lect4

1) Copy serv.c.

- Open putty terminal and connect to 165.246.38.151.

- Copy serv.c

$ cp ../../linuxer1/serv.c .

$ ls

serv.c

- Compile

$ gcc -o serv serv.c

2) Copy cli.c

- Open another putty terminal and connect to 165.246.38.151.

- Copy cli.c

$ cp ../../linuxer1/cli.c .

$ ls

cli.c



serv.c와 cli.c를 cp를 사용해 current directory에 복사해줬다.

3) Adjust port number both server and client. Recompile both and run the server first and run the client next. The client should talk first and then the server.

serv.c

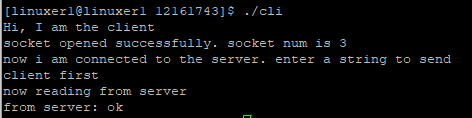


cli.c

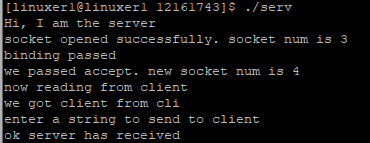




cli



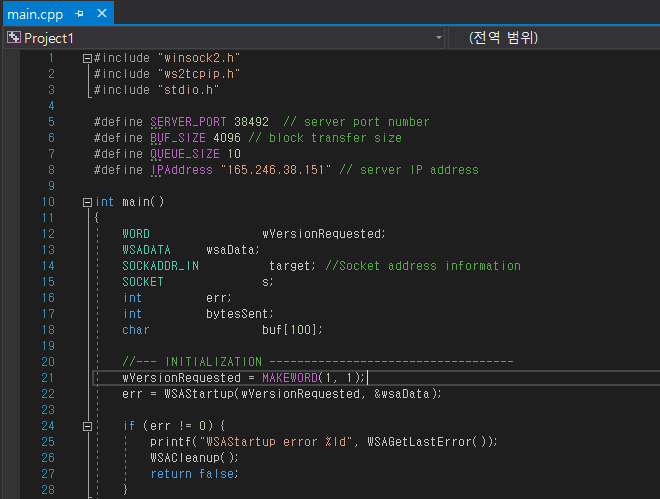
serv



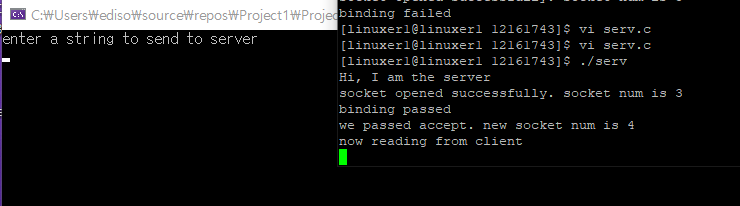
포트 넘버는 63636으로 바꿔주고 주소는 우리가 사용하는 165.246.38.151로 바꿔주었다. 다시 컴파일 한 후 서버부터 실행시켜서 성공적으로 메시지를 주고받을 수 있었다.

lect5

1) Make a client in your PC as follows.







visual studio를 열여서 강의노트에 있는 그대로 복사 붙여넣기를 한 다음 포트넘버와 IP주소를 바꿔주고 다시 한번 더 실행시켰다. 그 결과, 서버와 클라이언트가 성공적으로 연결됐다.

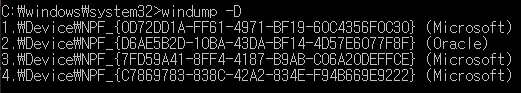
2) Download winpcap from iClass and install. Download windump.exe from iClass and run (in the command window; open command window as admin if needed) to monitor packets for specified port.

windump -D : check available network interfaces

windump -eSXX -i 2 -s 80 port 9924 : monitor packets at device 2 whose src or dest ports are 9924

(for wireshark: right mouse button click, run as admin)

(for MacOS, use tcpdump)



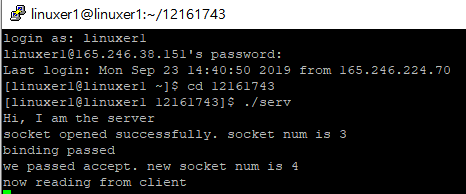
winpcap을 깔고 cmd창을 열어서 windump -D 명령을 쳐본 결과 들어오는 패킷을 모니터 할 수 있는 device들은 총 네개가 나왔다.

