

TCP 통신을 죽여봅시다

는 사실 가짜 제목입니다. 정확한 제목은

TCP 통신 자살(을 가장한 타살)시키기

ZeroPage 31기 김도엽

1시간짜리 목차

- TCP 소개
- Packet Injection
- packet 구성
- 원리 설명
- 코드 레벨 구현 방법

보험 하나만 깔게요

가르치신 분은 정석(정파, 무림맹)으로 해주셨는데, 제가 야매(사파, 마교)로 습득함

+

오늘 할 설명들도 기초를 야매로 때우고 넘어감

=

말 끊어도 되니까 질문 많이 해주세요

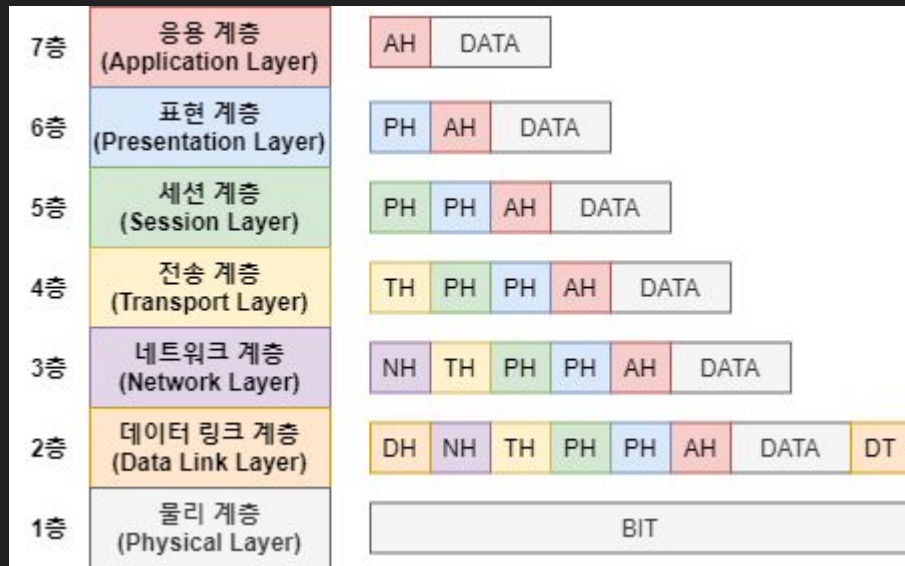
TCP?

