

## 서론

Wireshark라는 앱이 패킷을 찾기 위한 도구라는 것만 알고, 막상 실행해보면 여러 문자와 숫자의 향연에 이 앱을 어떻게 사용해야 할 지 몰랐다. 어떤 기능들이 있는지도 잘 모르고, 실행했을 때 뜨는 줄마다 색이 다 다른 것엔 어떤 이유가 있는지, 어떤 기준으로 인해 색이 달라지는지, 프로토콜의 종류는 무엇이 있는지 등 wireshark를 실행했을 때 화면에 뜨는 기본적인 것부터 알아가며 어떤 기능을 하고, 어떻게 작동하는지, 어떤 의미인지 알아가는 것부터 시작하려고 한다.

Wireshark를 이해하고 조작하는것에 익숙해지면 wireshark를 사용해 패킷을 분석하고 캡쳐하는것부터 시작해서 간단한 위게임을 연습해볼 예정이다.

## 개요

1. Wireshark란?
  - 1.1 Wireshark란?
  - 1.2 Wireshark 주요 기능
2. 패킷이란?
  - 2.1 패킷이란?
  - 2.2 패킷의 구성요소
3. 패킷 컬러
4. Wireshark 인터페이스
  - 4.1 시작, 종료, 다시시작
  - 4.2 실행 창 분류
  - 4.3 no., time, source, destination, protocol, length, info
5. 패킷 필터링 기능
  - 5.1 프로토콜 별 분리

## 4.2 논리 연산자 필터 검색 (ip.addr : ip 필터)

### 본론

#### 1. Wireshark 란?

네트워크 패킷을 캡처 및 분석 소프트웨어

네트워크 인터페이스에서 트래픽을 실시간으로 캡처하고, 패킷의 상세정보를 확인할 수 있다.

주요 기능 :

패킷 내용 분석, 패킷 필터링/검색, 캡처 데이터를 열거나 저장, 네트워크 인터페이스 RAW 패킷 캡처 등이 있다.

#### 2. 패킷이란?

패킷은 패키지(package)와 덩어리를 뜻하는 버킷(bucket)의 합성어로 통신망을 통해 전송하기 쉽도록 데이터를 잘게 나눈 전송 단위이다.

↳ 네트워크에서 데이터를 전송할 때 작은 단위로 나누어진 데이터

원래 패킷은 소포를 뜻하는 용어인데 우체국에서 화물을 나누어 행선지를 표시하여 꼬리표를 붙이는 작업을 데이터 통신에 접목한 용어로 사용하고 있다.

