

Network Monitoring Tool in Shell Programming

Mohit Jaiswal¹, Pallavi Kaur Jauhar^{2*}, Shubhendra Singh³, Rashi Agarwal⁴

Abstract

Network monitoring is the systematic effort of a computer network to detect slow or failing network components such as overloaded or crashed/frozen servers, failing routers, failed switches, or other problematic devices. The network monitoring system notifies the network administrator in the event of a network failure or other outage. Network management is a subset of network monitoring. Network monitoring is typically accomplished through the use of software applications and tools. Network monitoring services are widely used to determine whether a given Web server is operational and properly connected to networks around the world. Many servers that perform this function offer a more comprehensive view of the Internet and networks. While constantly monitoring the reliability of a network and searching for trends, the monitoring system tracks and logs network parameters. Among them are the data transmission rate (throughput), error rates, downtime/uptime, use-time percentages, and response time to user and automated inputs and requests. When predefined parameter thresholds are exceeded, alarms are triggered, and network fault management processes are initiated.

Keywords: Shell Script, Operating System, MySQL, LAMP, tcpdump, User Datagram Protocol (UDP)

INTRODUCTION

In the following project we have developed a packet analyzer tool which is used capture packets and provide filtered information about the packets. A packet analyzer (also known as a packet sniffer) is a piece of computer hardware or software that can intercept and log network traffic. The process of intercepting and logging traffic is known as packet capture. As data streams pass through the network, the sniffer captures each packet and, if necessary, decodes the raw data of the packet, displaying the values of various fields in the packet, and analysing its content in accordance with the relevant RFC or other specifications.

A wireless analyzer, also known as a Wi-Fi analyzer, is a packet analyzer used to intercept traffic on wireless networks. A packet analyzer is also known as a network analyzer or protocol analyzer, though these terms have different meanings.

*Author for Correspondence

Pallavi Kaur Jauhar

¹⁻³Student, Department of Information Technology, University Institute of Engineering and Technology, Kanpur, Uttar Pradesh, India

⁴HOD, Department of Information Technology, University Institute of Engineering and Technology, Kanpur, Uttar Pradesh, India

Received Date: March 14, 2022

Accepted Date: March 31, 2022

Published Date: April 07, 2022

Citation: Mohit Jaiswal, Pallavi Kaur Jauhar, Shubhendra Singh, Rashi Agarwal. Network Monitoring Tool in Shell Programming. Journal of Advances in Shell Programming. 2021; 8(3): 19–39p.

WHY PACKET ANALYZERS ARE USED?

A packet analyzer can be used for both legal and illegal purposes. A legal packet analyzer is a commercial device that aids in network management and maintenance as well as network security. It is also used to diagnose network backup systems and to inspect the network system for security flaws.

A hacker will use an illegal packet analyzer to gain unauthorized access to sensitive information and data on a network. An illegal packet analyzer is installed without the IT administrator's

knowledge and hides in various areas of the network to spy on and steal information packets that pass over the network.

WHO ALL USES PACKET ANALYZER?

ISPs use packet analyzers to monitor all your activities, including:

- Who is receiver of your email
- What is content of that email
- What you download
- Sites you visit
- What you looked on that website
- Downloads from a site
- Streaming events like video, audio, etc.

Advertising agencies or internet advertising agencies are compensated based on:

- Number of ads shown by them.
- Number of clicks on their ads also called PPC (pay per click).

Government agencies use packet sniffing to:

- Ensure security of data over the network.
- Track an organization's unencrypted data.

PROJECT GOALS

Main objective of project was to develop a packet analyzer tool which can analyze LAN network packets. There are different types of packets such as TCP, UDP, ARP, ICMP etc. In this project we are analyzing only TCP, UDP and ARP packets. There is variety of information inside packets such as Flags, Sequence number, length, acknowledgement, source MAC address, destination MAC address, source IP address, destination IP address etc. A packet analyzer used for intercepting traffic on wireless networks is known as a wireless analyzer or Wi-Fi analyzer.

We have developed a webpage which is used to create systematic table for different kind of packets captured from LAN network. Shell script plays an important role in backend of filtering packets according to requirement of user. I have implemented PHP for server-side scripting language.

Bundled Software

Bundled software can be a collection of single software program sold together, or it can be one or more software program sold in conjunction with a piece of hardware [1]. Operating systems, utilities, and accessories sold with desktop or laptop computers, as well as mobile devices, are examples of bundled software. Other types of bundled software include multiple programs sold as a single software service or product with multiple uses.

Bundled software *LAMP* is being used to develop the project.

The Linux operating system, the Apache HTTP Server, the MySQL relational database management system (RDBMS), and the PHP programming language are all acronyms for LAMP, which is an archetypal model of web service stacks. The LAMP components are mostly interchangeable and not restricted to the original selection. LAMP is a solution stack that can be used to create dynamic web sites and web applications (Figure 1).

- L: Linux (Operating System)
- A: Apache (HTTP Server)
- M: MySQL (RDBMS)
- P: PHP (Server-Side Scripting Language)

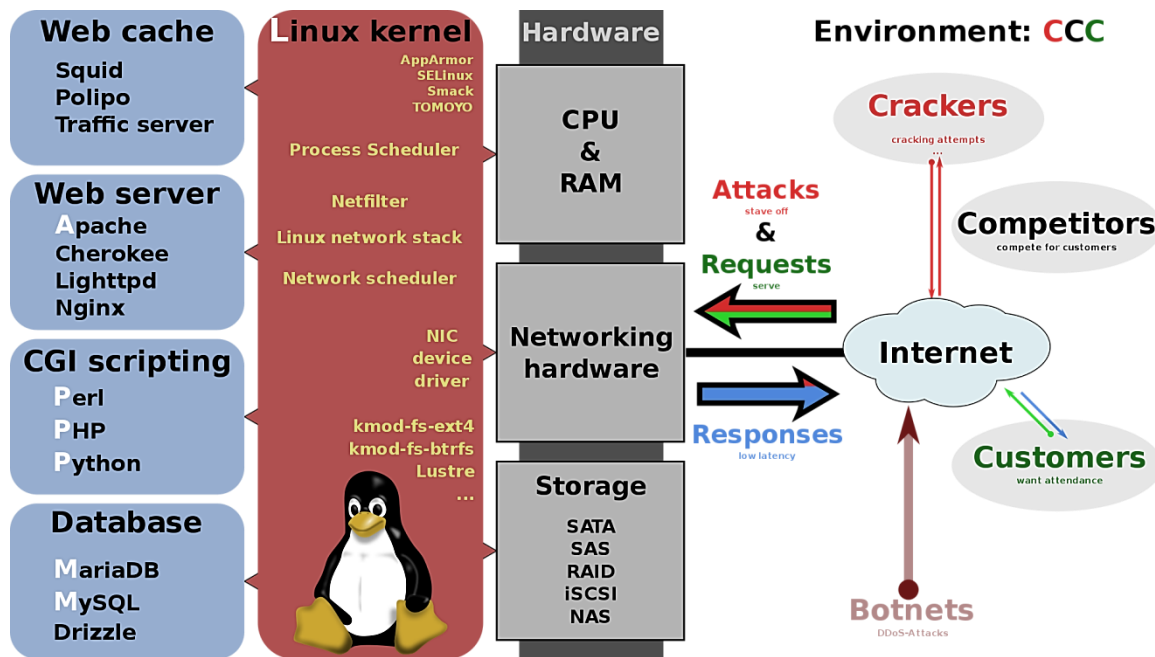


Figure 1. A high-level overview of LAMPs building blocks and overall system environment.

Linux

Linux is a family of open-source Unix-like operating systems based on the Linux kernel, which was first released by Linus Torvalds on September 17, 1991. Linux is usually packaged as part of a Linux distribution.

Ubuntu is a Debian-based free and open-source Linux distribution. Ubuntu is available in three editions: desktop, server, and core (for internet of things devices and robots). Ubuntu is a popular cloud computing operating system that supports OpenStack. Ubuntu is released every six months, with LTS releases every two years. The most recent short-term support release is 19.04 ("Disco Dingo"), and the most recent long-term support release is 18.04 LTS ("Bionic Beaver"), which is supported until 2028.

Ubuntu version 16.04 LTS is being used in the project to develop it.

Apache

Apache is an open-source and free web server software that powers roughly 46 percent of the world's websites. The Apache Software Foundation maintains and develops the Apache HTTP Server, which is the official name. It enables website owners to serve content on the internet, hence the term "web server."