3) Run your server again. Run the client in your PC.

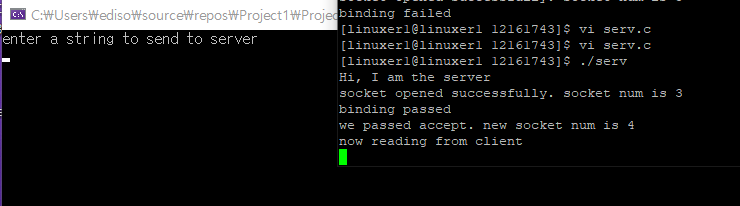
cmd



server



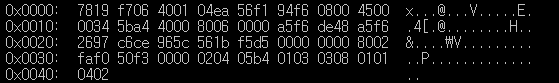
client



네개의 device중에 네번째 device에 패킷이 들어오기 때문에 windump에서 명령을 칠 때 -i 4를 해주었다. 그 후 서버와 클라이언트를 차례대로 실행시켜주었다.

4) Find the first packet which is a SYN packet sent by the client to the server in the windump window. Extract all packet header information. Refer TCP packet structure in Section 6 below.

SYN packet



Ethernet header = Dest addr(6) + Src addr(6) + type(2)

(빨간 박스로 표시)

Dest addr => 7819f7064001 (서버)

Src addr => 04ea56f194f6 (클라이언트)

Type => 0x0800 (IP packet)

IP header=version(4 bit)+header length(4 bit)+type of service(8 bit)+total length(16 bit)+identification(16 bit)+flags(3 bit)+fragment offset(13 bit)+time to live(8 bit)+protocol(8 bit)+header checksum(16 bit)+source IP address(32 bit)+dest IP address(32 bit)

(노란 박스로 표시)

version => 4 (version IPV4)

header length => 5\*4 = 20 (하나당 4바이트라서 5\*4)

type of service => 0x00

total length => 20(IP header length) + 32(TCP header length) = 52 (0x34)

identification => 5ba4

flags => 4

fragment offset => 0

time to live => 80

protocol => 06 (TCP 패킷)

header checksum => 0x0000

source IP address => a5f6de48

dest IP address => a5f62697 (165.246.38.151)

TCP header = source port number(16) + dest port number(16) + sequence number(32) + acknowledgment number(32) + header length(4) + reserved(6) + URG + ACK + PSH + RST + SYN + PIN + window size(16) + TCP checksum(16) + urgent pointer(16)

(초록 박스로 표시)

source port number => c6ce

dest port number => 965c (38492)

sequence number => 561bf5d5 (모든 sequence number는 다 unique하다.)

acknowledgment number => 0

header length => 8\*4 = 32

reserved =>0x000000 (002를 0000 0000 0010으로 바꿨을 때 앞의 000000)

URG => 0 ACK => 0 PSH => 0 RST => 0 SYN => 1 PIN => 0

(000010 = SYN packet)

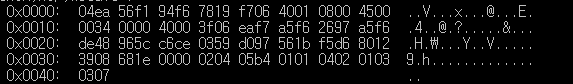
window size => faf0

TCP checksum => 50f3

urgent pointer => 0x0000

5) Analyze rest of the packets similarly.

S/ACK packet



Ethernet header

Dest addr => 04ea56f194f6 (클라이언트)

Src addr => 7819f7064001 (서버)

Type => 0x0800 (IP packet)

IP header

version => 4 (version IPV4)

header length => 5\*4 = 20

type of service => 0x00

total length => 20(IP header length) + 32(TCP header length) = 52 (0x34)

identification => 0000

flags => 4

fragment offset => 0

time to live => 3f

protocol => 06 (TCP 패킷)

header checksum => eaf7

source IP address => a5f62697 (165.246.38.151)

dest IP address => a5f6de48

TCP header

source port number => 965c (38492)

dest port number => c6ce

sequence number => 0359d097

acknowledgment number => 561bf5d6 (서버에서 SYN 패킷의 sequence number에 1을 더해서 여기로 보냄)

header length => 8\*4 = 32

reserved =>0x000000 (012를 0000 0001 0010으로 바꿨을 때 앞의 000000)

