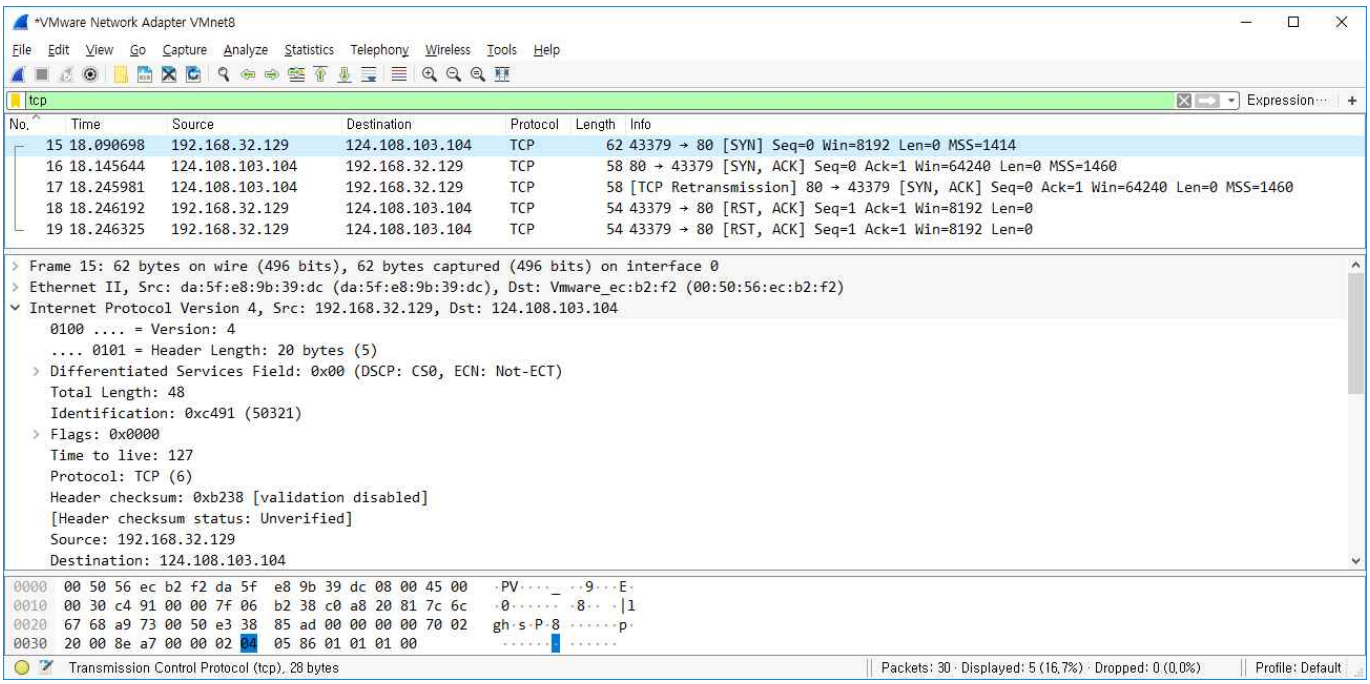
컴퓨터 네트워크

레포트 # 4

이름(학번)	황 준 일 (32131766)
담당교수	조 경 산 교수님
제출일	2018. 06. 11

1. Capture Ethernet frames using Wireshark, and explain the connection process in TCP.
   Explain fields in the header of TCP SYN packet



00 50 56 ec b2 f2 da 5f e8 9b 39 dc 08 00 => Ethernet Header

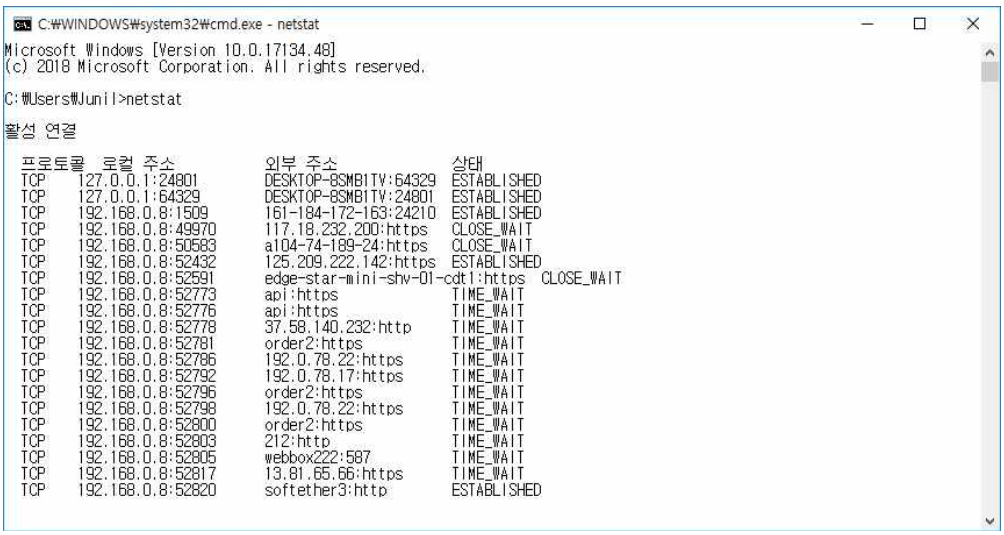
45 00 00 30 c4 91 00 00 7f 06 b2 38 c0 a8 20 81 7c 6c 67 68 => IP Header

a9 73 00 50 e3 38 85 ad 00 00 00 00 70 02 20 00 8e a7 00 00 02 04 05 86 01 01 01 00 => TCP Header

1) Source Port : 0xA973 = 43379
2) Destination Port : 0x0050 = 80 (HTTP)
3) Sequence number : 0xE33885AD = 3812132269 이지만 wireshark에선 0으로 표기됨
4) Acknowledge number : 0x000000 = 0
5) Header Length : 0x7 = 0111
6) Flags : 0x2 = 0000 0010 => SYN
7) Window Size : 0x2000 = 8192 (Bytes)
8) Checksum : 0x8EA7

패킷의 내용을 분석해본 결과, SYN의 경우 송신자 주소와 Window size를 전달하여 수신자에게 연결을 시도 하는 것이 목적이다.