Apache is a web server that is software that runs on a server rather than a physical server. Its role is to establish a connection between a server and website visitors' browsers (Firefox, Google Chrome, Safari, and so on) while delivering files back and forth between them (client-server structure). Apache is a cross-platform software; therefore, it works on both Unix and Windows servers.

MySQL

MySQL is free and open-source software under the terms of the GNU General Public License and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by MySQL AB, a Swedish company that was purchased by Sun Microsystems (now Oracle Corporation). MySQL is written in the C and C++ programming languages. Its SQL parser is written in yacc, but it employs a custom lexical analyzer. AIX, BSDi, FreeBSD, HP-UX, eComStation,

i5/OS, IRIX, Linux, macOS, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Oracle Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos, and Tru64 are among the system platforms supported by MySQL. MySQL for OpenVMS is also available.

PHP

Hypertext Preprocessor (or simply PHP) is a general-purpose programming language originally designed for web development. Rasmus Lerdorf created it in 1994, and The PHP Group now produces the PHP reference implementation. PHP was originally an abbreviation for Personal Home Page, but it is now abbreviated to the recursive initialism PHP: Hypertext Preprocessor. PHP code can be run via a command line interface (CLI), embedded in HTML code, or used in conjunction with a variety of web template systems, web content management systems, and web frameworks. PHP code is typically processed by a PHP interpreter, which is either a web server module or a Common Gateway Interface (CGI) executable. The results of the interpreted and executed PHP code, which can be any type of data, including images, are combined with the generated web page by the web server.

After the installation of LAMP next step is to capture packets from LAN, here shell script come into play. There are certain commands which are used capture packets and refine them into proper tabular form.

SHELL SCRIPTING

Shells are typically interactive, which means they accept commands from users and execute them. However, if we want to run a series of commands on a regular basis, we must enter all of them into the terminal each time. Because shell can accept commands from a file as input, we can write these commands in a file and execute them in shell to avoid this repetitive work. Shell Scripts and Shell Programs are the names given to these files. Shell scripts are similar to MS-DOS batch files. Each shell script is saved with the .sh file extension, for example, myscrip.sh.

What is Kernel?

The kernel is a computer program that serves as the heart of a computer's operating system, controlling everything in the system [2]. It is in charge of the following Linux system resources:

- File management
- Process management
- I/O management
- Memory management
- Device management

What is Shell?

A shell is a type of user programme that allows users to interact with operating system services. Shell accepts user-supplied human-readable commands and converts them into something that the kernel can understand. It's a command language interpreter that runs commands read from input devices like keyboards or files (Figure 2) [3].

Command Line Shell

A command line interface is used to access the shell. A special programme called Terminal in Linux/macOS, or Command Prompt in Windows OS is provided to type in human-readable commands such as "cat", "ls", and so on, which are then executed. The result is then displayed to the user on the terminal [4].

Commands Used

tcpdump

It prints a description of the contents of packets on a network interface that match the Boolean expression; the description is followed by a time stamp, which is printed as hours, minutes, seconds,

and fractions of a second since midnight by default. It is used to capture packets from the specified network interface (Figure 3).

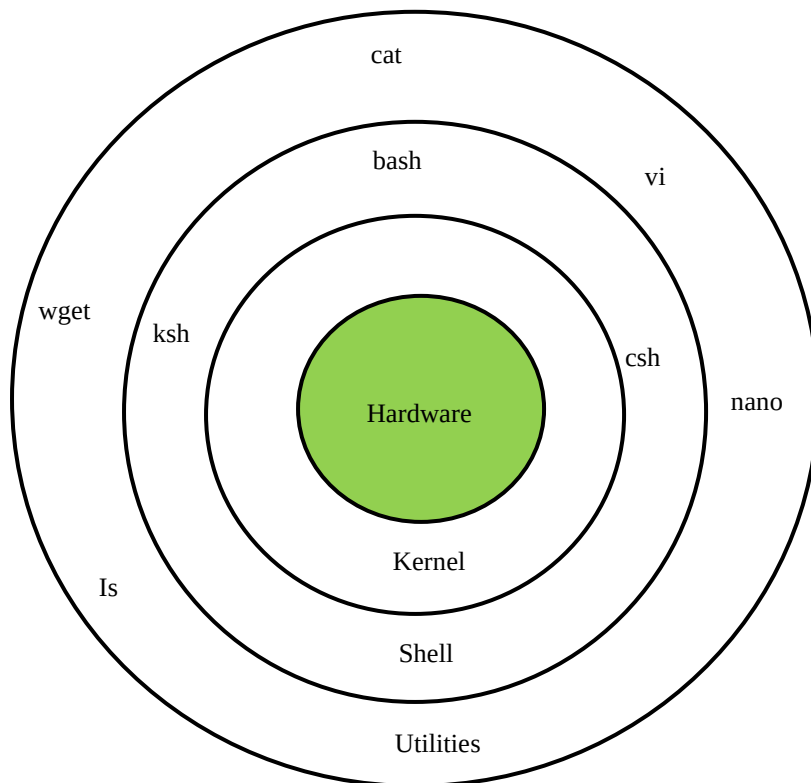


Figure 2. Lifecycle of shell scripting.

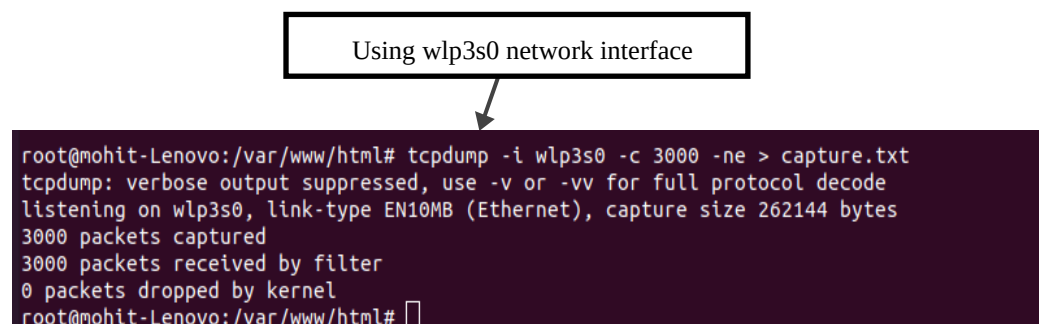


Figure 3. Packet captured using tcpdump.

Step 1

Number of packets captured using tcpdump is 3000 and writing into file named as main_packet.pcap. So, command for following function is:

tcpdump -i wlp3s0 -c 3000 -w capture.txt

Where,

- i: Flag represents the interface
- c: Flag captured specified number of packet
- w: Flag to save packet in file

Step 2

TCP, UDP, and ARP packets are of particular interest to us. So, the next step is to extract TCP, UDP, and ARP from the main packet.pcap file (Figure 4).

To separate out TCP packet:

```
tcpdump -r main_packet.pcap -w tcp_packet.pcap tcp
```

Where,

-r: Flag reads packets from file

tcp is used to specify type of packet (here tcp) to filtered from file.