URG => 0 ACK => 1 PSH => 0 RST => 0 SYN => 1 PIN => 0

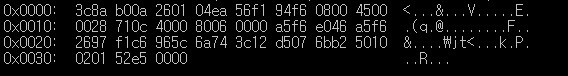
(010010 = S/ACK packet)

window size => 3908

TCP checksum => 681e

urgent pointer => 0x0000

ACK packet



Ethernet header

Dest addr => 3c8ab00a2601

Src addr => 04ea56f194f6 (클라이언트)

Type => 0x0800 (IP packet)

IP header

version => 4 (version IPV4)

header length => 5\*4 = 20

type of service => 0x00

total length => 20(IP header length) + 20(TCP header length) = 40 (0x28)

identification => 710c

flags => 4

fragment offset => 0

time to live => 80

protocol => 06 (TCP 패킷)

header checksum => 0x0000

source IP address => a5f6e046

dest IP address => a5f62697

TCP header

source port number => f1c6

dest port number => 965c (38492)

sequence number => 6a743c12

acknowledgment number => d5076bb2

header length => 5\*4 = 20

reserved =>0x000000 (010를 0000 0001 0000으로 바꿨을 때 앞의 000000)

URG => 0 ACK => 1 PSH => 0 RST => 0 SYN => 0 PIN => 0

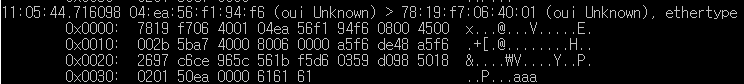
(0100000 = ACK packet)

window size => 0201

TCP checksum => 52e5

urgent pointer => 0x0000

Data packet



Ethernet header

Dest addr => 7819f7064001 (서버)

Src addr => 04ea56f194f6 (클라이언트)

Type => 0x0800 (IP packet)

IP header

version => 4 (version IPV4)

header length => 5\*4 = 20

type of service => 0x00

total length => 20(IP header length) + 23(TCP header length) = 43 (0x2b)

identification => 5ba7

flags => 4

fragment offset => 0

time to live => 80

protocol => 06

header checksum => 0x0000

source IP address => a5f6de48

dest IP address => a5f62697

TCP header

source port number => c6ce

dest port number => 965c (38492)

sequence number => 561bf5d6

acknowledgment number => 0359d098

header length => 5\*4 = 20

reserved =>0x000000 (018를 0000 0001 1000으로 바꿨을 때 앞의 000000)

URG => 0 ACK => 1 PSH => 1 RST => 0 SYN => 0 PIN => 0

window size => 0201

TCP checksum => 50ea

urgent pointer => 0x0000

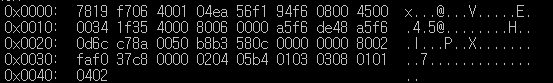
6) Connect to www.inha.ac.kr and analyze SYN, S/ACK, ACK packets between the web browser and www.inha.ac.kr. You may need "-c num" option to capture the first num packets as below.

windump –eSXX –c 20 –i 2 –s 80 host [www.inha.ac.kr](http://www.inha.ac.kr)

cmd에서 나오는 많은 패킷 중에서 SYN, S/ACK, ACK 패킷을 하나씩 대표로 캡쳐해서 분석했다

.

SYN packet



Ethernet header

Dest addr => 7819f7064001 (서버(www.inha.ac.kr))

Src addr => 04ea56f194f6 (클라이언트(chrome))

Type => 0x0800 (IP packet)

IP header

version => 4 (version IPV4)

header length => 5\*4 = 20

type of service => 0x00

total length => 20(IP header length) + 32(TCP header length) = 52 (0x34)

identification => 1f35

flags => 4

fragment offset => 0

time to live => 80

protocol => 06 (TCP 패킷)

header checksum => 0x0000

source IP address => a5f6de48

dest IP address => a5f60d6c

TCP header

source port number => c78a

dest port number => 0050

sequence number => b8b3580c

acknowledgment number => 0x00000000

header length => 8\*4 = 32

reserved =>0x000000 (002를 0000 0000 0010으로 바꿨을 때 앞의 000000)

