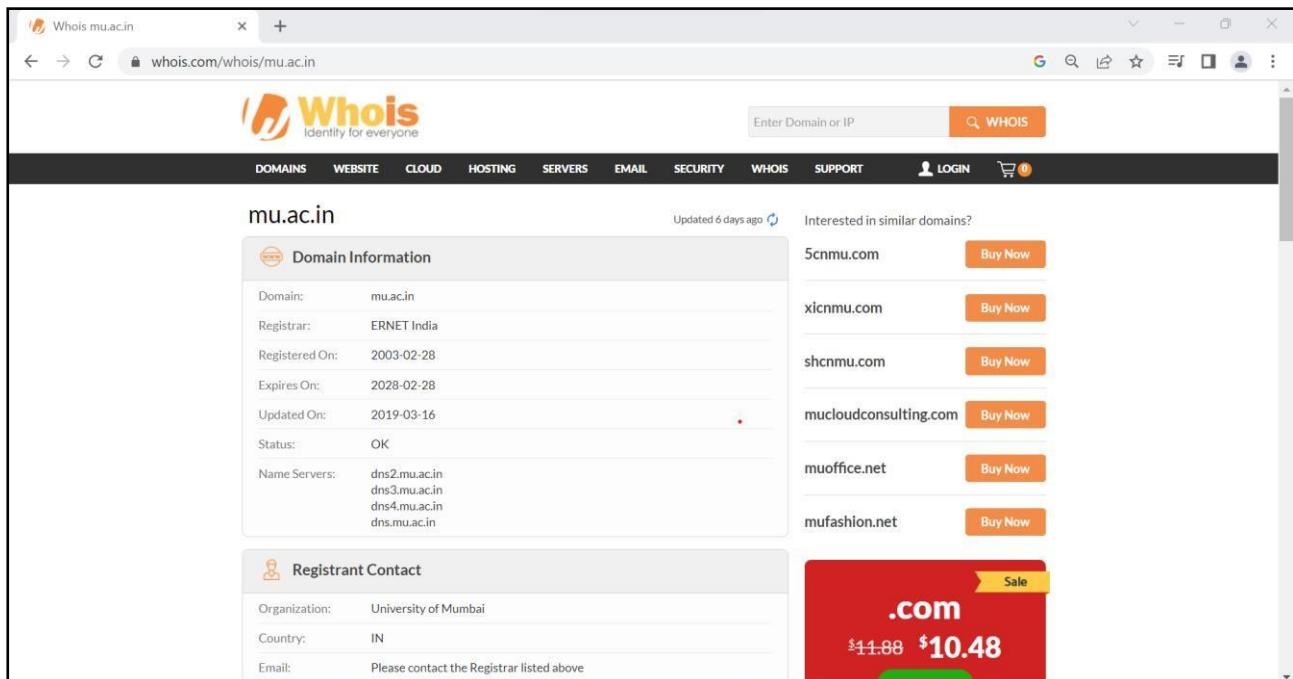
Step 1: Search for '<https://www.whois.com/>' on the internet. The following page will appear.



Step 2: Search for the desired domain name for which you need the information. Here we are using ‘mu.ac.in’.



Step 3: It displays all the details for the domain name ‘mu.ac.in’. Here it’s showing the “Domain Information” and “Registrant Contact” and some other details.



INSTITUTE OF DISTANCE AND OPEN LEARNING (IDOL) , UNIVERSITY OF MUMBAI

COURSE: S.Y.MCA (2YRS) SUBJECT: ETHICAL HACKING

The screenshot shows a web browser window for the WHOIS search of the domain `mu.ac.in`. The main content area displays the following sections:

- Administrative Contact:** Email: Please contact the Registrar listed above.
- Technical Contact:** Email: Please contact the Registrar listed above.
- Raw Whois Data:** This section contains a large amount of redacted WHOIS data. Key visible fields include:
 - Domain Name: `mu.ac.in`
 - Registry Domain ID: D12825-IN
 - Registrar WHOIS Server: <http://www.ernet.in>
 - Updated Date: 2019-03-16T09:41:18Z
 - Registry Update Date: 2003-02-28T05:00:00Z
 - Registry Expire Date: 2028-02-28T05:00:00Z
 - Registrar: ERNET India
 - Registrar IANA ID: 890068
 - Registrar Abuse Contact Email: abuse@mu.ac.in
 - Registrar Abuse Contact Phone: +91 22 2742 6400
 - Domain Status: ok <http://www.icann.org/epp#OK>
 - Registry Registrant ID: REDACTED FOR PRIVACY
 - Registrant Name: REDACTED FOR PRIVACY
 - Registrant Organization: University of Mumbai
 - Registrant Street: REDACTED FOR PRIVACY
 - Registrant Street2: REDACTED FOR PRIVACY
 - Registrant City: REDACTED FOR PRIVACY
 - Registrant State/Province: REDACTED FOR PRIVACY
 - Registrant Postal Code: REDACTED FOR PRIVACY
 - Registrant Country: IN
 - Registrant Phone: REDACTED FOR PRIVACY
 - Registrant Phone Ext: REDACTED FOR PRIVACY
 - Registrant Fax: REDACTED FOR PRIVACY
 - Registrant Fax Ext: REDACTED FOR PRIVACY
 - Registrant Email: Please contact the Registrar listed above
 - Admin Admin: REDACTED FOR PRIVACY
 - Admin Name: REDACTED FOR PRIVACY
 - Admin Organization: REDACTED FOR PRIVACY

On the right side of the browser window, there are two promotional banners:

- .site**: An offer for ".SITE @ \$3.98" with a "VIEW MORE" button.
- WORDPRESS HOSTING**: An offer for "\$3.58/mo" with a "VIEW MORE" button.

This screenshot shows a detailed view of the WHOIS data for the domain `mu.ac.in`. The page is filled with numerous redacted fields, indicating privacy protection for the registrant and administrative contacts. Key visible details include:

- Registrant Fax: REDACTED FOR PRIVACY
- Registrant Fax Ext: REDACTED FOR PRIVACY
- Registrant Email: Please contact the Registrar listed above
- Registry Admin ID: REDACTED FOR PRIVACY
- Admin Name: REDACTED FOR PRIVACY
- Admin Street: REDACTED FOR PRIVACY
- Admin Street2: REDACTED FOR PRIVACY
- Admin City: REDACTED FOR PRIVACY
- Admin State/Province: REDACTED FOR PRIVACY
- Admin Postal Code: REDACTED FOR PRIVACY
- Admin Country: REDACTED FOR PRIVACY
- Admin Phone: REDACTED FOR PRIVACY
- Admin Phone Ext: REDACTED FOR PRIVACY
- Admin Admin: REDACTED FOR PRIVACY
- Admin Email: Please contact the Registrar listed above
- Registry Tech ID: REDACTED FOR PRIVACY
- Tech Name: REDACTED FOR PRIVACY
- Tech Organization: REDACTED FOR PRIVACY
- Tech Street: REDACTED FOR PRIVACY
- Tech Street2: REDACTED FOR PRIVACY
- Tech City: REDACTED FOR PRIVACY
- Tech State/Province: REDACTED FOR PRIVACY
- Tech Postal Code: REDACTED FOR PRIVACY
- Tech Country: REDACTED FOR PRIVACY
- Tech Phone: REDACTED FOR PRIVACY
- Tech Phone Ext: REDACTED FOR PRIVACY
- Tech Fax: REDACTED FOR PRIVACY
- Tech Fax Ext: REDACTED FOR PRIVACY
- Tech Email: Please contact the Registrar listed above
- Name Server: dns2.mu.ac.in
- Name Server: dns3.mu.ac.in
- Name Server: dns4.mu.ac.in
- Name Server: dns.mu.ac.in
- DNSSEC: unsigned

At the bottom of the page, there is a note about the ICANN Whois Inaccuracy Complaint Form and a link to the URL of the form. There is also a message about the last update of the WHOIS database and a note about the purpose of WHOIS information.

Data for 'www.google.com'

The screenshot shows a web browser window with two tabs: "Ethical-Hacking-Lab.pdf" and "Whois.google.com". The main content area displays the Whois information for the domain "google.com".

Domain Information:

- Domain: google.com
- Registrar: MarkMonitor Inc.
- Registered On: 1997-09-15
- Expires On: 2028-09-13
- Updated On: 2019-09-09
- Status: clientDeleteProhibited, clientTransferProhibited, clientUpdateProhibited, serverDeleteProhibited, serverTransferProhibited, serverUpdateProhibited
- Name Servers: ns1.google.com, ns2.google.com, ns3.google.com, ns4.google.com

Registrant Contact:

- Organization: Google LLC
- State: CA
- Country: US
- Email: Select Request Email Form at <https://domains.markmonitor.com/whois/google.com>

Administrative Contact:

- Organization: Google LLC
- State: CA
- Country: US
- Email: Select Request Email Form at <https://domains.markmonitor.com/whois/google.com>

Technical Contact:

- Organization: Google LLC
- State: CA
- Country: US
- Email: Select Request Email Form at <https://domains.markmonitor.com/whois/google.com>

On the right side of the page, there are promotional banners for ".com" and ".fun" domains, as well as an offer for "WORDPRESS HOSTING \$3.58/mo".

This screenshot shows the same Whois.google.com interface, but the contact details for Google LLC have been expanded.

Registrant Contact:

- Organization: Google LLC
- State: CA
- Country: US
- Email: Select Request Email Form at <https://domains.markmonitor.com/whois/google.com>

Administrative Contact:

- Organization: Google LLC
- State: CA
- Country: US
- Email: Select Request Email Form at <https://domains.markmonitor.com/whois/google.com>

Technical Contact:

- Organization: Google LLC
- State: CA
- Country: US
- Email: Select Request Email Form at <https://domains.markmonitor.com/whois/google.com>

The right side of the page features a prominent "WORDPRESS HOSTING \$3.58/mo" offer banner.

Raw Whois Data

Domain Name: google.com
Registry Domain ID: 2138514_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: <http://www.markmonitor.com>
Updated Date: 2019-09-09T15:39:04+0000
Creation Date: 1997-09-15T07:00:00+0000
Registrar Registration Expiration Date: 2028-09-13T07:00:00+0000
Registrar: MarkMonitor, Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: **abusecomplaints**@markmonitor.com
Registrar Abuse Contact Phone: +1.2086851750
Domain Status: clientUpdateProhibited
(<https://www.icann.org/epp#clientUpdateProhibited>)
Domain Status: clientTransferProhibited
(<https://www.icann.org/epp#clientTransferProhibited>)
Domain Status: clientDeleteProhibited
(<https://www.icann.org/epp#clientDeleteProhibited>)
Domain Status: serverUpdateProhibited
(<https://www.icann.org/epp#serverUpdateProhibited>)
Domain Status: serverTransferProhibited
(<https://www.icann.org/epp#serverTransferProhibited>)
Domain Status: serverDeleteProhibited
(<https://www.icann.org/epp#serverDeleteProhibited>)
Registrant Organization: Google LLC
Registrant State/Province: CA
Registrant Country: US
Registrant Email: Select Request Email Form at
<https://domains.markmonitor.com/whois/google.com>
Admin Organization: Google LLC
Admin State/Province: CA
Admin Country: US
Admin Email: Select Request Email Form at
<https://domains.markmonitor.com/whois/google.com>
Tech Organization: Google LLC
Tech State/Province: CA
Tech Country: US
Tech Email: Select Request Email Form at
<https://domains.markmonitor.com/whois/google.com>
Name Server: ns2.google.com
Name Server: ns4.google.com
Name Server: ns1.google.com
Name Server: ns3.google.com
DNSSEC: unsigned
URL of the ICANN WHOIS Data Problem Reporting System:
<http://wdprs.internic.net/>
>>> Last update of WHOIS database: 2023-03-20T08:47:05+0000 <<<

For more information on WHOIS status codes, please visit:
<https://www.icann.org/resources/pages/epp-status-codes>

If you wish to contact this domain's Registrant, Administrative, or Technical contact, and such email address is not visible above, you may do so via our web form, pursuant to ICANN's Temporary Specification. To verify that you are not a robot, please enter your email address to receive a link to a page that facilitates email communication with the relevant contact(s).

Web-based WHOIS:

<https://domains.markmonitor.com/whois>

If you have a legitimate interest in viewing the non-public WHOIS details, send your request and the reasons for your request to **whoisrequest**@markmonitor.com and specify the domain name in the subject line. We will review that request and may ask for supporting documentation and explanation.

The data in MarkMonitor's WHOIS database is provided for information purposes, and to assist persons in obtaining information about or related to a domain name's registration record. While MarkMonitor believes the data to be accurate, the data is provided "as is" with no guarantee or warranties regarding its accuracy.

By submitting a WHOIS query, you agree that you will use this data only for lawful purposes and that, under no circumstances will you use this data to:

- (1) allow, enable, or otherwise support the transmission by email, telephone, or facsimile of mass, unsolicited, commercial advertising, or spam; or
- (2) enable high volume, automated, or electronic processes that send queries, data, or email to MarkMonitor (or its systems) or the domain name contacts (or its systems).

MarkMonitor reserves the right to modify these terms at any time.

By submitting this query, you agree to abide by this policy.

MarkMonitor Domain Management (TM)

Protecting companies and consumers in a digital world.

Visit MarkMonitor at <https://www.markmonitor.com>

Contact us at +1.8007459229

In Europe, at +44.02032062220

--

3. Using 'NS lookup' to perform foot printing.

- Name server lookup (nslookup) is a command-line tool that lets you find the internet protocol (IP) address or domain name system (DNS) record of a specific hostname.
- This command also allows reverse DNS lookup by inputting the IP addresses of the corresponding domains.

a. Using nslookup command to get IP

In command prompt type nslookup and hit enter then type any of the desired domain name to get IP or write nslookup and domain name together.

Command: nslookup <any domain name>

Example: nslookup www.google.com

```
C:\>nslookup www.google.com
Server: Unknown
Address: 10.0.0.1

Non-authoritative answer:
Name: www.google.com
Addresses: 2404:6800:4009:822::2004
           142.250.67.228
```

```
C:\>nslookup
Default Server: Unknown
Address: 10.0.0.1

> www.google.com
Server: Unknown
Address: 10.0.0.1

Non-authoritative answer:
Name: www.google.com
Addresses: 2404:6800:4009:814::2004
           142.250.67.228

> mu.ac.in
Server: Unknown
Address: 10.0.0.1

Non-authoritative answer:
Name: mu.ac.in
Addresses: 121.241.25.2
           121.241.25.1
           14.139.125.195

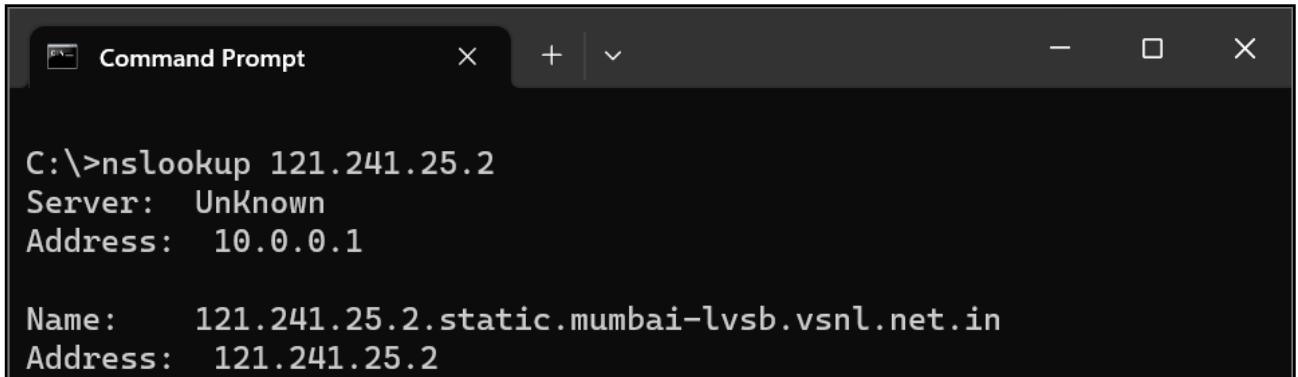
> whois.com
Server: Unknown
Address: 10.0.0.1

Non-authoritative answer:
Name: whois.com
Address: 64.91.226.82
```

b. Using nslookup command in a reverse manner.

In this case we get the name from the IP.