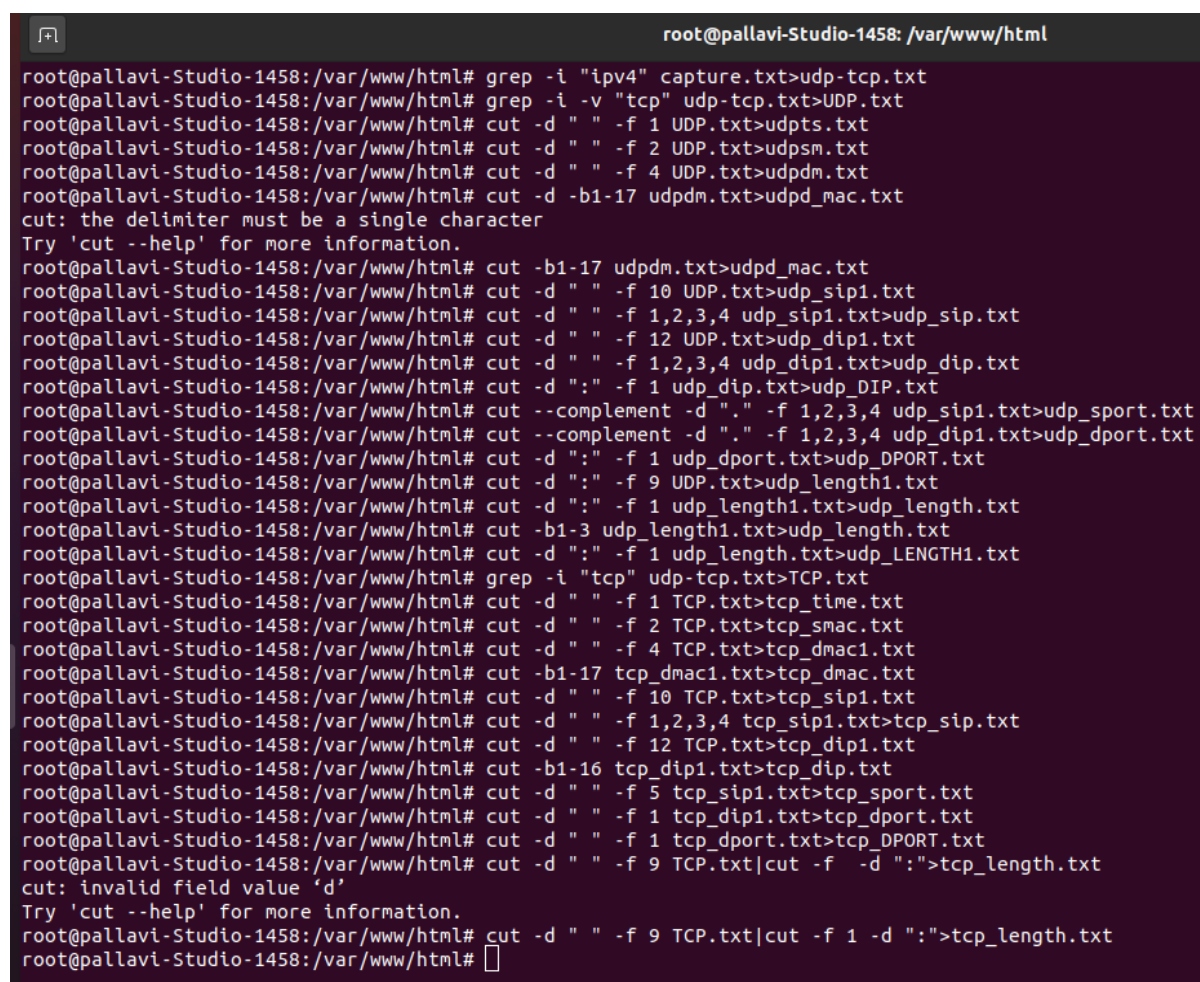
A new file tcp_packet.pcap file is created which store all tcp packet captured.

Similarly, for UDP,

```
tcpdump -r main_packet.pcap -w udp_packet.pcap udp
```

Similarly for ARP,

```
tcpdump -r main_packet.pcap -w arp_packet.pcap arp
```



```

root@pallavi-Studio-1458: /var/www/html
root@pallavi-Studio-1458:/var/www/html# grep -i "ipv4" capture.txt>udp-tcp.txt
root@pallavi-Studio-1458:/var/www/html# grep -i -v "tcp" udp-tcp.txt>UDP.txt
root@pallavi-Studio-1458:/var/www/html# cut -d " " -f 1 UDP.txt>udppts.txt
root@pallavi-Studio-1458:/var/www/html# cut -d " " -f 2 UDP.txt>udpdm.txt
root@pallavi-Studio-1458:/var/www/html# cut -d " " -f 4 UDP.txt>udpdm.txt
root@pallavi-Studio-1458:/var/www/html# cut -d -b1-17 udpdm.txt>udpdm_mac.txt
cut: the delimiter must be a single character
Try 'cut --help' for more information.
root@pallavi-Studio-1458:/var/www/html# cut -b1-17 udpdm.txt>udpdm_mac.txt
root@pallavi-Studio-1458:/var/www/html# cut -d " " -f 10 UDP.txt>udp_sip1.txt
root@pallavi-Studio-1458:/var/www/html# cut -d " " -f 1,2,3,4 udp_sip1.txt>udp_sip.txt
root@pallavi-Studio-1458:/var/www/html# cut -d " " -f 12 UDP.txt>udp_dip1.txt
root@pallavi-Studio-1458:/var/www/html# cut -d " " -f 1,2,3,4 udp_dip1.txt>udp_dip.txt
root@pallavi-Studio-1458:/var/www/html# cut -d ":" -f 1 udp_dip.txt>udp_DIP.txt
root@pallavi-Studio-1458:/var/www/html# cut --complement -d "." -f 1,2,3,4 udp_sip1.txt>udp_sport.txt
root@pallavi-Studio-1458:/var/www/html# cut --complement -d "." -f 1,2,3,4 udp_dip1.txt>udp_dport.txt
root@pallavi-Studio-1458:/var/www/html# cut -d ":" -f 1 udp_dport.txt>udp_DPORT.txt
root@pallavi-Studio-1458:/var/www/html# cut -d ":" -f 9 UDP.txt>udp_length1.txt
root@pallavi-Studio-1458:/var/www/html# cut -d ":" -f 1 udp_length1.txt>udp_length.txt
root@pallavi-Studio-1458:/var/www/html# cut -b1-3 udp_length1.txt>udp_length.txt
root@pallavi-Studio-1458:/var/www/html# cut -d ":" -f 1 udp_length.txt>udp_LENGTH1.txt
root@pallavi-Studio-1458:/var/www/html# grep -i "tcp" udp-tcp.txt>TCP.txt
root@pallavi-Studio-1458:/var/www/html# cut -d " " -f 1 TCP.txt>tcp_time.txt
root@pallavi-Studio-1458:/var/www/html# cut -d " " -f 2 TCP.txt>tcp_smac.txt
root@pallavi-Studio-1458:/var/www/html# cut -d " " -f 4 TCP.txt>tcp_dmac1.txt
root@pallavi-Studio-1458:/var/www/html# cut -b1-17 tcp_dmac1.txt>tcp_dmac.txt
root@pallavi-Studio-1458:/var/www/html# cut -d " " -f 10 TCP.txt>tcp_sip1.txt
root@pallavi-Studio-1458:/var/www/html# cut -d " " -f 1,2,3,4 tcp_sip1.txt>tcp_sip.txt
root@pallavi-Studio-1458:/var/www/html# cut -d " " -f 12 TCP.txt>tcp_dip1.txt
root@pallavi-Studio-1458:/var/www/html# cut -b1-16 tcp_dip1.txt>tcp_dip.txt
root@pallavi-Studio-1458:/var/www/html# cut -d " " -f 5 tcp_sip1.txt>tcp_sport.txt
root@pallavi-Studio-1458:/var/www/html# cut -d " " -f 1 tcp_dip1.txt>tcp_dport.txt
root@pallavi-Studio-1458:/var/www/html# cut -d " " -f 1 tcp_dport.txt>tcp_DPORT.txt
root@pallavi-Studio-1458:/var/www/html# cut -d " " -f 9 TCP.txt|cut -f -d ":">tcp_length.txt
cut: invalid field value 'd'
Try 'cut --help' for more information.
root@pallavi-Studio-1458:/var/www/html# cut -d " " -f 9 TCP.txt|cut -f 1 -d ":">tcp_length.txt
root@pallavi-Studio-1458:/var/www/html#

```

Figure 4. Packet separation.

Step 3

After the packets have been separated, the next step is to filter out important attributes from the packets, which are listed below:

- Time Stamp
- Source MAC address
- Destination MAC address
- Source IP Address

- Source Port
- Destination IP Address
- Destination Port
- Length of packet

To filter the data from file commands used are following:

- Sed
- Cut
- Grep

Sed

In UNIX, it is known as the stream editor. The sed command is used to perform searches, find, and replace, insertion, and deletion. You can use sed to edit files without opening them, which is a much faster way to find and replace something in a file than opening it in VI Editor and then changing it. Sed also supports regular expression, which can be used to perform complex tasks.

Syntax

```
sed OPTIONS... [SCRIPT] [INPUTFILE...]  
sed 's/old character/new character/position' where, s is for substitute
```

Cut

The cut command is used to remove a section from each line of a file. It can be used to cut sections of a line based on their byte position, character, and field. The cut command basically slices a line and extracts the text. It is necessary to include an option with the command or else an error will occur. It divides the file into columns.

Syntax

```
cut OPTION... [FILE]...  
cut -d'delimiter' -f [fields to be extracted] where,  
-d represents delimiter for field  
-f represents column or field to be extracted from file
```

Grep

The grep filter searches a file for a specific character pattern and displays all lines that match that pattern. The regular expression is the pattern that is searched for in the file.

Syntax

```
grep [options] pattern [files]  
grep -v 'pattern' file name  
This command is used to remove all lines from a file that match the pattern.  
grep -h 'pattern' file name  
This command is used to include all lines in a file that match the pattern.
```

Following the discussion of packet types, we will look at step 3 in greater detail.