URG => 0 ACK => 0 PSH => 0 RST => 0 SYN => 1 PIN => 0

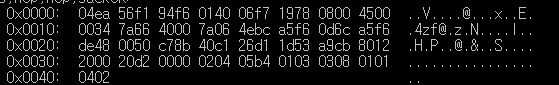
(SYN packet)

window size => faf0

TCP checksum => 37c8

urgent pointer => 0x0000

S/ACK packet



Ethernet header

Dest addr => 04ea56f194f6 (클라이언트)

Src addr => 014006f71978

Type => 0x0800 (IP packet)

IP header

version => 4 (version IPV4)

header length => 5\*4 = 20

type of service => 0x00

total length => 20(IP header length) + 32(TCP header length) = 52 (0x34)

identification => 7a66

flags => 4

fragment offset => 0

time to live => 7a

protocol => 06 (TCP 패킷)

header checksum => 4ebc

source IP address => a5f60d6c

dest IP address => a5f6de48

TCP header

source port number => 0050

dest port number => c78b

sequence number => 40c126d1

acknowledgment number => 1d53a9cb

header length => 8\*4 = 32

reserved =>0x000000 (012를 0000 0001 0010으로 바꿨을 때 앞의 000000)

URG => 0 ACK => 1 PSH => 0 RST => 0 SYN => 1 PIN => 0

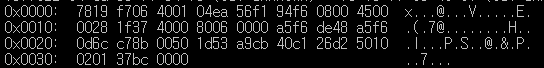
(010010 = S/ACK packet)

window size => 2000

TCP checksum => 20d2

urgent pointer => 0x0000

ACK packet



Ethernet header

Dest addr => 7819f7064001 (서버)

Src addr => 04ea56f194f6 (클라이언트)

Type => 0x0800 (IP packet)

IP header

version => 4 (version IPV4)

header length => 5\*4 = 20

type of service => 0x00

total length => 20(IP header length) + 20(TCP header length) = 40 (0x28)

identification => 1f37

flags => 4

fragment offset => 0

time to live => 80

protocol => 06 (TCP 패킷)

header checksum => 0x0000

source IP address => a5f6de48

dest IP address => a5f60d6c

TCP header

source port number => c78b

dest port number => 0050

sequence number => 1d53a9cb

acknowledgment number => 40c126d2

header length => 5\*4 = 20

reserved =>0x000000 (010를 0000 0001 0000으로 바꿨을 때 앞의 000000)

URG => 0 ACK => 1 PSH => 0 RST => 0 SYN => 0 PIN => 0

(0100000 = ACK packet)

window size => 0201

TCP checksum => 37bc

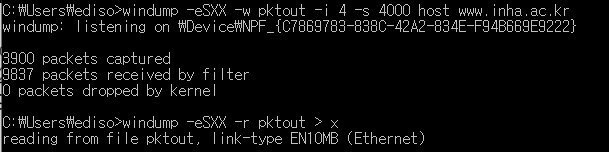
urgent pointer => 0x0000

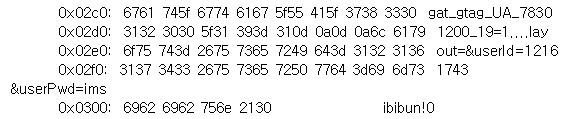
7) Click login menu(로그인) and enter id and password. Find the packet that contains your login ID and password. To capture login ID and password, make the capture size larger, e.g. 3000. Use –w option to save the result in a file (e.g. pktout) and use –r option to read packets from a file.

windump –eSXX –w pktout –i 2 –s 3000 host www.inha.ac.kr

windump –eSXX –r pktout > x

vi x

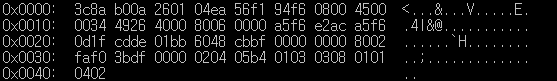




windump를 사용해서 캡쳐한 내용을 pktout에 넣은 뒤 그 내용을 binary file로 볼 수 있도록 -r을 사용해서 파일 x에 pktout의 내용을 넣고, 그것을 워드패드를 이용해서 열었다. 그 뒤 찾기 기능을 이용해 내 아이디와 비밀번호를 찾아본 결과, 내 아이디와 비밀번호가 적나라하게 노출돼있는걸 볼 수 있었다.

8) Connect to portal.inha.ac.kr and do the same thing as in Problem 6) and 7).