Command: nslookup <any ip address>
Example: nslookup 121.241.25.2 (mu.ac.in)



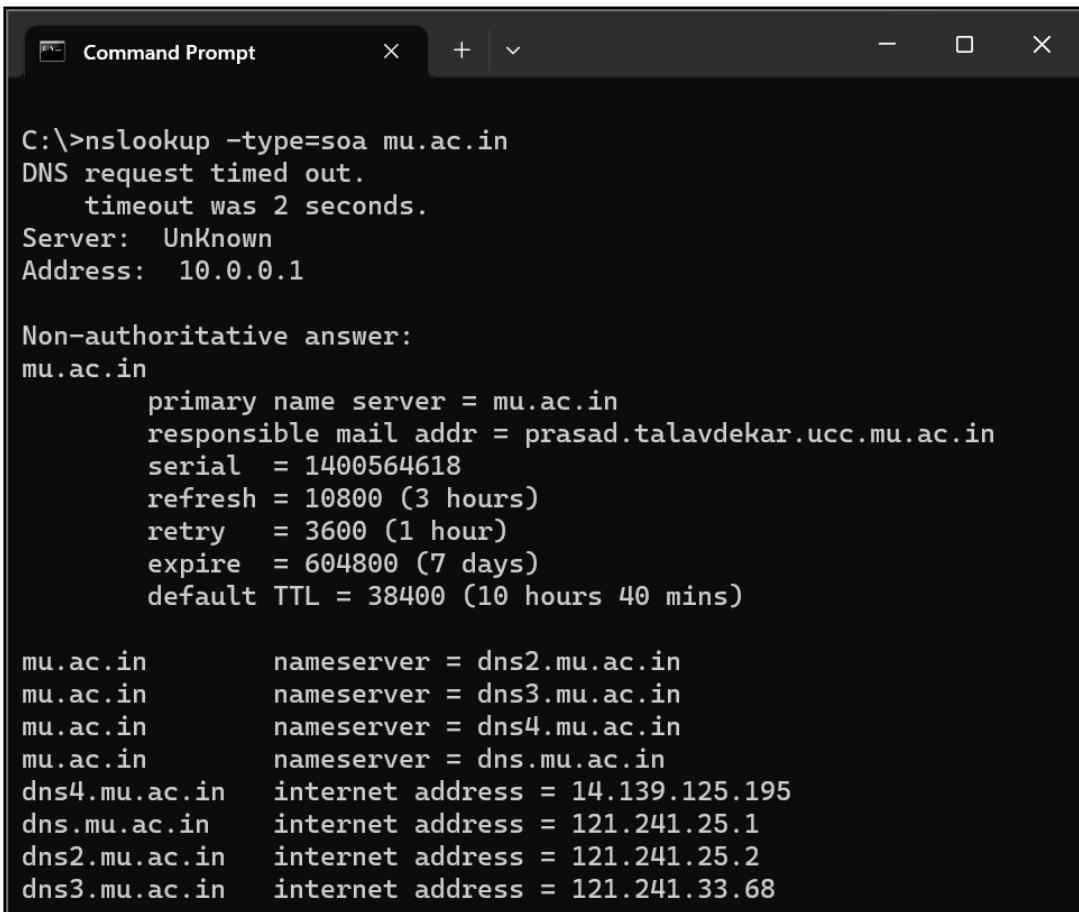
```
C:\>nslookup 121.241.25.2
Server: Unknown
Address: 10.0.0.1

Name: 121.241.25.2.static.mumbai-lvsl.vsnl.net.in
Address: 121.241.25.2
```

c. Using nslookup command to get important information regarding the domain.

This will query the standard of authority (SOA) record containing important information about the specified domain. For example, you want to get an authoritative response for the domain mu.ac.in:

Command: nslookup -type=soa domainname.tld
Example: nslookup -type=soa mu.ac.in



```
C:\>nslookup -type=soa mu.ac.in
DNS request timed out.
    timeout was 2 seconds.
Server: Unknown
Address: 10.0.0.1

Non-authoritative answer:
mu.ac.in
    primary name server = mu.ac.in
    responsible mail addr = prasad.talavdekar.ucc.mu.ac.in
    serial = 1400564618
    refresh = 10800 (3 hours)
    retry = 3600 (1 hour)
    expire = 604800 (7 days)
    default TTL = 38400 (10 hours 40 mins)

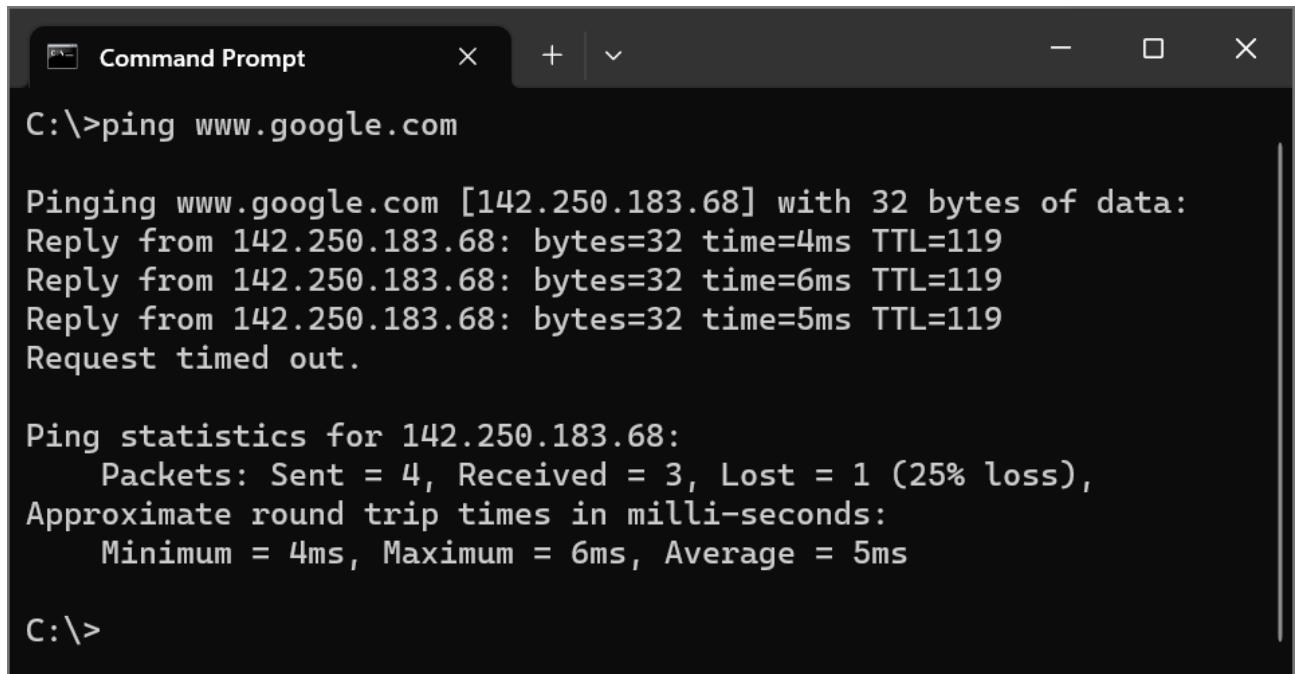
mu.ac.in      nameserver = dns2.mu.ac.in
mu.ac.in      nameserver = dns3.mu.ac.in
mu.ac.in      nameserver = dns4.mu.ac.in
mu.ac.in      nameserver = dns5.mu.ac.in
dns4.mu.ac.in internet address = 14.139.125.195
dns5.mu.ac.in internet address = 121.241.25.1
dns2.mu.ac.in internet address = 121.241.25.2
dns3.mu.ac.in internet address = 121.241.33.68
```

4. Using 'ping' to perform foot printing.

- Ping is a command-line utility, available on virtually any operating system with network connectivity, that acts as a test to see if a networked device is reachable.
- The ping command sends a request over the network to a specific device. A successful ping results in a response from the computer that was pinged back to the originating computer.

Command: ping <domain name>

Example: ping www.google.com or ping mu.ac.in

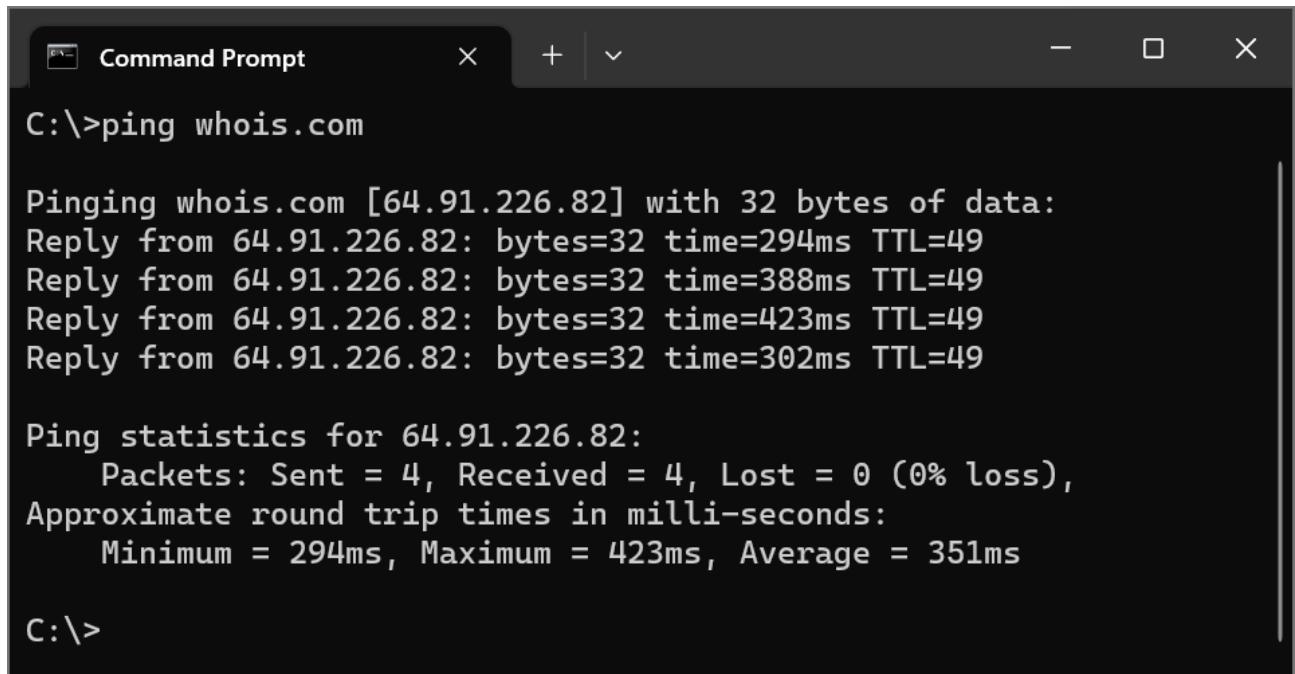


```
Command Prompt
C:\>ping www.google.com

Pinging www.google.com [142.250.183.68] with 32 bytes of data:
Reply from 142.250.183.68: bytes=32 time=4ms TTL=119
Reply from 142.250.183.68: bytes=32 time=6ms TTL=119
Reply from 142.250.183.68: bytes=32 time=5ms TTL=119
Request timed out.

Ping statistics for 142.250.183.68:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 4ms, Maximum = 6ms, Average = 5ms

C:\>
```



```
Command Prompt
C:\>ping whois.com

Pinging whois.com [64.91.226.82] with 32 bytes of data:
Reply from 64.91.226.82: bytes=32 time=294ms TTL=49
Reply from 64.91.226.82: bytes=32 time=388ms TTL=49
Reply from 64.91.226.82: bytes=32 time=423ms TTL=49
Reply from 64.91.226.82: bytes=32 time=302ms TTL=49

Ping statistics for 64.91.226.82:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 294ms, Maximum = 423ms, Average = 351ms

C:\>
```

SCANNING NETWORKS, ENUMERATION AND SNIFFING

- **Port Scanning:**

- A port is a virtual location where networking communication starts and ends (in a nutshell).
- A port scanner is a computer program that examines network ports for one of three possible condition – open, closed, or filtered.
- Port scanning can provide information such as:
 - Services that are running.
 - Users who own services.
 - Whether unknown logins are allowed.
 - Which network services require authentication.
- Port scanners are valuable tools in diagnosing network and connectivity issues.
- However, attackers use port scanners to detect possible access points for intrusion and to identify what kinds of devices you are running on the network, like firewalls, proxy servers or VPN servers.
- Some of the Port Scanning Tools are as follows:
 - Nmap
 - SolarWinds Port Scanner
 - Netcat
 - Advanced Port Scanner
 - Net Scan Tools

- **Nmap Tool**

- Nmap is a free, open source and multi-platform network security scanner used for network discovery and security auditing.
- Nmap can be extremely useful for helping you get to the root of the problem you are investigating, verify firewall rules or validate your routing tables are configured correctly.

1. Using 'Nmap' to perform scan for ports.

- i. Scan open ports: This functionality is used to scan for open ports.

Command: nmap -open <ip address / URL>

Example: nmap -open mu.ac.in

```
C:\>nmap -open mu.ac.in
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-27 12:06 India Standard Time
Nmap scan report for mu.ac.in (14.139.125.195)
Host is up (0.043s latency).
Other addresses for mu.ac.in (not scanned): 121.241.25.1 121.241.25.2
Not shown: 997 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 13.32 seconds
```

Example: nmap -open 142.250.183.68

```
C:\>nmap -open 142.250.183.68
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-27 12:10 India Standard Time
Nmap scan report for bom12s12-in-f4.1e100.net (142.250.183.68)
Host is up (0.072s latency).
Not shown: 998 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 45.32 seconds
```

- ii. Scan single ports: This functionality is used to scan for specified port.

Command: nmap -p <port number> <ip address / URL>

Example: nmap -p 80 www.google.com

```
C:\>nmap -p 80 www.google.com
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-27 12:13 India Standard Time
Nmap scan report for www.google.com (142.250.67.228)
Host is up (0.0079s latency).
rDNS record for 142.250.67.228: bom07s24-in-f4.1e100.net

PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 1.13 seconds
```

Example: nmap -p 80 www.google.com

```
C:\>nmap -p 80 mu.ac.in
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-27 12:13 India Standard Time
Nmap scan report for mu.ac.in (14.139.125.195)
Host is up (0.011s latency).
Other addresses for mu.ac.in (not scanned): 121.241.25.2 121.241.25.1

PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 1.16 seconds
```

iii. Scan specified range of ports: This functionality scans the specified range of ports.

Command: nmap -p <port range> <ip address / URL>
Example: nmap -p 1-200 www.google.com

```
C:\>nmap -p 1-200 www.google.com
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-27 12:15 India Standard Time
Nmap scan report for www.google.com (142.250.67.228)
Host is up (0.0091s latency).
rDNS record for 142.250.67.228: bom07s24-in-f4.1e100.net
Not shown: 199 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 3.16 seconds
```

Example: nmap -p 1-200 mu.ac.in

```
C:\>nmap -p 1-200 mu.ac.in
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-27 12:16 India Standard Time
Nmap scan report for mu.ac.in (14.139.125.195)
Host is up (0.015s latency).
Other addresses for mu.ac.in (not scanned): 121.241.25.2 121.241.25.1
Not shown: 198 filtered tcp ports (no-response)
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 10.16 seconds
```

iv. Scan entire range of ports: This functionality scans the entire range of ports.