About Packets

TCP/IP

In network transmission, a packet is the basic unit of information. TCP/IP is the network protocol, or set of rules, that most networks use for communication between devices, and the rules of TCP/IP require information to be split into packets that contain both a segment of data to be transferred and the address to which the data is to be sent [5].

TCP/IP is a two-tiered protocol. Transmission Control Protocol (TCP) manages the breaking down of a message or file into smaller packets that are transmitted over the Internet and received by a TCP

layer that reassembles the packets into the original message. The lower layer, Internet Protocol (IP), handles the address portion of each packet to ensure that it reaches its intended destination (Figures 5–7).

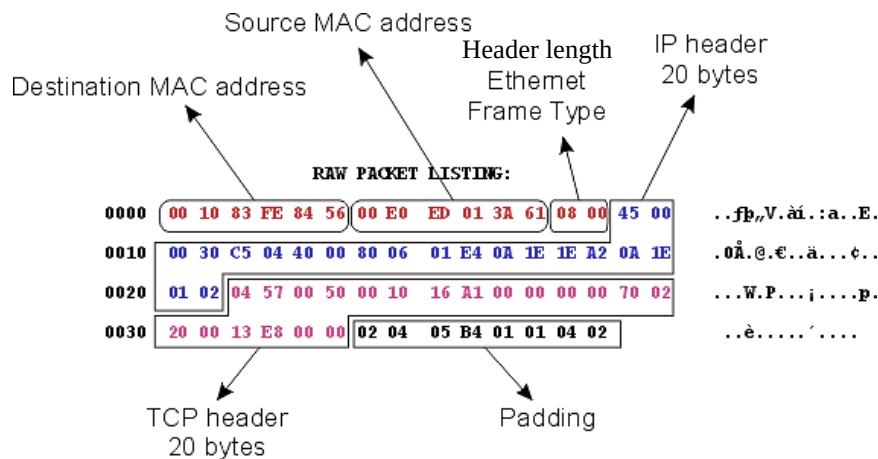


Figure 5. TCP/IP data packet.

Let's look at an example of a packet in more detail:

Explanation of Header Fields

Source and Destination Port

These fields identify the connection's local endpoint. Each host is free to choose how to allocate its own ports, beginning with 1024. The connection is identified by the source and destination socket numbers [6].

Sequence and ACK number: This field is used to assign a sequential number to each byte transferred. This has an advantage over assigning sequence numbers to each packet because data from multiple small packets can be combined into one during retransmission if necessary. The ACK indicates that the next byte expected from the source, not the last byte received, has been received. ACKs are cumulative rather than selective [7]. Although the sequence number space is as large as 32 bits, 17 bits would have sufficed if the packets were delivered in order. If the packets arrive in the correct order, then use the following formula:

$$(\text{sender's window size}) + (\text{receiver's window size}) < (\text{sequence number space})$$

Header length: This field indicates the number of 32-bit words in the TCP header. Because the options field has a variable length, this is required.

Flags: There are 6 one-bit flags.

1. **URG:** This bit indicates whether or not this packet's urgent pointer field is being used.
2. **ACK:** This bit is set to indicate that the packet's ACK number field is valid.
3. **PSH:** This bit indicates that the data has been PUSHed. The receiver is instructed to deliver the data to the application as soon as it arrives and not to buffer it until the buffer is full.
4. **RST:** This flag is used to re-establish a connection that has become confused due to a host crash or other cause. It can also be used to reject an invalid segment or a connection attempt. If there was a connection, it would come to an abrupt end as a result of this.
5. **SYN:** This bit is used to connect things together. SYN=1 and ACK=0 is assigned to the connection request (the first packet in the three-way handshake). SYN=1 and ACK=1 are the values in the connection reply (the second packet in the three-way handshake).
6. **FIN:** This bit is used to disconnect a connection. It indicates that the sender has no new data to send. It will, however, retransmit any lost or delayed packet.

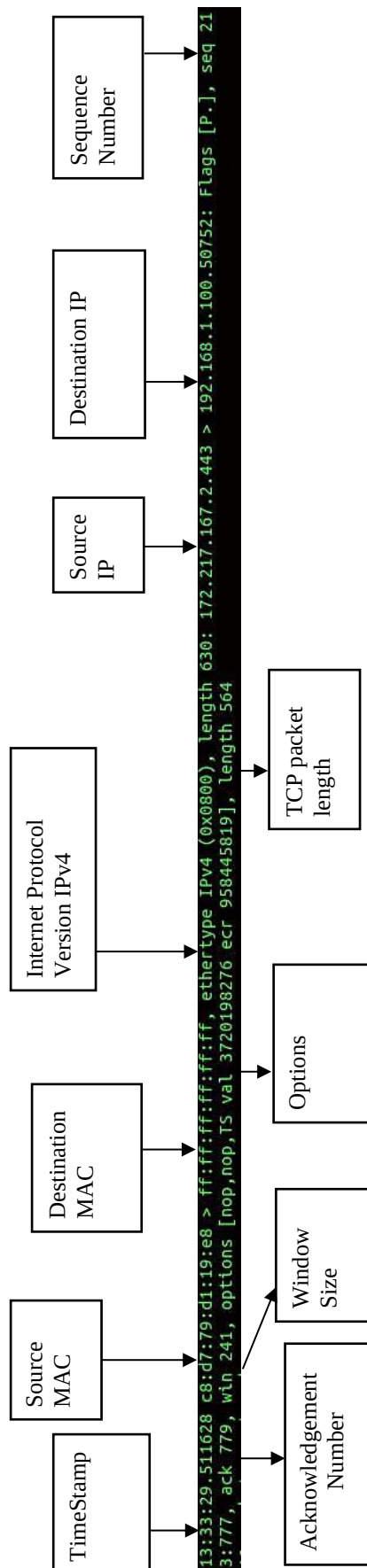
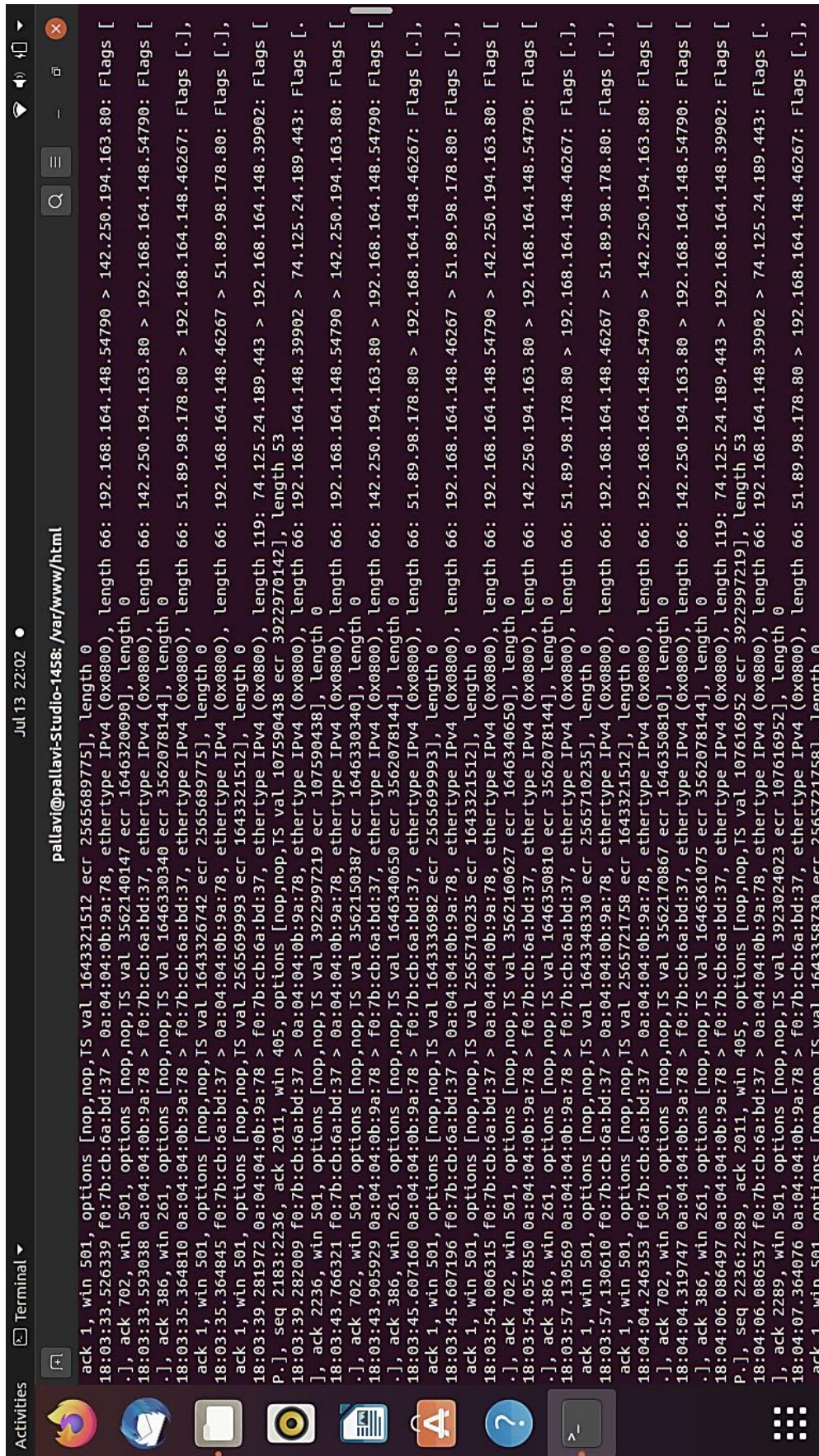