SYN packet



Ethernet header

Dest addr => 3c8ab00a2601 (서버(portal.inha.ac.kr))

Src addr => 04ea56f194f6 (클라이언트(chrome))

Type => 0x0800 (IP packet)

IP header

version => 4 (version IPV4)

header length => 5\*4 = 20

type of service => 0x00

total length => 20(IP header length) + 32(TCP header length) = 52 (0x34)

identification => 4926

flags => 4

fragment offset => 0

time to live => 80

protocol => 06 (TCP 패킷)

header checksum => 0x0000

source IP address => a5f6e2ac

dest IP address => a5f60d1f

TCP header

source port number => cdde

dest port number => 01bb

sequence number => 6048cbbf

acknowledgment number => 0x00000000

header length => 8\*4 = 32

reserved =>0x000000 (002를 0000 0000 0010으로 바꿨을 때 앞의 000000)

URG => 0 ACK => 0 PSH => 0 RST => 0 SYN => 1 PIN => 0

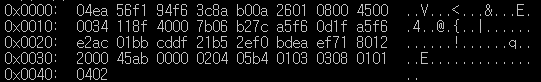
(SYN packet)

window size => faf0

TCP checksum => 3bdf

urgent pointer => 0x0000

S/ACK packet



Ethernet header

Dest addr => 04ea56f194f6 (클라이언트)

Src addr => 3c8ab00a2601 (서버)

(여기에서는 6번과 다르게 S/ACK 패킷에서 서버가 클라이언트에게 제대로 보내준다.)

Type => 0x0800 (IP packet)

IP header

version => 4 (version IPV4)

header length => 5\*4 = 20

type of service => 0x00

total length => 20(IP header length) + 32(TCP header length) = 52 (0x34)

identification => 118f

flags => 4

fragment offset => 0

time to live => 7b

protocol => 06 (TCP 패킷)

header checksum => b27c

source IP address => a5f60d1f

dest IP address => a5f6e2ac

TCP header

source port number => 01bb

dest port number => cddf

sequence number => 21b52ef0

acknowledgment number => bdeaef71

header length => 8\*4 = 32

reserved =>0x000000 (012를 0000 0001 0010으로 바꿨을 때 앞의 000000)

URG => 0 ACK => 1 PSH => 0 RST => 0 SYN => 1 PIN => 0

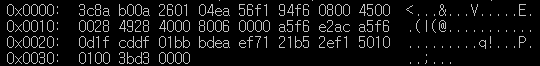
(010010 = S/ACK packet)

window size => 2000

TCP checksum => 45ab

urgent pointer => 0x0000

ACK packet



Ethernet header

Dest addr => 3c8ab00a2601 (서버)

Src addr => 04ea56f194f6 (클라이언트)

Type => 0x0800 (IP packet)

IP header

version => 4 (version IPV4)

header length => 5\*4 = 20

type of service => 0x00

total length => 20(IP header length) + 20(TCP header length) = 40 (0x28)

identification => 4928

flags => 4

fragment offset => 0

time to live => 80

protocol => 06 (TCP 패킷)

header checksum => 0x0000

source IP address => a5f6e2ac

dest IP address => a5f60d1f

TCP header

source port number => cddf

dest port number => 01bb

sequence number => bdeaef71

acknowledgment number => 21b52ef1

header length => 5\*4 = 20

reserved =>0x000000 (010를 0000 0001 0000으로 바꿨을 때 앞의 000000)

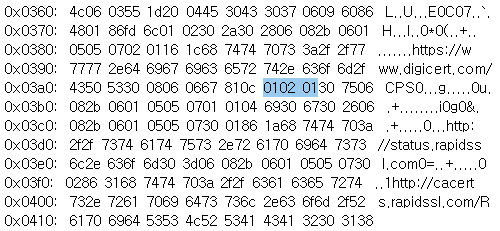
URG => 0 ACK => 1 PSH => 0 RST => 0 SYN => 0 PIN => 0

(0100000 = ACK packet)

window size => 0100

TCP checksum => 3bd3

urgent pointer => 0x0000



이번엔 [www.inha.ac.kr](http://www.inha.ac.kr)이 아닌 portal.inha.ac.kr로 들어가서 6번과 7번을 다시 해보았다. portal.inha.ac.kr은 www.inha.ac.kr보다 더 secure해서 모든 정보가 암호화 돼있기 때문에 학번과 비번 자체를 검색해보기도 하고 아스키코드로 찾아보려해도 찾을 수 없었다.