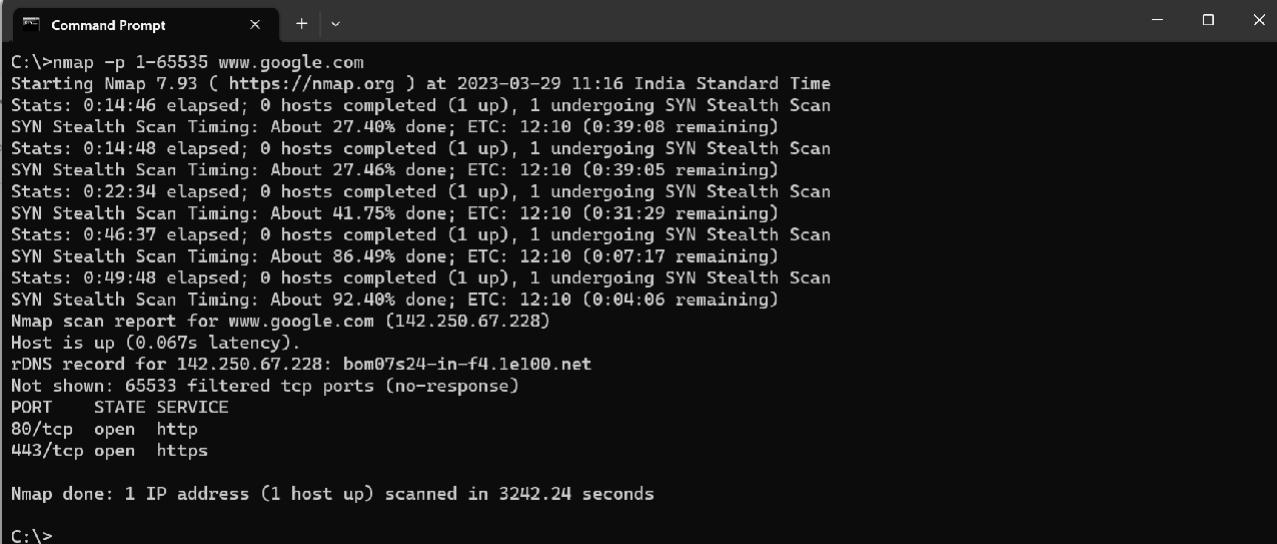
Command: nmap -p <port range> <ip address / URL>
Example: nmap -p 1-65535 mu.ac.in

```
C:\>nmap -p 1-65535 mu.ac.in
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-27 12:19 India Standard Time
Stats: 0:07:45 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 19.22% done; ETC: 13:00 (0:32:34 remaining)
Stats: 0:07:45 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 19.27% done; ETC: 12:59 (0:32:28 remaining)
Stats: 0:07:45 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 19.31% done; ETC: 12:59 (0:32:23 remaining)
Stats: 0:07:46 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 19.42% done; ETC: 12:59 (0:32:13 remaining)
Stats: 0:07:46 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 19.43% done; ETC: 12:59 (0:32:12 remaining)
Stats: 0:07:46 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 19.44% done; ETC: 12:59 (0:32:11 remaining)

Stats: 0:18:16 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 56.43% done; ETC: 12:52 (0:14:06 remaining)
Nmap scan report for mu.ac.in (121.241.25.1)
Host is up (0.046s latency).
Other addresses for mu.ac.in (not scanned): 14.139.125.195 121.241.25.2
rDNS record for 121.241.25.1: 121.241.25.1.static.mumbai-lvsl.vsnl.net.in
Not shown: 65532 filtered tcp ports (no-response)
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 1617.97 seconds
```

Example: nmap -p 1-65535 www.google.com



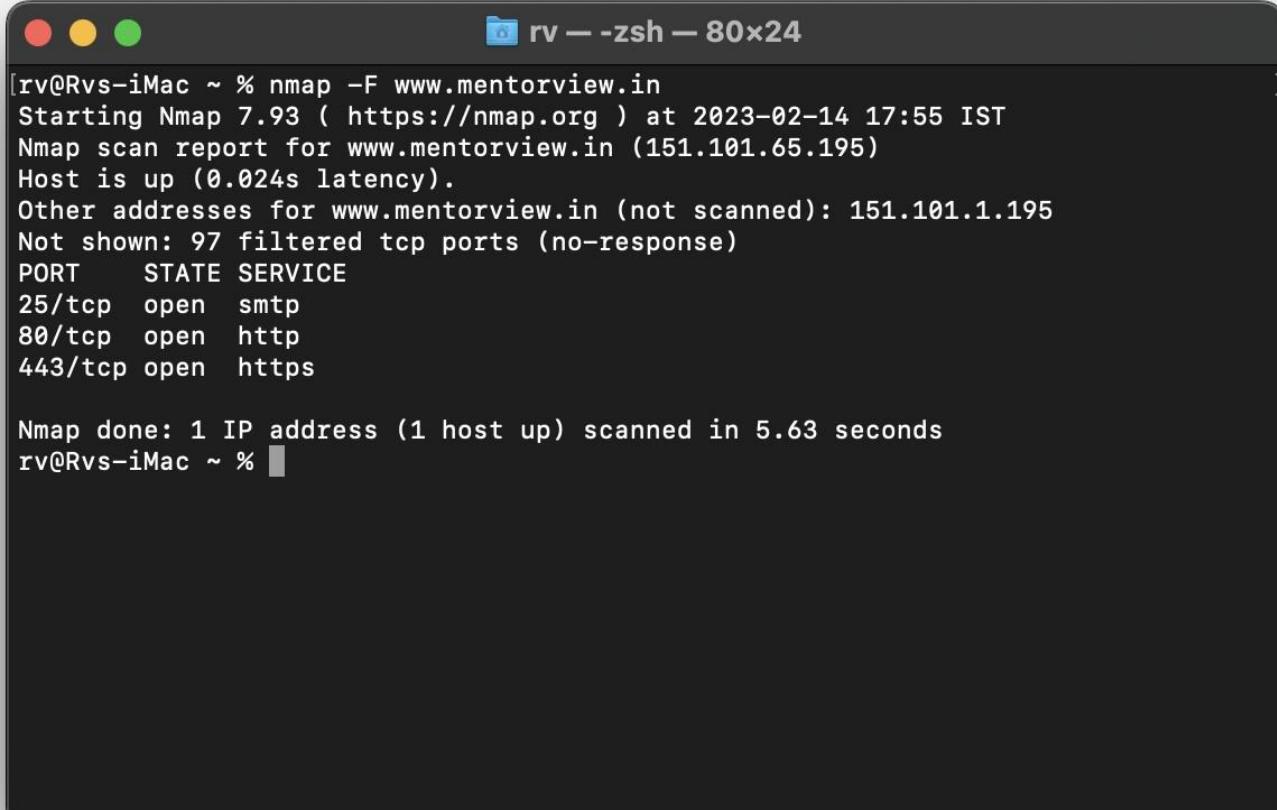
```
C:\>nmap -p 1-65535 www.google.com
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-29 11:16 India Standard Time
Stats: 0:14:46 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 27.40% done; ETC: 12:10 (0:39:08 remaining)
Stats: 0:14:48 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 27.46% done; ETC: 12:10 (0:39:05 remaining)
Stats: 0:22:34 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 41.75% done; ETC: 12:10 (0:31:29 remaining)
Stats: 0:46:37 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 86.49% done; ETC: 12:10 (0:07:17 remaining)
Stats: 0:49:48 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 92.40% done; ETC: 12:10 (0:04:06 remaining)
Nmap scan report for www.google.com (142.250.67.228)
Host is up (0.067s latency).
rDNS record for 142.250.67.228: bom07s24-in-f4.1e100.net
Not shown: 65533 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 3242.24 seconds
C:\>
```

- v. Scan top 100 ports (fast scan): This functionality scans the top 100 ports and its done very fast.

Command: nmap -F <ip address / URL>

Example: nmap -F www.mentorview.in



```
[rv@Rvs-iMac ~ % nmap -F www.mentorview.in
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-14 17:55 IST
Nmap scan report for www.mentorview.in (151.101.65.195)
Host is up (0.024s latency).
Other addresses for www.mentorview.in (not scanned): 151.101.1.195
Not shown: 97 filtered tcp ports (no-response)
PORT      STATE SERVICE
25/tcp    open  smtp
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 5.63 seconds
rv@Rvs-iMac ~ %
```

• **Network Scanning:**

- Network scanning is a technique that is used to gather information regarding computing systems by making the use of a computer network.
- Network scanning is mainly used for security assessment, system maintenance, and also for performing attacks by hackers.
- The purpose of network scanning is as follows:
 - Recognize available UDP and TCP network services running on the targeted hosts.
 - Recognize filtering systems between the user and the targeted hosts.
 - Determine the operating systems (OSs) in use by assessing IP responses.
 - Evaluate the target host's TCP sequence number predictability to determine sequence prediction attack and TCP spoofing.
- Some of the Top Network Scanning Tools (IP and Network Scanner) are as follows:
 - SolarWinds Network Device Scanner
 - Wireshark
 - Angry IP Scanner
 - Advanced IP Scanner
 - Qualys Freescan
 - Nmap

• **Nmap Tool:**

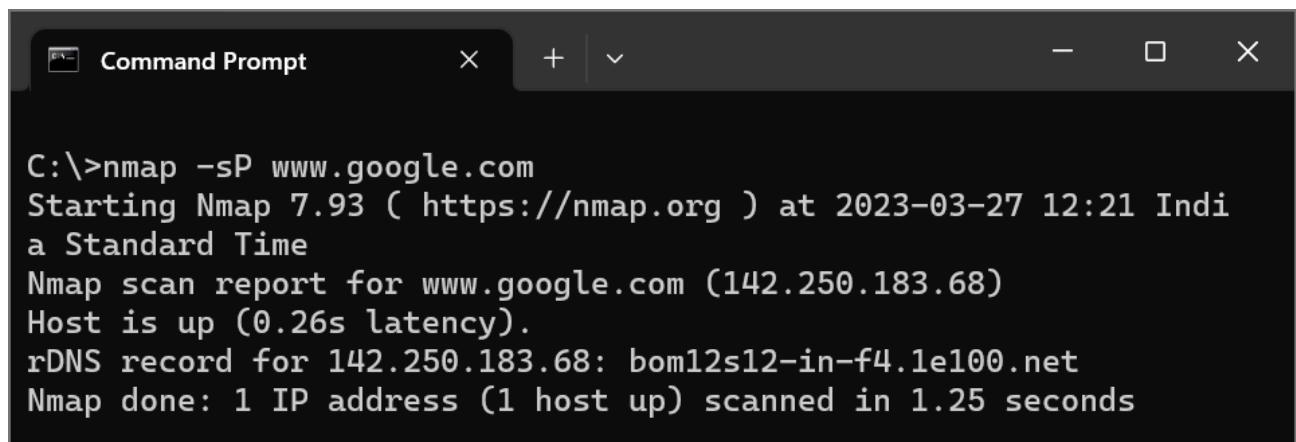
- Nmap is also used to scan networks.
- Nmap is now one of the core tools used by network administrators to map their networks.
- The program can be used to find:
 - Live hosts on a network
 - Perform port scanning
 - Ping sweeps
 - OS detection
 - Version detection

2. Using 'Nmap' to perform scan for network.

- i. Ping Scan: It returns a list of hosts on your network and the total number of assigned IP addresses. If you spot any hosts or IP addresses on this list that you cannot account for, you can then run further commands to investigate them further.

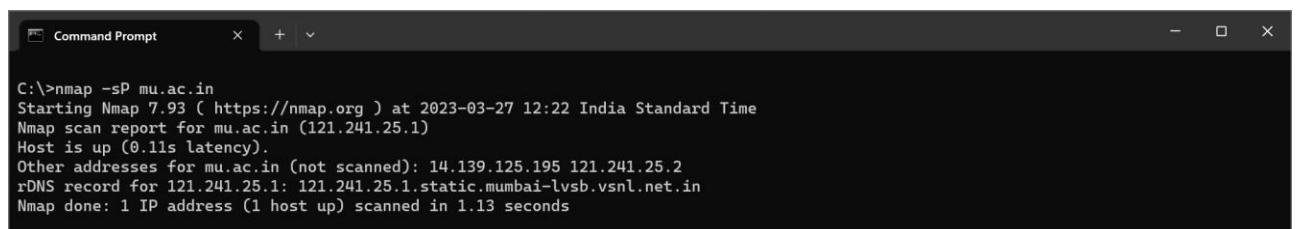
Command: nmap -sP <ip address / URL>

Example: nmap -sP www.google.com



```
C:\>nmap -sP www.google.com
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-27 12:21 India Standard Time
Nmap scan report for www.google.com (142.250.183.68)
Host is up (0.26s latency).
rDNS record for 142.250.183.68: bom12s12-in-f4.1e100.net
Nmap done: 1 IP address (1 host up) scanned in 1.25 seconds
```

Example: nmap -sP mu.ac.in

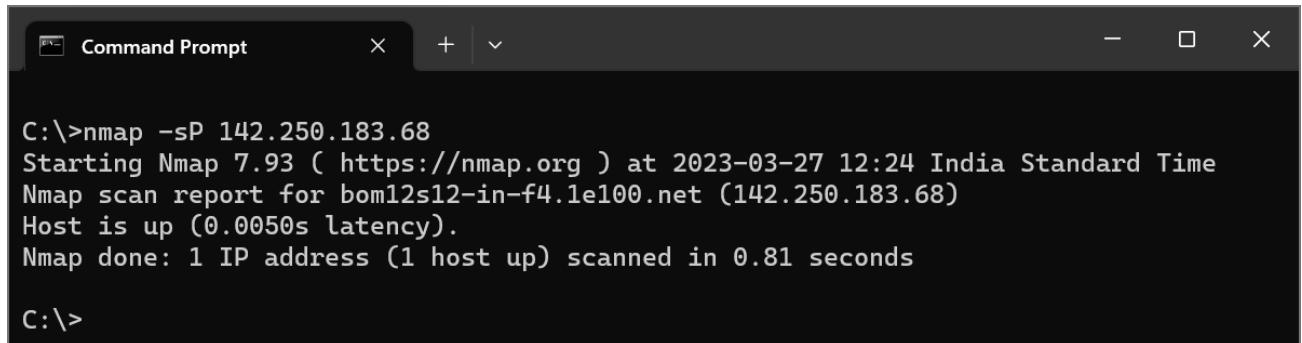


```
C:\>nmap -sP mu.ac.in
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-27 12:22 India Standard Time
Nmap scan report for mu.ac.in (121.241.25.1)
Host is up (0.11s latency).
Other addresses for mu.ac.in (not scanned): 14.139.125.195 121.241.25.2
rDNS record for 121.241.25.1: 121.241.25.1.static.mumbai-lvsb.vsnl.net.in
Nmap done: 1 IP address (1 host up) scanned in 1.13 seconds
```

- ii. ***Host Scan:*** A host scan actively sends ARP request packets to all the hosts connected to your network. Each host then responds to this packet with another ARP packet containing its status and MAC address. This can be a powerful way of spotting suspicious hosts connected to your network.

Command: nmap -sP <ip address / URL>

Example: nmap -sP 142.250.183.68 (www.google.com ip)



```
C:\>nmap -sP 142.250.183.68
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-27 12:24 India Standard Time
Nmap scan report for bom12s12-in-f4.1e100.net (142.250.183.68)
Host is up (0.0050s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.81 seconds

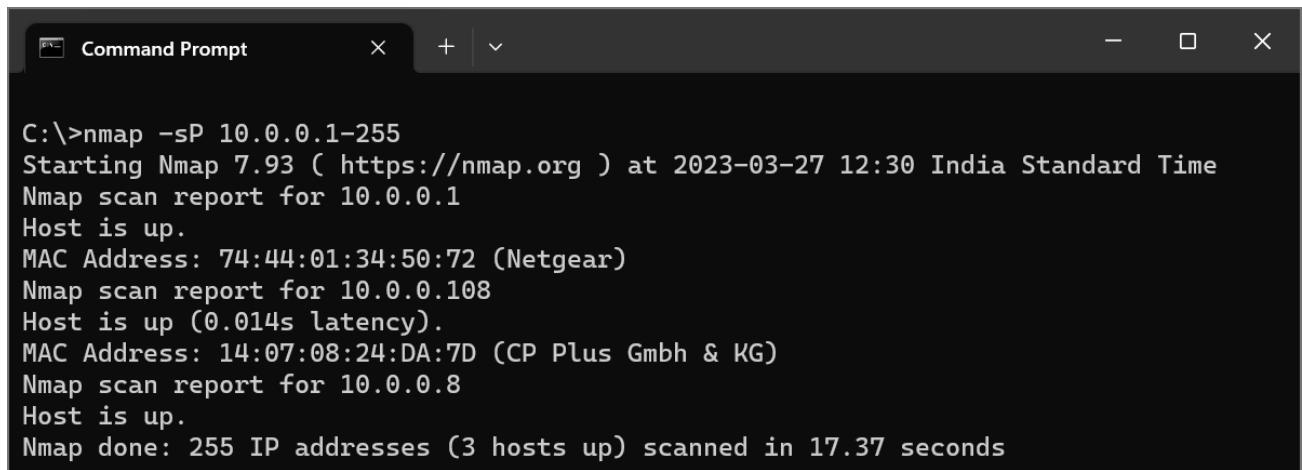
C:\>
```

Host scan can also be used to:

- Identify active host(s) in a network.
- It sends ARP request packets to all systems in the target.
- Host Scan Results, “Host is up” by receiving MAC address from each active host.

