## Lab Assignment – 8

### **Name:** Srujan Patwardhan

### **STD:** TY-D

### **Roll no.:** 02

### **PRN:** 12210847

### **Subject:** Computer Networks

#### Q1. Understanding protocol stack of Intranet a) Analyze packet formats of Ethernet, IP, TCP and UDP captured using Wireshark/Fiddler for traffic analysis tool in peer-to-peer mode for wired and wireless networks. b) Use any tool for custom packet generation (Packet Sender Tool) or write your own code for packet generation and analyze the packets.

##### CODE:

import *java.nio.ByteBuffer*;

import *java.util.Arrays*;

*public* *class* UDPPacketGenerator {

*public* *static* void main(String[] args) {

        byte[] udpPacket = createUDPPacket();

        parseUDPPacket(udpPacket);

    }

*public* *static* byte[] createUDPPacket() {

        byte[] ipHeader = createIPHeader();

        byte[] udpHeader = createUDPHeader();

        ByteBuffer packet = ByteBuffer.allocate(ipHeader.length + udpHeader.length);

        packet.put(ipHeader);

        packet.put(udpHeader);

        System.out.println("Generated UDP Packet: " + Arrays.toString(packet.array()));

        return packet.array();

    }

*public* *static* byte[] createIPHeader() {

        ByteBuffer ipHeader = ByteBuffer.allocate(20);  *// IP header is 20 bytes*

        ipHeader.put((byte) 0x45);  *// Version (4) + IHL (5)*

        ipHeader.put((byte) 0x00);  *// Type of Service*

        ipHeader.putShort((short) 28);  *// Total Length (20 for IP + 8 for UDP)*

        ipHeader.putShort((short) 0x1c46);  *// Identification*

        ipHeader.putShort((short) 0x4000);  *// Flags and Fragment Offset*

        ipHeader.put((byte) 64);  *// TTL (Time to Live)*

        ipHeader.put((byte) 17);  *// Protocol (UDP = 17)*

        ipHeader.putShort((short) 0x0000);  *// Header Checksum (set to 0 for simplicity)*

        ipHeader.put(new byte[]{(byte) 192, (byte) 168, 0, 1});  *// Source IP: 192.168.0.1*

        ipHeader.put(new byte[]{(byte) 192, (byte) 168, 0, 2});  *// Destination IP: 192.168.0.2*

        return ipHeader.array();

    }

*public* *static* byte[] createUDPHeader() {

        ByteBuffer udpHeader = ByteBuffer.allocate(8);

        udpHeader.putShort((short) 12345);  *// Source Port*

        udpHeader.putShort((short) 9876);   *// Destination Port*

        udpHeader.putShort((short) 8);      *// Length (8 bytes for UDP header, no data)*

        udpHeader.putShort((short) 0x0000); *// Checksum (set to 0 for simplicity)*

        return udpHeader.array();

    }

*public* *static* void parseUDPPacket(byte[] packet) {

        ByteBuffer buffer = ByteBuffer.wrap(packet);

        System.out.println("----- IP Header -----");

        byte versionAndIHL = buffer.get();  *// First byte: Version (4 bits) + IHL (4 bits)*

        int version = (versionAndIHL >> 4) & 0xF;  *// Extract version*

        int ihl = versionAndIHL & 0xF;  *// Extract IHL (Internet Header Length)*

        System.out.println("Version: " + version);

        System.out.println("IHL: " + ihl + " (in 32-bit words, " + (ihl \* 4) + " bytes)");

        byte typeOfService = buffer.get();

        System.out.println("Type of Service: " + (typeOfService & 0xFF));

        short totalLength = buffer.getShort();

        System.out.println("Total Length: " + totalLength + " bytes");

        short identification = buffer.getShort();

        System.out.println("Identification: " + identification);

        short flagsAndFragmentOffset = buffer.getShort();

        System.out.println("Flags and Fragment Offset: " + flagsAndFragmentOffset);

        byte ttl = buffer.get();

        System.out.println("TTL: " + ttl);

        byte protocol = buffer.get();

        System.out.println("Protocol: " + protocol + " (UDP)");

        short headerChecksum = buffer.getShort();

        System.out.println("Header Checksum: 0x" + Integer.toHexString(headerChecksum & 0xFFFF));

        byte[] sourceIP = new byte[4];

        buffer.get(sourceIP);

        System.out.println("Source IP: " + (sourceIP[0] & 0xFF) + "." + (sourceIP[1] & 0xFF) + "." +

                (sourceIP[2] & 0xFF) + "." + (sourceIP[3] & 0xFF));

        byte[] destinationIP = new byte[4];

        buffer.get(destinationIP);

        System.out.println("Destination IP: " + (destinationIP[0] & 0xFF) + "." + (destinationIP[1] & 0xFF) + "." +

                (destinationIP[2] & 0xFF) + "." + (destinationIP[3] & 0xFF));

        System.out.println("----- UDP Header -----");

        short sourcePort = buffer.getShort();

        System.out.println("Source Port: " + (sourcePort & 0xFFFF));

        short destinationPort = buffer.getShort();

        System.out.println("Destination Port: " + (destinationPort & 0xFFFF));

        short udpLength = buffer.getShort();

        System.out.println("UDP Length: " + udpLength + " bytes");

        short udpChecksum = buffer.getShort();

        System.out.println("UDP Checksum: 0x" + Integer.toHexString(udpChecksum & 0xFFFF));

    }

}

##### OUTPUT:

