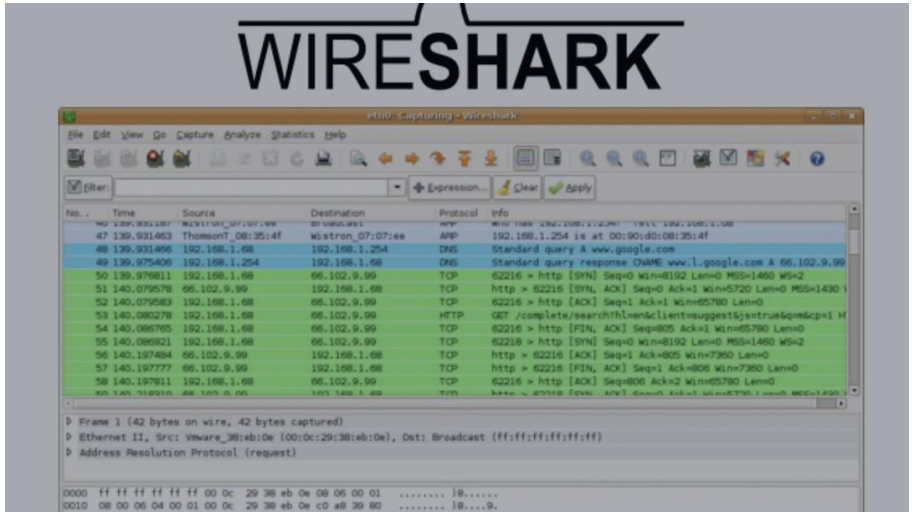
Wireshark

설치부터 패킷 캡처/분석 기본 사용법



네트워크 패킷 캡처

Display Filter

HTTP/TCP 실습

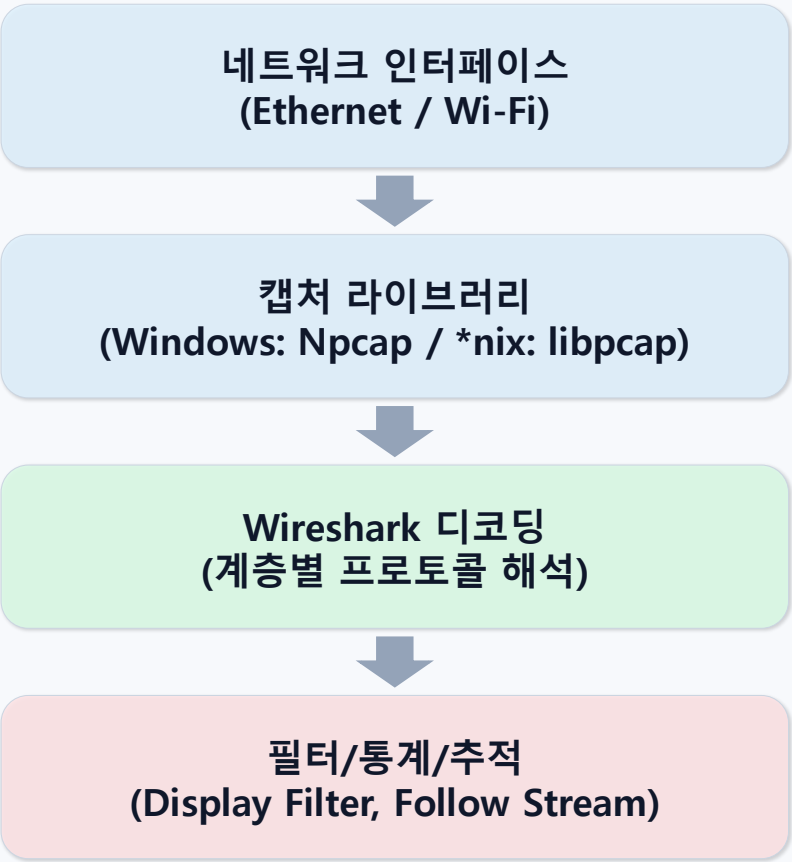
Wireshark와 패킷 캡처

패킷 캡처(Packet Capture)는 네트워크 인터페이스를 통과하는 데이터 패킷을 기록하고 분석하는 기술입니다.

Wireshark는 가장 널리 쓰이는 오픈 소스 프로토콜 분석기로 패킷을 계층별로 디코딩해 보여줍니다.

대표 활용 사례

- 네트워크 문제 해결(지연/끊김/접속 실패 원인 파악)
- 프로토콜 동작 학습(TCP 3-way, HTTP 구조 등)
- 보안 분석(의심 트래픽/악성 행위 단서 찾기)
- 성능 분석(지연 시간, 전송량, 재전송 등)



설치 전 체크리스트

핵심 포인트: "캡처 권한/드라이버"가 준비되어야 실제 캡처가 됩니다.

Npcap 필요

Windows

- Wireshark 설치 파일에 Npcap 포함
- Npcap 없으면 라이브 캡처 불가(파일 열기는 가능)
- 기본값으로 설치하면 대부분 OK

권한 설정

macOS

- DMG에서 /Applications로 드래그
- 처음 실행 시 권한 관련 안내 가능
- 필요 시 ChmodBPF로 캡처 권한 부여

libpcap

Linux/UNIX

- 보통 패키지 매니저로 설치
- 캡처는 root 권한/캡처 그룹 설정 필요
- 배포판 정책에 따라 권한 방식이 다름

TIP: 캡처는 "권한 + 캡처 드라이버/라이브러리"가 모두 필요합니다. 설치 후 인터페이스 목록이 비어있으면 드라이버/권한을 먼저 점검하세요.

설치: Windows

핵심: 설치 중 “Install Npcap” 체크(기본값) — 없으면 라이브 캡처가 안 됩니다.

