4주차 연구일지

-Wireshark

패킷 필터링 예시

- tcp.stream eq [number]
    1. 패킷들을 tcp stream 단위로 분석 가능.
    2. Request, Response 패킷 색으로 구분 가능.

- http.request.method == [method]
    1. http 프로토콜 패킷 중 특정 메소드를 요청한 패킷 필터링.
    2. GET 메소드 : 페이지 요청이나 검색 결과등을 찾을 수 있음.
    3. POST 메소드 : 로그인이나 어떠한 정보 전달의 내용을 찾을 수 있음.

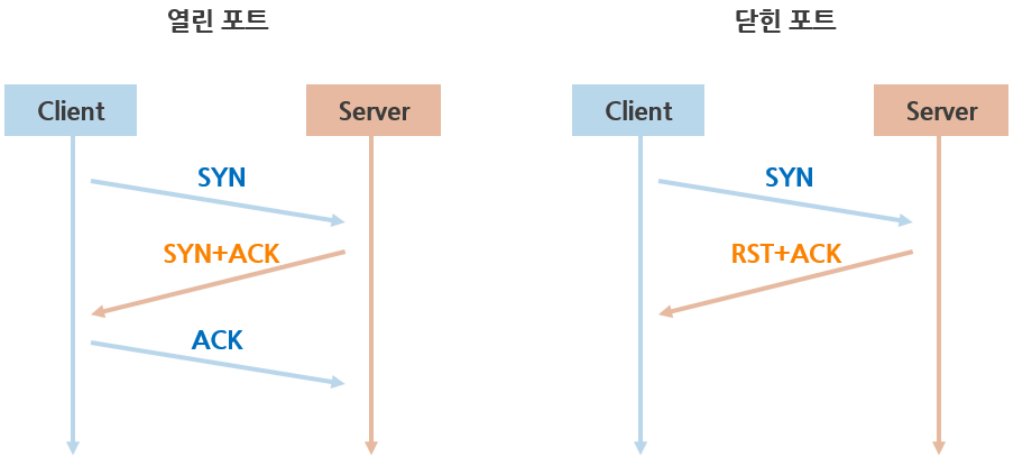
- frame {contains / matches} [string]
    1. 패킷의 문자열이 포함되어 있는 패킷만 필터링.
    2. contains : 정확하게 일치하는 것만 찾음.
    3. matches : 대소문자 구분없이 비슷한 내용을 찾고 정규표현식 사용.

- tcp.flags == [hex]
    1. TCP 패킷의 플래그별로 검색 가능
    2. flags는 각 1 bit씩 총 6 bit로 구성
    3. URG - ACK - PSH - SYN - FIN 순서로 되어있다.
    4. 만약 [SYN, ACK] 패킷 필터링 시 ob0110010(2진수) / 18(10진수) / 0x12 표현 가능 ⇒ tcp.flags == 0x12

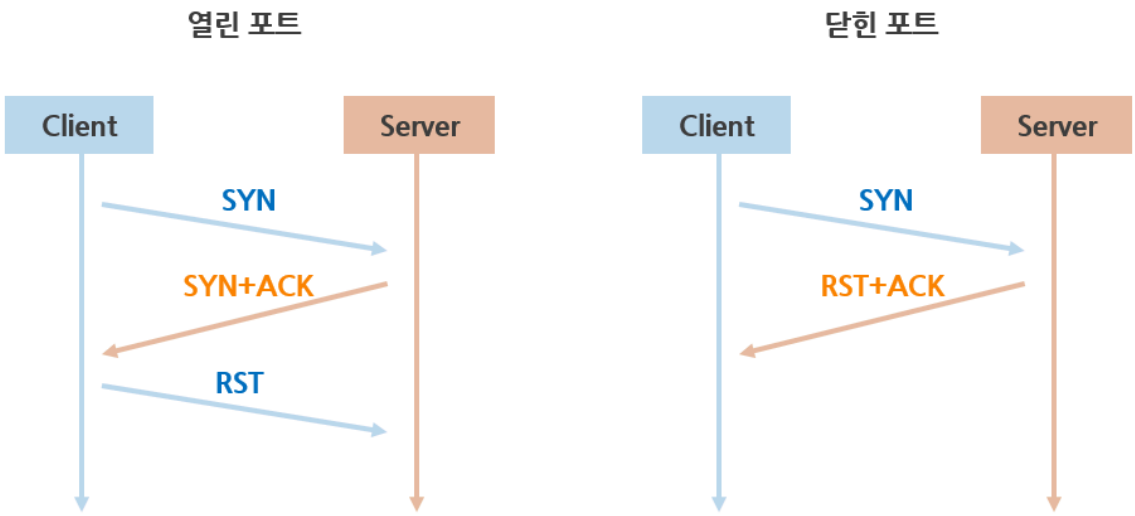
- Edit - Find packet - packet detail / Ctrl + F
    1. 패킷 내부의 문자열 쉽게 찾아줌.
    2. 단, 드롭박스에서 packet list를 선택하면 패킷의 info에서만 찾아 내부 확인을 위해서는 packet detail로 설정해줘야함.

