Batch: B-1 **Experiment Number: 2 - Application layer protocols.**

Roll Number: 16010422234 **Name: Chandana Ramesh Galgali**

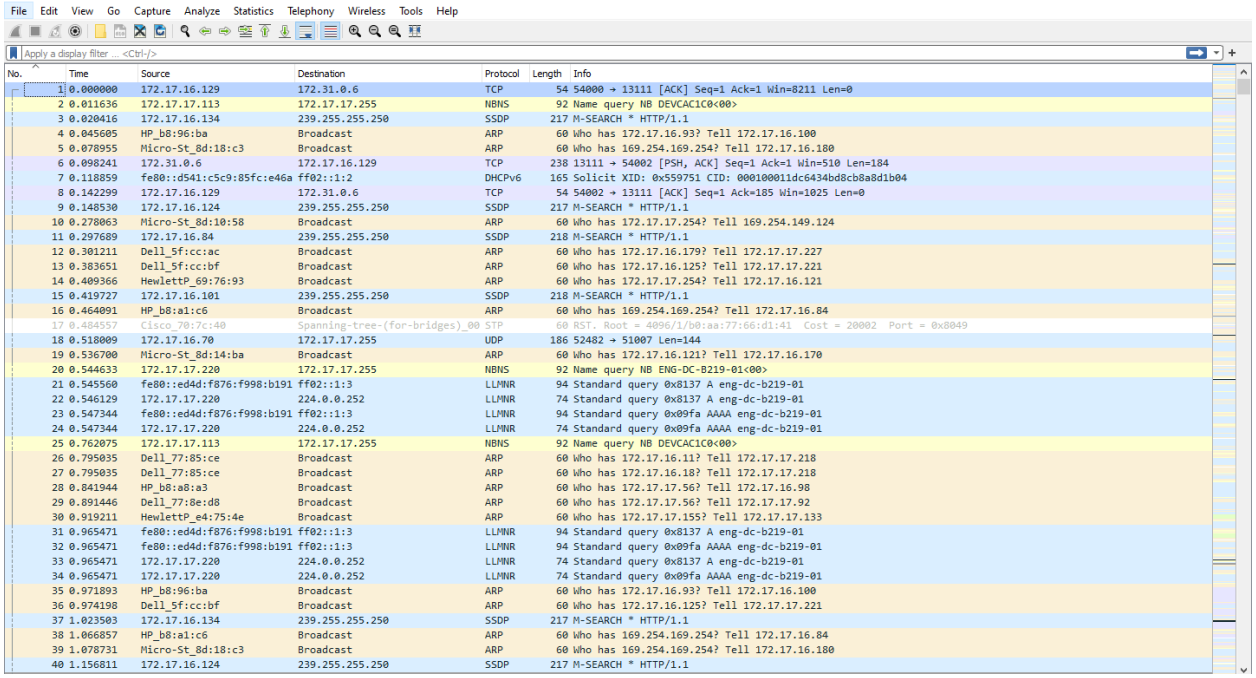
Aim of the Experiment: To explore application layer protocols with packet analysis using Wireshark.

Program/ Steps:

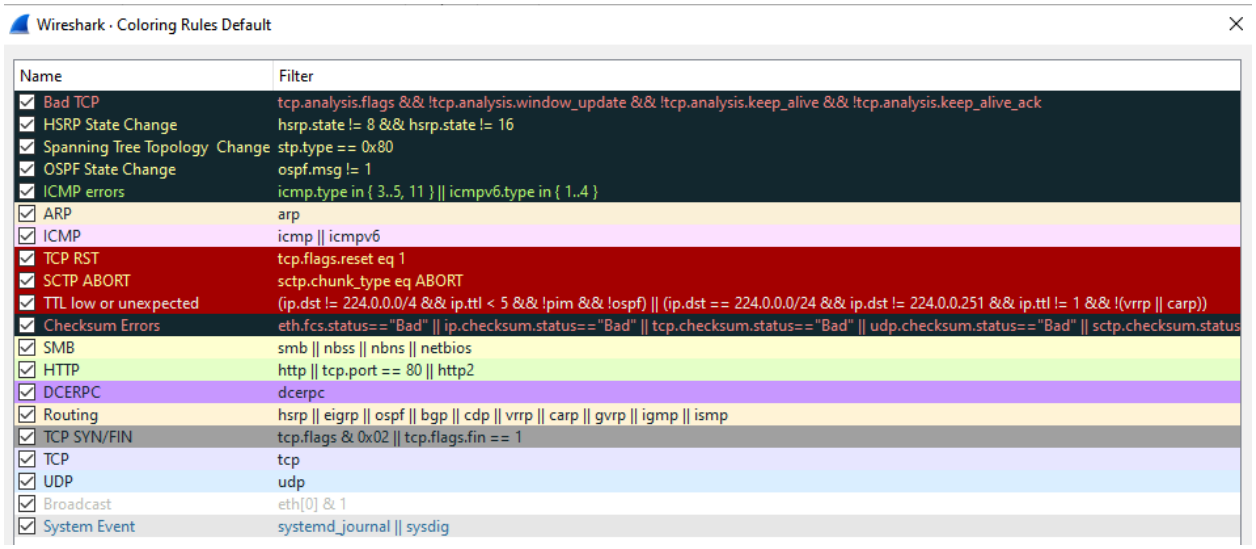
1. Start the machine as an administrator.
2. Start the internet.
3. Go to the official website of Wireshark. (www.wireshark.org) and download the old stable version of Wireshark for 32 bit windows operating system.
4. After successful installation you will get the blue icon of Wireshark on the desktop.
5. Click on the icon and start the software.
6. Choose an interface and start capturing the packets.
7. Study the packet details of any one application layer protocols.
8. Understand color code in detail.
9. Perform the statistics for captured application layer protocol packets. (Every student should perform for different protocols.)
10. Show the output to the teacher and get it approved.

Output/Result:

1. Capturing a packet.



2. Color coding of different protocols.



3. Statistics for the application layer protocol you have chosen.

DNS:

Wireshark · Packet 520 · cg.pcapng

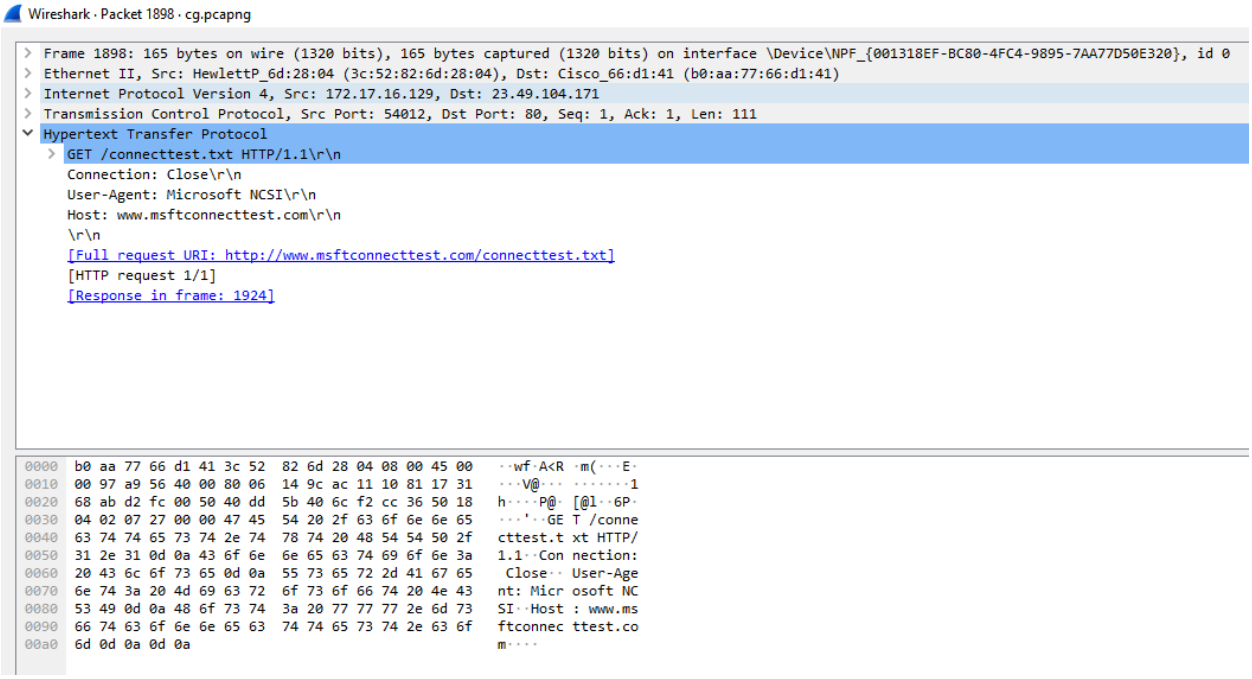
```
> Frame 520: 227 bytes on wire (1816 bits), 227 bytes captured (1816 bits) on interface \Device\NPF_{001318EF-BC80-4FC4-9895-7AA77D50E320}, id 0
> Ethernet II, Src: Cisco_66:d1:41 (b0:aa:77:66:d1:41), Dst: HewlettP_6d:28:04 (3c:52:82:6d:28:04)
> Internet Protocol Version 4, Src: 172.31.0.25, Dst: 172.17.16.129
> User Datagram Protocol, Src Port: 53, Dst Port: 49868
v Domain Name System (response)
    Transaction ID: 0x409e
  > Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 5
    Authority RRs: 0
    Additional RRs: 0
  > Queries
  > Answers
    [Request In: 519]
    [Time: 0.000578000 seconds]
```

```
0000  3c 52 82 6d 28 04 b0 aa  77 66 d1 41 08 00 45 00   <R·m(··· wf·A··E·
0010  00 d5 43 f2 40 00 7e 11  4f 5b ac 1f 00 19 ac 11   ··C·@·~· O[······
0020  10 81 00 35 c2 cc 00 c1  54 a5 40 9e 81 80 00 01   ···5···· T·@·····
0030  00 05 00 00 00 00 03 77  77 77 0f 6d 73 66 74 63   ·······w ww·msftc
0040  6f 6e 6e 65 63 74 74 65  73 74 03 63 6f 6d 00 00   onnectte st·com··
0050  01 00 01 c0 0c 00 05 00  01 00 00 00 94 00 1d 08   ········ ········
0060  6e 63 73 69 2d 67 65 6f  0e 74 72 61 66 66 69 63   ncsi-geo ·traffic
0070  6d 61 6e 61 67 65 72 03  6e 65 74 00 c0 35 00 05   manager· net··5··
0080  00 01 00 00 00 6c 00 1d  03 77 77 77 08 6d 73 66   ·····l·· ·www·msf
0090  74 6e 63 73 69 03 63 6f  6d 09 65 64 67 65 73 75   tncsi·co m·edgesu
00a0  69 74 65 c0 4d c0 5e 00  05 00 01 00 00 00 05 00   ite·M·^· ········
00b0  12 05 61 31 39 36 31 02  67 32 06 61 6b 61 6d 61   ··a1961· g2·akama
00c0  69 c0 4d c0 87 00 01 00  01 00 00 00 27 00 04 17   i·M····· ····'···
00d0  31 68 ab c0 87 00 01 00  01 00 00 00 27 00 04 17   1h····· ····'···
00e0  31 68 aa                                          1h·
```