Transmission Control Protocol

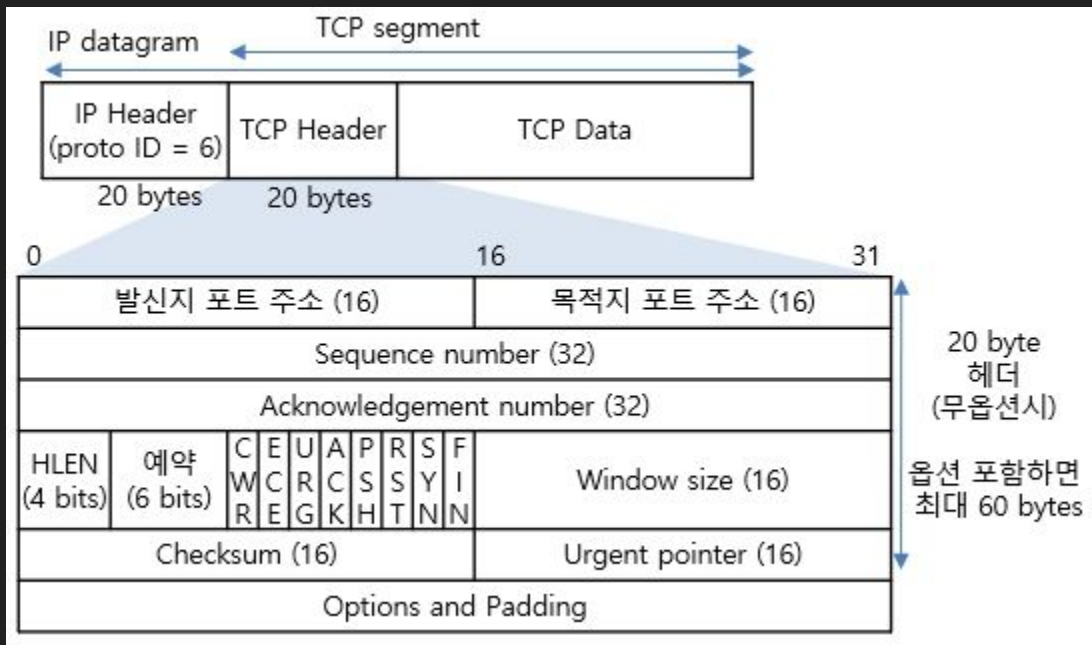
전송 제어 프로토콜

OSI 7계층 중 4층, 전송 계층에 해당

★신뢰성 있는 연결을 지향★



TCP vs UDP



0	16	31 비트
발신 포트번호	수신 포트번호	
(패킷전체) 길이	(패킷전체) 체크섬	
... 데이터 ...		

역시 그림으로만 보면 몰라요

직접 까봅시다



Capturing from Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter -- <Ctrl-/>

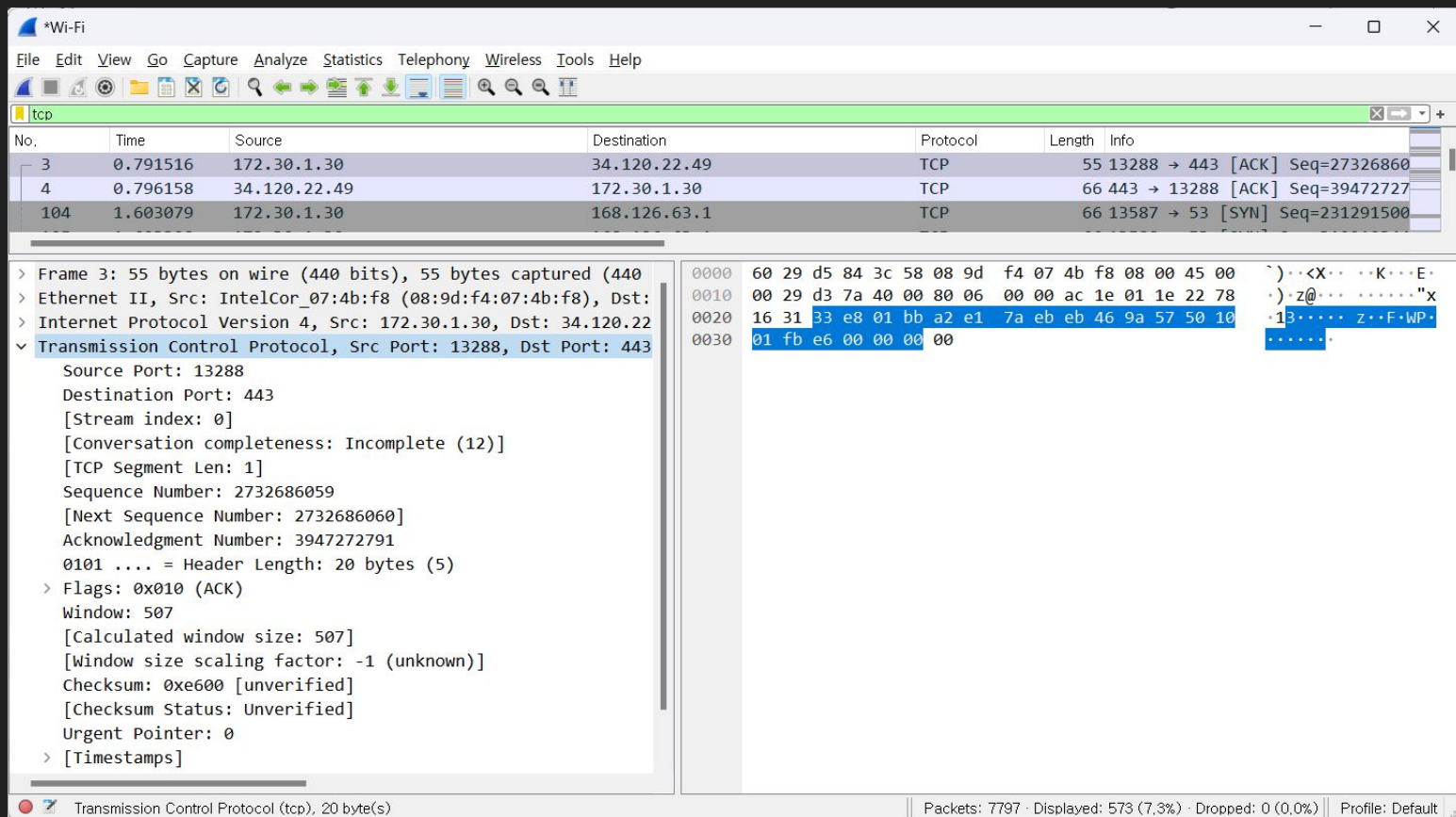
No.	Time	Source	Destination	Protocol	Length	Info
2039	8.190475	172.217.161.227	172.30.1.30	QUIC	66	Protected Payload (KP0)
2040	8.200804	172.217.161.227	172.30.1.30	QUIC	63	Protected Payload (KP0)
2041	8.201322	172.30.1.30	172.217.161.227	QUIC	75	Protected Payload (KP0), DCID=f1f
2042	8.264977	34.17.18.17	172.30.1.30	TLSv1.3	78	Application Data
2043	8.264977	34.17.18.17	172.30.1.30	TCP	54	443 → 13591 [FIN, ACK] Seq=380426
2044	8.265099	172.30.1.30	34.17.18.17	TCP	54	13591 → 443 [ACK] Seq=1941778202
2045	8.265158	172.30.1.30	34.17.18.17	TCP	54	13591 → 443 [RST, ACK] Seq=194177
2046	8.455168	IntelCor_07:4b:f8	Broadcast	ARP	42	Who has 172.30.1.96? Tell 172.30.
2047	8.728836	172.64.148.154	172.30.1.30	TLSv1.2	79	Application Data
2048	8.730372	172.30.1.30	172.64.148.154	TLSv1.2	83	Application Data
2049	8.733752	172.64.148.154	172.30.1.30	TCP	54	443 → 5768 [ACK] Seq=3405476032 A