설치 절차(권장)

- 공식 사이트에서 최신 Stable 버전 다운로드
- 설치 마법사 대부분 “Default”로 진행
- Npcap 설치 옵션은 반드시 포함(기본 체크)
- USB 트래픽 분석이 필요할 때만 USBPcap 옵션 고려
- 설치 완료 후 Wireshark 실행 → 인터페이스 목록 확인

Npcap이 필요한 이유

Windows에서는 패킷을 “가로채어” 사용자 공간으로 전달하는 캡처 드라이버가 필요합니다.

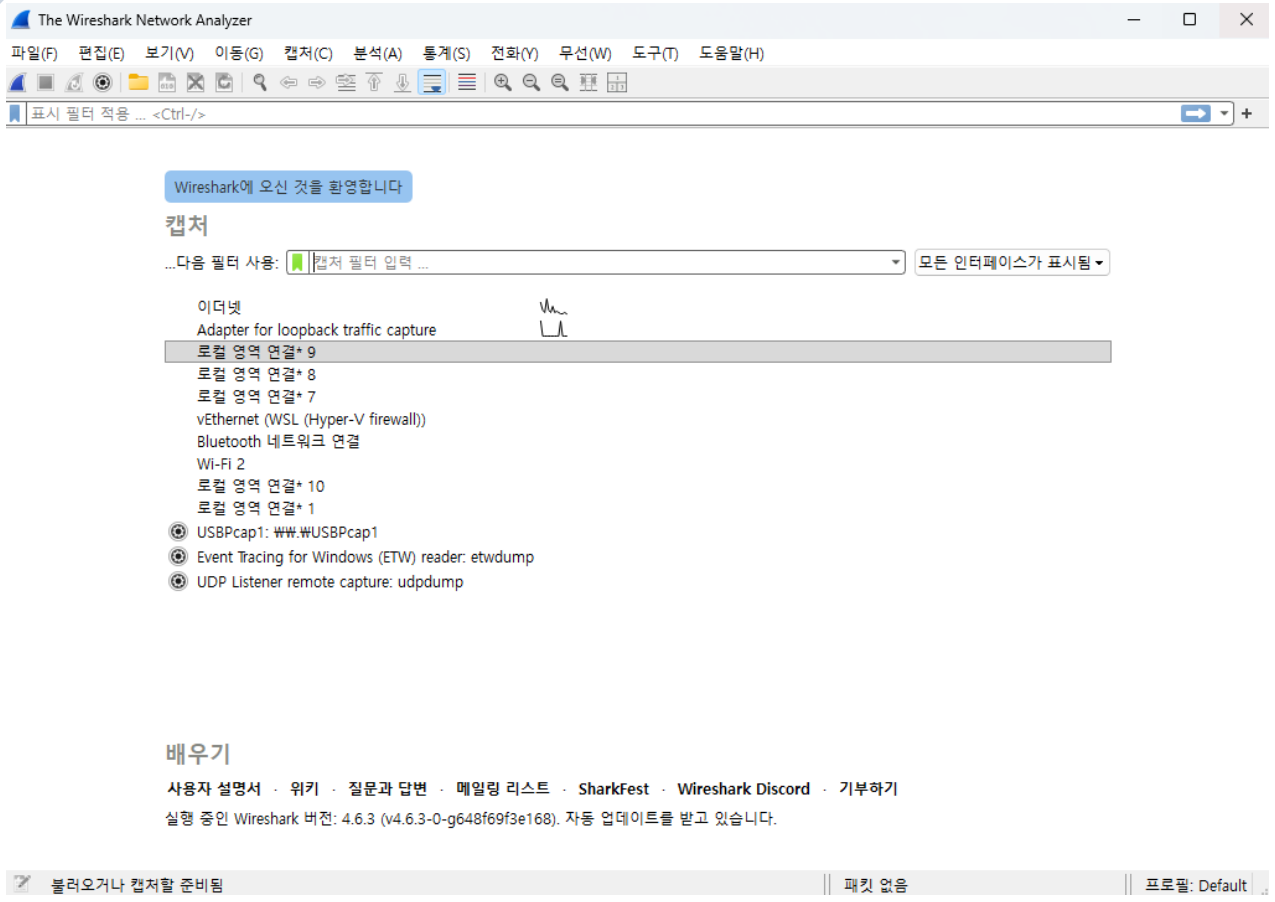
Wireshark는 Npcap을 통해 라이브 캡처를 수행합니다.

설치 후 문제(자주 겪는 것)

- 인터페이스가 안 보임 → Npcap 설치/서비스 상태 확인
- 캡처가 0 packets → 올바른 인터페이스(활동 그래프) 선택
- 권한 오류 → 관리자 권한으로 실행(환경에 따라 필요)

첫 실행: 인터페이스 선택과 캡처 시작

인터넷에 연결된 "활성 인터페이스"를 선택하세요 (그래프가 흔들립니다).