Figure 6. TCP Command.



```

ack 1, win 501, options [nop,nop,TS val 1643321512 ecr 2565689775], length 0
18:03:33.526339 f0:7b:cb:6a:bd:37 > 0a:04:04:0b:9a:78, ethertype IPv4 (0x0800), length 66: 192.168.164.148.54790 > 142.250.194.163.80: Flags [
.] , ack 702, win 501, options [nop,nop,TS val 3562140147 ecr 1646320090], length 0
18:03:33.593038 0a:04:04:0b:9a:78 > f0:7b:cb:6a:bd:37, ethertype IPv4 (0x0800), length 66: 142.250.194.163.80 > 192.168.164.148.54790: Flags [
.] , ack 386, win 261, options [nop,nop,TS val 1646330340 ecr 3562078144], length 0
18:03:35.364810 0a:04:04:0b:9a:78 > f0:7b:cb:6a:bd:37, ethertype IPv4 (0x0800), length 66: 51.89.98.178.80 > 192.168.164.148.46267: Flags [
.] ,
ack 1, win 501, options [nop,nop,TS val 1643326742 ecr 2565689775], length 0
18:03:35.364845 f0:7b:cb:6a:bd:37 > 0a:04:04:0b:9a:78, ethertype IPv4 (0x0800), length 66: 192.168.164.148.46267 > 51.89.98.178.80: Flags [
.] ,
ack 1, win 501, options [nop,nop,TS val 2565699993 ecr 1643321512], length 0
18:03:39.281972 0a:04:04:0b:9a:78 > f0:7b:cb:6a:bd:37, ethertype IPv4 (0x0800), length 119: 74.125.24.189.443 > 192.168.164.148.39902: Flags [
P.] , seq 2183:2236, ack 2011, win 405, options [nop,nop,TS val 107590438 ecr 3922970142], length 53
18:03:39.282009 f0:7b:cb:6a:bd:37 > 0a:04:04:0b:9a:78, ethertype IPv4 (0x0800), length 66: 192.168.164.148.39902 > 74.125.24.189.443: Flags [
.] , ack 2236, win 501, options [nop,nop,TS val 3922997219 ecr 107590438], length 0
18:03:43.766321 f0:7b:cb:6a:bd:37 > 0a:04:04:0b:9a:78, ethertype IPv4 (0x0800), length 66: 192.168.164.148.54790 > 142.250.194.163.80: Flags [
.] , ack 702, win 501, options [nop,nop,TS val 3562150387 ecr 1646330340], length 0
18:03:43.905929 0a:04:04:0b:9a:78 > f0:7b:cb:6a:bd:37, ethertype IPv4 (0x0800), length 66: 142.250.194.163.80 > 192.168.164.148.54790: Flags [
.] , ack 386, win 261, options [nop,nop,TS val 1646330650 ecr 3562078144], length 0
18:03:45.607160 0a:04:04:0b:9a:78 > f0:7b:cb:6a:bd:37, ethertype IPv4 (0x0800), length 66: 51.89.98.178.80 > 192.168.164.148.46267: Flags [
.] ,
ack 1, win 501, options [nop,nop,TS val 1643336982 ecr 2565699993], length 0
18:03:45.607196 f0:7b:cb:6a:bd:37 > 0a:04:04:0b:9a:78, ethertype IPv4 (0x0800), length 66: 192.168.164.148.46267 > 51.89.98.178.80: Flags [
.] ,
ack 1, win 501, options [nop,nop,TS val 2565710235 ecr 1643321512], length 0
18:03:54.006315 f0:7b:cb:6a:bd:37 > 0a:04:04:0b:9a:78, ethertype IPv4 (0x0800), length 66: 192.168.164.148.54790 > 142.250.194.163.80: Flags [
.] , ack 702, win 501, options [nop,nop,TS val 3562160627 ecr 1646340650], length 0
18:03:54.057850 0a:04:04:0b:9a:78 > f0:7b:cb:6a:bd:37, ethertype IPv4 (0x0800), length 66: 142.250.194.163.80 > 192.168.164.148.54790: Flags [
.] , ack 386, win 261, options [nop,nop,TS val 1646330810 ecr 3562078144], length 0
18:03:57.130569 0a:04:04:0b:9a:78 > f0:7b:cb:6a:bd:37, ethertype IPv4 (0x0800), length 66: 51.89.98.178.80 > 192.168.164.148.46267: Flags [
.] ,
ack 1, win 501, options [nop,nop,TS val 1643348330 ecr 2565710235], length 0
18:03:57.130610 f0:7b:cb:6a:bd:37 > 0a:04:04:0b:9a:78, ethertype IPv4 (0x0800), length 66: 192.168.164.148.46267 > 51.89.98.178.80: Flags [
.] ,
ack 1, win 501, options [nop,nop,TS val 2565721758 ecr 1643321512], length 0
18:04:04.246353 f0:7b:cb:6a:bd:37 > 0a:04:04:0b:9a:78, ethertype IPv4 (0x0800), length 66: 192.168.164.148.54790 > 142.250.194.163.80: Flags [
.] , ack 702, win 501, options [nop,nop,TS val 3562170867 ecr 1646350810], length 0
18:04:04.319747 0a:04:04:0b:9a:78 > f0:7b:cb:6a:bd:37, ethertype IPv4 (0x0800), length 66: 142.250.194.163.80 > 192.168.164.148.54790: Flags [
.] , ack 386, win 261, options [nop,nop,TS val 1646361075 ecr 3562078144], length 0
18:04:06.086497 0a:04:04:0b:9a:78 > f0:7b:cb:6a:bd:37, ethertype IPv4 (0x0800), length 119: 74.125.24.189.443 > 192.168.164.148.39902: Flags [
P.] , seq 2236:2289, ack 2011, win 405, options [nop,nop,TS val 107616952 ecr 3922997219], length 53
18:04:06.086537 f0:7b:cb:6a:bd:37 > 0a:04:04:0b:9a:78, ethertype IPv4 (0x0800), length 66: 192.168.164.148.39902 > 74.125.24.189.443: Flags [
.] , ack 2289, win 501, options [nop,nop,TS val 3923024023 ecr 107616952], length 0
18:04:07.364076 0a:04:04:0b:9a:78 > f0:7b:cb:6a:bd:37, ethertype IPv4 (0x0800), length 66: 51.89.98.178.80 > 192.168.164.148.46267: Flags [
.] ,
ack 1, win 501, options [nop,nop,TS val 1643358730 ecr 2565721758], length 0

```

Figure 7. Command for shell scripting.

Window Size

TCP handles flow control with a variable-size sliding window. The Window Size field specifies how many bytes may be sent beginning with the acknowledged byte. The sender can send bytes with sequence numbers ranging from (ACK#) to (ACK# + window size - 1). A window size of zero is permissible and indicates that bytes up to and including ACK# - 1 have been received, but the receiver would like no more data for the time being. Later, permission to send can be obtained by sending a segment with the same ACK number and a nonzero Window Size field.

Checksum

This is provided for maximum dependability. It computes checksums for the header, data, and conceptual pseudo header. The pseudo header contains the source and destination machines' 32-bit IP addresses, the TCP protocol number (6), and the byte count for the TCP segment (including the header). Incorporating the pseudo header into TCP checksum computation aids in the detection of misdelivered packets, but it violates the protocol hierarchy because the IP addresses contained within it belong to the IP layer, not the TCP layer.

Urgent Pointer: Indicates the byte offset from the current sequence number where urgent data should be found. Urgent data is available until the end of the segment. In practice, this is not used. The same effect can be obtained by using two TCP connections, one for transferring urgent data and the other for transferring non-urgent data.

Options: Allows you to add extra facilities that aren't covered by the regular header.