> Frame 1: 120 bytes on wire (960 bits), 120 bytes captured (960 bits) on interface
> Ethernet II, Src: IntelCor_07:4b:f8 (08:9d:f4:07:4b:f8), Dst: DAVOLINK_84:3c:58
> Internet Protocol Version 4, Src: 172.30.1.30, Dst: 192.168.0.202
> User Datagram Protocol, Src Port: 53224, Dst Port: 161
> Simple Network Management Protocol

0000 60 29 d5 84 3c 58 08 9d f4 07 4b f8 08 00 45 00 ~
0010 00 6a 75 b1 00 00 80 11 00 00 ac 1e 01 1e c0 a8 ~ju
0020 00 ca cf e8 00 a1 00 56 6f 16 30 4c 02 01 00 04 ~
0030 06 70 75 62 6c 69 63 a0 3f 02 02 05 8a 02 01 00 ~put
0040 02 01 00 30 33 30 0f 06 0b 2b 06 01 02 01 19 03 ~6
0050 02 01 05 01 05 00 30 0f 06 0b 2b 06 01 02 01 19 ~
0060 03 05 01 01 01 05 00 30 0f 06 0b 2b 06 01 02 01 ~
0070 19 03 05 01 01 02 01 05 00 ~

Wi-Fi: <live capture in progress> Packets: 2049 - Displayed: 2049 (100.0%) Profile: Default