Command: nmap -sP <ip address / URL>

Example: nmap -sP 10.0.0.1-255 (it's current working network ip)



```
C:\>nmap -sP 10.0.0.1-255
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-27 12:30 India Standard Time
Nmap scan report for 10.0.0.1
Host is up.
MAC Address: 74:44:01:34:50:72 (Netgear)
Nmap scan report for 10.0.0.108
Host is up (0.014s latency).
MAC Address: 14:07:08:24:DA:7D (CP Plus GmbH & KG)
Nmap scan report for 10.0.0.8
Host is up.
Nmap done: 255 IP addresses (3 hosts up) scanned in 17.37 seconds
```

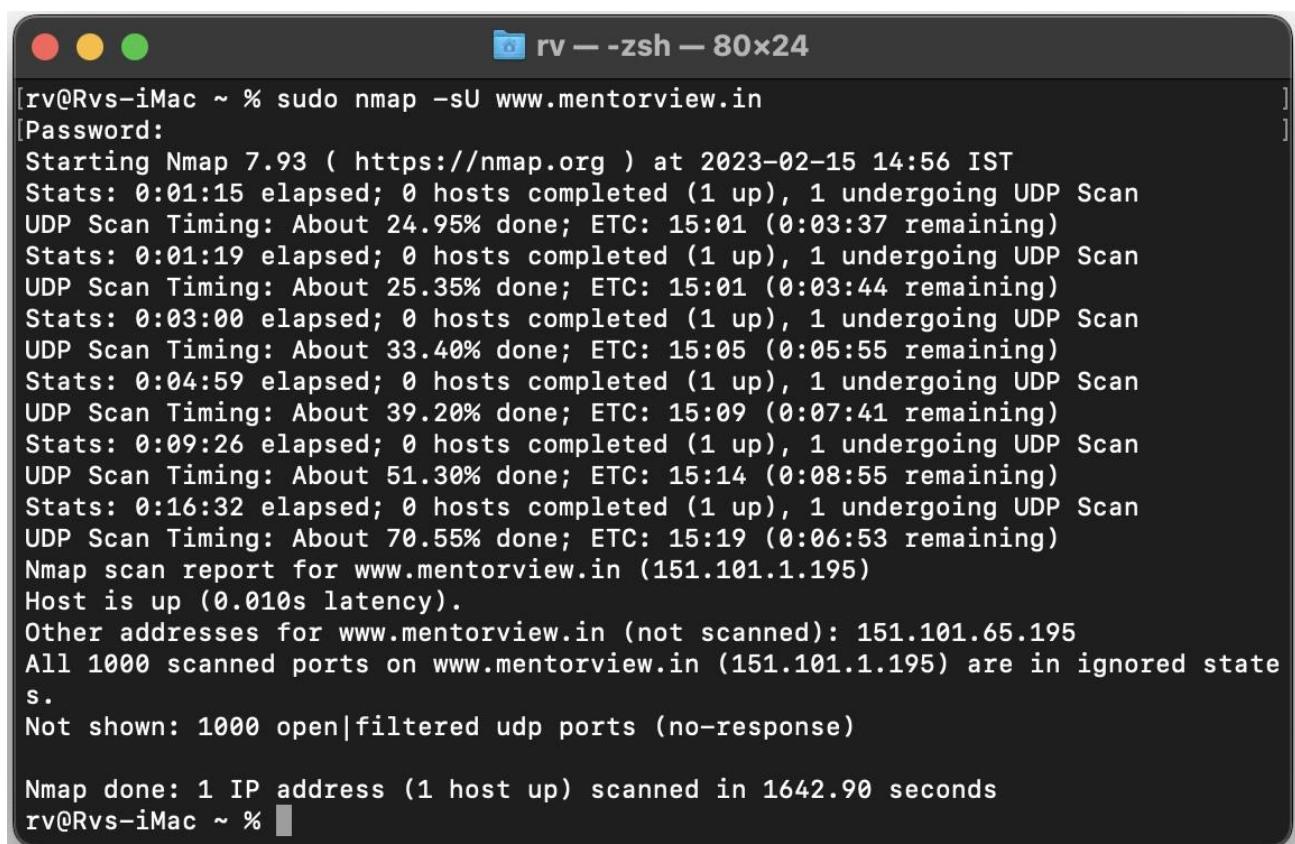
The output shows all the host connected to current network with its MAC address.

iii. UDP Scan:

- UDP scan works by sending a UDP packet to every destination port and analyses the response to determine the port's state; it is a connection-less protocol.
- For some common ports such as 53 and 161, a protocol-specific payload is sent to increase the response rate:
 - A service will respond with a UDP packet, proving that it is “open”.
 - If the port is “closed”, an ICMP Port Unreachable message is received from the target.
 - If no response is received after retransmissions, the port is classified as “open | filtered”. This means that the port could be open, or perhaps packet filters are blocking the communication.
 -

Command: nmap -sU <ip address / URL>

Example: nmap -sU www.mentorview.com



The terminal window shows the command `nmap -sU www.mentorview.in` being run. The output details the progress of the UDP scan, which took approximately 1642.90 seconds. It shows that 1 host was up and 1000 ports were open/filtered. The host IP is 151.101.1.195.

```
[rv@Rvs-iMac ~ % sudo nmap -sU www.mentorview.in
[Password:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-15 14:56 IST
Stats: 0:01:15 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 24.95% done; ETC: 15:01 (0:03:37 remaining)
Stats: 0:01:19 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 25.35% done; ETC: 15:01 (0:03:44 remaining)
Stats: 0:03:00 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 33.40% done; ETC: 15:05 (0:05:55 remaining)
Stats: 0:04:59 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 39.20% done; ETC: 15:09 (0:07:41 remaining)
Stats: 0:09:26 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 51.30% done; ETC: 15:14 (0:08:55 remaining)
Stats: 0:16:32 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 70.55% done; ETC: 15:19 (0:06:53 remaining)
Nmap scan report for www.mentorview.in (151.101.1.195)
Host is up (0.010s latency).
Other addresses for www.mentorview.in (not scanned): 151.101.65.195
All 1000 scanned ports on www.mentorview.in (151.101.1.195) are in ignored state
s.
Not shown: 1000 open|filtered udp ports (no-response)

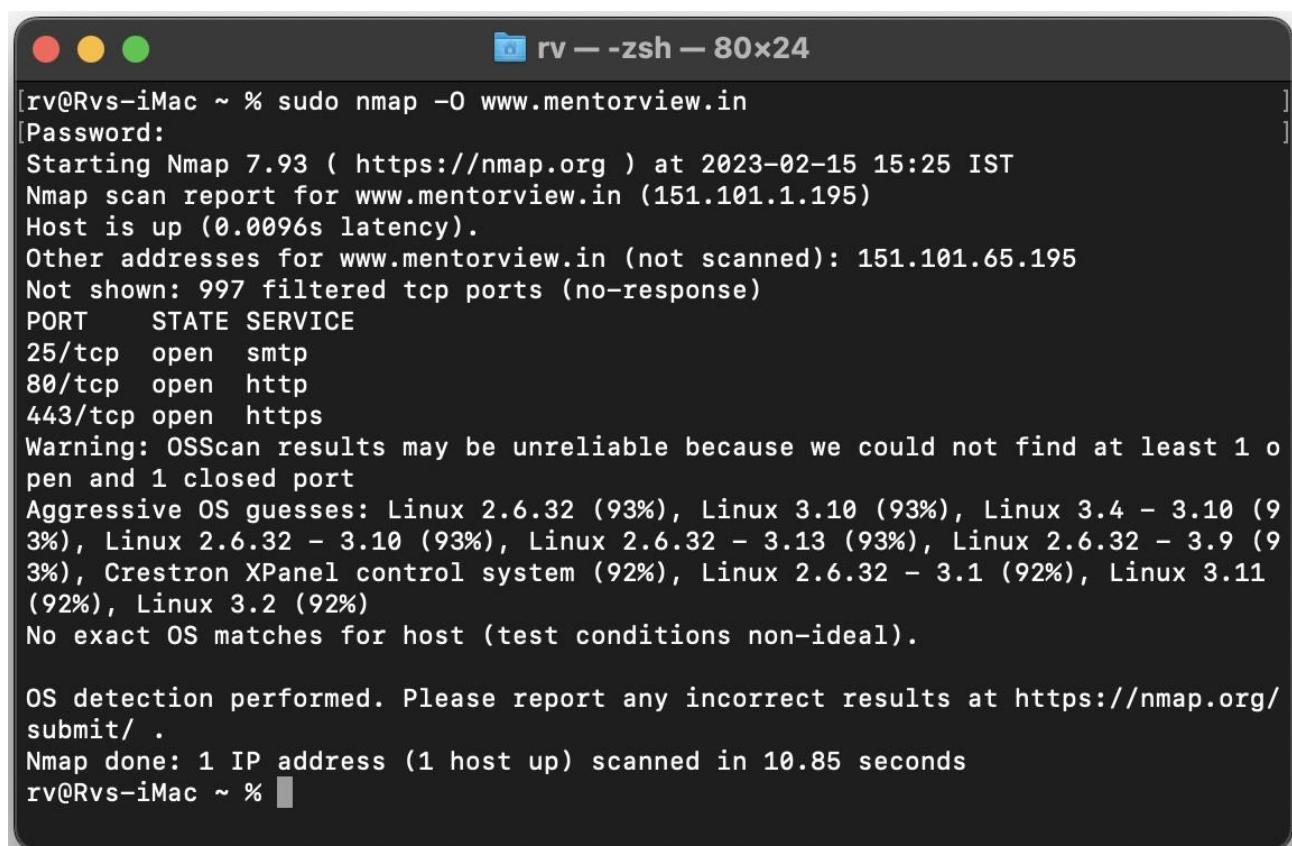
Nmap done: 1 IP address (1 host up) scanned in 1642.90 seconds
rv@Rvs-iMac ~ %
```

iv. *OS Detection Scan:* - nmap -O <target>

- Apart from the open port enumeration Nmap is quite useful in OS fingerprinting
- After running the command, we will get the information about:
 - Device type
 - Running
 - OS CPE (Common Platform Enumeration)
 - cpe:/o → OS and cpe:/h → hardware
 - OS details → human readable report of the operating system.
- The option -O inform Nmap to enable OS detection that identifies a wide variety of systems, including residential routers, IP webcams, operating systems, and many other hardware devices.
- You can also execute the following command for os detection:
 - nmap -O -p- --osscan-guess in case OS identification fails, try to guess the operating system.
 - nmap -O --osscan-limit try to launch OS detection if scan conditions are ideal.

Command: nmap -O <ip address / URL>

Example: nmap -O www.mentorview.com



The screenshot shows a terminal window titled "rv — zsh — 80x24". The command entered was "sudo nmap -O www.mentorview.in". The output shows the host is up with 0.0096s latency. It lists open ports 25/tcp (smtp), 80/tcp (http), and 443/tcp (https). A warning is given about OSScan results being unreliable. Aggressive OS guesses are listed for Linux versions 2.6.32 through 3.10, and for various other systems like Crestron XPanel control system and Linux 3.11. No exact OS matches were found due to non-ideal test conditions. The scan took 10.85 seconds and scanned 1 IP address (1 host up).

```
[rv@Rvs-iMac ~ % sudo nmap -O www.mentorview.in
[Password:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-15 15:25 IST
Nmap scan report for www.mentorview.in (151.101.1.195)
Host is up (0.0096s latency).
Other addresses for www.mentorview.in (not scanned): 151.101.65.195
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE
25/tcp    open  smtp
80/tcp    open  http
443/tcp   open  https
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 2.6.32 (93%), Linux 3.10 (93%), Linux 3.4 - 3.10 (93%), Linux 2.6.32 - 3.10 (93%), Linux 2.6.32 - 3.13 (93%), Linux 2.6.32 - 3.9 (93%), Crestron XPanel control system (92%), Linux 2.6.32 - 3.1 (92%), Linux 3.11 (92%), Linux 3.2 (92%)
No exact OS matches for host (test conditions non-ideal).

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 10.85 seconds
rv@Rvs-iMac ~ %
```

v. Version Scan:

- When doing vulnerability assessments of your companies or clients, you really want to know which mail and DNS servers and versions are running.
- Having an accurate version number helps dramatically in determining which exploits a server is vulnerable to.
- Fingerprinting a service may also reveal additional information about a target, such as available modules and specific protocol information.
- Version scan is also categorized as Banner Grabbing in penetration testing.

Command: nmap -O <ip address / URL>

Example: nmap -O www.mentorview.com

```
[rv@Rvs-iMac ~ % sudo nmap -sV www.mentorview.in
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-15 15:27 IST
NSOCK ERROR [0.0690s] ssl_init_helper(): OpenSSL legacy provider failed to load.

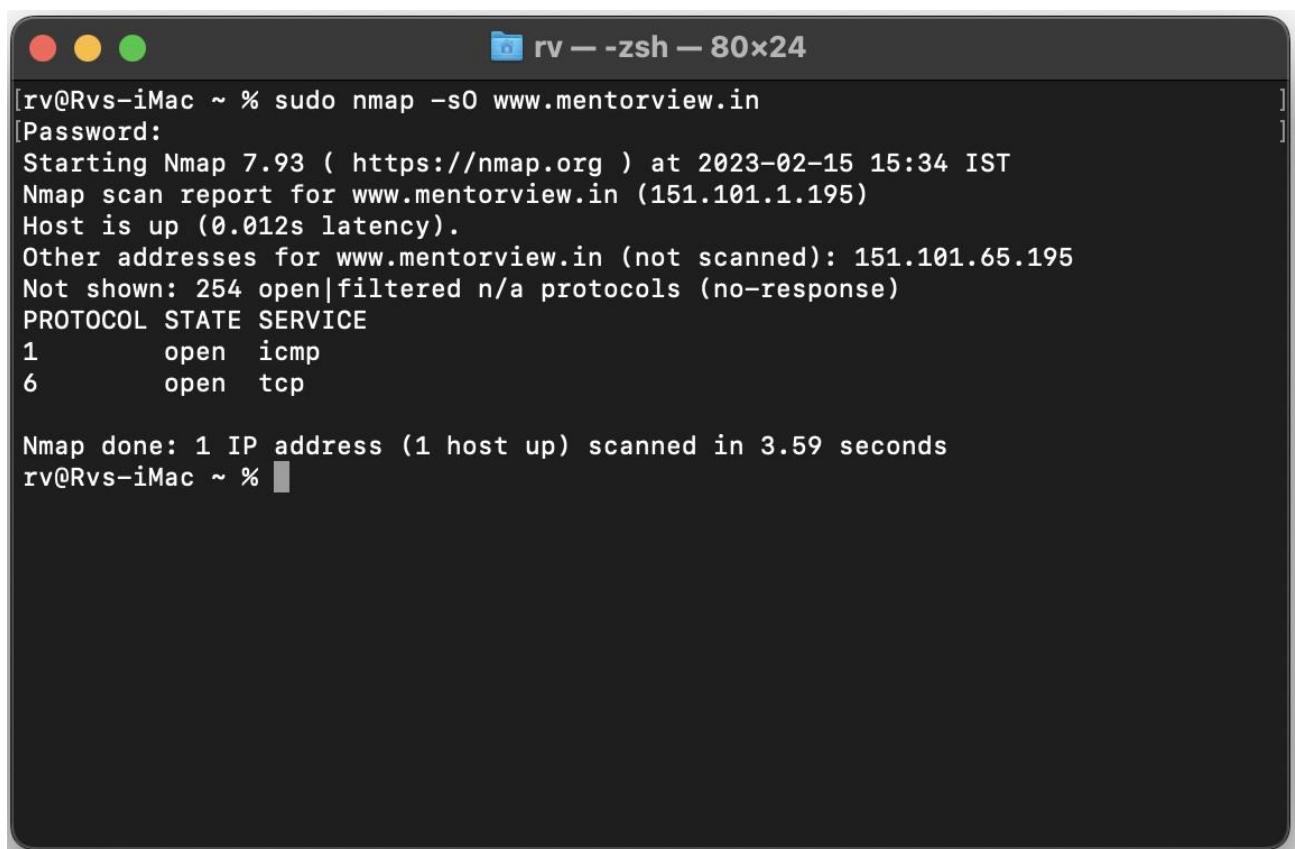
Nmap scan report for www.mentorview.in (151.101.1.195)
Host is up (0.0075s latency).
Other addresses for www.mentorview.in (not scanned): 151.101.65.195
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
25/tcp    open  smtp?
80/tcp    open  http        Varnish
443/tcp   open  ssl/https
3 services unrecognized despite returning data. If you know the service/version,
please submit the following fingerprints at https://nmap.org/cgi-bin/submit.cgi
?new-service :
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
SF-Port25-TCP:V=7.93%I=7%D=2/15%Time=63ECACB5%P=x86_64-apple-darwin17.7.0%
SF:r(GenericLines,28,"500\x20Syntax\x20error,\x20command\x20unrecognized\r
SF:\n");
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
SF-Port80-TCP:V=7.93%I=7%D=2/15%Time=63ECACA6%P=x86_64-apple-darwin17.7.0%
SF:r(GetRequest,167,"HTTP/1\.1\x20301\x20Moved\x20Permanently\r\nConnectio
SF:n:\x20close\r\nContent-Length:\x200\r\nServer:\x20Varnish\r\nRetry-Afte
```

vi. Protocol Scan:

- IP Protocol scan is very helpful for determining what communication protocols are being used by a host.
- This method shows how to use Nmap to compute all of the IP protocols, where sends a raw IP packet without any additional protocol header, to each protocol on the target machine.
- For the IP protocols TCP, ICMP, UDP, IGMP, and SCTP, Nmap will set valid header values but for the rest, an empty IP packet will be used.