즉, 패킷은 UDP, TCP, IP 등 모두가 가지고 있는 데이터 조각이라고 이해하면 편하다

- UDP : 빠른 데이터 전송을 위한 비연결형 프로토콜이다. TCP에 비해 안전성은 떨어지지만 더 빠르고 간단하다. 일부 패킷이 누락되더라도 데이터를 전송하므로 패킷 손실로 인해 전체 전송이 중단되지 않는다.
- TCP : 데이터 송신의 신뢰성을 위해 사용하는 프로토콜이다. 데이터를 작은 조각으로 나눈 후 보내면서 각 조각이 제대로 도착했는지 확인한다. 만약 데이터가 누락되거나 순서가 어긋났을 경우, TCP는 해당 데이터를 다시 전송하거나 올바른 순서로 배열해준다. (신뢰성 있는 데이터 전송을 보장하는 프로토콜)
- IP : 인터넷 주소 관리 역할을 담당하는 프로토콜이다. 데이터를 목적지까지 전달하는 역할을 하지만 데이터의 손실이 가능하다. (신뢰성 보장을 위해 TCP같은 상위 프로토콜과 함께 사용) 이라는 설명이 어울릴까? 패킷을 설명할때

```
명령 프롬프트
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ixlix>ping google.com

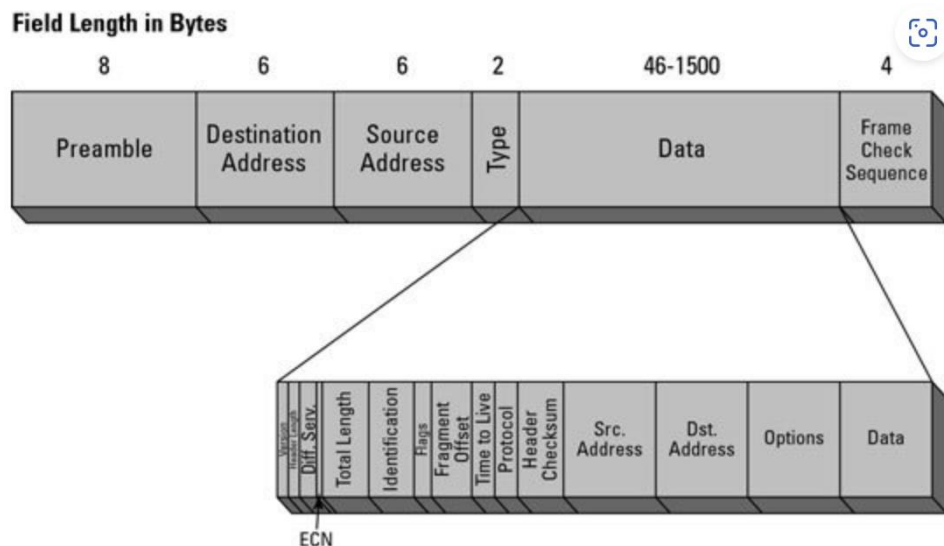
Ping google.com [142.250.196.142] 32바이트 데이터 사용 :
142.250.196.142의 응답 : 바이트=32 시간=76ms TTL=111
142.250.196.142의 응답 : 바이트=32 시간=93ms TTL=111
142.250.196.142의 응답 : 바이트=32 시간=95ms TTL=111
142.250.196.142의 응답 : 바이트=32 시간=91ms TTL=111

142.250.196.142에 대한 Ping 통계 :
    패킷 : 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
    왕복 시간(밀리초):
        최소 = 76ms, 최대 = 95ms, 평균 = 88ms
```

컴퓨터에서 간단하게 cmd를 이용해 구글에 ping(네트워크에서 다른 컴퓨터나 서버와의 연결 상태를 확인하는 명령어) 요청을 보내서 네트워크가 잘 이어지고 있는지 패킷을 보내고 받음으로써 검증한다.

이처럼 패킷은 네트워크간 주고받는 '무언가'이다.

### 패킷의 구성요소



패킷은 헤더, 페이로드, 트레일러 로 구성된다. 각 부분은 패킷 전송과 관련된 다양한 기능을 수행한다.

- 헤더(Header) :

헤더는 패킷의 시작 부분으로 네트워크에서 패킷의 올바른 전송을 위해 필수적인 역할을 한다. 헤더는 송신자와 수신자의 주소, 전송상태가 어떤지 등을 나타내는 정보를 포함한다.

헤더에 포함되는 주요 구성요소 :

송신자 ip 주소, 수신자 ip주소, 프로토콜, 패킷 순서 번호, 시간정보(TTL), 등

- 페이로드(Payload) :

페이로드는 패킷의 중간 부분으로, 패킷 내에서 실제 데이터가 담겨져 있는 부분이다. 웹 페이지 요청, 이메일 내용, 파일 데이터 등이 페이로드에 포함된다. 페이로드는 사용자가 전송하려는 핵심 데이터이다.

페이로드에 포함되는 주요 구성요소 :

웹 브라우징 이메일, 파일전송, http 요청 등

- 트레일러(Trailer) :

트레일러는 패킷의 끝 부분으로, 패킷이 정상적으로 전송 되었는지 오류 검출 및 패킷 전송의 종료 신호를 제공한다.

트레일러의 주요 구성요소

FCS(오류 검출 코드), 프레임 종료 플래그 등

정리 :

헤더 : 패킷을 전송하는 데 필요한 정보(출발지, 목적지, 프로토콜 등)

페이로드 : 실제로 전송하는 데이터(웹 페이지, 파일 내용)

트레일러 : 데이터 오류 검출 및 종료 신호

### 3. 패킷 컬러

Wireshark에서 캡처한 패킷을 보면 여러 색상으로 나오는 것을 확인할 수 있다.

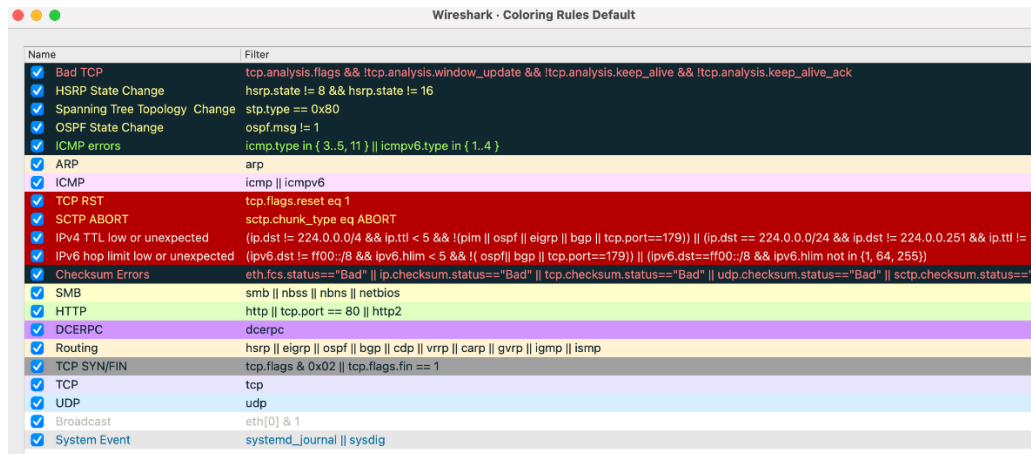
패킷마다 다른 컬러가 있는데 각 컬러는 패킷의 프로토콜을 반영한다

Ex) 모든 HTTP 트래픽은 녹색이다.

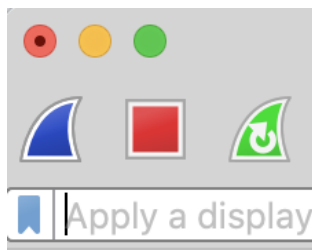
패킷 컬러 구분으로 신속하게 다양한 프로토콜을 구분할 수 있다는 장점이 있다.

자신만의 컬러링 규칙을 정의할 수 있다. (기존 컬러 규칙 변경 가능!)

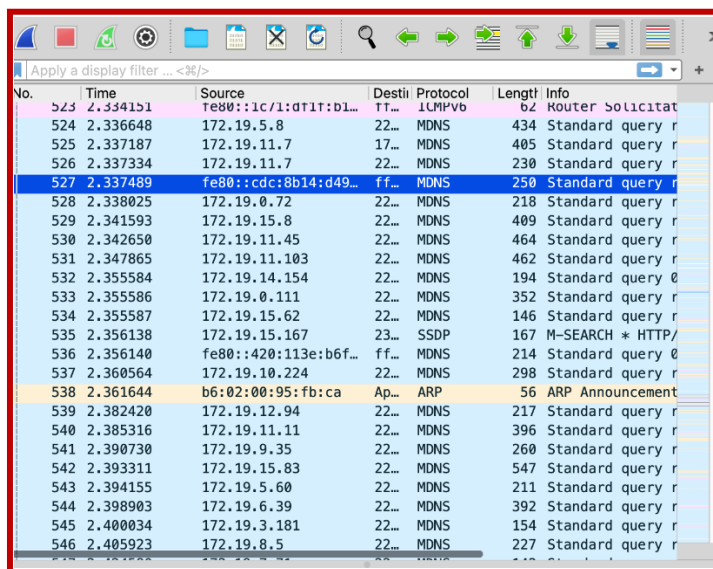
Wireshark를 실행한 후 View -> Coloring Rules를 클릭하면 현재 설정된 규칙을 확인할 수 있다.