Wireshark를 통해 보는 TCP 구조



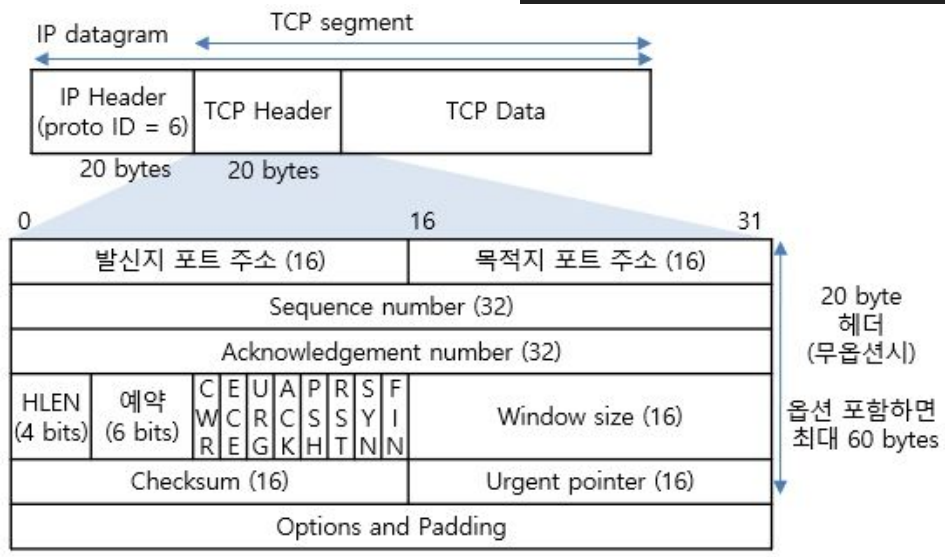
Wireshark를 통해 보는 TCP 구조

0000	60	29	d5	84	3c	58	08	9d	f4	07	4b	f8	08	00	45	00
0010	00	29	d3	7a	40	00	80	06	00	00	ac	1e	01	1e	22	78
0020	16	31	33	e8	01	bb	a2	e1	7a	eb	eb	46	9a	57	50	10
0030	01	fb	e6	00	00	00	00	00	00	00	00	00	00	00	00	00

총 20바이트 구성

바이트 String으로 이루어져 있음

컴퓨터 메모리는 little-endian (host order)
패킷은 big-endian (network order)



번외) 그럼 UDP는?

단 8바이트

포트 2개랑

패킷 길이

체크섬만 있으면 끝

The image shows a Wireshark packet capture window titled '*Wi-Fi'. The packet list on the left shows three packets. The first packet (No. 1877) is a UDP packet from 172.30.1.30 to 172.217.161.228, length 77. The second packet (No. 1896) is a QUIC packet from 172.30.1.30 to 172.217.161.238, length 1292. The third packet (No. 1897) is a QUIC packet from 172.30.1.30 to 172.217.161.238, length 121. The packet details pane on the right shows the structure of the first packet: Frame 1877: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface 0. The packet is an Ethernet II frame, an Internet Protocol Version 4 packet, and a User Datagram Protocol (UDP) packet. The UDP packet details show Source Port: 51964, Destination Port: 443, Length: 43, Checksum: 0xfc36 [unverified], [Checksum Status: Unverified], [Stream index: 9], [Timestamps], UDP payload (35 bytes), and Data (35 bytes). The packet bytes pane on the right shows the raw data in hexadecimal and ASCII. The status bar at the bottom indicates 'User Datagram Protocol (udp), 8 byte(s)' and 'Packets: 7797 · Displayed: 7216 (92.5%) · Dropped: 0 (0.0%) · Profile: Default'.

No.	Time	Source	Destination	Protocol	Length	Info
1877	6.653310	172.30.1.30	172.217.161.228	UDP	77	51964 → 443 Len=35
1896	6.663890	172.30.1.30	172.217.161.238	QUIC	1292	Initial, DCID=3f111101da515687
1897	6.664013	172.30.1.30	172.217.161.238	QUIC	121	0-RTT, DCID=3f111101da515687

> Frame 1877: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface 0

> Ethernet II, Src: IntelCor_07:4b:f8 (08:9d:f4:07:4b:f8), Dst: D...

> Internet Protocol Version 4, Src: 172.30.1.30, Dst: 172.217.161...

> User Datagram Protocol, Src Port: 51964, Dst Port: 443

- Source Port: 51964
- Destination Port: 443
- Length: 43
- Checksum: 0xfc36 [unverified]
- [Checksum Status: Unverified]
- [Stream index: 9]
- [Timestamps]
- UDP payload (35 bytes)

> Data (35 bytes)

0000 60 29 d5 84 3c 58 08 9d f4 07 4b f8 08 00 45 00 ~) <X...K...E...