Command: nmap -sO <ip address / URL>

Example: nmap -sO www.mentorview.com



The terminal window title is "rv --zsh -- 80x24". The command entered is "sudo nmap -sO www.mentorview.in". The output shows the Nmap version (7.93), the start time (2023-02-15 15:34 IST), and the target host (151.101.1.195). It indicates the host is up with 0.012s latency. Other addresses for the target are listed as 151.101.65.195. A note says "Not shown: 254 open|filtered n/a protocols (no-response)". The scan results table shows two ports: port 1 is open and listening for ICMP, and port 6 is open and listening for TCP. The total scan time is 3.59 seconds.

```
[rv@Rvs-iMac ~ % sudo nmap -sO www.mentorview.in
[Password:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-15 15:34 IST
Nmap scan report for www.mentorview.in (151.101.1.195)
Host is up (0.012s latency).
Other addresses for www.mentorview.in (not scanned): 151.101.65.195
Not shown: 254 open|filtered n/a protocols (no-response)
PROTOCOL STATE SERVICE
1       open  icmp
6       open  tcp

Nmap done: 1 IP address (1 host up) scanned in 3.59 seconds
rv@Rvs-iMac ~ % ]]
```

• **IDS (Intrusion Detection Systems) Tool:**

- An Intrusion Detection System (IDS) monitors network traffic for unusual or suspicious activity and sends an alert to the administrator.
- Detection of strange activity and reporting it to the network administrator is the primary function of IDS.
- However, some IDS software can take action based on rules when malicious activity is detected, for example blocking certain incoming traffic.
- Some of the best Intrusion Detection System Software and Tools are as follows:
 - **Snort:** Provided by Cisco Systems and free to use, leading network-based intrusion detection system software.
 - **Solar Winds Security Event Manager EDITOR'S CHOICE:** Analyses logs from Windows, Unix, Linux, and Mac OS systems. It manages data collected by Snort, including real-time data. SEM is also an intrusion prevention system, shipping with over 700 rules to shut down malicious activity.
 - **Security Onion:** Network monitoring and security tool made up of elements pulled in from other free tools.
- Types of Intrusion Detection Systems:
 - There are two main types of intrusion detection systems:
 - **Host-based Intrusion Detection System (HIDS):**
 - This system will examine events on a computer on your network rather than the traffic that passes around the system.
 - **Network-based Intrusion Detection System (NIDS):**
 - This system will examine the traffic on your network.
- **Snort:**
 - Snort is a free open-source network intrusion detection system (NIDS) and intrusion prevention system (IPS).
 - Snort IPS uses a series of rules that help define malicious network activity and uses those rules to find packets that match against them and generates alerts for users.
 - Snort can be configured in three main modes:
 - **Sniffer Mode:** The program will read network packets and display them on the console.
 - **Packet Logger Mode:** The program will log packets to the disk.
 - **Network Intrusion Detection System Mode:** The program will monitor network traffic and analyse it against a rule set defined by the user. The program will then perform a specific action based on what has been identified.

- In order to start working with the ‘Snort’ we need to change the current working direct to `C:\Snort\bin`

i. Command: To check version of ‘snort’ → snort -V

```
C:\Snort\bin>snort -V

    --> Snort! <--  
Version 2.9.20-WIN64 GRE (Build 82)  
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team  
Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.  
Copyright (C) 1998-2013 Sourcefire, Inc., et al.  
Using PCRE version: 8.10 2010-06-25  
Using ZLIB version: 1.2.11

C:\Snort\bin>
```

ii. Command: To get help → snort --help

```
C:\Snort\bin>Snort --help

    --> Snort! <--  
Version 2.9.20-WIN64 GRE (Build 82)  
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team  
Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.  
Copyright (C) 1998-2013 Sourcefire, Inc., et al.  
Using PCRE version: 8.10 2010-06-25  
Using ZLIB version: 1.2.11

USAGE: Snort [<options>] <filter options>  
Snort /SERVICE /INSTALL [<options>] <filter options>  
Snort /SERVICE /UNINSTALL  
Snort /SERVICE /SHOW  
Options:  
-A      Set alert mode: fast, full, console, test or none (alert file alerts only)  
-b      Log packets in tcpdump format (much faster!)  
-B <mask> Obfuscates IP addresses in alerts and packet dumps using CIDR mask  
-c <rules> Use Rules File <rules>  
-C      Print out payloads with character data only (no hex)  
-d      Dump the Application Layer  
-e      Display the second layer header info  
-E      Log alert messages to NT Eventlog. (Win32 only)  
-f      Turn off fflush() calls after binary log writes  
-F <bpf> Read BPF filters from file <bpf>  
-G <0xid> Log Identifier (to uniquely id events for multiple snorts)  
-h <hn> Set home network = <hn>  
       (for use with -l or -B, does NOT change $HOME_NET in IDS mode)  
-H      Make hash tables deterministic.  
-i <if> Listen on interface <if>  
-I      Add Interface name to alert output  
-k <mode> Checksum mode (all,noip,notcp,noudp,noicmp,none)  
-K <mode> Logging mode (pcap[default],ascii,none)  
-l <ld> Log to directory <ld>  
-L <file> Log to this tcpdump file  
-n <cnt> Exit after receiving <cnt> packets  
-N      Turn off logging (alerts still work)
```

iii. Command: To find the list of interfaces running → snort -W

```
C:\Snort\bin>snort -W

--> Snort! <--
Version 2.9.20-WIN64 GRE (Build 82)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using PCRE version: 8.10 2010-06-25
Using ZLIB version: 1.2.11

Index Physical Address IP Address Device Name Description
---- -----
1 00:00:00:00:00:00 disabled \Device\NPF_{87637146-349B-49E2-82D3-3685B7716F26} WAN Miniport (Network Monitor)
2 00:00:00:00:00:00 disabled \Device\NPF_{867714B3-4A6C-4931-A535-C5A318EF44BB} WAN Miniport (IPv6)
3 00:00:00:00:00:00 disabled \Device\NPF_{0FB99408-02AF-4758-BD3C-A542796A761E} WAN Miniport (IP)
4 5C:BA:EF:08:05:B6 169.254.34.132 \Device\NPF_{C1C75E81-C10E-4422-BA74-07227FCDF533} Bluetooth Device (Personal Area Network)
5 5C:BA:EF:08:05:B5 10.0.0.8 \Device\NPF_{8E966D0B-93E4-4436-AF00-90D37A862115} Realtek 8822BE Wireless LAN 802.11ac PCI-E NIC
6 00:50:56:C0:00:08 192.168.16.1 \Device\NPF_{E3585292-011D-439E-A36F-DAE2B5525E9C} VMware Virtual Ethernet Adapter for VMnet8
7 00:50:56:C0:00:01 192.168.10.1 \Device\NPF_{3BC08BD7-4F27-4572-97DC-0894BF0810C2} VMware Virtual Ethernet Adapter for VMnet1
8 DE:BA:EF:08:05:B5 169.254.152.78 \Device\NPF_{0E7272545-BA71-4EEC-A320-6601AD887026} Microsoft Wi-Fi Direct Virtual Adapter #4
9 5C:BA:EF:08:05:B5 169.254.46.118 \Device\NPF_{0695E965-AF34-4A00-BCC2-FC6B845EC8EB0} Microsoft Wi-Fi Direct Virtual Adapter #3
10 00:00:00:00:00:00 0000:0000:0000:0000:0000:0000 \Device\NPF_Loopback Adapter for loopback traffic capture

C:\Snort\bin>
```

vi. Command: To carry out work of sniffing the network interface → snort -i 5 -c c:\Snort\etc\snort.conf -A console (here 5 indicates current working network)

o Where:

- i - stands for interface, here is where you tell snort what network interface it should sniff on (it should be followed by the interface number which is active that we have get information from above command)
- c - is where you tell snort the location of the file you want it to run
- A - means print output in the terminal

Command run:

```
C:\Snort\bin>snort -i 5 -c c:\Snort\etc\snort.conf -A console
Running in IDS mode

--== Initializing Snort ==-
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "c:\Snort\etc\snort.conf"
PortVar 'HTTP_PORTS' defined : [ 36 80:90 311 323 383 443:444 555 591 593 623 631 664 801 808 818 901 972 1158 1220 1270 1414 1533 1581 1719:1720 1741 1801 1812 1830 194
2 2231 2301 2375 2381 2578 2869 2988 3000 3029 3037 3057 3128 3323 3443 3702 4000 4343 4444 4848 5000 5054 5060:5061 5117 5222 5256 5416 5443 5456 5480 5555 5600 581
4 5894 5986 6064 6080 6173 6988 7000:7001 7005 7076:7071 7080 7144:7145 7180:7181 7510 7776 7777:7779 8000:8001 8008 8014:8015 8020 8028 8040 8080:8082 8085 8088 809
8 8091 8118 8123 8161 8180:8182 8222 8243 8280 8300 8333 8344 8393 8400 8443 8484 8500 8509 8511 8694 8787 8800 8848 8852 8888 8899 8983 9000:9002 9050 9060 9088 909
0:9091 9111 9200:9201 9290 9443 9447 9502 9700 9710 9788 9830 9850 9999 9999:10000 10080 10100 10250 10297 10443 11371 12601 13014 15489 16000 16992:16995 17000 180
81 19981 20000 29991 30007 30018 30888 33308 34412 34443:34444 36099 37215 40007 41088 44449 49152:49153 50000 50002 50452 51423 53331 54444 55252 55555 56712 ]
PortVar 'SHLLCODE_PORTS' defined : [ 0:79 81:65535 ]
PortVar 'ORACLE_PORTS' defined : [ 1024:65535 ]
PortVar 'SSH_PORTS' defined : [ 22 ]
PortVar 'FTP_PORTS' defined : [ 21 2100 3535 ]
PortVar 'SIP_PORTS' defined : [ 5060:5061 5066 ]
PortVar 'FILE_DATA_PORTS' defined : [ 36 80:90 110 143 311 323 383 443:444 555 591 593 623 631 664 801 808 818 901 972 1158 1220 1270 1414 1533 1581 1719:1720 1741 1801
1812 1830 1942 2231 2301 2375 2381 2578 2869 2988 3000 3029 3037 3057 3128 3323 3443 3702 4000 4343 4444 4848 5000 5054 5060:5061 5117 5222 5256 5416 5443 5456 5480
5555 5600 5814 5894 5986 6064 6080 6173 6988 7000:7001 7005 7076:7071 7080 7144:7145 7180:7181 7510 7776 7777:7779 8000:8001 8008 8014:8015 8020 8028 8040 8080:8082 8085 8088 809
8088 8098 8095 8118 8123 8161 8180:8182 8222 8243 8280 8300 8333 8344 8393 8400 8443 8484 8500 8509 8511 8694 8787 8800 8848 8852 8888 8899 8983 9000:9002 9050 9060 9088 909
0:9091 9111 9200:9201 9290 9443 9447 9502 9700 9710 9788 9830 9850 9999 9999:10000 10080 10100 10250 10297 10443 11371 12601 13014 15489 16000 16992:16995 17000 180
995 17000 18081 19981 20000 29991 30007 30018 30888 33308 34412 34443:34444 36099 37215 40007 41088 44449 49152:49153 50000 50002 50452 51423 53331 54444 55252 55555 56712 ]
2 ]
PortVar 'GTP_PORTS' defined : [ 2123 2152 3386 ]
Detection:
    Search-Method = AC-Full-Q
    Split Any/Any group = enabled
    Search-Method=Optimizations = enabled
    Maximum pattern length = 20
Taged Packet Limit: 256
Loading dynamic engine C:\Snort\lib\snort_dynamicengine\sf_engine.dll... done
Loading all dynamic preprocessor libs from C:\Snort\lib\snort_dynamicpreprocessor...
    Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_dce2.dll... done
    Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_dnp3.dll... done
    Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_dns.dll... done
    Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_ftptelnet.dll... done
```

Packet sniffing in progress:

```

Command Prompt - snort -i! + - v

Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_DCERP2 Version 1.0 <Build 3>
Commenting packet processing (pid=15864)
03/29/11:32:48.564142 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52997 -> 142.250.67.228:45833
03/29/11:32:48.579735 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:29832
03/29/11:32:48.595693 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:27974
03/29/11:32:48.610968 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52997 -> 142.250.67.228:52791
03/29/11:32:48.626651 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52997 -> 142.250.67.228:640
03/29/11:32:48.642134 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52997 -> 142.250.67.228:53559
03/29/11:32:48.657613 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:43836
03/29/11:32:48.673613 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:45833
03/29/11:32:48.689303 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:53001 -> 142.250.67.228:29832
03/29/11:32:48.704653 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:53001 -> 142.250.67.228:27974
03/29/11:32:48.728195 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:52791
03/29/11:32:48.735945 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:53001 -> 142.250.67.228:640
03/29/11:32:48.751647 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:53559
03/29/11:32:48.767135 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52997 -> 142.250.67.228:40913
03/29/11:32:48.782838 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:53001 -> 142.250.67.228:45833
03/29/11:32:48.788562 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52997 -> 142.250.67.228:26117
03/29/11:32:48.814126 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52997 -> 142.250.67.228:18768
03/29/11:32:48.842803 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 142.250.67.228:443 -> 10.0.0.8:53198
03/29/11:32:48.8485269 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52997 -> 142.250.67.228:46604
03/29/11:32:48.861238 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:53001 -> 142.250.67.228:53559
03/29/11:32:48.876698 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:40913
03/29/11:32:48.892286 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52997 -> 142.250.67.228:38280
03/29/11:32:48.967737 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:26117
03/29/11:32:48.923416 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:53001 -> 142.250.67.228:52791
03/29/11:32:48.940944 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 142.250.67.228:443 -> 10.0.0.8:53198
03/29/11:32:48.954737 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52997 -> 142.250.67.228:46604
03/29/11:32:48.970847 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52997 -> 142.250.67.228:29404
03/29/11:32:48.985795 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:53001 -> 142.250.67.228:40913
03/29/11:32:49.001397 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:38280
03/29/11:32:49.017172 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:53001 -> 142.250.67.228:26117
03/29/11:32:49.032169 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:53001 -> 142.250.67.228:18768
03/29/11:32:49.048595 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:6394
03/29/11:32:49.064138 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:53001 -> 142.250.67.228:46604
03/29/11:32:49.079938 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:2946

```

Unidentified traffic (the longest line in the output):

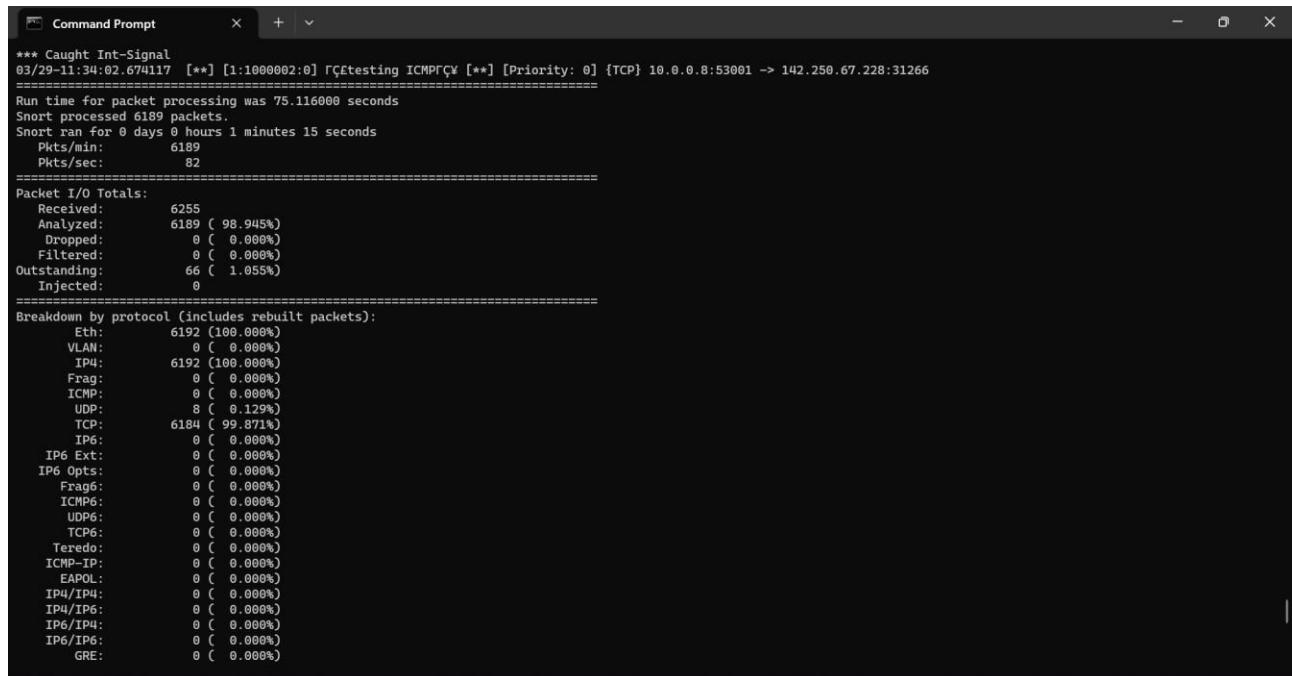
```

Command Prompt - snort -i! + - v

03/29/11:32:51.5508404 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52997 -> 142.250.67.228:63733
03/29/11:32:51.566362 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52997 -> 142.250.67.228:59899
03/29/11:32:51.581936 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:36017
03/29/11:32:51.597245 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52997 -> 142.250.67.228:48562
03/29/11:32:51.612576 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:44072
03/29/11:32:51.628622 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:34674
03/29/11:32:51.644218 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:53001 -> 142.250.67.228:34456
03/29/11:32:51.660251 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:63733
03/29/11:32:51.675848 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:59899
03/29/11:32:51.691132 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:53001 -> 142.250.67.228:36017
03/29/11:32:51.741366 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 119.133.21. (http_inspect) UNESCAPED SPACE IN HTTP URI [**] [Classification: Unknown Traffic] [Priority: 3] {TCP} 10.0.0.8:51271 -> 13.107.42.1 2:443
03/29/11:32:51.691873 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 13.107.42.12:443 -> 10.0.0.8:51271
03/29/11:32:51.691873 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 13.107.42.12:443 -> 10.0.0.8:51271
03/29/11:32:51.692416 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 13.107.42.12:443 -> 10.0.0.8:51271
03/29/11:32:51.692568 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 13.107.42.12:443 -> 13.107.42.12:443
03/29/11:32:51.706774 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:48562
03/29/11:32:51.722272 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:53001 -> 142.250.67.228:44072
03/29/11:32:51.737992 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:53001 -> 142.250.67.228:34674
03/29/11:32:51.756365 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52997 -> 142.250.67.228:32831
03/29/11:32:51.769246 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:53001 -> 142.250.67.228:63733
03/29/11:32:51.784917 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:53001 -> 142.250.67.228:59899
03/29/11:32:51.800654 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52997 -> 142.250.67.228:3677
03/29/11:32:51.816167 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:53001 -> 142.250.67.228:48562
03/29/11:32:51.831925 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52997 -> 142.250.67.228:38331
03/29/11:32:51.847617 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52997 -> 142.250.67.228:38042
03/29/11:32:51.862938 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:32831
03/29/11:32:51.878653 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52997 -> 142.250.67.228:10192
03/29/11:32:51.894299 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52997 -> 142.250.67.228:28562
03/29/11:32:51.969977 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:3677
03/29/11:32:51.9725530 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52997 -> 142.250.67.228:38493
03/29/11:32:51.941194 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:38331
03/29/11:32:51.956736 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:38042
03/29/11:32:51.972491 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:53001 -> 142.250.67.228:32831
03/29/11:32:51.988160 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:10192
03/29/11:32:52.003662 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:28562
03/29/11:32:52.019272 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:53001 -> 142.250.67.228:3677
03/29/11:32:52.034970 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:52999 -> 142.250.67.228:38493
03/29/11:32:52.050535 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:53001 -> 142.250.67.228:38331
03/29/11:32:52.066200 [**] [1:0000002:0] ↗Ctesting ICMPv4 [**] [Priority: 0] {TCP} 10.0.0.8:53001 -> 142.250.67.228:38042