Wireshark · DNS · cg.pcapng

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
Total Packets	26				0.0004	100%	0.1400	42.754
rcode	26				0.0004	100.00%	0.1400	42.754
No error	26				0.0004	100.00%	0.1400	42.754
opcodes	26				0.0004	100.00%	0.1400	42.754
Standard query	26				0.0004	100.00%	0.1400	42.754
Query/Response	26				0.0004	100.00%	0.1400	42.754
Response	13				0.0002	50.00%	0.0700	42.784
Query	13				0.0002	50.00%	0.0700	42.754
Query Type	26				0.0004	100.00%	0.1400	42.754
AAAA (IPv6 Address)	4				0.0001	15.38%	0.0400	42.790
A (Host Address)	22				0.0004	84.62%	0.1000	42.754
Class	26				0.0004	100.00%	0.1400	42.754
IN	26				0.0004	100.00%	0.1400	42.754
Service Stats	0				0.0000	100%	-	-
request-response time (msec)	12	1206.52	0.310000	11626.712891	0.0002		0.0700	42.784
no. of unsolicited responses	0				0.0000		-	-
no. of retransmissions	1				0.0000		0.0100	53.968
Response Stats	0				0.0000	100%	-	-
no. of questions	26	1.00	1	1	0.0004		0.1400	42.784
no. of authorities	26	0.15	0	1	0.0004		0.1400	42.784
no. of answers	26	4.54	1	8	0.0004		0.1400	42.784
no. of additionals	26	0.00	0	0	0.0004		0.1400	42.784
Query Stats	0				0.0000	100%	-	-
Qname Len	13	22.08	16	31	0.0002		0.0700	42.754
Label Stats	0				0.0000		-	-
4th Level or more	4				0.0001		0.0200	51.229
3rd Level	9				0.0001		0.0600	42.754
2nd Level	0				0.0000		-	-
1st Level	0				0.0000		-	-
Payload size	26	103.73	34	191	0.0004	100%	0.1400	42.754

HTTP:



Wireshark · Packet Counter · cg.pcapng

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
Total HTTP Packets	312				0.0030	100%	0.0400	74.205
Other HTTP Packets	0				0.0000	0.00%	-	-
HTTP Response Packets	3				0.0000	0.96%	0.0100	13.893
???: broken	0				0.0000	0.00%	-	-
5xx: Server Error	0				0.0000	0.00%	-	-
4xx: Client Error	0				0.0000	0.00%	-	-
3xx: Redirection	0				0.0000	0.00%	-	-
2xx: Success	3				0.0000	100.00%	0.0100	13.893
200 OK	3				0.0000	100.00%	0.0100	13.893
1xx: Informational	0				0.0000	0.00%	-	-
HTTP Request Packets	309				0.0030	99.04%	0.0400	74.205
SEARCH	306				0.0030	99.03%	0.0400	75.682
GET	3				0.0000	0.97%	0.0100	13.577

Post Lab Question-Answers:

1. What is the difference between Wireshark software and NMAP software?

Ans: Wireshark and Nmap are both popular network analysis tools, but they serve different purposes and have distinct features. Here's a breakdown of the key differences between the two:

1. Functionality:

- Wireshark: Wireshark is a network protocol analyzer that captures and analyzes network traffic in real-time. It allows you to inspect packets at a granular level, providing detailed information about network protocols, packet headers, and payload contents. Wireshark is primarily used for network troubleshooting, protocol analysis, and security auditing.
- Nmap: Nmap (Network Mapper) is a network scanning tool used for host discovery, port scanning, and service enumeration. It helps identify open ports, detect operating systems, and gather information about network services running on target systems. Nmap is commonly used for network reconnaissance, vulnerability assessment, and penetration testing.

2. Scope:
    - Wireshark: Wireshark focuses on capturing and analyzing network traffic on a specific network interface or network segment. It provides a comprehensive view of the network communication between devices, allowing you to inspect individual packets and analyze network behavior.
    - Nmap: Nmap is designed for scanning and mapping networks to identify hosts, open ports, and services. It can scan large networks or specific IP ranges to discover active hosts and gather information about their network services.
3. User Interface:
    - Wireshark: Wireshark provides a graphical user interface (GUI) that allows users to interactively capture, filter, and analyze network traffic. It offers powerful filtering capabilities, customizable displays, and various analysis tools to dissect captured packets.
    - Nmap: Nmap primarily operates from the command-line interface (CLI), although there are graphical frontends available. It provides a wide range of command-line options and scripting capabilities, making it highly flexible and suitable for automation and scripting purposes.
4. Use Cases:
    - Wireshark: Wireshark is commonly used by network administrators, security analysts, and developers for troubleshooting network issues, analyzing network protocols, and investigating security incidents. It helps in diagnosing network performance problems, identifying malicious activities, and understanding network behavior.
    - Nmap: Nmap is widely used by network administrators, security professionals, and ethical hackers for network exploration, vulnerability scanning, and penetration testing. It helps in identifying open ports, assessing network security, and discovering potential vulnerabilities in systems.

In summary, Wireshark is primarily used for capturing and analyzing network traffic, while Nmap focuses on network scanning and host discovery. Both tools have their own strengths and are often used together to gain a comprehensive understanding of network behavior and security.

2. At which of the OSI layer Wireshark runs?

Ans: Wireshark operates at the lowest three layers of the OSI (Open Systems Interconnection) model, namely the Physical layer, Data Link layer, and Network layer.

1. Physical Layer: Wireshark can capture and analyze raw network traffic at the physical layer, including bits and electrical signals transmitted over the network medium. However, it is important to note that Wireshark typically captures data after it has been converted into a more readable format by the network interface card (NIC).

2. Data Link Layer: Wireshark can dissect and interpret data link layer protocols such as Ethernet, Wi-Fi (802.11), and others. It can analyze the frames exchanged between devices on a local network, including MAC addresses, frame types, and error detection.

3. Network Layer: Wireshark can capture and analyze network layer protocols such as IP (Internet Protocol), ICMP (Internet Control Message Protocol), and routing protocols like OSPF (Open Shortest Path First). It can provide insights into IP addressing, routing information, and network behavior.

While Wireshark primarily focuses on these lower layers, it also has the capability to analyze higher-layer protocols such as TCP (Transmission Control Protocol), UDP (User Datagram Protocol), HTTP (Hypertext Transfer Protocol), and many others. This allows for comprehensive analysis of network traffic across multiple layers of the OSI model.

3. Just write down the names of the softwares which have similar functionality as Wireshark. (open source or proprietary)

Ans:

1. tcpdump: tcpdump is a command-line packet analyzer that captures and analyzes network traffic. It is available for various operating systems, including Linux, macOS, and Windows.
2. Microsoft Network Monitor: Microsoft Network Monitor is a proprietary packet analyzer developed by Microsoft. It allows for capturing and analyzing network traffic on Windows-based systems.
3. Tshark: Tshark is a command-line tool that is part of the Wireshark suite. It provides similar functionality to Wireshark but operates in a terminal environment, making it suitable for scripting and automation.
4. Colasoft Capsa: Capsa is a proprietary network analyzer that offers real-time packet capturing and analysis. It provides a graphical user interface and is available for Windows operating systems.
5. NetworkMiner: NetworkMiner is an open-source network forensic analysis tool that captures and parses network traffic. It focuses on extracting files, emails, and other artifacts from captured packets.

6. EtherApe: EtherApe is an open-source graphical network monitor that visualizes network activity in real-time. It provides a visual representation of network traffic flows and can be used for basic packet analysis.
7. PRTG Network Monitor: PRTG is a proprietary network monitoring tool that includes packet sniffing capabilities. It allows for capturing and analyzing network traffic, along with monitoring various network parameters.

Outcomes:

Enumerate the layers of the OSI model and TCP/IP model, their functions and Protocols.

Conclusion (based on the Results and outcomes achieved):

The experiment successfully explored application layer protocols using packet analysis with Wireshark. The insights gained from this analysis can contribute to network troubleshooting, performance optimization, and security enhancement.

References:

Books/ Journals/ Websites:

1. Behrouz A Forouzan, Data Communication and Networking, Tata Mc Graw hill, India, 4th Edition
2. A. S. Tanenbaum, ”Computer Networks”, 4th edition, Prentice Hall
3. Behrouz A Forouzan, Data Communication and Networking, Tata Mc Graw hill, India, 4th Edition
4. A. S. Tanenbaum, ”Computer Networks”, 4th edition, Prentice Hall