0010 00 3f 0b 96 40 00 80 11 00 00 ac 1e 01 1e ac d9 ..?..@...

0020 a1 e4 ca fc 01 bb 00 2b fc 36 58 ee bd d6 a9 c7 ..+++++6X.....

0030 52 7c 5c 19 66 2d 9c cf b0 ea 38 52 53 e0 c9 18 R|\-f-...8RS...

0040 81 05 e6 ea 71 ac 1a e6 b5 a9 4f 7b c6q...O{...

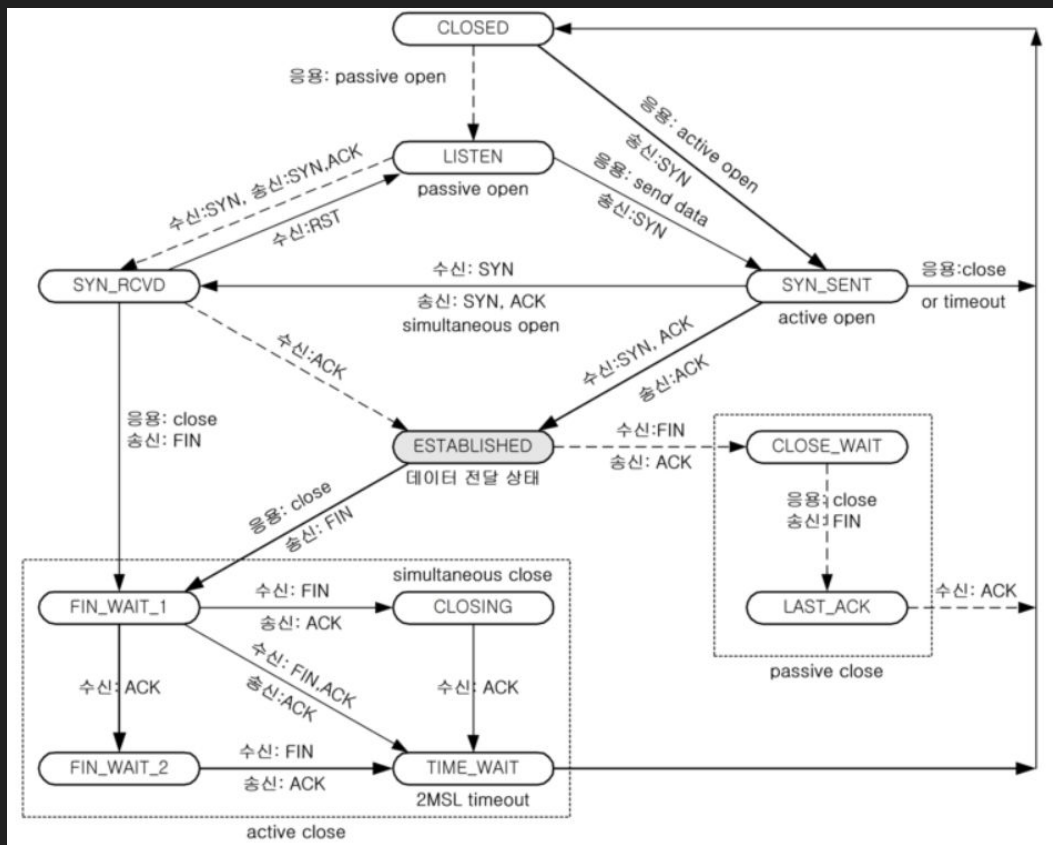
그래서, 뭐 해야하는데

Packet Injection을 사용할 것. 잘 통신하던 둘한테 각각을 사칭해서 꼬사리 끼기.

tcp 연결이 끊기는 피해자를 **victim**, 공격자를 **attacker**라고 하고
공격이 성공하기 위해 필요한 조건

1. packet sniffer
2. attacker가 victim에게 패킷을 보낼 수 있어야 함
3. seq, ack number가 정상처럼 보여야 함
4. checksum이 정확해야 함

죄다 CLOSED로 변화



구체적으로 패킷을 어떻게 만들어야 하죠

신경써야 하는 주요 필드 2가지

seq와 ack number

이 패킷이 제대로 된 순서를 지켰는지 + 주고받은 페이로드 크기가 정확한지 확인

checksum

패킷에 네트워크 문제 등으로 발생한 오류가 있는지 탐지하기 위한 수.
우리는 임의로 패킷을 만들어서 보내기 때문에 따로 계산해줘야 됨.

