



TCS2351 NETWORK SECURITY

Trimester 1, 2023/2024

Lecture: FCI

Lecturer: Dr. Chan Wai Kok

No	Student Name	Student ID
1	Kristina Geetha Menon a/p Christopher Menon	1201201474

Question 10

Output file = xyz.pcap

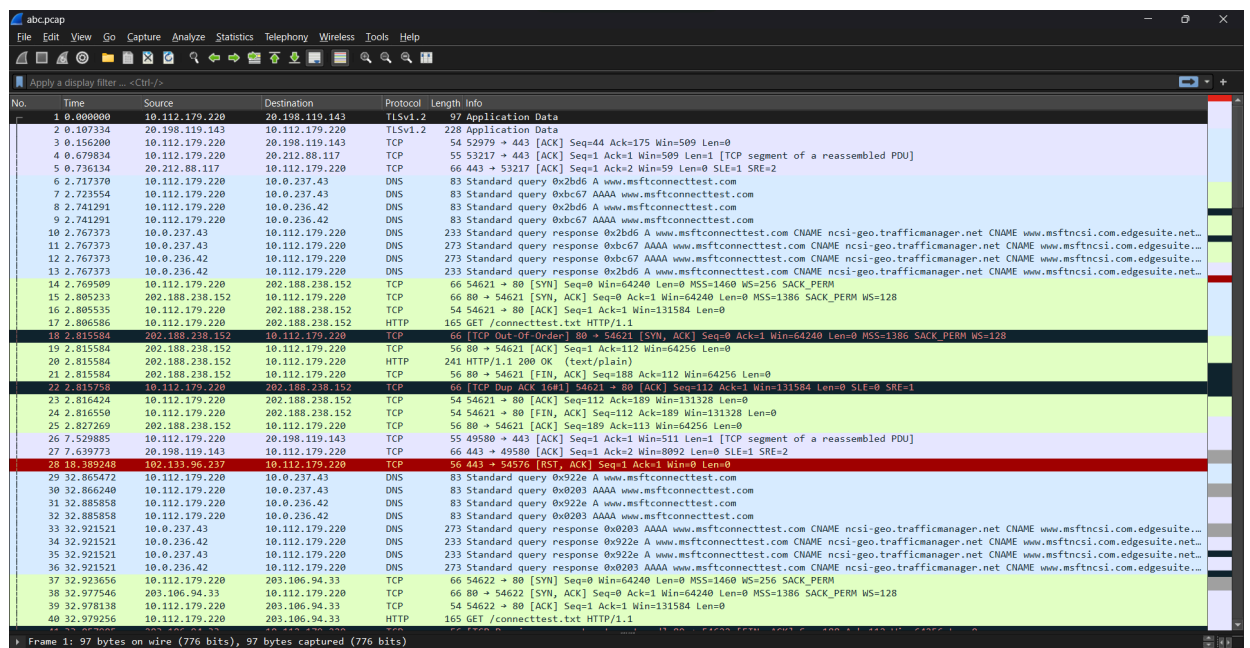
Change all source ip addresses to 18.X.X.X.

And rewrite the output into a file called xyz.pcap. (tcpdump format)

Other packets remain unchanged and are written to file xyz.pcap as well.

Bear in mind you have to recalculate the IP checksum for that packet.

1) Capture the packet in Wireshark and save as abc.pcap.