- Maximum TCP payload that the sender is willing to accept. MSS stands for maximum segment size (Maximum Segment Size). Both parties exchange capacity information during the handshake. The minimum of two is honoured. This information is sent in the options of the three-way handshake's SYN packets.
- The window scale option can be used to increase the size of the window. It can be specified by instructing the receiver to interpret the window size by shifting it left by the specified number of bits. This header option allows for a window size of up to 230 pixels.

Data: This can be of varying sizes. TCP determines its size by inspecting the IP size header.

UDP

The User Datagram Protocol (UDP) is a fundamental component of the Internet protocol suite. David P. Reed created the protocol in 1980, and it was formally defined in RFC 768. UDP employs a straightforward connectionless communication model with few protocol mechanisms. UDP includes checksums for data integrity and port numbers for addressing different functions at the datagram's source and destination [8]. It lacks handshaking dialogues, exposing the user's program to the unreliability of the underlying network; there is no guarantee of delivery, ordering, or duplicate protection, as illustrated in Figures 8–10.

Let's look at an example of a packet in more detail:

ARP: The Address Resolution Protocol (ARP) is a communication protocol that is used to discover the link layer address, such as a MAC address, that is associated with a given internet layer address, which is typically an IPv4 address. This mapping is an essential part of the Internet protocol suite. RFC 826, which is Internet Standard STD 37, defined ARP in 1982. The destination node's IP address is broadcast, and the destination node informs the source of its MAC address, as shown in Figures 11 and 12 [9].

1. Assume that the LAN is broadcast.
2. The destination's broadcast IP address
3. The destination responds with its MAC address.
4. The source keeps a cache of IP and MAC address bindings.

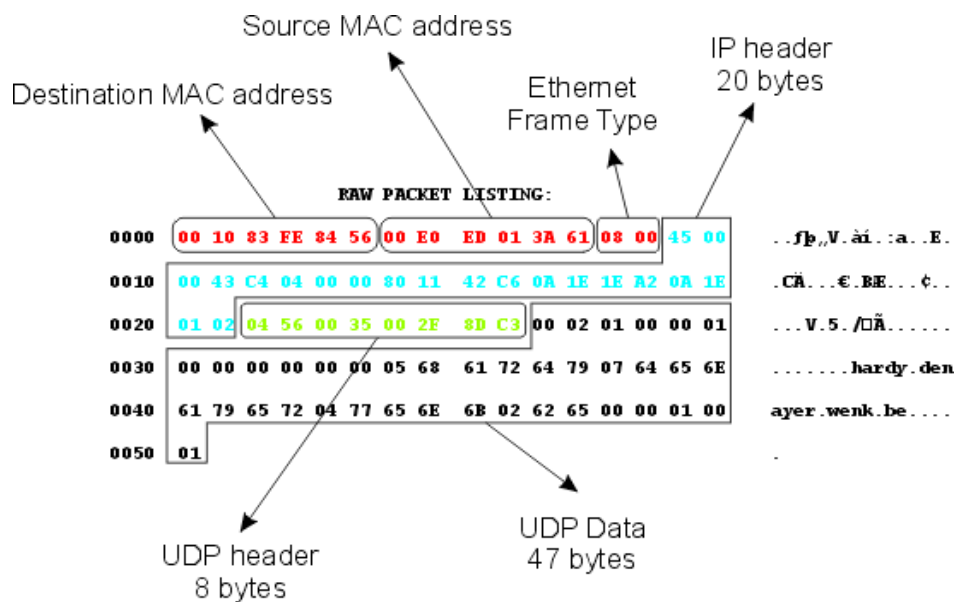


Figure 8. UDP data packet.

As shown in Figures 13 and 14, the operation code specifies the type of message, 1 for Request and 2 for Reply.

Shell Scripting

For Linux systems, there are several shells available, such as:

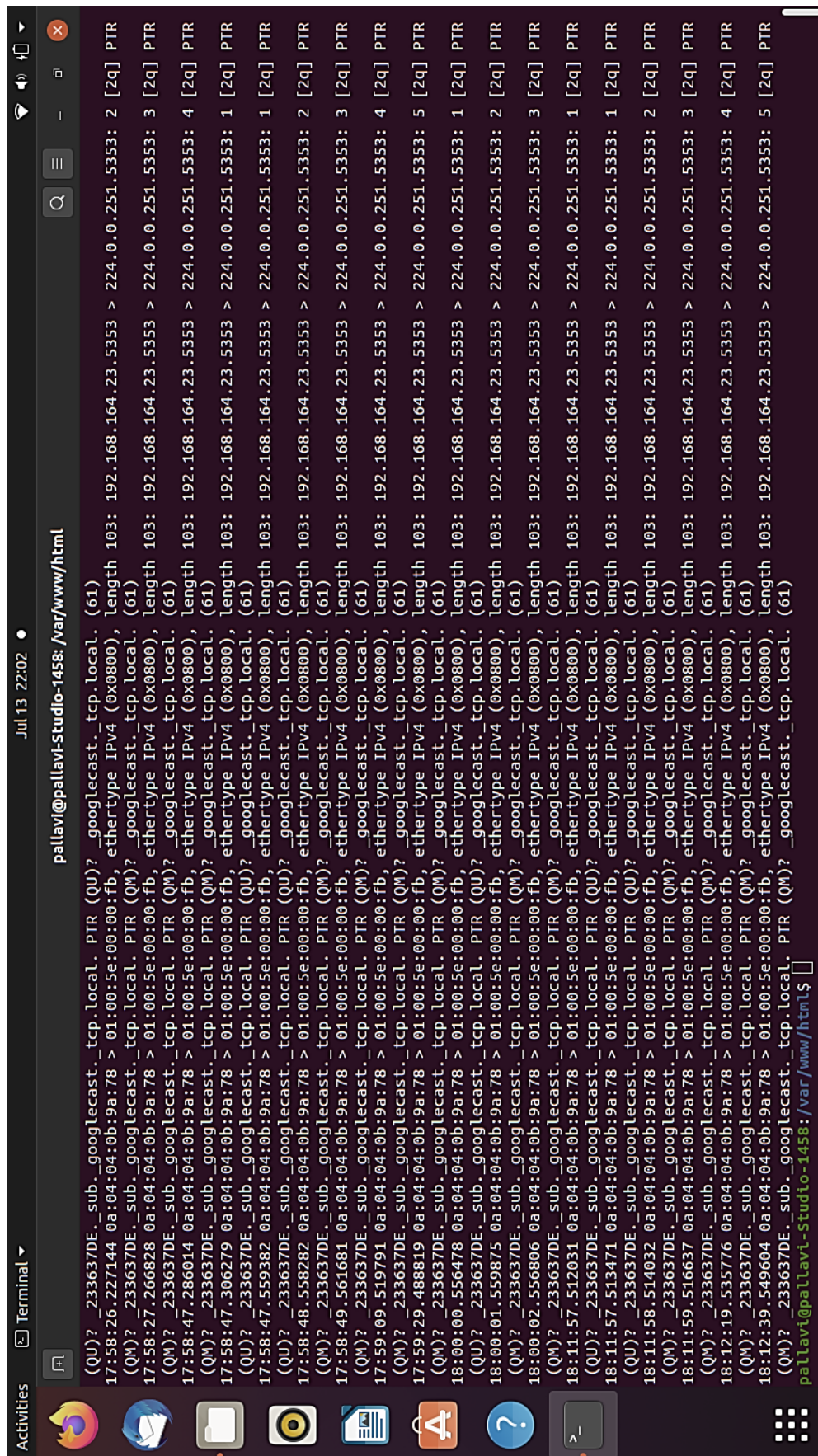
- **BASH (Bourne Again Shell):** It is the most popular shell in Linux systems. It is the default login shell in Linux and macOS systems. It is also compatible with the Windows operating system.
- **CSH (C Shell):** The syntax and usage of the C shell are very similar to those of the C programming language.
- **KSH (Korn Shell):** The Korn Shell also served as the foundation for the POSIX Shell standard specifications, among other things.

Each shell performs the same function but understands different commands and has a different set of built-in functions [10]. Shells are interactive in the sense that they accept commands from users and execute them. However, if we want to run a series of commands on a regular basis, we must enter all of them into the terminal each time.