빠른 절차

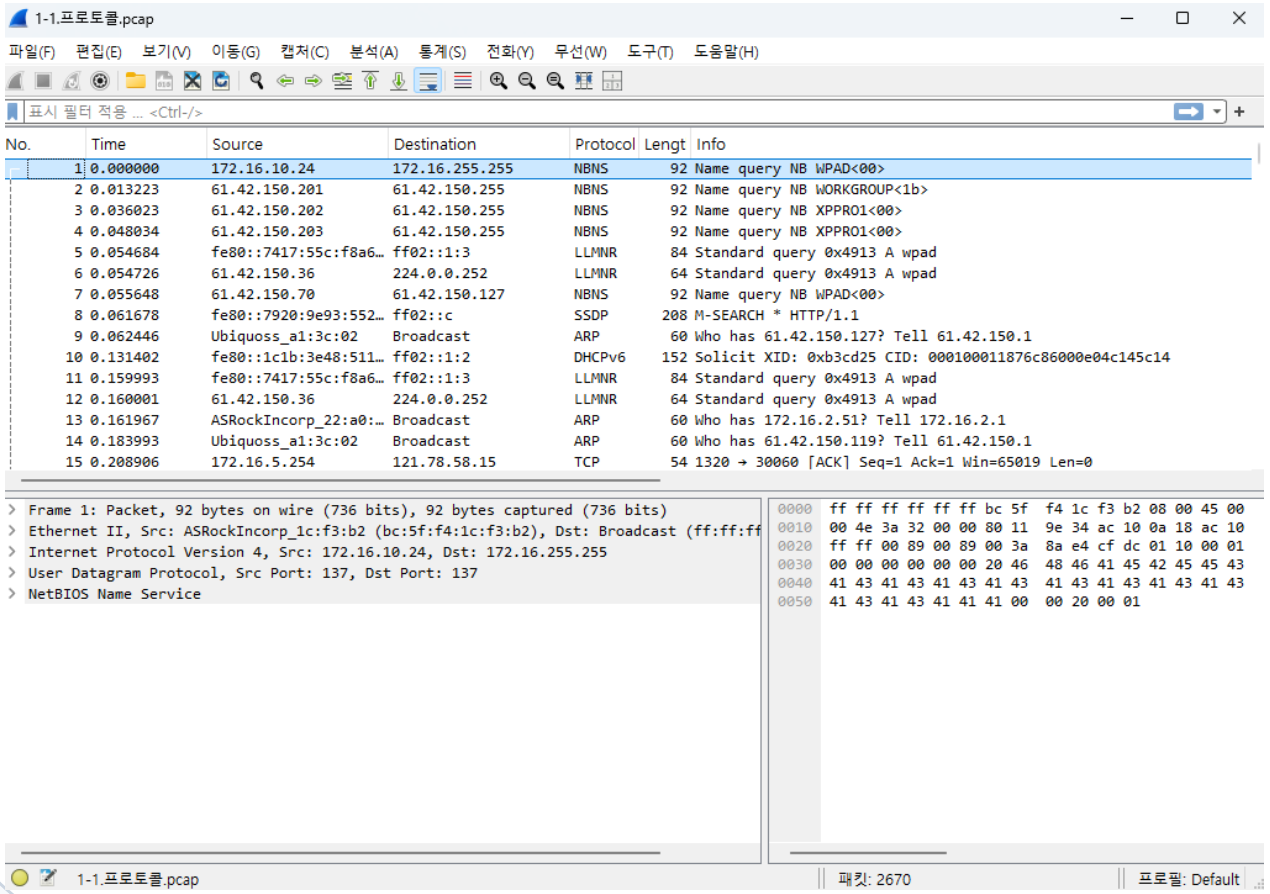
- (1) 인터페이스 선택
- (2) 상어 지느러미(Start) 클릭
- (3) 트래픽 발생(웹 접속 등)
- (4) Stop으로 중지
- (5) Display Filter로 추리기

TIP

원치 않는 패킷까지
모두 잡힙니다.
실습은 짧게 캡처하고
필터로 분석하세요.

기본 화면 구성(3 Pane) - Windows

인터넷에 연결된 "활성 인터페이스"를 선택하세요 (그래프가 흔들립니다).



빠른 절차

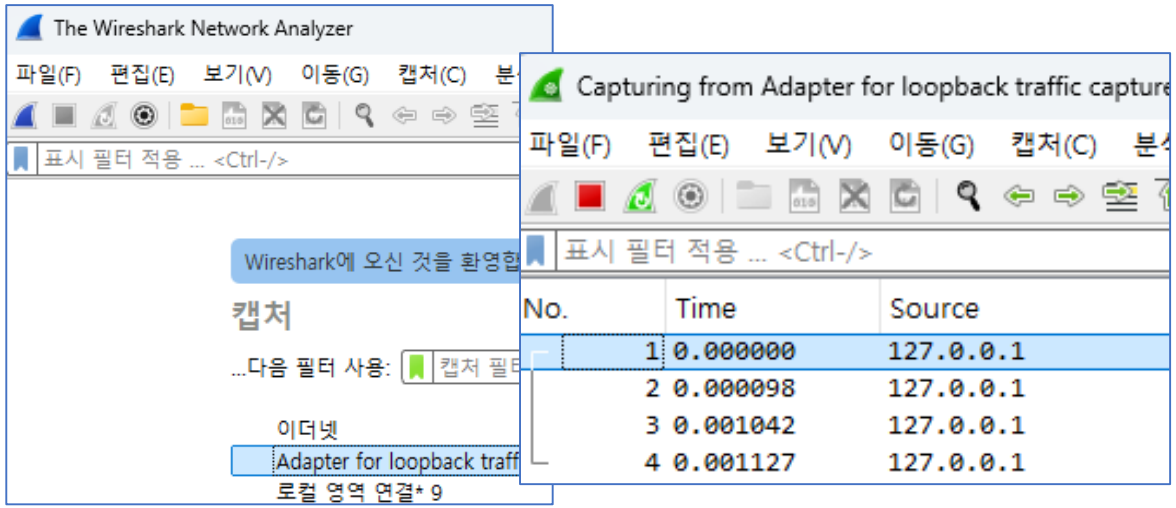
- (1) 인터페이스 선택
- (2) 상어 지느러미(Start) 클릭
- (3) 트래픽 발생(웹 접속 등)
- (4) Stop으로 중지
- (5) Display Filter로 추리기

TIP

원치 않는 패킷까지
모두 잡힙니다.
실습은 짧게 캡처하고
필터로 분석하세요.