2. Explain the information provided by "netstat" command.



TCP의 Connection state를 확인할 수 있다.

3. The following is a dump of a UDP header; 0045DF0000580000

1) Is the packet directed from a client to a server or vice versa?

: 0x0045 => 0xDF00

: 69 => 57,088 이므로 Sever(1024~49151) to Client(49152 ~ 65535)

2) What is the length of the data?

: 0x0058 = 88

3) How the sender handled checksum for this packet?

: 0x0000 => 검사하지 않는다

4. In TCP, how many sequence numbers are consumed by each segments?

가. SYN : 1 increment

나. ACK : no increment

다. FIN+ACK : fin - 1 increment, ack - no increment => total 1 increment

라. ACK+data(100bytes)

: increment 100 (100 bytes를 보낼 땐 push plag가 켜진다)

5. The intruder sends a SYN segment to the server using 철수's IP address. Can the intruder create a TCP connection with the server by pretending that he is 철수? Assume that the server uses 1) a different ISN(Initial Sequence Number) for each connection or 2) the same ISN for each connection.

편의상 instruder는 '준일', 수신자를 '경산'이라고 칭하겠습니다.

1) a different ISN(Initial Sequence Number) for each connection

: '준일'이 '철수'의 IP주소로 변조하여 '경산'에게 SYN 신호를 보내도, 결국 ACK 신호를 받는건 '철수'이며, '준일'은 철수가 받은 ACK 신호를 모르기 때문에 연결은 불가능하다

2) the same ISN for each connection

: 각 연결에 대하여 ISN 주소가 똑같다면 초기의 ACK 신호의 Sequence Nubmer는 항상 동일하므로, '준일'은 '경산'에게 신호를 받은 척 하며 연결이 완료되었다는 ACK 신호를 보낼 수 있다. 따라서 연결이 가능하다.

6. Following is the output from netstat command.

Proto	Local Address	Foreign Address	State
TCP	192.13.201.215:61032	219.240.16.226:80	ESTABLISHED
TCP	192.13.201.215:1059	211.234.249.226:59004	LISTENING
TCP	192.13.201.215:62029	211.233.16.71:80	TIME_WAIT

1) Explain the values of state - LISTENING, ESTABLISHED, TIME_WAIT.

ESTABLISHED : 연결 활.

TIME_WAIT : 연결 종료. but, 완전한 종료가 아니라 일정 시간이 흐른 뒤에 수신 신호가 없으면 종료

LISTENING : 접속 대기. 연결을 기다리는 중

2) Explain "219.240.16.226:80 " in Foreign Address in two parts.

http://219.240.16.226 에 연결되었다. 즉, 웹 서버에 연결되었다는 뜻

3) Show the server and client program in each connection.

전자(61032) / 후자(59004) / 전자(62029)

7. An HTTP clients opens a TCP connection using an ISN of 100 and port number of 50,000. The server opens the connection with an ISN of 200. If the client defines receive buffer of 1024 and the server defines receive buffer of 4096, show the header of 2nd segment during the connection establishment. Ignore the calculation of checksum field.
   1) 2번째 segment는 ECE flag
   2) ECE가 1이고 SYN이 1일 경우 : traffic 제어가 가능하다
   3) ECE가 1이고 SYN이 0일 경우 : traffic 제어가 불하니 window 크기를 줄여서 전송해달라고 요청한다.

8. Explain the use of congestion window(cwnd) and receive window(rwnd) in TCP. How the actual size of the send window is determined from cwnd and rwnd?
   1) cwnd : 현재 traffic 용량에 대한 제어. traffic이 수월해질 때 까지 cwnd를 통하여 window 크기를 제어한다.
   2) rwnd : 현재 수신 가능한 window 크기. 상대방이 아무리 많이 보내봤자 rwnd 만큼만 받을 수 있다. 따라서 상대방이 rwnd 만큼 보내도록 요청한다.
   3) 실제 window 크기는 cwnd와 rwnd중 작은 것으로 수신해야 한다.

9. Compare the TCP header and the UDP header. List the fields in the TCP header that are not part of the UDP header, and list the fields in the UDP header that are not part of the TCP header. Give the reason for each missing field.
   1) UDP : Source Port / Destination Port / Message Length / Checksum
   2) TCP : Source Port / Destination Port / Sequence Number / ACK Number / Flags / window size / Checksum
   3) UDP는 비연결지향이라서 보내기만 하면 된다. 따라서 데이터의 신뢰도는 중요하지 않다
      TCP는 연결지향이기 때문에 데이터가 정확히 전달되어야 하므로, 정확성에 대한 옵션들이 존재한다.

10. Which of UDP and TCP is better for the communication between DNS server and client. (consists of two packets - DNS request, DNS reply)
   1) UDP는 TCP 같은 통신의 신뢰성 확보를 위한 처리가 없다 => Client-Sever 통신에서 오버 헤드가 작아진다 => UDP를 사용하는 것이 더 효율적이다.
   2) DNS는 교환되는 데이터가 작은 것부터 하는 것을 원칙으로 Request / Response(Reply)을 하나의 UDP Pacaket 에서 처리할 수 있게 설계되었다.