Because shell can take commands from a file as input, we can write these commands in a file and execute them in shell to avoid this repetitive work. These files are known as Shell Scripts or Shell Programs. Shell scripts are similar to batch files in MS-DOS. Each shell script is saved with the .sh file extension, as shown in Figure 15. Main Shell Script for capturing and filtering packet is:

Three webpages have been designed to make the project user interactive:

1. **Login Form:** The user can log in on this page by entering their username and password. When a user clicks the login button, the main.sh (shell script file) is executed, and the filtering process is performed only on this page, as shown in Figure 16.
2. **User option form:** The user's credentials are saved in a database linked to a php file. When a user presses the submit button, a query is extracted from the database's details table and compared to the user's value, as shown in Figure 17.
On the next page, the user can select the protocol and attributes that they want to view. To select a protocol, a radio button is used, and to select attributes, options are used.
3. **Table of the attributes:** This is a table of the attributes that the user selected on the previous page. It also informs the user of the number of packets captured, as shown in Figure 18.



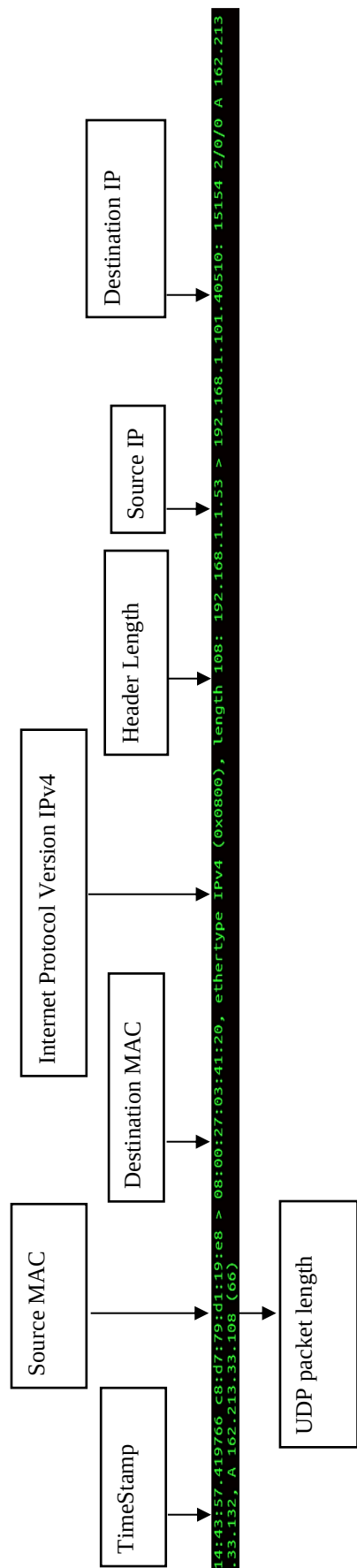


Figure 10. UDP packet length.

physical layer header		x bytes
A R P	hardware address space	2 bytes
	protocol address space	2 bytes
P a c k e t	hardware address	2 bytes
	byte length (n)	2 bytes
operation code		2 bytes
hardware address of sender		n bytes
protocol address of sender		m bytes
hardware address of target		n bytes
protocol address of target		m bytes

Figure 11. ARP Packet format.

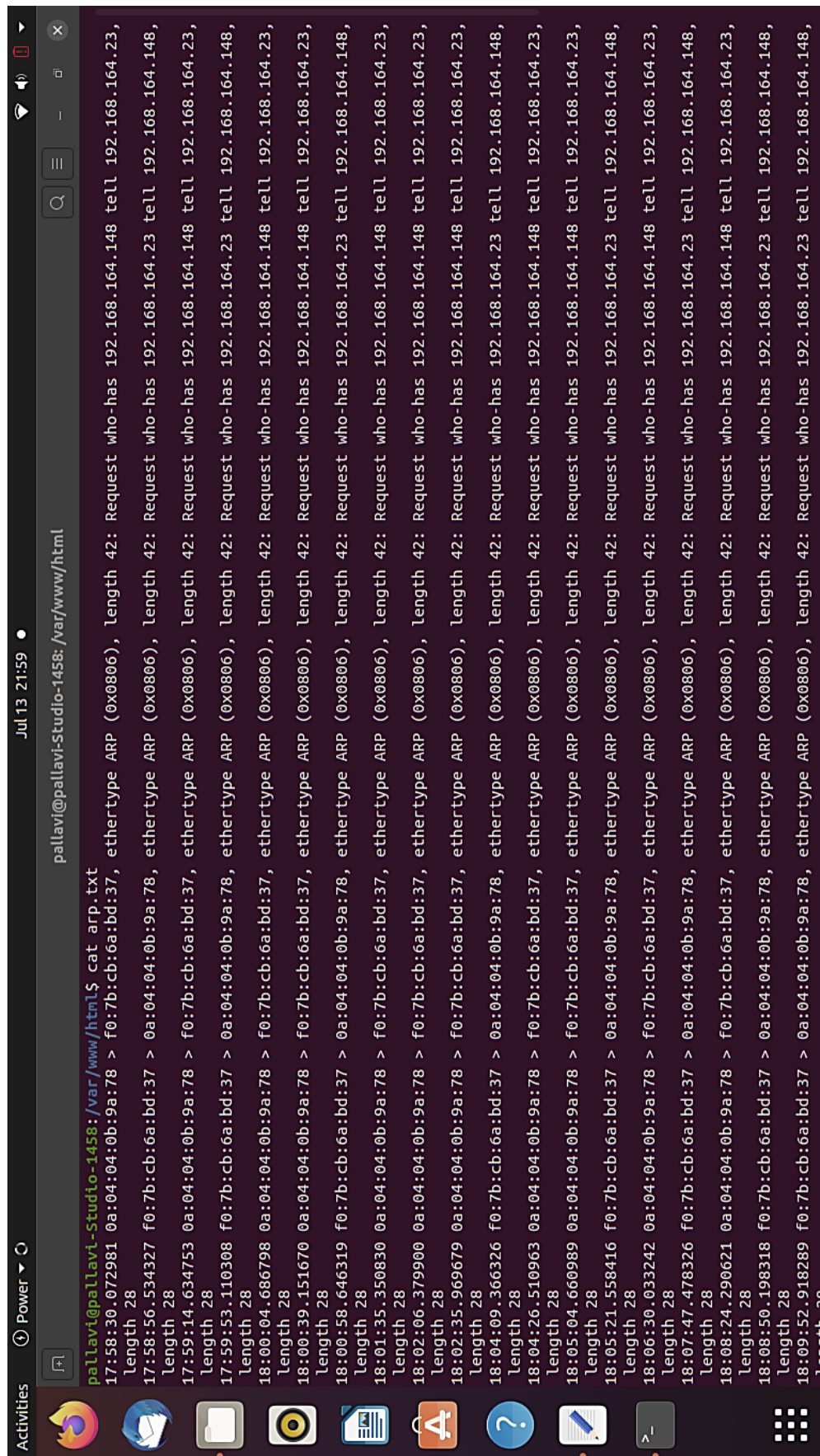


Figure 12. Captured ARP packet.

```
16:12:51.099591 c8:d7:79:d1:19:e8 > ff:ff:ff:ff:ff:ff, ethernet ARP (0x0806), length 60: Request who-has 192.168.1.100 tell 192.168.1.1, length 46
```

Figure 13. Reply message

```
16:13:03.341809 08:00:27:03:41:20 > c8:d7:79:d1:19:e8, ethernet ARP (0x0806), length 42: Reply 192.168.1.101 is-at 08:00:27:03:41:20, length 28
```

Figure 14. Request message

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mydatabase |
+-----+
2 rows in set (0.00 sec)

mysql> USE mydatabase;
Database changed
mysql> create table mytable values(username varchar(20),password varchar(20));
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax
to use near '(username varchar(20),password varchar(20))' at line 1
mysql> ytable values(username varchar(20),password varchar(20));
ytable values(username varchar(20),password varchar(20));
^C
mysql> create table mytable (username varchar(20),password varchar(20));
Query OK, 0 rows affected (1.73 sec)

mysql> insert into mytable values ('pallavi','ITLAB');
Query OK, 1 row affected (0.15 sec)

mysql> insert into mytable values ('mohit','1');
Query OK, 1 row affected (0.18 sec)

mysql> select * from mytable;
+-----+-----+
| username | password |
+-----+-----+
| pallavi | ITLAB |
| mohit | 1 |
+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

Figure 15. Shell Script Filter

NETWORK MONITORING TOOL

login form

Enter Username:

Enter Password:

Figure 16. Login form.

NETWORK MONITORING TOOL

TCP ☒

UDP ☐

ARP ☐

Time Slot

Figure 17. User form.

Permission of File

The tcpdump command requires sudo privileges and must be executed as the root user. Because main.sh (shell script file) contains the tcpdump command, which is used to extract packets from a LAN using an interface, the user must enter the password each time the code is executed. However, we do not want the user to be prompted for their password when they enter their username and password, so we must change the permissions of the main.sh file.