캡처 컨트롤(시작/중지/재시작)

상단 툴바의 아이콘으로 캡처를 제어합니다. (Start / Stop / Restart)



미니 실습

Start → 10초 동안 웹사이트 1회 접속 → Stop

버튼 의미

Start	선택한 인터페이스에서 캡처 시작
Stop	캡처 중지(분석은 계속 가능)
Restart	캡처를 초기화하고 다시 시작
Options	캡처 옵션(인터페이스/모니터 모드 등)

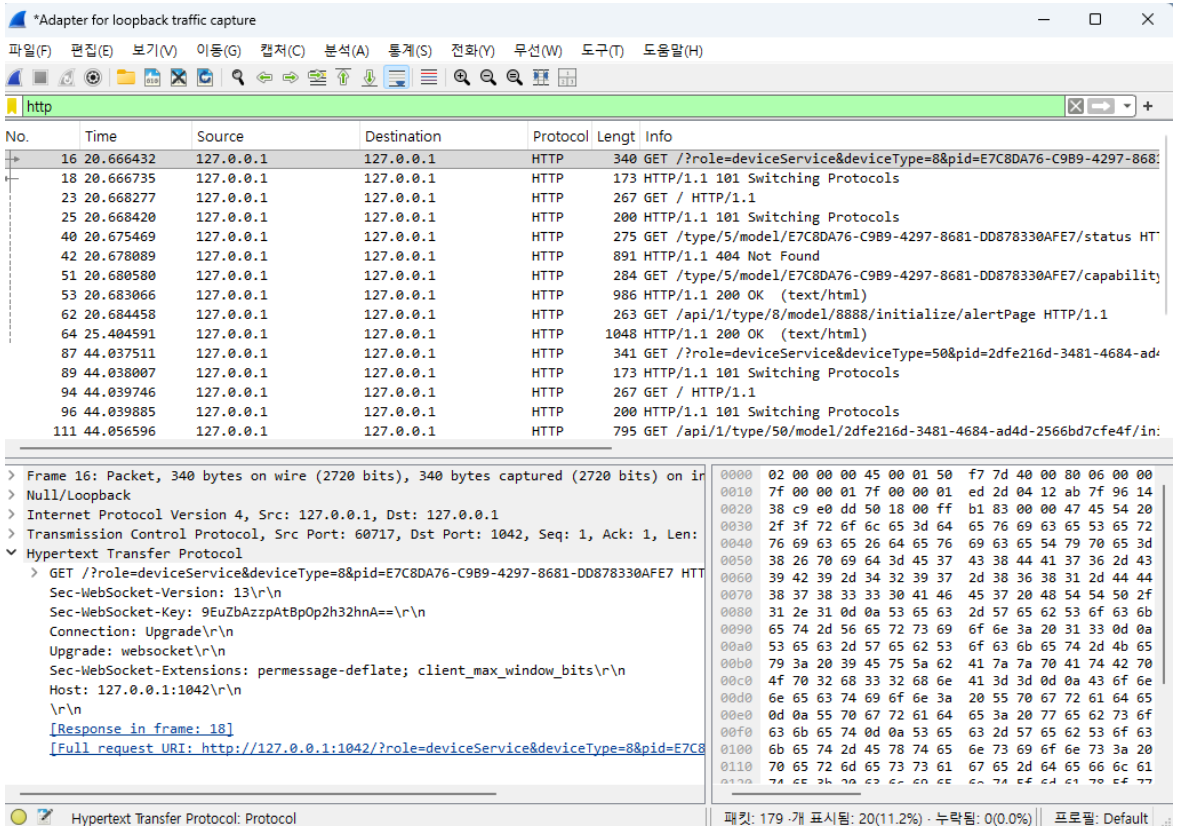
필터링: Capture Filter vs Display Filter

Capture Filter

캡처 "전에" 거르기 (저장되는 패킷 자체를 줄임)

Display Filter

캡처 "후에" 화면에서 골라 보기 (분석의 핵심)



Display Filter 예시

```
http
dns
ip.addr == 192.168.0.1
tcp.port == 80
!(arp)
```

필터 입력창은 초록색이면 "문법 OK"

Wireshark 디스플레이 필터(필터 명령어) 표

항목	필터명령어
TCP 패킷만	tcp
UDP 패킷만	udp
ICMP 패킷만	icmp
ARP 패킷만	arp
IPv4 패킷만	ip
IPv6 패킷만	ipv6
출발지 IP	ip.src == 192.168.0.10
목적지 IP	ip.dst == 192.168.0.20
IP 대역(서브넷) 포함	ip.addr == 192.168.0.0/24
특정 IP 포함(출발지/목적지 모두)	ip.addr == 192.168.0.10
출발지 TCP 포트	tcp.srcport == 443
목적지 TCP 포트	tcp.dstport == 80
TCP 포트(출발지/목적지 포함)	tcp.port == 22
출발지 UDP 포트	udp.srcport == 53
목적지 UDP 포트	udp.dstport == 500
UDP 포트(출발지/목적지 포함)	udp.port == 123
출발지 MAC	eth.src == aa:bb:cc:dd:ee:ff
목적지 MAC	eth.dst == 11:22:33:44:55:66
특정 MAC 포함(출발지/목적지)	eth.addr == aa:bb:cc:dd:ee:ff