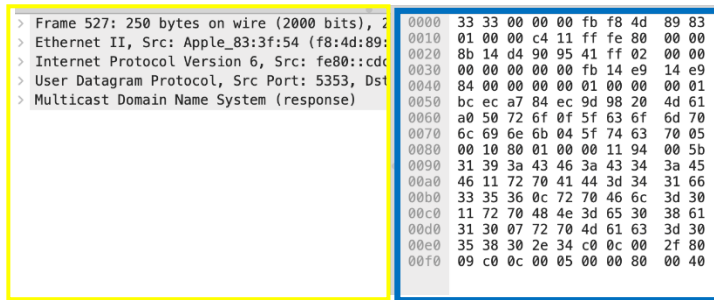


#### 4. Wireshark 인터페이스



↳ 차례대로 시작, 종료, 다시시작

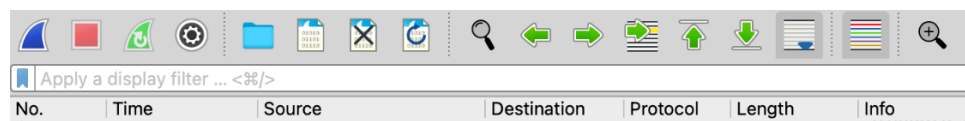




빨간색 부분 - 호스트 간에 주고받은 패킷을 나열해 놓는 부분

노란색 부분 - 빨간색 부분에서 주고받은 패킷의 내부 헤더 정보, 즉 이 패킷들을 어떤 규칙이나 전송 기준으로 주고받았는지를 보여줌

파란색 부분 - 실제 주고받은 내용을 16진수로 보여줌



No. : 패킷을 수집한 순서

Time : 패킷이 수집된 시간

Source : 패킷을 보낸 주소

Destination : 패킷 도착 주소

Protocol : 프로토콜 정보

Length : 패킷의 길이

Info : 패킷 정보

## 5. 패킷 필터링 기능

- 프로토콜 별 분리

Tcp, udp, arp 프로토콜 선별 예시

tcp							
No.	Time	Source	Destination	Protocol	Length	Info	
1265	3.824998	172.19.11.7	172.175.23...	TCP	78	60	
1321	4.022044	172.175.234.12	172.19.11.7	TCP	74	44	
1325	4.022044	172.19.11.7	172.175.23...	TCP	66	60	
1326	4.022050	172.19.11.7	172.175.23...	TLSv1.2	583	CL	
1328	4.033204	172.19.11.7	172.19.15...	TCP	78	52	
1331	4.045664	172.19.15.202	172.19.11.7	TCP	78	70	
1332	4.045807	172.19.11.7	172.19.15...	TCP	66	52	
udp							
No.	Time	Source	Destination	Protocol	Length	Info	
1396	4.312748	172.19.11.7	224.0.0.251	MDNS	339	Standard quer...	
1397	4.314420	172.19.11.7	224.0.0.251	MDNS	120	Standard quer...	
1398	4.315203	172.19.11.7	224.0.0.251	MDNS	364	Standard quer...	
1399	4.319163	172.19.11.7	224.0.0.251	MDNS	104	Standard quer...	
1400	4.321941	172.19.11.7	224.0.0.251	MDNS	259	Standard quer...	
1401	4.326127	172.19.11.7	224.0.0.251	MDNS	158	Standard quer...	
1402	4.331265	172.19.11.7	224.0.0.251	MDNS	398	Standard quer...	
1404	4.335107	172.19.11.7	224.0.0.251	MDNS	386	Standard quer...	
arp							
No.	Time	Source	Destination	Protocol	Length	Info	
1117	3.315879	Apple_83:3f...	Apple_83:3f...	ARP	56	ARP Announcem...	
1126	3.338041	62:a...	Apple_83:3f...	ARP	56	Who has 172.1...	
1144	3.380066	Apple_83:3f...	Apple_83:3f...	ARP	56	Who has 172.1...	
1151	3.448947	Juni...	Apple_83:3f...	ARP	60	Who has 172.1...	
1152	3.449164	Juni...	Apple_83:3f...	ARP	60	Who has 172.1...	
1168	3.472886	72:8...	Apple_83:3f...	ARP	56	Who has 172.1...	
1175	3.489930	Clou...	Apple_83:3f...	ARP	56	Who has 172.1...	
1199	3.549000	Inte...	Apple_83:3f...	ARP	56	Who has 169.2...	
dns							
No.	Time	Source	Destination	Protocol	Length	Info	
1257	3.814298	172.19.11.7	168.126.63.1	DNS	72	Standard quer...	
1258	3.814463	172.19.11.7	168.126.63.1	DNS	72	Standard quer...	
1260	3.818134	168.126.63.1	172.19.11.7	DNS	155	Standard quer...	
1262	3.819015	168.126.63.1	172.19.11.7	DNS	215	Standard quer...	
1263	3.820054	172.19.11.7	168.126.63.1	DNS	113	Standard quer...	