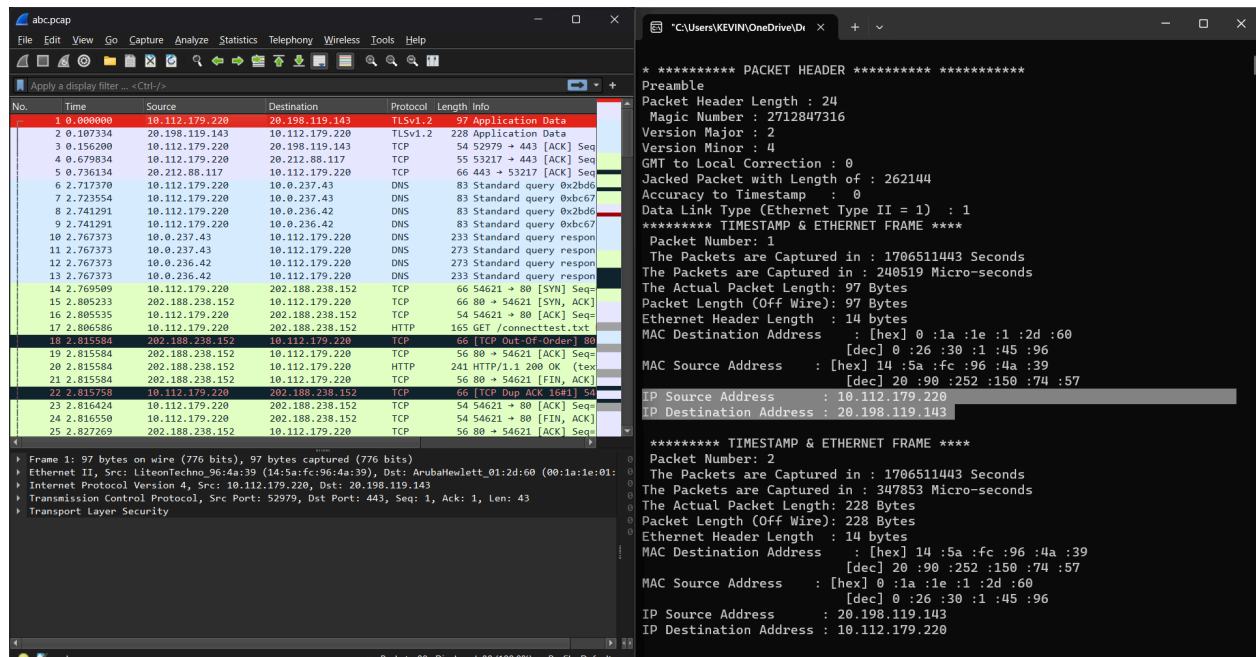
2) Modify the code (1201201474_net1.cpp) to capture the IP source address and IP Destination address.

```
// IP Header to capture IP Source address and Destination address
typedef struct ip_header
{
    unsigned char version_ihl;           // Version (4 bits) + Internet header length (4 bits)
    unsigned char dscp_ecn;              // DSCP (6 bits) + ECN (2 bits)
    unsigned short total_length;          // Total length
    unsigned short identification;        // Identification
    unsigned short flags_fragoffset;     // Flags (3 bits) + Fragment offset (13 bits)
    unsigned char ttl;                   // Time to Live
    unsigned char protocol;               // Protocol
    unsigned short checksum;              // Header checksum
    unsigned int src_ip;                  // Source IP address
    unsigned int dest_ip;                 // Destination IP address
} ip_hdr;
```

```
// Read and display IP header information
fread((char *)&ip, sizeof(ip), 1, input);
cout << "IP Source Address : " << ((ip.src_ip >> 0) & 0xFF) << "." << ((ip.src_ip >> 8) & 0xFF)
<< "." << ((ip.src_ip >> 16) & 0xFF) << "." << ((ip.src_ip >> 24) & 0xFF) << endl;

cout << "IP Destination Address : " << ((ip.dest_ip >> 0) & 0xFF) << "." << ((ip.dest_ip >> 8) & 0xFF)
<< "." << ((ip.dest_ip >> 16) & 0xFF) << "." << ((ip.dest_ip >> 24) & 0xFF) << endl;
```

- 3) Run the 1201201474_net1.cpp file to compare the IP Source and Destination address with Wireshark.



Note: The IP source and destination address is similar compared to Wireshark. Hence, it is successful to locate all the IP packets.

- 4) **Task: I will need to modify my code in which:**

- a) Change all source ip captured in abc.pcap to 18.X.X.X. The code below represents the implementation on modifying the captured ip source address.

```
// Modify the source IP address -> 18.X.X.X
unsigned char *src_ip_bytes = reinterpret_cast<unsigned char*>(&ip.src_ip);

src_ip_bytes[0] = 18;
```