```

Report of Output:



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The content of the window is the output of the Snort packet analyzer. The output includes statistics like packet counts, rates, and protocol breakdowns.

```
*** Caught Int-Signal
03/29-11:34:02.674117 [**] [1:1000002:0] ICMPFCW [**] [Priority: 0] {TCP} 10.0.0.8:53001 -> 142.250.67.228:31266
=====
Run time for packet processing was 75.116000 seconds
Snort processed 6189 packets.
Snort ran for 0 days 0 hours 1 minutes 15 seconds
Pkts/min: 6189
Pkts/sec: 82
=====
Packet I/O Totals:
Received: 6255
Analyzed: 6189 ( 98.945%)
Dropped: 0 ( 0.000%)
Filtered: 0 ( 0.000%)
Outstanding: 66 ( 1.055%)
Injected: 0
=====
Breakdown by protocol (includes rebuilt packets):
Eth: 6192 (100.000%)
VLAN: 0 ( 0.000%)
IP4: 6192 (100.000%)
Frag: 0 ( 0.000%)
ICMP: 0 ( 0.000%)
UDP: 8 ( 0.129%)
TCP: 6184 ( 99.871%)
IP6: 0 ( 0.000%)
IP6 Ext: 0 ( 0.000%)
IP6 Opts: 0 ( 0.000%)
Frag6: 0 ( 0.000%)
ICMP6: 0 ( 0.000%)
UDP6: 0 ( 0.000%)
TCP6: 0 ( 0.000%)
Teredo: 0 ( 0.000%)
ICMP-IP: 0 ( 0.000%)
EAPOL: 0 ( 0.000%)
IP4/IP4: 0 ( 0.000%)
IP4/IP6: 0 ( 0.000%)
IP6/IP4: 0 ( 0.000%)
IP6/IP6: 0 ( 0.000%)
GRE: 0 ( 0.000%)
```

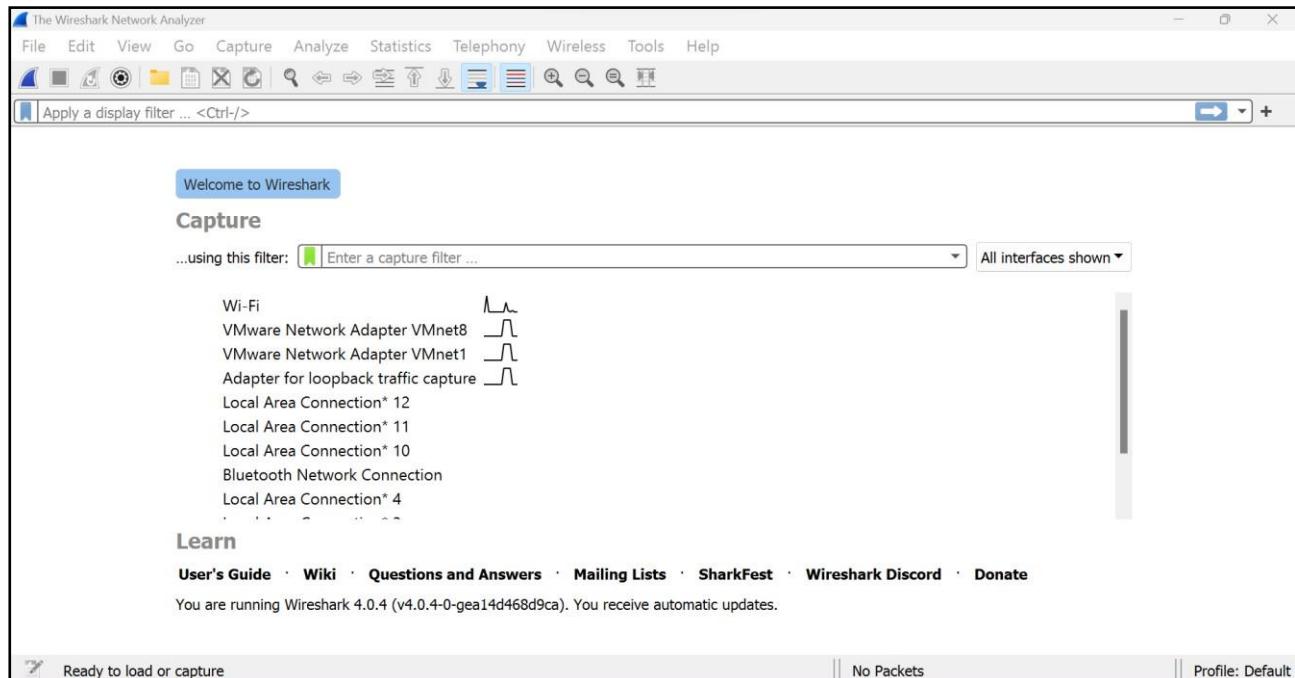
• **Network Sniffing:**

- Computers communicate using networks. These networks could be on a local area network LAN or exposed to the internet.
- Network Sniffers are programs that capture low-level package data that is transmitted over a network.
- An attacker can analyse this information to discover valuable information such as user ids and passwords.
- Network sniffing is the process of capturing data packets sent over a network.
- This can be done by the specialized software program or hardware equipment.
- Sniffing can be used to:
 - Capture sensitive data such as login credentials
 - Eavesdrop on chat messages
 - Capture files that have been transmitted over a network.
- The following are protocols that are vulnerable to sniffing:
 - Telnet
 - Rlogin
 - HTTP
 - SMTP
 - NNTP
 - POP
 - FTP
 - IMAP
- The above protocols are vulnerable if login details are sent in plain text.

• **Network sniffing using Wireshark:**

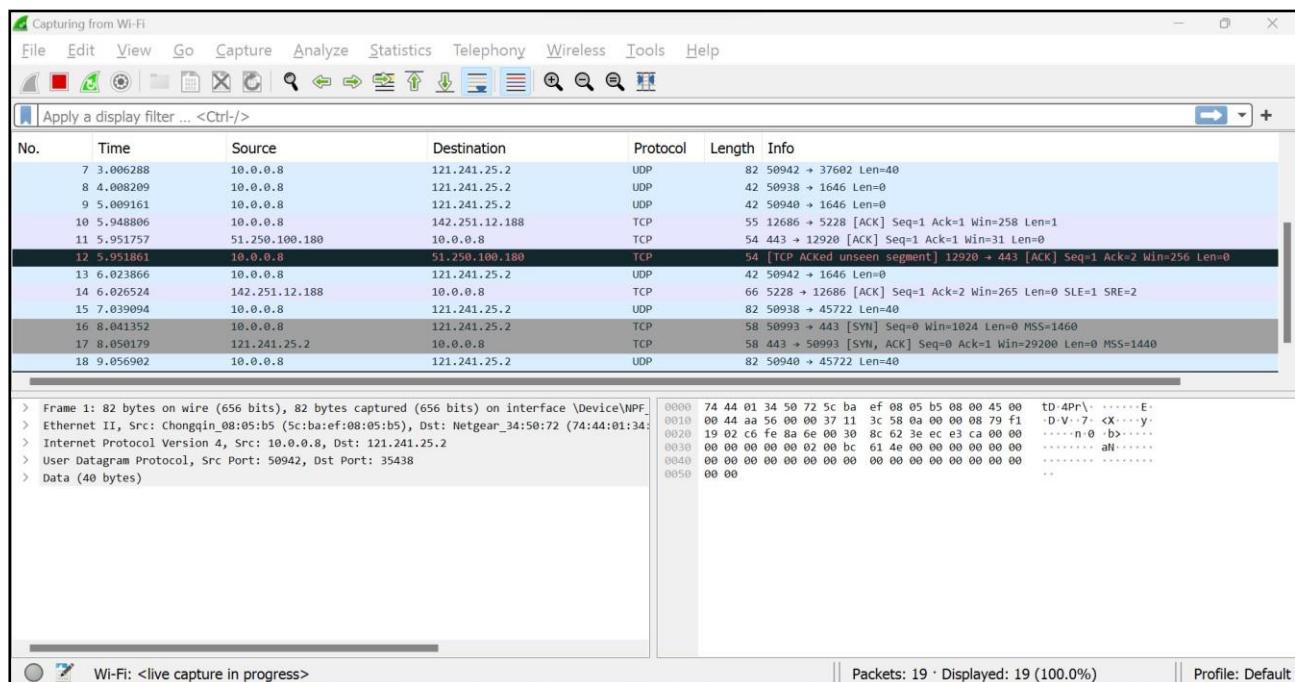
- Wireshark is a free and open-source packet analyser.
- It is used for network troubleshooting, analysis, software and communications protocol development, and education.
- Wireshark is cross-platform, using the Qt widget toolkit in current releases to implement its user interface, and using pcap to capture packets.
- It runs on Linux, macOS, BSD, Solaris, some other Unix-like operating systems, and Microsoft Windows.
- There is also a terminal-based (non-GUI) version called TShark.
- Wireshark is used to capture and analyse packets in network.
- It is also used as a sniffer, network protocol analyser, and network analyser.
- We can also apply specific filter on network traffic to get more filtered data packets.

i . Wireshark User Interface:

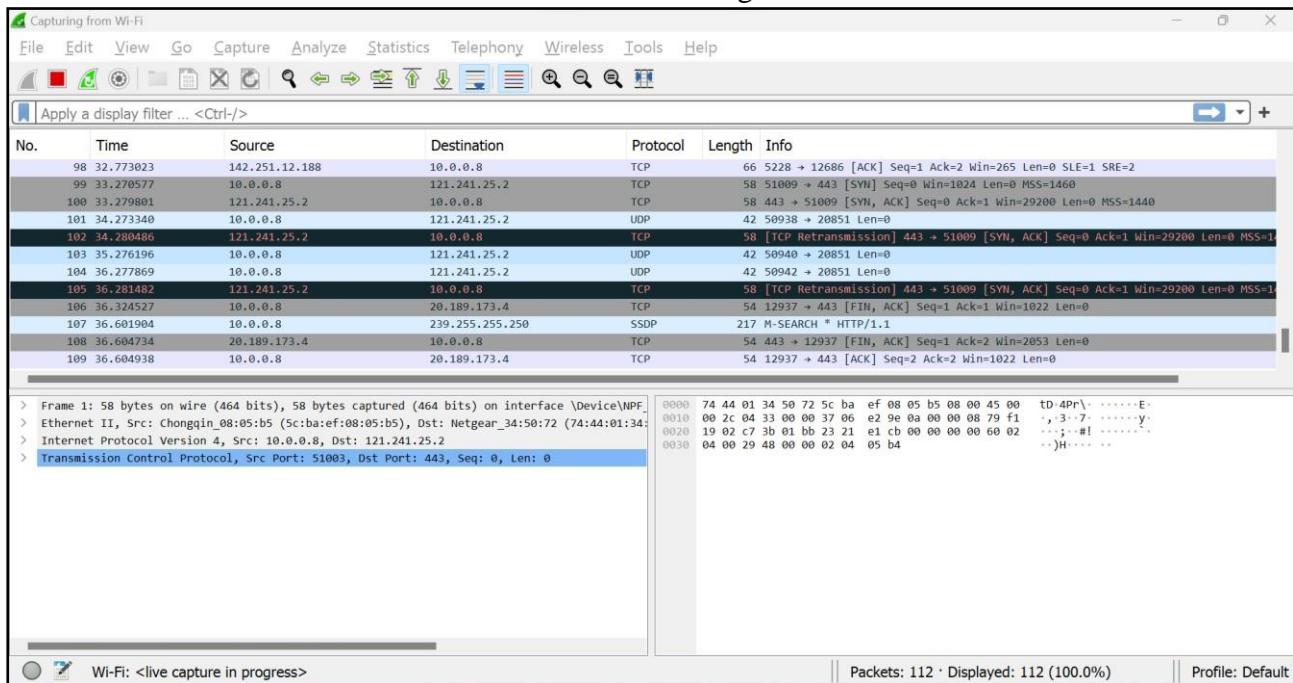


ii . Capturing Live Network Data:

- To capture Live Network Data double, click on any of the interface in the above UI screen.
- Once you doble click on the inface you will start getting packet detail entering and leaving the network as shown below:

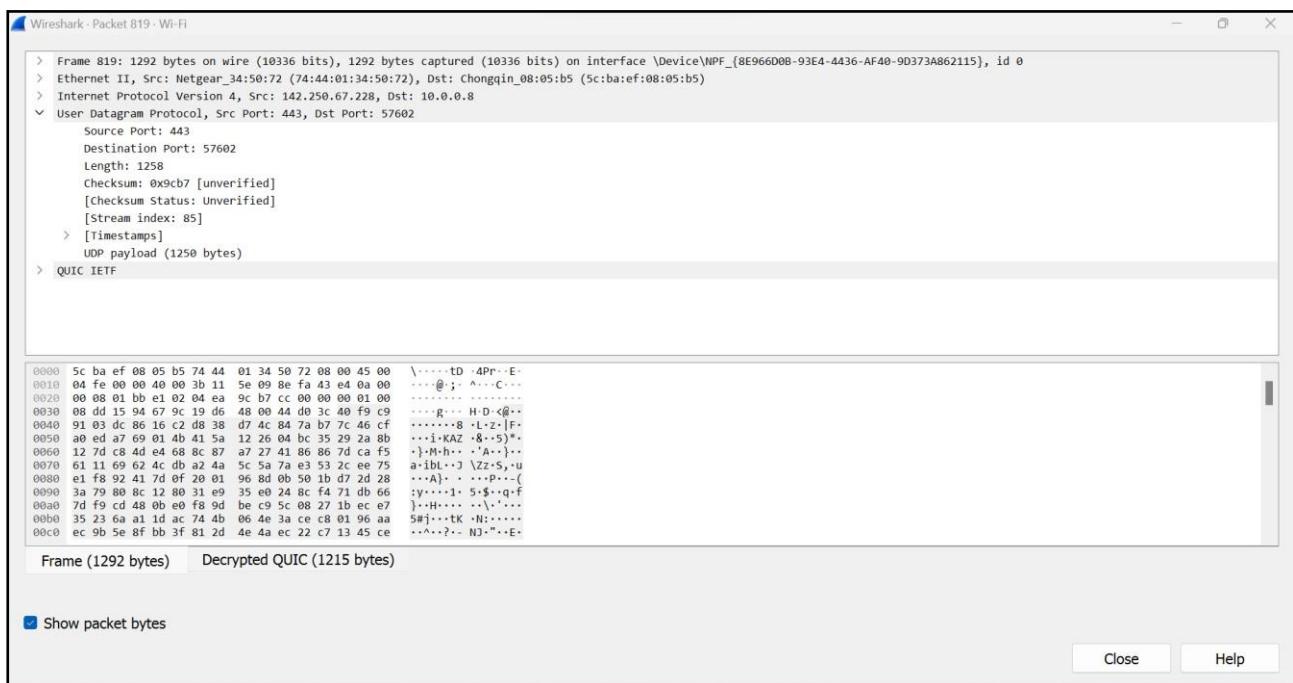


c. Packet based on different colour coding scheme.

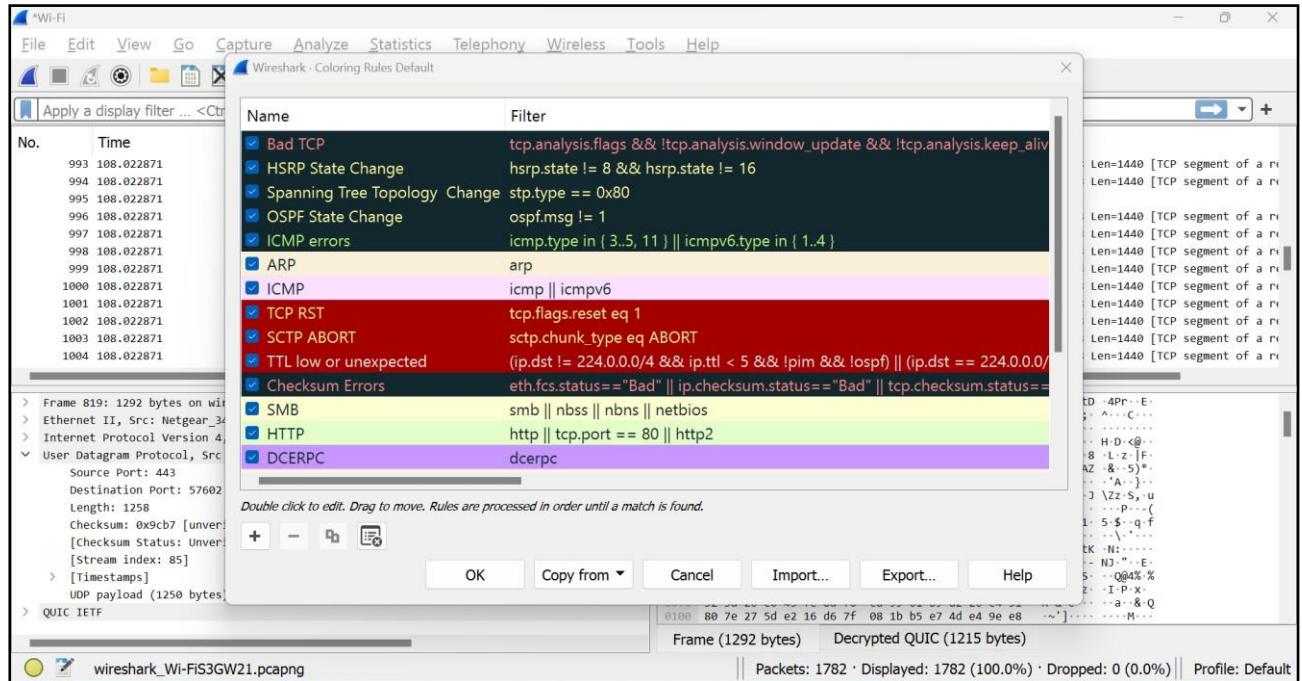


iii. Viewing Captured Packets:

- a. Double click on any of the packet that you want to view. Another window will open, showing the details of the selected packet as shown below:



iv. Colour Coding Scheme for various Protocols in Wireshark:



ENCRYPTION AND DECRYPTION

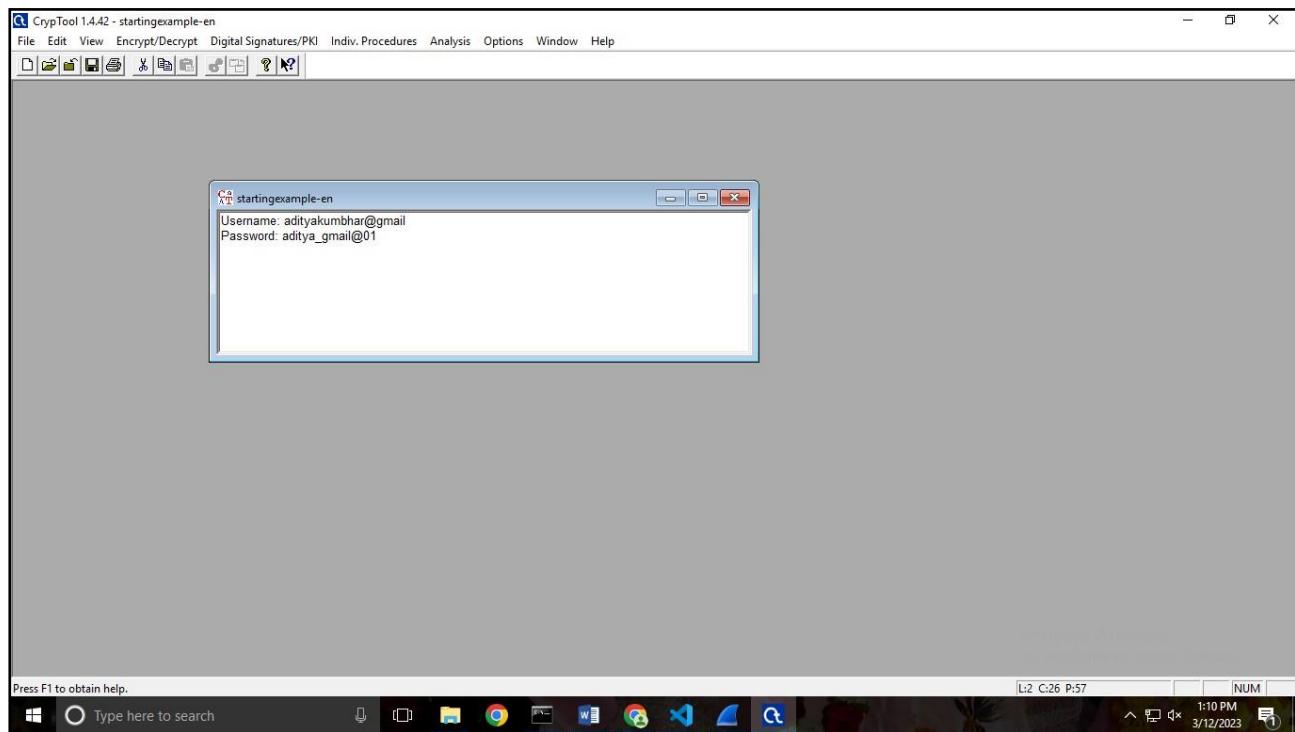
- **Using CrypTool:**

- CrypTool 1 (CT1) is a comprehensive and free educational program about cryptography and cryptanalysis offering extensive online help and many visualizations.
- We are basically using it to the process of encryption and decryption techniques.
- CrypTool provides a variety of range of Encryption methods some are:
 - Symmetric (Classic):
 - Caesar / Rot – 13
 - Symmetric (Modern):
 - RC2
 - RC4
 - DES
 - Triple DES
 - Asymmetric:
 - RSA Encryption
 - RSA Decryption

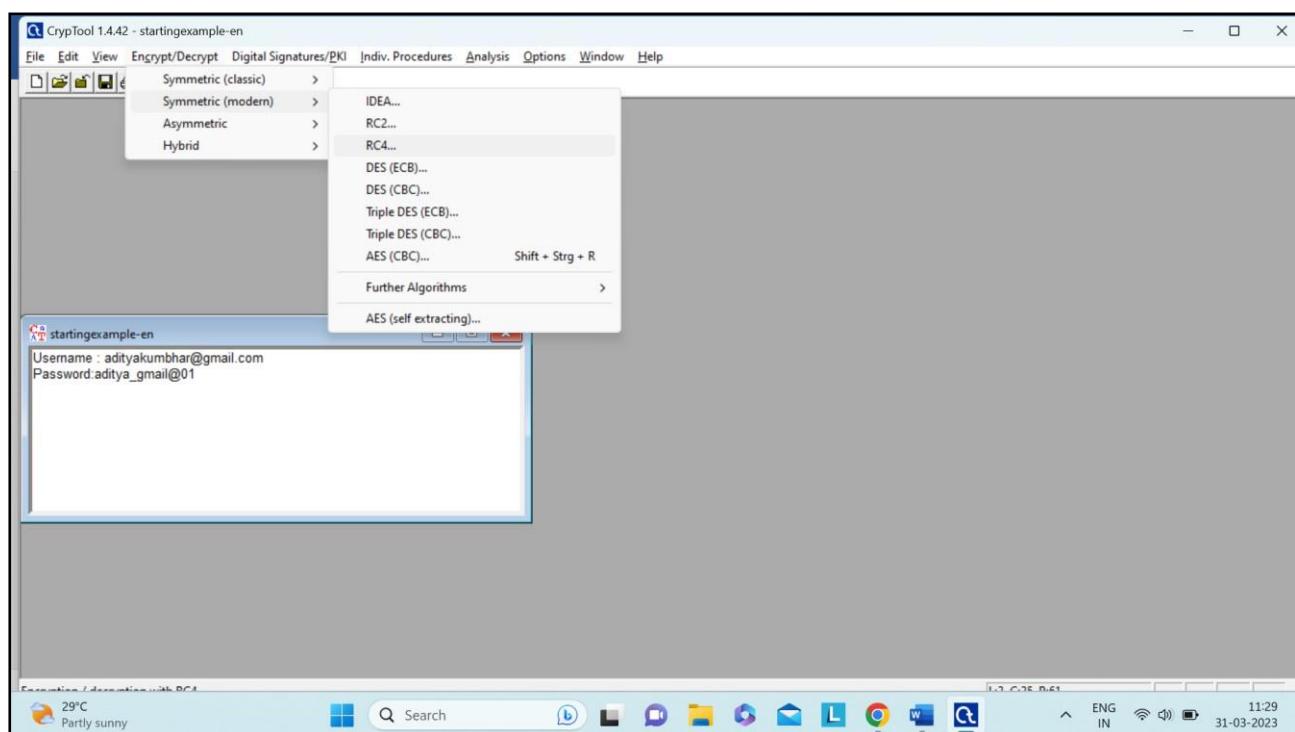
RC4 Algorithm:

Practical Work: Performing Encryption using RC4 algorithm

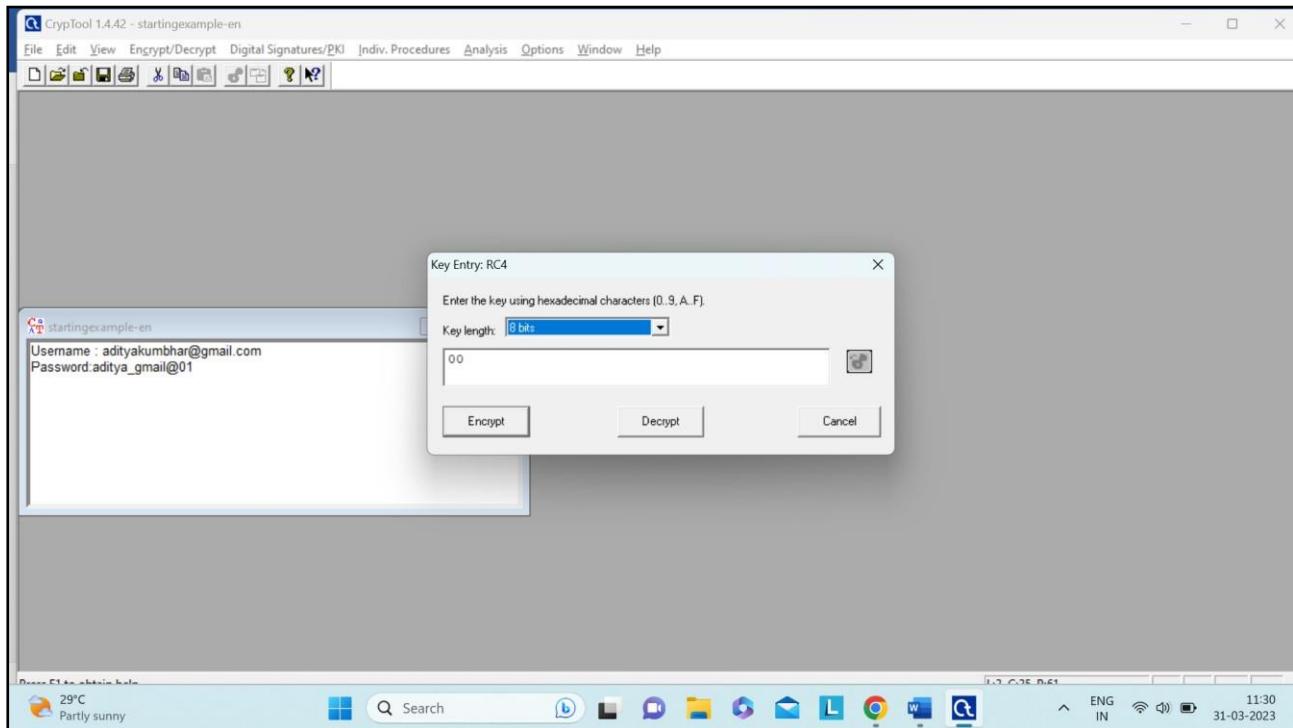
Step 1: Type the content that you need to encrypt.



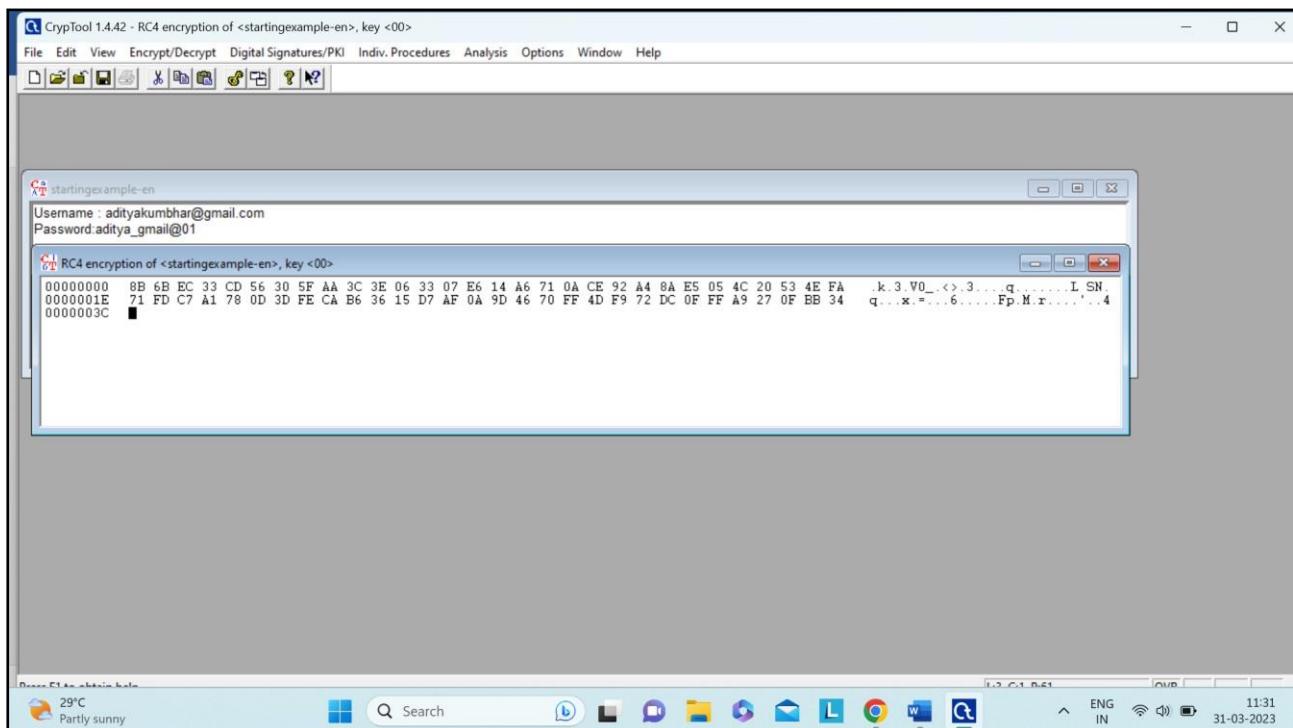
Step 2: Select the type of Encryption technique from the available list. (Here we are using RC4 algorithm for encryption)



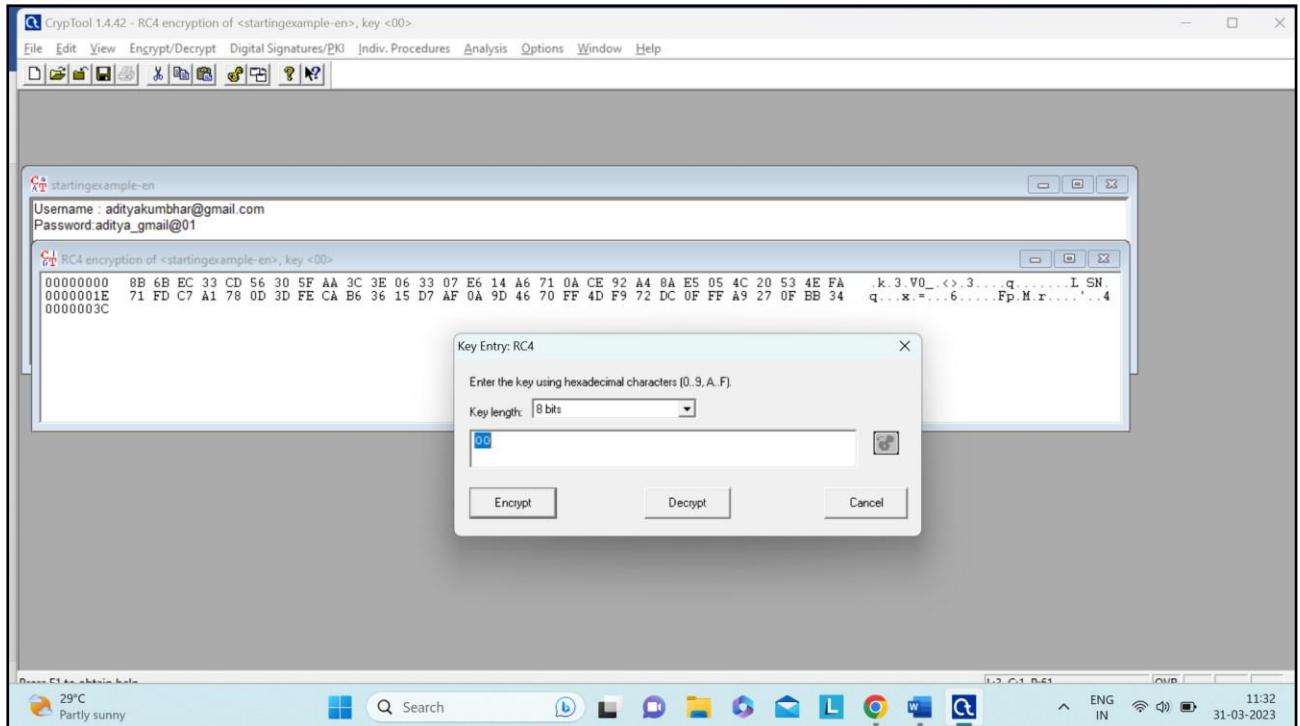
Encryption goes here:



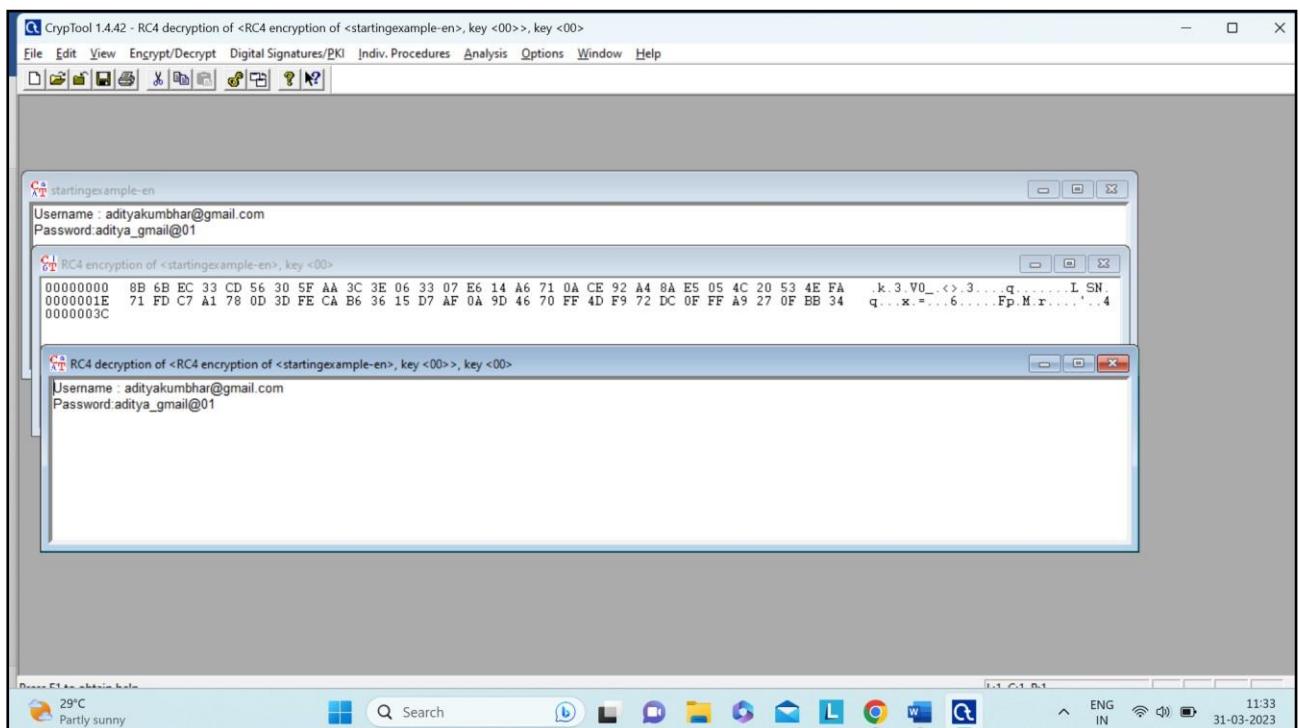
Step 3: Select Encrypt and the encrypted message will appear



Decryption goes here:



Decrypted Message:



Using traceroute, ping, ipconfig, netstat Command

1. tracert:

- a. The ‘tracert’ command is a Command Prompt command that's used to show several details about the path that a packet takes from the computer or device you're on to whatever destination you specify.
- b. Traceroute is a command which can show you the path a packet of information takes from your computer to one you specify.
- c. It will list all the routers it passes through until it reaches its destination, or fails to and is discarded.
- d. In addition to this, it will tell you how long each 'hop' from router to router takes.

Command: tracert <ip address / URL>

Example: tracert www.facebook.com

```
C:\>tracert www.facebook.com
Tracing route to star-mini.c10r.facebook.com [157.240.16.35]
over a maximum of 30 hops:
1  3 ms   1 ms   1 ms  10.0.0.1
2  2 ms   2 ms   2 ms  192.168.1.1
3  2 ms   2 ms   2 ms  10.10.200.89
4  *       *       * Request timed out.
5  60 ms   3 ms   3 ms  ae19.pr03.bom1.tfbnw.net [157.240.68.32]
6  8 ms    2 ms   2 ms  po103.psw01.bom1.tfbnw.net [157.240.52.211]
7  4 ms    2 ms   2 ms  173.252.67.39
8  6 ms    2 ms   5 ms  edge-star-mini-shv-01-bom1.facebook.com [157.240.16.35]

Trace complete.

C:\>
```

2. ping:

- a. The ping command is a Command Prompt command used to test the ability of the source computer to reach a specified destination computer.
- b. The ping command is usually used as a simple way to verify that a computer can communicate over the network with another computer or network device.

Command: tracert <ip address / URL>

Example: tracert www.facebook.com

```
C:\>ping www.facebook.com
Pinging star-mini.c10r.facebook.com [157.240.16.35] with 32 bytes of data:
Reply from 157.240.16.35: bytes=32 time=29ms TTL=57
Reply from 157.240.16.35: bytes=32 time=4ms TTL=57
Reply from 157.240.16.35: bytes=32 time=4ms TTL=57
Reply from 157.240.16.35: bytes=32 time=14ms TTL=57

Ping statistics for 157.240.16.35:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 29ms, Average = 12ms

C:\>
```

3. ipconfig:

- a. Ipconfig is a DOS utility that can be used from MS-DOS and the Windows command line to display the network settings currently assigned and given by a network.
- b. This command can be utilized to verify a network connection as well as to verify your network settings.

Command: ipconfig
Example: ipconfig

```
C:\>ipconfig

Windows IP Configuration

Wireless LAN adapter Local Area Connection* 3:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

Wireless LAN adapter Local Area Connection* 4:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

Ethernet adapter VMware Network Adapter VMnet1:
  Connection-specific DNS Suffix . :
  Link-local IPv6 Address . . . . . : fe80::cba2:79ed:8a2d:4d96%12
  IPv4 Address . . . . . : 192.168.10.1
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . :

Ethernet adapter VMware Network Adapter VMnet8:
  Connection-specific DNS Suffix . :
  Link-local IPv6 Address . . . . . : fe80::63e9:8d49:4519:f87%22
  IPv4 Address . . . . . : 192.168.186.1
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . :

Wireless LAN adapter Wi-Fi:
  Connection-specific DNS Suffix . :
  Link-local IPv6 Address . . . . . : fe80::4fed:c154:c52:ee24%17
  IPv4 Address . . . . . : 10.0.0.8
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . :

Ethernet adapter Bluetooth Network Connection:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

C:\>
```

4. netstat:

- a. The netstat command, meaning network statistics, is a Command Prompt command used to display very detailed information about how your computer is communicating with other computers or network devices.
- b. Specifically, the netstat command can show details about individual network connections, overall and protocol-specific networking statistics, and much more, all of which could help troubleshoot certain kinds of networking issues.

Command: netstat

Example: netstat

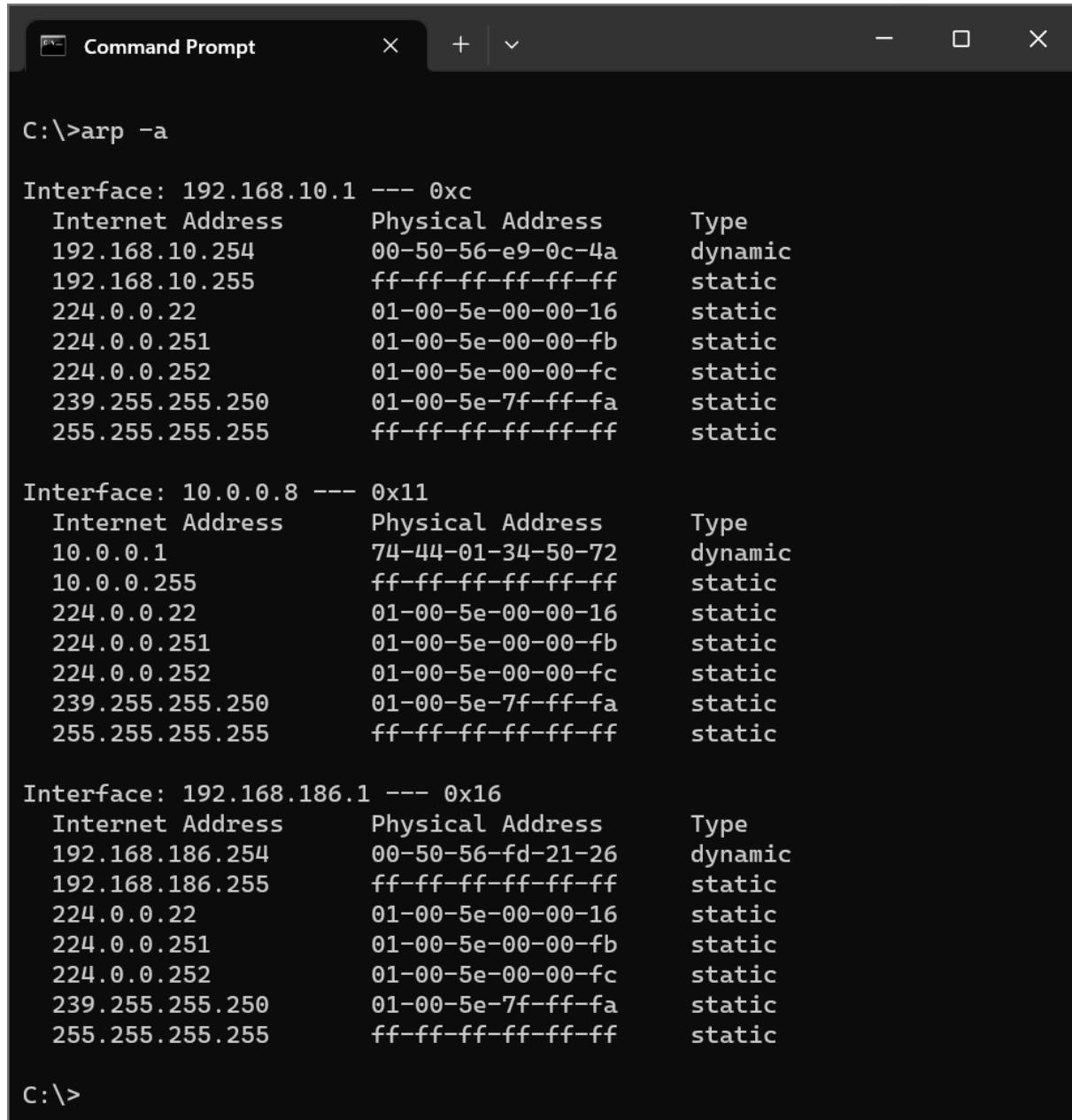
Proto	Local Address	Foreign Address	State
TCP	10.0.0.8:50882	20.198.119.143:https	ESTABLISHED
TCP	10.0.0.8:50951	20.198.119.143:https	ESTABLISHED
TCP	10.0.0.8:50953	20.198.119.143:https	ESTABLISHED
TCP	10.0.0.8:51058	20.210.169.67:https	ESTABLISHED
TCP	10.0.0.8:51178	52.108.44.14:https	ESTABLISHED
TCP	10.0.0.8:51201	se-in-f188:5228	ESTABLISHED
TCP	10.0.0.8:51595	1drv:https	ESTABLISHED
TCP	10.0.0.8:51600	52.98.123.210:https	ESTABLISHED
TCP	10.0.0.8:51601	52.98.123.210:https	ESTABLISHED
TCP	10.0.0.8:51604	52.98.123.210:https	ESTABLISHED
TCP	10.0.0.8:51605	52.98.123.210:https	ESTABLISHED
TCP	10.0.0.8:51606	52.98.123.210:https	ESTABLISHED
TCP	10.0.0.8:51607	52.98.123.210:https	ESTABLISHED
TCP	10.0.0.8:51656	137.193.65.237:https	ESTABLISHED
TCP	10.0.0.8:51702	1drv:https	ESTABLISHED
TCP	10.0.0.8:51704	1drv:https	ESTABLISHED
TCP	10.0.0.8:51708	20.189.173.2:https	TIME_WAIT
TCP	10.0.0.8:51710	20.50.73.9:https	ESTABLISHED
TCP	10.0.0.8:51711	20.189.173.7:https	ESTABLISHED
TCP	10.0.0.8:51713	40.79.141.154:https	TIME_WAIT
TCP	127.0.0.1:49669	LAPTOP-BAJVM52A:49670	ESTABLISHED
TCP	127.0.0.1:49670	LAPTOP-BAJVM52A:49669	ESTABLISHED
TCP	127.0.0.1:49671	LAPTOP-BAJVM52A:49672	ESTABLISHED
TCP	127.0.0.1:49672	LAPTOP-BAJVM52A:49671	ESTABLISHED

5. arp:

- a. ARP command to view and modify the ARP (Address Resolution Protocol) table entries on the local computer.
- b. This may display all the known connections on your local area network segment (if they have been active and, in the cache,). The arp command is useful for viewing the ARP cache and resolving address resolution problems.

Command: arp -a (to display the arp table)

Example: arp -a (to display the arp table)



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command "arp -a" is entered at the prompt. The output displays three network interfaces (0xc, 0x11, and 0x16) with their respective Internet Addresses, Physical Addresses, and Type (dynamic or static).

Interface	Internet Address	Physical Address	Type
0xc	192.168.10.254	00-50-56-e9-0c-4a	dynamic
0xc	192.168.10.255	ff-ff-ff-ff-ff-ff	static
0x11	224.0.0.22	01-00-5e-00-00-16	static
0x11	224.0.0.251	01-00-5e-00-00-fb	static
0x11	224.0.0.252	01-00-5e-00-00-fc	static
0x11	239.255.255.250	01-00-5e-7f-ff-fa	static
0x11	255.255.255.255	ff-ff-ff-ff-ff-ff	static
0x16	10.0.0.1	74-44-01-34-50-72	dynamic
0x16	10.0.0.255	ff-ff-ff-ff-ff-ff	static
0x16	224.0.0.22	01-00-5e-00-00-16	static
0x16	224.0.0.251	01-00-5e-00-00-fb	static
0x16	224.0.0.252	01-00-5e-00-00-fc	static
0x16	239.255.255.250	01-00-5e-7f-ff-fa	static
0x16	255.255.255.255	ff-ff-ff-ff-ff-ff	static