# NSS Assignment-5 Report

By Kaustav Vats (2016048)

Objective of the assignment was to create your small little rudimentary port scanner which should emulate two different TCP port scans supported by NMAP. I tested its functionality on a VM. I tested it by probing another VM using the former, using your own tool.

Raw sockets allow new IPv4 protocols to be implemented in user space. A raw socket receives or sends the raw datagram not including link level headers.

I created my own IP headers, TCP headers and UDP header[Bonus].

TCP pseudo-header for checksum computation (IPv4)

| Bit offset | 0-3                    | 4–7      | 8–15     | 16–31            |
|------------|------------------------|----------|----------|------------------|
| 0          | Source address         |          |          |                  |
| 32         | Destination address    |          |          |                  |
| 64         | Zeros                  |          | Protocol | TCP length       |
| 96         | Source port            |          |          | Destination port |
| 128        | Sequence number        |          |          |                  |
| 160        | Acknowledgement number |          |          |                  |
| 192        | Data<br>offset         | Reserved | Flags    | Window           |
| 224        | Checksum               |          |          | Urgent pointer   |
| 256        | Options (optional)     |          |          |                  |
| 256/288+   | Data                   |          |          |                  |

In IP header i added source, destination address and TCP protocol 6. I also added length of TCP header. For TCP headers i added Source and Destination Port with sequence number of the TCP packet. The IPv4 layer generates an IP header when sending a packet unless the **IP\_HDRINCL** socket option is enabled on the socket. When it is enabled, the packet must contain an IP header. For receiving, the IP header is always included in the packet.

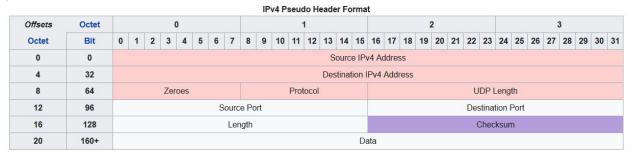
# **SYN Port Scan**

- Set syn bit in flag to 1.
- For open port i received a packet with ack set to 1.

### **FIN Port Scan**

- Set FIN bit in flag to 1.
- For Closed port i received a TCP packet with ACK 1 and RST 1

For Bonus i crafted my own UDP packet by setting the Source port, destination port number, UDP packet length and UDP checksum. For unreachable port i received a icmp packet which had type = 3 and code = 3 set. It means that the port is unreachable.



## ICMP\_FILTER

Enable a special filter for raw sockets bound to the **IPPROTO\_ICMP** protocol. The value has a bit set for each ICMP message type which should be filtered out. The default is to filter no ICMP messages.

# **Assumptions**

- Only open ports need to be shown in the assignment.
- No server file is required for this assignment.
- Code needs to be exit with CTRL+C to exit.

### References:-

- [1] http://man7.org/linux/man-pages/man7/raw.7.html
- [2] https://opensourceforu.com/2015/03/a-quide-to-using-raw-sockets/
- [3] https://en.wikipedia.org/wiki/Transmission\_Control\_Protocol
- [4] https://www.cnblogs.com/rollenholt/articles/2590959.html
- [5] https://en.wikipedia.org/wiki/User\_Datagram\_Protocol