- b) Recalculate the IP checksum.

```
// Recalculate the IP checksum - call function
ip.checksum = calculateIPChecksum(&ip);
```

```
// Function to recalculate IP checksum
unsigned short calculateIPChecksum(ip_header *ip_hdr)
{
    unsigned long sum = 0;
    unsigned short *ip_header_ptr = (unsigned short *)ip_hdr;
    int header_length = sizeof(ip_header) / 2; // Divide by 2 to get the number of 16-bit words

    ip_hdr->checksum = 0; // Clear the checksum field before calculation

    // Calculate the sum of all 16-bit words in the IP header
    for (int i = 0; i < header_length; i++)
    {
        sum += *ip_header_ptr++;
    }

    // Fold carry bits back into the sum
    while (sum >> 16)
    {
        sum = (sum & 0xFFFF) + (sum >> 16);
    }

    // Take the one's complement
    sum = ~sum;

    return (unsigned short)sum;
}
```

The IP checksum has been validated at the xyz.pcap (output file) indicating that it has been calculated correctly and ensures the integrity of the packets. The output of the checksum is as per screenshot below.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.112.179.220	20.198.119.143	TLSv1.2	97	Application Data
2	0.107334	20.198.119.143	10.112.179.220	TLSv1.2	228	Application Data
3	0.156200	10.112.179.220	20.198.119.143	TCP	54	52979 → 443 [ACK] Seq=553217 → 415 [ACK] Seq=553217
4	0.670334	10.112.179.220	20.198.119.143	TCP	66	443 → 53217 [ACK] Seq=553217 → 415 [ACK] Seq=553217
5	0.736134	20.212.88.117	10.112.179.220	TCP	66	443 → 53217 [ACK] Seq=553217 → 415 [ACK] Seq=553217
6	2.717370	10.112.179.220	10.0.237.43	DNS	83	Standard query 0xbcb67
7	2.723554	10.112.179.220	10.0.237.43	DNS	83	Standard query 0xbcb67
8	2.741291	10.112.179.220	10.0.236.42	DNS	83	Standard query 0xbcb67
9	2.741291	10.112.179.220	10.0.236.42	DNS	83	Standard query 0xbcb67
10	2.767373	10.0.237.43	10.112.179.220	DNS	233	Standard query response
11	2.767373	10.0.237.43	10.112.179.220	DNS	273	Standard query response
12	2.767373	10.0.236.42	10.112.179.220	DNS	273	Standard query response
13	2.767373	10.0.236.42	10.112.179.220	DNS	233	Standard query response
14	2.769509	10.112.179.220	202.188.238.152	TCP	66	54621 → 80 [SYN] Seq=66 80 → 54621 [SYN, ACK] Seq=54621 → 80 [ACK] Seq=54621
15	2.805233	202.188.238.152	10.112.179.220	TCP	66	80 → 54621 [SYN, ACK] Seq=54621 → 80 [ACK] Seq=54621
16	2.805535	10.112.179.220	202.188.238.152	TCP	54	54621 → 80 [ACK] Seq=54621
17	2.806586	10.112.179.220	202.188.238.152	HTTP	165	GET /connecttest.txt
18	2.815584	202.188.238.152	10.112.179.220	TCP	66	80 → 54621 [ACK] Seq=54621
19	2.815584	202.188.238.152	10.112.179.220	TCP	56	80 → 54621 [ACK] Seq=54621
20	2.815584	202.188.238.152	10.112.179.220	HTTP	241	HTTP/1.1 200 OK (text/css)
21	2.815584	202.188.238.152	10.112.179.220	TCP	56	80 → 54621 [FIN, ACK] Seq=54621
22	2.815584	10.112.179.220	202.188.238.152	TCP	60	80 → 54621 [FIN, ACK] Seq=54621
23	2.816424	10.112.179.220	202.188.238.152	TCP	54	54621 → 80 [ACK] Seq=54621
24	2.816550	10.112.179.220	202.188.238.152	TCP	54	54621 → 80 [FIN, ACK] Seq=54621
25	2.827269	202.188.238.152	10.112.179.220	TCP	56	80 → 54621 [ACK] Seq=54621

Packet details for selected packet (No. 17):

- Ethernet II, Src: Intel(R) Ethernet Controller (10:00:00:00:00:00), Dst: Intel(R) Ethernet Controller (10:00:00:00:00:00)
- Internet Protocol Version 4, Src: 10.112.179.220, Dst: 20.212.88.117
- Transmission Control Protocol, Src Port: 53217, Dst Port: 443, Seq: 1, Ack: 1, Len: 1
- Hypertext Transfer Protocol, GET /connecttest.txt