Dns : 도메인 이름을 조회하는 패킷

dhcp							
No.	Time	Source	Destination	Protocol	Length	Info	
688	1.834878	0.0.0.0	255.255.255...	DHCP	342	DHCP Request ...	
1064	3.142443	0.0.0.0	255.255.255...	DHCP	342	DHCP Discover...	
1071	3.153381	0.0.0.0	255.255.255...	DHCP	342	DHCP Discover...	
1113	3.310533	0.0.0.0	255.255.255...	DHCP	342	DHCP Request ...	
1132	3.353268	0.0.0.0	255.255.255...	DHCP	342	DHCP Discover...	
1418	4.388268	0.0.0.0	255.255.255...	DHCP	342	DHCP Request ...	

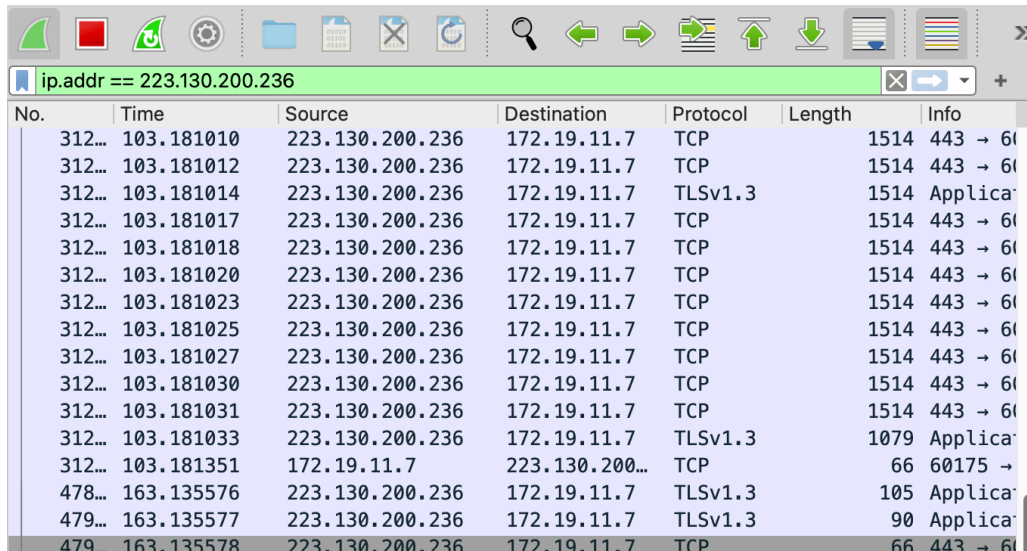
Dhcp : 동적 ip 주소 할당 요청을 위해 발생하는 패킷

- 논리 연산자 필터 검색 (ip.addr : ip 필터)