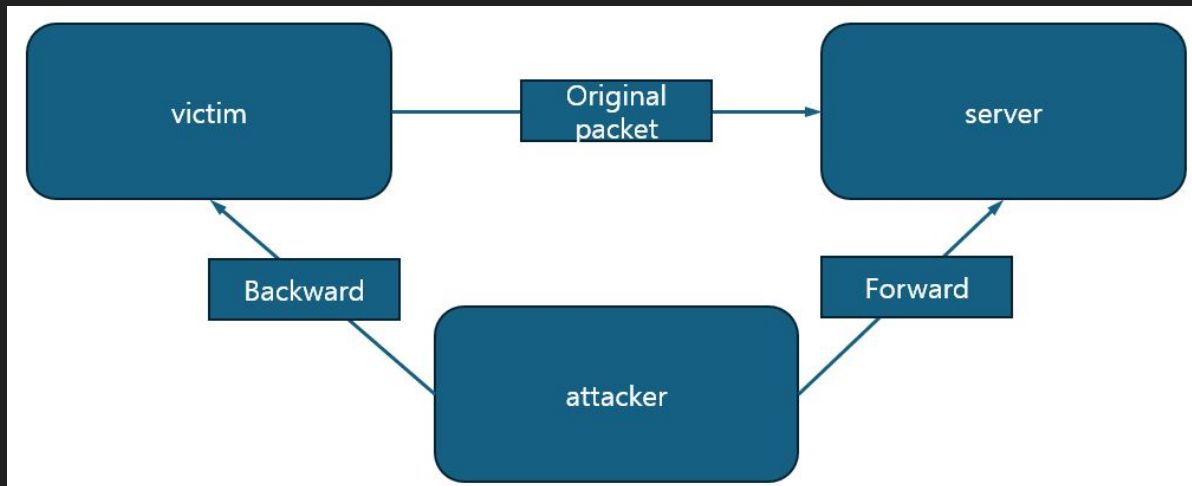
구체적으로 패킷을 어떻게 만들어야 하죠

일단 패킷의 정의부터

original packet과 같은 방향의 패킷 = forward packet

“”

과 반대 방향의 패킷 = backward packet



구체적으로 패킷을 어떻게 만들어야 하죠

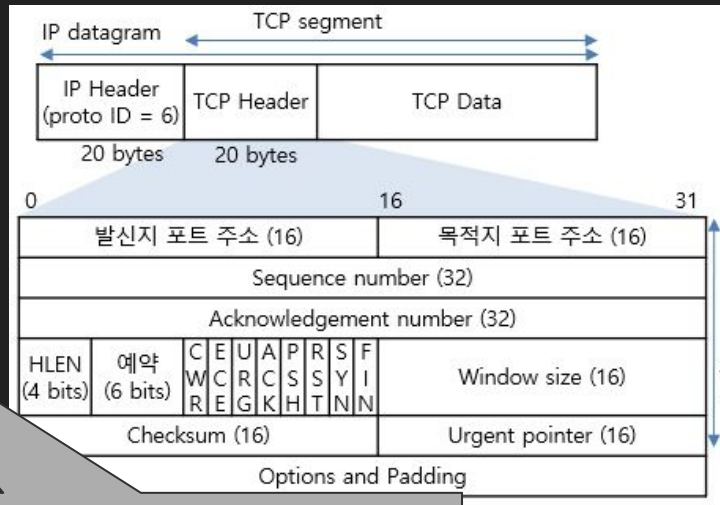
Forward와 Backward를 따로 구성해줘야 함

이 부분은 ethernet, ip, tcp 헤더의 구조와 각 필드의 역할을 알아야 할 수 있음

이번 시간에는 TCP 헤더 구성 방법만 알아봄

구체적으로 패킷을 어떻게 만들어야 하죠

필드명	Forward	Backward
port	src와 dst 동일하게	swap(src, dst)
seq number	seq + payload length	ack
ack	ack	seq
flag	RST flag on	
checksum	직접 계산	



server에 original packet이
도착하기 전에 RST 패킷을
보낸 것으로 가정

근데 왜 이게 성공하는 거임?