항목	필터명령어
HTTP 트래픽	http
HTTP 요청(Method)	http.request.method == "GET"
HTTP Host 헤더	http.host == "example.com"
HTTP URI 포함	http.request.uri contains "/login"
HTTPS(TLS) 트래픽	tls
TLS SNI(서버 이름)	tls.handshake.extensions_server_name == "example.com"
DNS 트래픽	dns
DNS 질의 이름	dns.qry.name == "example.com"
DNS 응답 IP 포함	dns.a == 1.2.3.4
DHCP(BOOTP)	dhcp (또는 bootp)
SSH	ssh
FTP	ftp
SMTP	smtp
TCP SYN 패킷만(연결 시작)	tcp.flags.syn == 1 and tcp.flags.ack == 0
SYN/ACK 패킷만	tcp.flags.syn == 1 and tcp.flags.ack == 1
RST 패킷만(강제 종료/리셋)	tcp.flags.reset == 1
FIN 패킷만(정상 종료)	tcp.flags.fin == 1
재전송 표시(분석 플래그)	tcp.analysis.retransmission
중복 ACK(분석 플래그)	tcp.analysis.duplicate_ack
프레임 길이(바이트) 조건	frame.len > 1000
에러/경고 패킷(Expert Info)	expert.severity >= 3
특정 문자열 포함(패킷 전체)	frame contains "password"
특정 바이트 시퀀스 포함(hex)	frame contains 0xdeadbeef
특정 시간 이후 패킷(예: epoch)	frame.time_epoch > 1700000000

필터 연산 기호(연산자) 표

항목	내용
&&	AND (그리고) (=and)
||	OR (또는) (=or)
!	NOT (부정) (=not)
==	같다
!=	같지 않다
>	크다
<	작다
>=	크거나 같다
<=	작거나 같다
contains	문자열/바이트 "포함"
matches	정규식 매칭(Regex)
in	집합/범위 포함(예: 포트 범위 등에서 활용)
and	AND (키워드 형태)
or	OR (키워드 형태)
not	NOT (키워드 형태
()	우선순위 묶기(그룹핑)

Wireshark Filters

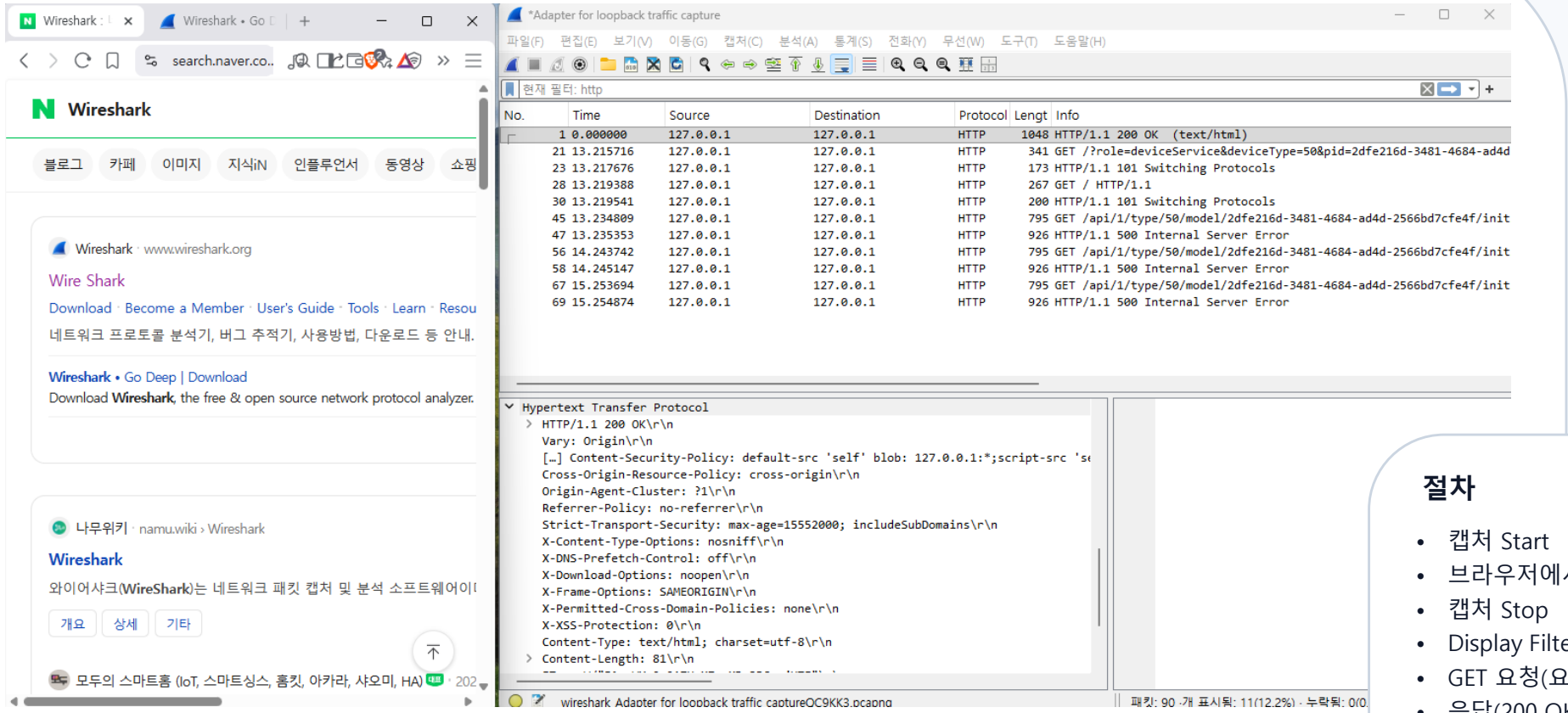
항목	필터명령어
특정 IP와 HTTP만 보기	ip.addr == 192.168.0.10 && http
특정 IP의 DNS 질의만 보기	ip.src == 192.168.0.10 && dns
특정 서버(목적지 IP)로 가는 TCP만 보기	ip.dst == 10.0.0.5 && tcp
특정 서버로 가는 웹(80/443)만 보기	`ip.dst == 10.0.0.5 && (tcp.port == 80
내 PC에서 나가는 트래픽만(출발지 IP)	ip.src == 192.168.0.10
내 PC로 들어오는 트래픽만(목적지 IP)	ip.dst == 192.168.0.10
특정 포트 "제외"하기(예: 443 제외)	tcp && tcp.port != 443
내 PC 트래픽 중 DNS만 제외하고 보기	ip.addr == 192.168.0.10 && !dns
브로드캐스트/멀티캐스트 제외하고 보기	eth.dst[0] & 1 == 0
TCP 3-way 핸드셰이크(SYN / SYN-ACK / ACK)	(tcp.flags.syn==1 && tcp.flags.ack==0)
TCP 연결 시작(SYN)만 모아보기	tcp.flags.syn == 1 && tcp.flags.ack == 0
RST 발생만 모아보기(연결 문제/차단 의심)	tcp.flags.reset == 1
재전송만 모아보기(품질 문제 분석)	tcp.analysis.retransmission
중복 ACK만 모아보기(혼잡/손실 징후)	tcp.analysis.duplicate_ack
지연/손실 의심(재전송 + dup ack 같이 보기)	tcp.analysis.retransmission
특정 문자열 포함(로그인/토큰 등 탐색)	frame contains "token"
POST 요청만 보기(웹 폼/로그인 시도)	http.request.method == "POST"
특정 Host로 가는 HTTP만 보기	http.host == "example.com"
특정 URI 경로 포함(예: /login)	http.request.uri contains "/login"
TLS SNI로 특정 도메인만 보기(HTTPS에서 유용)	tls.handshake.extensions_server_name == "example.com"
DNS에서 특정 도메인만 보기	dns.qry.name == "example.com"
DNS 응답에 특정 IP가 포함된 것만	dns.a == 1.2.3.4
패킷 크기 큰 것만(예: 1400B 초과)	frame.len > 1400
짧은 패킷만(예: ACK-only 분석)	frame.len < 100
대역 전체에서 특정 포트 트래픽(예: 사내망 10.0.0.0/8의 3389)	ip.addr == 10.0.0.0/8 && tcp.port == 3389

Wireshark Filters

항목	필터명령어
특정 IP와 HTTP만 보기	ip.addr == 192.168.0.10 && http
특정 IP의 DNS 질의만 보기	ip.src == 192.168.0.10 && dns
특정 서버(목적지 IP)로 가는 TCP만 보기	ip.dst == 10.0.0.5 && tcp
특정 서버로 가는 웹(80/443)만 보기	`ip.dst == 10.0.0.5 && (tcp.port == 80
내 PC에서 나가는 트래픽만(출발지 IP)	ip.src == 192.168.0.10
내 PC로 들어오는 트래픽만(목적지 IP)	ip.dst == 192.168.0.10
특정 포트 "제외"하기(예: 443 제외)	tcp && tcp.port != 443
내 PC 트래픽 중 DNS만 제외하고 보기	ip.addr == 192.168.0.10 && !dns
브로드캐스트/멀티캐스트 제외하고 보기	eth.dst[0] & 1 == 0
TCP 3-way 핸드셰이크(SYN / SYN-ACK / ACK)	(tcp.flags.syn==1 && tcp.flags.ack==0)
TCP 연결 시작(SYN)만 모아보기	tcp.flags.syn == 1 && tcp.flags.ack == 0
RST 발생만 모아보기(연결 문제/차단 의심)	tcp.flags.reset == 1
재전송만 모아보기(품질 문제 분석)	tcp.analysis.retransmission
중복 ACK만 모아보기(혼잡/손실 징후)	tcp.analysis.duplicate_ack
지연/손실 의심(재전송 + dup ack 같이 보기)	tcp.analysis.retransmission
특정 문자열 포함(로그인/토큰 등 탐색)	frame contains "token"
POST 요청만 보기(웹 폼/로그인 시도)	http.request.method == "POST"
특정 Host로 가는 HTTP만 보기	http.host == "example.com"
특정 URI 경로 포함(예: /login)	http.request.uri contains "/login"
TLS SNI로 특정 도메인만 보기(HTTPS에서 유용)	tls.handshake.extensions_server_name == "example.com"
DNS에서 특정 도메인만 보기	dns.qry.name == "example.com"
DNS 응답에 특정 IP가 포함된 것만	dns.a == 1.2.3.4
패킷 크기 큰 것만(예: 1400B 초과)	frame.len > 1400
짧은 패킷만(예: ACK-only 분석)	frame.len < 100
대역 전체에서 특정 포트 트래픽(예: 사내망 10.0.0.0/8의 3389)	ip.addr == 10.0.0.0/8 && tcp.port == 3389

실습 1: HTTP 요청/응답 분석하기

목표: 웹사이트 접속 시 발생하는 HTTP 패킷을 찾아 "요청 라인/헤더"를 확인한다.



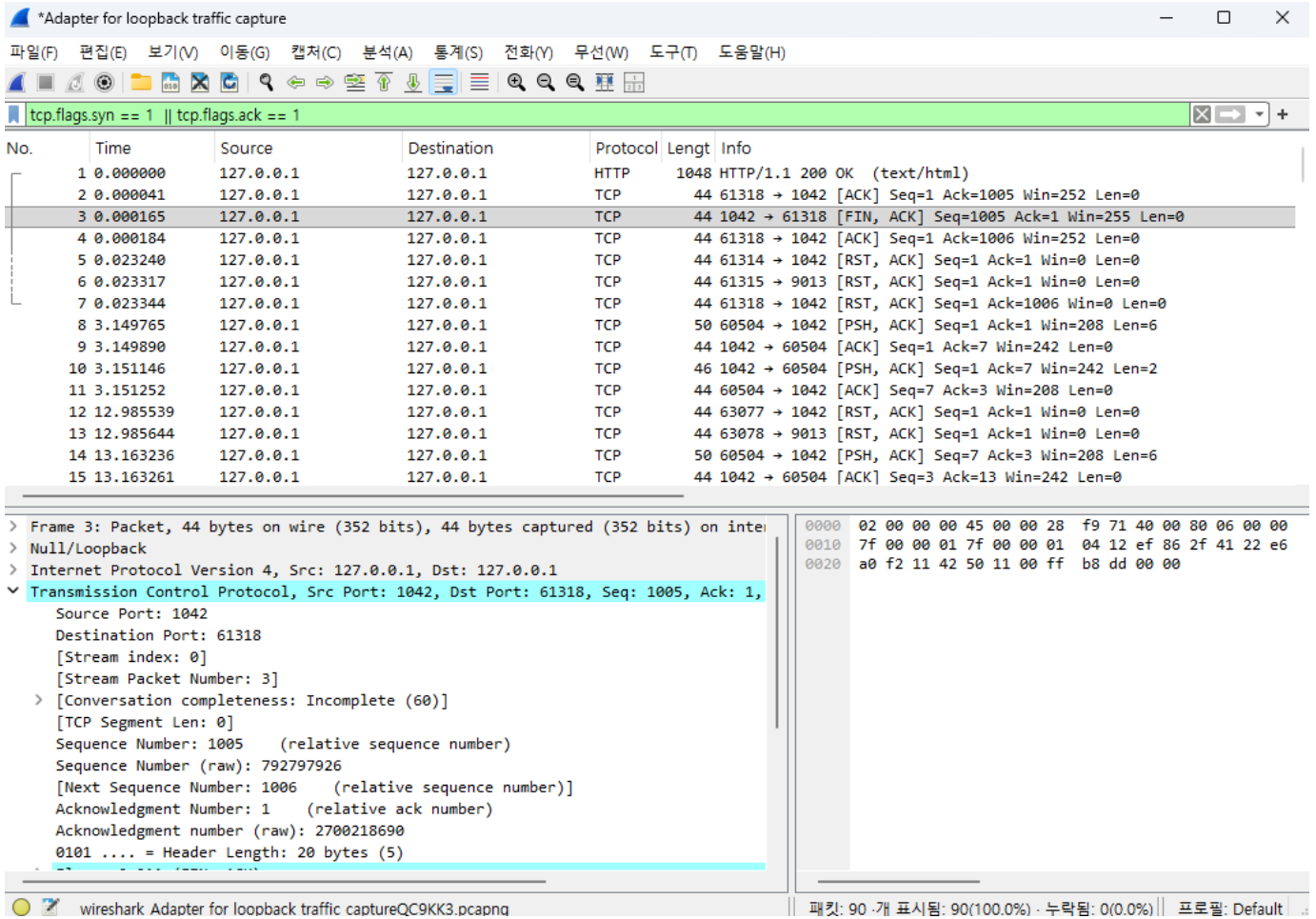
절차

- 캡처 Start
- 브라우저에서 HTTP 사이트 접속
- 캡처 Stop
- Display Filter에 http 입력
- GET 요청(요청 라인/Host/User-Agent) 확인
- 응답(200 OK 등) 확인

HTTPS는 페이로드가 암호화되어 내용 확인이 어렵습니다(예외: TLS 분석).

실습 2: TCP 3-way Handshake 확인

목표: SYN → SYN/ACK → ACK 패킷을 찾아 연결 성립 과정을 확인한다.



필터 예시

```
tcp.port == 80 &&
(tcp.flags.syn == 1
 || tcp.flags.ack == 1)
```

해석

SYN: 연결 요청
SYN/ACK: 수락
ACK: 최종 확인

핸드셰이크가 보이면 "연결은 됐다"는 의미입니다.

연결 실패 분석은 RST/재전송/타임아웃 등을 추가로 확인

문제 상황 예

1) IP를 못 받음 / “네트워크에 연결 됨”인데 인터넷 안 됨 (DHCP)

항목	내용
현실 증상 예시	”새 노트북이 사내(어떤 장소) Wi-Fi에 붙었는데 ‘인터넷 없음’ / IP가 169.254.x.x로 잡힘”
필터	dhcp
핵심 체크 항목	Discover → Offer → Request → Ack 흐름이 있는지, 어디서 끊기는지
해석 포인트	Discover만 있고 Offer가 없음 → DHCP 서버까지 요청이 안 가거나(라우팅/VLAN/릴레이), DHCP 서버 문제. Offer는 오는데 Ack가 없음 → DHCP 서버에서 임대 거부/정책/스코프 부족 가능
추가	arp(게이트웨이 MAC 해석), dns(IP 받은 뒤 DNS 되는지)

2) “사이트가 안 열림”인데 IP로는 접속됨 (DNS 문제)

항목	내용
현실 증상 예시	”브라우저로 example.com은 안 열리는데 1.2.3.4로는 접속됨”
필터	dns
핵심 체크 항목	Query는 나가는데 Response가 오는지, Response의 rcode(NXDOMAIN 등), 응답 지연
해석 포인트	Query만 반복 → DNS 서버 응답 없음(방화벽/경로/서버 장애). NXDOMAIN → 도메인 자체 문제 또는 내부 DNS split-brain 설정 문제. 응답이 매우 늦음 → DNS 서버 과부하/원격 DNS 사용
실전 팁	Wireshark에서 DNS는 dns.time(질의~응답 시간)을 보면 ”느림”을 바로 수치로 확인 가능

문제 상황 예

3) “핑은 되는데 접속이 안 되는 경우” (TCP 3-way handshake 실패)