Header Checksum (ip checksum), 2 bytes

The header checksum of Frame 4 from abc.pcap (original).

```
.... 0101 = Header Length: 20 bytes (5)
▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 41
  Identification: 0xacfa (44282)
▶ 010. .... = Flags: 0x2, Don't fragment
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 128
  Protocol: TCP (6)
  Header Checksum: 0x223f [correct]
  [Header checksum status: Good]
  [Calculated Checksum: 0x223f]
  Source Address: 10.112.179.220
  Destination Address: 20.212.88.117
Transmission Control Protocol, Src Port: 53217, Dst Port: 443, Seq: 1, Ack: 1, Len: 1
```

The header checksum of Frame 4 from xyz.pcap (modified IP source address).

```
.... 0101 = Header Length: 20 bytes (5)
▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 41
  Identification: 0xacfa (44282)
▶ 010. .... = Flags: 0x2, Don't fragment
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 128
  Protocol: TCP (6)
  Header Checksum: 0x1a3f [correct]
  [Header checksum status: Good]
  [Calculated Checksum: 0x1a3f]
  Source Address: 18.112.179.220
  Destination Address: 20.212.88.117
▶ Transmission Control Protocol, Src Port: 53217, Dst Port: 443, Seq: 1, Ack: 1, Len: 1
```

c) And rewrite the output into a file called xyz.pcap. (tcpdump format).

```
// Write the modified packet to the output file -> xyz.pcap
fwrite((char *)&tt, sizeof(tt), 1, output);
fwrite((char *)&eth, sizeof(eth), 1, output);
fwrite((char *)&ip, sizeof(ip), 1, output);

// Write the remaining data from the original packet
for (i = 0; i < tt.caplen - sizeof(eth) - sizeof(ip); i++)
{
    fread((char *)&buff, sizeof(buff), 1, input);
    fwrite((char *)&buff, sizeof(buff), 1, output);
}
```

The output of xyz.pcap file is as per below screenshot.

xyz.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	18.112.179.220	20.198.119.143	TLSv1.2	97	Application Data
2	0.107334	18.198.119.143	10.112.179.220	TLSv1.2	228	Application Data
3	0.156200	18.112.179.220	20.198.119.143	TCP	54	52979 → 443 [ACK] Seq=
4	0.679834	18.112.179.220	20.212.88.117	TCP	55	53217 → 443 [ACK] Seq=
5	0.736134	18.212.88.117	10.112.179.220	TCP	66	443 → 53217 [ACK] Seq=
6	2.717370	18.112.179.220	10.0.237.43	DNS	83	Standard query 0x2bd6
7	2.723554	18.112.179.220	10.0.237.43	DNS	83	Standard query 0xbc67
8	2.741291	18.112.179.220	10.0.236.42	DNS	83	Standard query 0x2bd6
9	2.741291	18.112.179.220	10.0.236.42	DNS	83	Standard query 0xbc67
10	2.767373	18.0.237.43	10.112.179.220	DNS	233	Standard query respon
11	2.767373	18.0.237.43	10.112.179.220	DNS	273	Standard query respon
12	2.767373	18.0.236.42	10.112.179.220	DNS	273	Standard query respon
13	2.767373	18.0.236.42	10.112.179.220	DNS	233	Standard query respon
14	2.769509	18.112.179.220	202.188.238.152	TCP	66	54621 → 80 [SYN] Seq=
15	2.805233	18.188.238.152	10.112.179.220	TCP	66	80 → 54621 [SYN, ACK]
16	2.805535	18.112.179.220	202.188.238.152	TCP	54	54621 → 80 [ACK] Seq=
17	2.806586	18.112.179.220	202.188.238.152	HTTP	165	GET /connecttest.txt
18	2.815584	18.188.238.152	10.112.179.220	TCP	66	[TCP Retransmission]
19	2.815584	18.188.238.152	10.112.179.220	TCP	56	80 → 54621 [ACK] Seq=
20	2.815584	18.188.238.152	10.112.179.220	HTTP	241	HTTP/1.1 200 OK (tex
21	2.815584	18.188.238.152	10.112.179.220	TCP	56	80 → 54621 [FIN, ACK]
22	2.815758	18.112.179.220	202.188.238.152	TCP	66	[TCP Dup ACK 16#1] 54
23	2.816424	18.112.179.220	202.188.238.152	TCP	54	54621 → 80 [ACK] Seq=
24	2.816550	18.112.179.220	202.188.238.152	TCP	54	54621 → 80 [FIN, ACK]
25	2.827269	18.188.238.152	10.112.179.220	TCP	56	80 → 54621 [ACK] Seq=