Name: www.naver.com.nheos.com  
Address: 223.130.200.236

터미널에서 nslookup [www.naver.com](http://www.naver.com) 을 치면 위와 같이 NAVER의 아이피가 뜬다

- Nslookup : DNS(Domain Name System) 조회를 위한 명령어



No.	Time	Source	Destination	Protocol	Length	Info
312...	103.181010	223.130.200.236	172.19.11.7	TCP	1514	443 → 60
312...	103.181012	223.130.200.236	172.19.11.7	TCP	1514	443 → 60
312...	103.181014	223.130.200.236	172.19.11.7	TLSv1.3	1514	Applica
312...	103.181017	223.130.200.236	172.19.11.7	TCP	1514	443 → 60
312...	103.181018	223.130.200.236	172.19.11.7	TCP	1514	443 → 60
312...	103.181020	223.130.200.236	172.19.11.7	TCP	1514	443 → 60
312...	103.181023	223.130.200.236	172.19.11.7	TCP	1514	443 → 60
312...	103.181025	223.130.200.236	172.19.11.7	TCP	1514	443 → 60
312...	103.181027	223.130.200.236	172.19.11.7	TCP	1514	443 → 60
312...	103.181030	223.130.200.236	172.19.11.7	TCP	1514	443 → 60
312...	103.181031	223.130.200.236	172.19.11.7	TCP	1514	443 → 60
312...	103.181033	223.130.200.236	172.19.11.7	TLSv1.3	1079	Applica
312...	103.181351	172.19.11.7	223.130.200...	TCP	66	60175 →
478...	163.135576	223.130.200.236	172.19.11.7	TLSv1.3	105	Applica
479...	163.135577	223.130.200.236	172.19.11.7	TLSv1.3	90	Applica
479...	163.135578	223.130.200.236	172.19.11.7	TCP	66	443 → 60

논리 연산자 중 ip 필터를 활용해 와이어샤크의 패킷 검색 기능을 활용해

ip.addr == 223.130.200.236 을 입력해 필터를 검색하면

해당 아이피(네이버주소)와 통신했던 데이터 패킷만 출력되는 것을 볼 수 있다.

## 결론

Wireshark를 이해하기 위해 기본적인 것들을 알아보는 시간을 가져보았다.

Wireshark가 패킷을 분석하는 프로그램이기 때문에 패킷이라는 단어를 먼저 알아보는 것을 우선 순위로 두었다. 또한 궁금했던 패킷 컬러에 대해서도 알아보고 추후 Wireshark 활용을 위해 간단히 wireshark의 구조에 대해 알아보고 패킷 필터링 기능을 한 두가지 실행해보기도 하였는데 아직은 이러한 패킷 검색 기능이 있다는 것만 인지하는 것 같고, 이 패킷 필터링 기능을 능숙하게 사용하고 활용하는 방법에 대해선 미숙한 것 같다.

Wireshark를 잘 사용하려면 패킷 검색을 잘 활용해야 한다는 것을 보았는데, 앞으로 패킷 검색에 대한 것을 중점으로 먼저 공부해야할 것 같다.

오늘 실습한 논리 연산자의 종류가 많았는데 다음 SSR 시간에는 다양한 논리 연산자 필터링을 우선적으로 해보고 싶다. (오늘은 중간에 파일 날라감 이슈가 있어 많은 것을 해



보지 못했습니다ㅜㅜ...)

#### 참고자료

1. JD-pro (2023. 12. 8) *패킷(Packet)이란? (쉬운 설명, 구조, 헤더, 인캡슐레이션, 핑, Ping)*. Tistory

<https://jdcyber.tistory.com/12>

2. Hongpossible (2018. 11. 16) *Wireshark란? / 설치법* Tistory

<https://hongpossible.tistory.com/entry/Wireshark%EB%9E%80-%EC%84%A4%EC%B9%98%EB%B2%95>

3. 휴롱이 공부하는 작업공간 (2024. 8. 17) *패킷의 구조와 전송 방식에 대한 이해* Velog

<https://velog.io/@alsgur/%ED%8C%A8%ED%82%B7%EC%9D%98-%EA%B5%AC%EC%A1%B0%EC%99%80-%EC%A0%84%EC%86%A1-%EB%B0%A9%EC%8B%9D%EC%97%90-%EB%8C%80%ED%95%9C-%EC%9D%B4%ED%95%B4>

4. JD-pro (2023. 12. 7) *와이어샤크(Wireshark) 사용법 #2 (쉬운 설명, 필터, 캡처, 연산자)* Tistory

<https://jdcyber.tistory.com/9>