Chmod: The chmod command is used in Unix-like operating systems to change the access mode of a file.

After entering the folder, use the following command to change the access mode: `chmod 777 main.sh` / giving access mode permission to read, write, and execute files.

Chown: Different users in the operating system have ownership and permission to ensure the security of the files and to limit who can modify the contents of the files.

Types of file Permissions

- **User:** These file permissions have an impact on the file's owner.
- **Group:** These file permissions have an impact on the group that owns the file. If the owner user is a member of this group, the user permissions will take precedence over the group permissions.
- **Other:** These file permissions have an impact on all other users on the system.



To change the permissions from www-data to root, use the command `chown root main.sh`.

This will only change the user from www-data to root for the owner. So, the next command to execute to change the group user to root is `chgrp root main.sh` / for group

Visudo File

The visudo command on Unix-like operating systems edits the sudoers file, which is used by the sudo command. Run visudo to change which users and groups are permitted to use sudo.

As shown in Figure 19, we are allowing the user to access the main.sh file without entering a password.

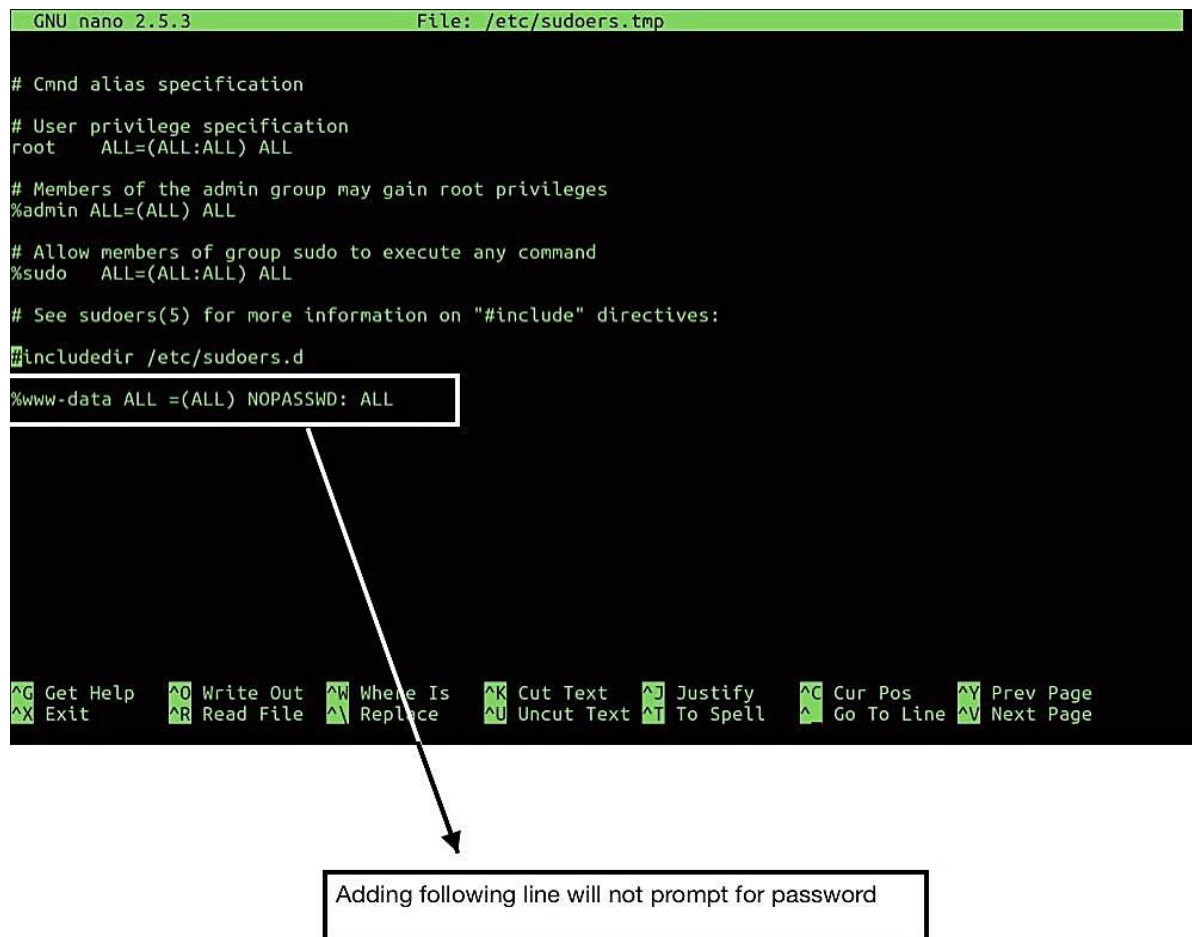


Figure 19. Shell execution command in PHP script.

To execute shell script in php script:

```
exec("/var/www/html/main.sh")
```

PHP Script

PHP Script for Connection with Database

```
<?php
define('DB_SERVER', 'localhost'); define('DB_USERNAME', 'root'); define('DB_PASSWORD',
'shshwat1234'); define('DB_DATABASE', 'login');
$db = mysqli_connect(DB_SERVER,DB_USERNAME,DB_PASSWORD,DB_DATABASE);
?>
```

PHP Script for Login Page

```

<?php include("config.php");
    session_start();

if($_SERVER["REQUEST_METHOD"] == "POST") {
    // username and password sent from form

    $myusername = mysqli_real_escape_string($db,$_POST['Username']);
    $mypassword = mysqli_real_escape_string($db,$_POST['Password']);

    $sql = "SELECT id FROM details WHERE username = '$myusername' and password = '$mypassword'";
    $result = mysqli_query($db,$sql);
    $row = mysqli_fetch_array($result,MYSQLI_ASSOC);
    $active = $row['active'];

    $count = mysqli_num_rows($result); if($count == 1) {
        $var_value=$myusername;
        $_SESSION['varname'] = $var_value;
        $output = exec('/var/www/html/main.sh 2<&1'); echo $output;
        header("location: choose.php");
    }
    else {
        $error = "Invalid Username or Password";
    }
}
?>

```

CONCLUSION

A wireless analyzer, also known as a Wi-Fi analyzer, is a packet analyzer used to intercept traffic on wireless networks. A packet analyzer is also known as a network analyzer or protocol analyzer, though these terms have different meanings.

Acknowledgment

The success and final outcome of this project required a lot of guidance and assistance from many people, and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I respect and thank Dr. Rashi Agarwal, for providing me an opportunity to do the project work and giving us all support and guidance, which made me complete the project duly. I am extremely thankful to her for providing such a nice support and guidance.

Owe my deep gratitude to our project guide Er. Shubhendra Singh, who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

I am thankful to and fortunate enough to get constant encouragement, support, and guidance from all instructors of Information Technology Department which helped us in successfully completing our project work.

REFERENCES

1. Techopedia. Bundled Software: What Does Bundled Software Mean? [Online]. Available from <https://www.techopedia.com/definition/4240/bundled-software>

2. Digital Ocean (31-Jul-2021). What is a Kernel? [Online]. Available from <https://www.digitalocean.com/community/tutorials/what-is-a-kernel>
3. TechTarget Contributor. Shell [Online]. Available from <https://www.techtarget.com/searchdatacenter/definition/shell>
4. How do I use the command-line shell? [Online]. Available from <https://courses.cs.washington.edu/courses/cse140/13wi/shell-usage.html>
5. Cloudflare. What is TCP/IP? [Online]. Available from <https://www.cloudflare.com/en-in/learning/ddos/glossary/tcp-ip/>
6. Paloalto Networks. Source and Destination NAT Example [Online]. Available from <https://docs.paloaltonetworks.com/pan-os/9-0/pan-os-admin/networking/nat/nat-configuration-examples/source-and-destination-nat-example.html>
7. Madpackets. TCP Sequence and Acknowledgement Numbers Explained [Online]. Available from <https://madpackets.com/2018/04/25/tcp-sequence-and-acknowledgement-numbers-explained/>
8. Khan Academy. User Datagram Protocol (UDP) [Online]. Available from <https://www.khanacademy.org/computing/computers-and-internet/xcae6f4a7ff015e7d:the-internet/xcae6f4a7ff015e7d:transporting-packets/a/user-datagram-protocol-udp>
9. Tech Target. Mariusz Zydyk. Address Resolution Protocol (ARP) [Online]. Available from <https://www.techtarget.com/searchnetworking/definition/Address-Resolution-Protocol-ARP>
10. Tutorials Point. Shell Scripting Tutorial [Online]. Available from https://www.tutorialspoint.com/unix/shell_scripting.htm