Frame 4: 55 bytes on wire (440 bits), 55 bytes captured (440 bits)

Ethernet II, Src: LiteonTechno_96:4a:39 (14:5a:fc:96:4a:39), Dst: ArubaHewlett_01:2d:60 (00:1a:1e:01:2d:60)

- Destination: ArubaHewlett_01:2d:60 (00:1a:1e:01:2d:60)
- Source: LiteonTechno_96:4a:39 (14:5a:fc:96:4a:39)
- Type: IPv4 (0x0800)

Internet Protocol Version 4, Src: 18.112.179.220, Dst: 20.212.88.117

Transmission Control Protocol, Src Port: 53217, Dst Port: 443, Seq: 1, Ack: 1, Len: 1

d) Display the modified IP source address to the console.

```

C:\Users\KEVIN\OneDrive\Dt  x  +  v

***** PACKET HEADER *****
Preamble
Packet Header Length : 24
Magic Number : 2712847316
Version Major : 2
Version Minor : 4
GMT to Local Correction : 0
Jacked Packet with Length of : 262144
Accuracy to Timestamp : 0
Data Link Type (Ethernet Type II = 1) : 1
***** TIMESTAMP & ETHERNET FRAME ****
Packet Number: 1
The Packets are Captured in : 1706511443 Seconds
The Packets are Captured in : 240519 Micro-seconds
The Actual Packet Length: 97 Bytes
Packet Length (Off Wire): 97 Bytes
Ethernet Header Length : 14 bytes
MAC Destination Address : [hex] 0 :1a :1e :1 :2d :60
                        [dec] 0 :26 :30 :1 :45 :96
MAC Source Address : [hex] 14 :5a :fc :96 :4a :39
                    [dec] 20 :90 :252 :150 :74 :57
IP Source Address : 10.112.179.220
IP Destination Address : 20.198.119.143

Modified Source Address: 18.112.179.220

***** TIMESTAMP & ETHERNET FRAME ****
Packet Number: 2
The Packets are Captured in : 1706511443 Seconds

```

- e) Additionally, the MAC address has been compared to make sure it is the same between the abc.pcap and xyz.pcap.

The MAC address of Frame 4 from abc.pcap.

```
▶ Frame 4: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface 0  
▼ Ethernet II, Src: LiteonTechno_96:4a:39 (14:5a:fc:96:4a:39), Dst: ArubaHewlett_01:2d:60 (00:1a:1e:01:2d:60)  
  ▶ Destination: ArubaHewlett_01:2d:60 (00:1a:1e:01:2d:60)  
  ▶ Source: LiteonTechno_96:4a:39 (14:5a:fc:96:4a:39)  
    Type: IPv4 (0x0800)  
▶ Internet Protocol Version 4, Src: 10.112.179.220, Dst: 20.212.88.117  
▶ Transmission Control Protocol, Src Port: 53217, Dst Port: 443, Seq: 1, Ack: 1, Len: 1
```

The MAC address of Frame 4 from xyz.pcap.

```
▶ Frame 4: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface 0  
▼ Ethernet II, Src: LiteonTechno_96:4a:39 (14:5a:fc:96:4a:39), Dst: ArubaHewlett_01:2d:60 (00:1a:1e:01:2d:60)  
  ▶ Destination: ArubaHewlett_01:2d:60 (00:1a:1e:01:2d:60)  
  ▶ Source: LiteonTechno_96:4a:39 (14:5a:fc:96:4a:39)  
    Type: IPv4 (0x0800)  
▶ Internet Protocol Version 4, Src: 18.112.179.220, Dst: 20.212.88.117  
▶ Transmission Control Protocol, Src Port: 53217, Dst Port: 443, Seq: 1, Ack: 1, Len: 1
```