이론상 **server**가 다시 패킷을 보내는 시간보다

같은 라우터 안에서 쏘는 내 패킷이 **client**한테 먼저 도착함

그러면 내 패킷을 신뢰하고 나중에 오는 **server**의 패킷은 **drop**하게 됨

이는 후술할 **warning.or.kr** 사이트에도 적용되는 이야기

코드 레벨로 구경하러 가볼까요

<https://github.com/bob12vpn/vpn-hater>

(스타 누르지 마세요)

왜 제목을 그렇게 정했을까

오늘의 진짜 제목 : TCP 통신 자살(을 가장한 타살)시키기

client 시점 -> 서버가 **RST** 보냈네. 연결 끊어야지.

server 시점 -> 클라이언트가 **RST** 보냈네. 연결 끊어야지.

FIN flag도 사용할 수 있으나, 부득이하게 이 방법은 사용해보지 않아서 **RST**로 소개하였음.

Packet Injection은 어디서 사용되고 있는가

유해 사이트 접속 시 해당 사이트 정보를 받기 전에
먼저 packet을 보내서 warning.or.kr 페이지를 띄워버림

당연히 외국보다는
국내에서 쓴 패킷이
먼저 도달하기 때문



The screenshot shows the Warning.or.kr website with a blue header. The header contains the KCSC logo, the word "Warning" in large white text, and the Korean National Police logo. Below the header, there is a warning message in Korean. The message states that the user is about to access a site with illegal content and that the site has been blocked by the Korea Communications Commission (KCC). It also mentions that the site is being monitored by the KCSC and that the user should be cautious. Below the message, there is a table with three columns: "사이트분야" (Site Category), "담당기관" (Responsible Agency), and "전화번호" (Phone Number). The table lists various categories of illegal sites and the corresponding agencies and phone numbers for reporting.

Warning
불법·유해 정보(사이트)에 대한 차단 안내

지금 접속하려고 하는 정보(사이트)에서 **불법·유해** 내용이 제검되고 있
이에 대한 접속이 차단되었음을 알려드립니다.

해당 정보(사이트)는 **방송통신심의위원회(KCSC)**의 심의를 거쳐
「방송통신위원회의 설치 및 운영에 관한 법률」에 따라 적법하게 차단된 것이오니
이에 관한 문의사항이 있으시면 아래의 담당기관으로 문의하여 주시기 바랍니다.

※ 차단안내페이지(warning.or.kr)를 이용한 피싱사이트가 발견되어 각별한 주의가 필요합니다.
(차단안내페이지는 개인정보를 요구하거나 프로그램 설치를 유도하지 않습니다.)

사이트분야	담당기관	전화번호
불법 도박	사이버 경찰청	1566-0112
	사행산업통합감독위원회	1855-0112
	사행산업통합감독위원회	1855-0112
불법 채굴진흥료 부과 판매	국민체육진흥공단	1899-1119
	올림픽스포츠 통합센터	1899-1119
불법 송자투표권 구매대행	국민체육진흥공단 경륜·경정 총괄본부	1899-0707
불법 마권 구매대행	한국마사회	080-8282-112
불법 의약품 판매	식품의약품안전처 사이버조서팀	(043)719-1913, 1921
불법 의약품 판매 및 허위과대광고	식품의약품안전처 사이버조서팀	(043)719-1921, 1943
	식품의약품안전처 사이버조서팀	(043)719-1912, 1914, 1919, 1944

마무리

질문 받습니다

출 처

- <https://huimang2.github.io/etc/iso-standard-7498>
- http://www.ktword.co.kr/test/view/view.php?m_temp1=1889
- <https://velog.io/@kimyeji203/%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC-TCPUDP>