항목	내용
현실 증상 예시	“서버 IP로 ping은 되는데, 웹/SSH 접속이 계속 실패”
필터	tcp.flags.syn == 1 && tcp.flags.ack == 0 (SYN만 먼저 확인)
핵심 체크 항목	SYN이 간 뒤 SYN/ACK이 돌아오는지, 아니면 SYN 재전송만 계속되는지
해석 포인트	SYN만 계속 → 서버 포트 미오픈 / 방화벽 차단 / 중간 장비에서 드롭. SYN/ACK은 오는데 이후가 꼬임 → 클라이언트 측 방화벽/보안SW, 비대칭 라우팅 가능
추가 필터	tcp.flags.syn == 1 && tcp.flags.ack == 1(SYN/ACK), tcp.flags.reset == 1(RST)

4) “접속하자마자 바로 끊기는 상황” (RST 발생)

항목	내용
현실 증상 예시	“로그인 누르면 바로 튕김 / SSH 접속하자마자 종료”
필터	tcp.flags.reset == 1
핵심 체크 항목	RST가 누가 보냈는지(Source/Destination), RST 직전에 어떤 패킷이 있었는지
해석 포인트	서버가 RST → 해당 포트 서비스가 없거나 앱이 즉시 연결 종료(정책/오류). 중간 장비가 RST처럼 보이는 패턴 → 방화벽/IPS가 세션 차단(특정 패턴 탐지) 가능
실전 팁	같은 시점에 http(응답코드), tls.alert_message(TLS 경고)도 함께 보면 원인 단서가 빨리 나옵니다

문제 상황 예

5) “느리고 / 가끔 끊기는 상황” (패킷 손실·재전송)

항목	내용
현실 증상 예시	"업로드가 느림", “화상회의가 끊김”, “RDP가 버벅임"
필터	`tcp.analysis.retransmission
핵심 체크 항목	재전송이 **특정 구간/특정 서버/특정 방향(업/다운)**에 집중되는지
해석 포인트	재전송↑/dup ack↑ → 손실/혼잡/무선 품질 저하 강력 의심. 특히 Wi-Fi에서 흔함. 한 방향만 심하면 업링크/다운링크 중 한쪽 문제일 수 있음
추가 체크	frame.len(큰 패킷에서만 손실?), 시간대별로 늘어나는지(혼잡)

6) “다운로드는 괜찮은데 업로드가 유독 느림” (TCP 윈도우/수신 버퍼 이슈)

항목	내용
현실 증상 예시	"파일 업로드만 느리고 다운로드는 빠름"
필터	`tcp.analysis.zero_window
핵심 체크 항목	Zero Window / Window Full가 반복되는지, 어느 쪽이 광고(advertise)하는지
해석 포인트	수신 측(보통 서버)이 Zero Window → 서버/앱이 데이터를 처리 못해 버퍼가 꽉 참(디스크/CPU/앱 병목). 클라이언트가 Zero Window → 클라이언트 PC 자원/보안SW/프로세스 병목 가능

문제 상황 예

7) “사내에서는 되는데 외부(또는 VPN)에서는 큰 파일만 실패” (MTU/조각화/PMTUD)

항목	내용
현실 증상 예시	"로그인은 되는데 다운로드가 중간에 멈춤", “VPN에서 대용량 전송만 실패"
필터(IPv4)	icmp.type==3 && icmp.code==4
필터(IPv6)	icmpv6.type==2
핵심 체크 항목	ICMP Fragmentation needed(IPv4) / **Packet Too Big(IPv6)**가 보이는지
해석 포인트	PMTUD가 막히면(중간 ICMP 차단) 큰 패킷 경로에서 전송이 깨짐. VPN/터널 환경에서 특히 흔함
추가	ip.flags.df == 1(DF 설정 확인), TCP MSS(너무 큰지)

8) “간헐적으로 다른 사람 PC가 내 IP를 쓰는 것 같아요” (IP 충돌/ARP 이상)

항목	내용
현실 증상 예시	"갑자기 연결이 끊기고 다시 되었다가 반복", “게이트웨이는 멀쩡한데 내 PC만 간헐 장애"
필터	arp
핵심 체크 항목	같은 Sender IP인데 Sender MAC이 바뀌는지, Gratuitous ARP가 반복되는지
해석 포인트	동일 IP에 대해 MAC이 번갈아 나타나면 IP 충돌 또는 ARP 스푸핑/보안장비 개입 의심
실전 팁	문제가 난 시간대 전후로 ARP Reply(opcode 2)가 "누가 내 IP를 주장했는지"를 보면 원인이 빨리 좁혀집니다

문제 상황 예

9) “특정 사이트만 안 되고 다른 건 돼요” (프록시/방화벽 정책/차단)

항목	내용
현실 증상 예시	"업무 사이트 A만 접속 불가(다른 사이트는 OK)”, “회사망에서만 막힘"
필터(HTTP)	http && ip.addr == 192.168.0.10
필터(HTTPS 프록시 흔한 경우)	http.request.method == "CONNECT"
핵심 체크 항목	HTTP 응답코드(예: 403/407), CONNECT 응답 상태, 특정 도메인에서만 실패하는지
해석 포인트	407 Proxy Auth Required → 프록시 인증 문제. 403/차단 페이지 → 정책 차단. CONNECT 실패 → 프록시가 대상 도메인/포트를 막는 경우
추가 체크	TLS SNI(tls.handshake.extensions_server_name)로 차단 도메인 추적

10) “로그인 실패 / 자격증명은 맞는데 반복 재시도” (HTTP 상태코드·세션/리다이렉션)

항목	내용
현실 증상 예시	"아이디/비번 맞는데 계속 로그인 화면으로 돌아감"
필터	http
핵심 체크 항목	302/301 리다이렉트 루프, 401/403, 쿠키/세션 관련 흐름(요청-응답 반복)
해석 포인트	302가 반복되면 세션/쿠키 문제(프록시/도메인/시간동기/보안장비). 401은 인증 실패, 403은 권한/정책 차단 가능
실전 팁	같은 tcp.stream만 골라 "한 세션"의 요청→응답 반복 패턴을 보는 게 빠릅니다