- 포트 스캐닝

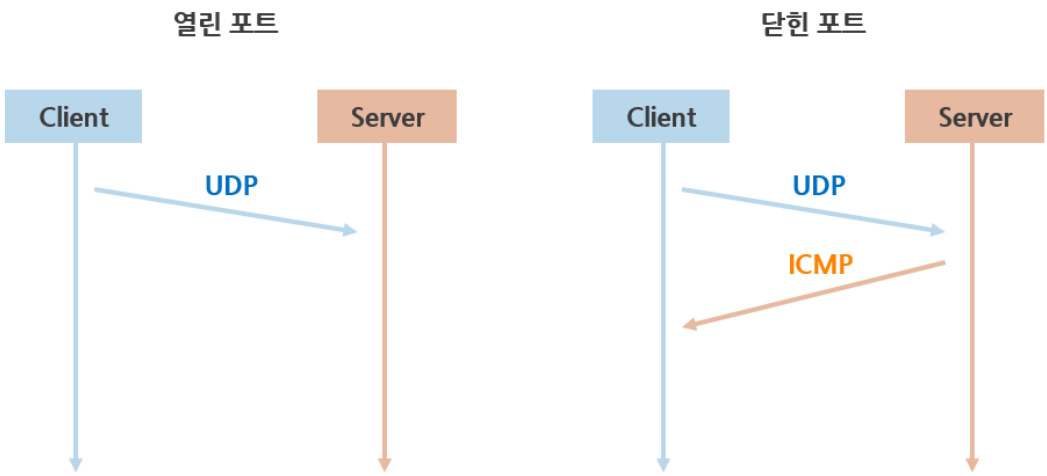
1. TCP Connect Scan(TCP Open Scan)



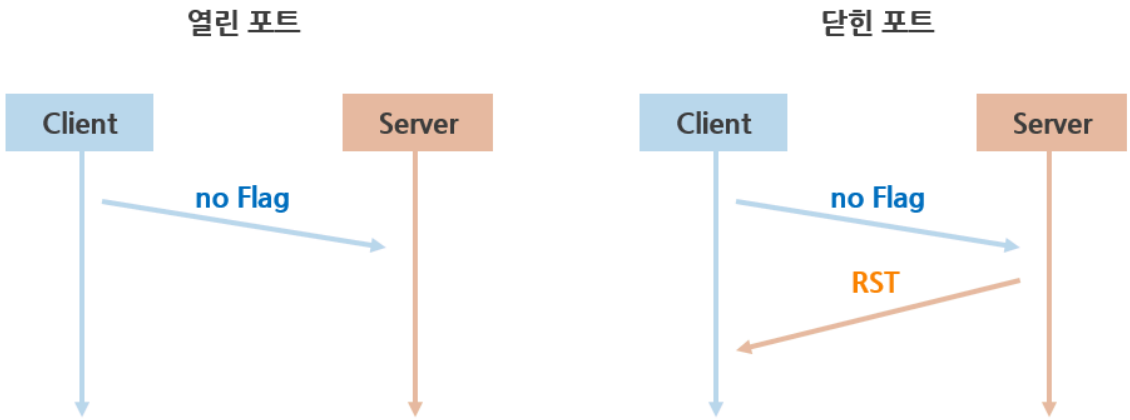
2. TCP Syn Scan(Half-Open Scan)



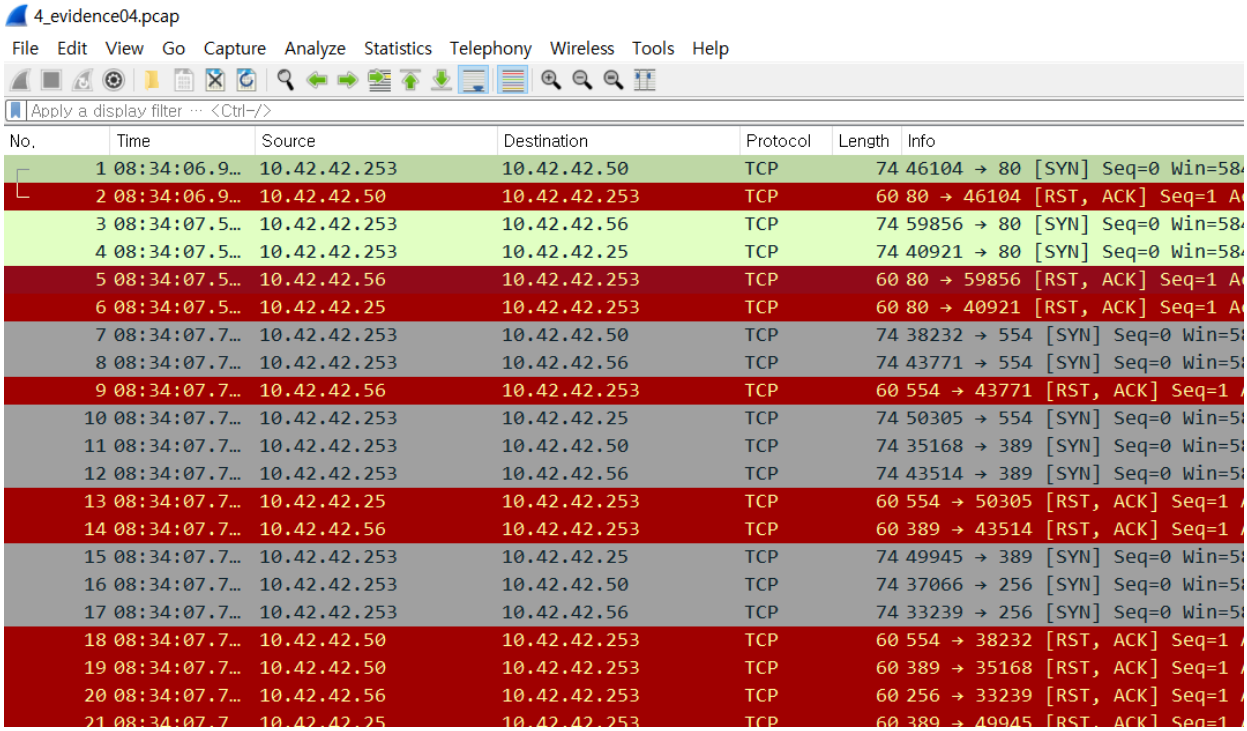
3. UDP Scan



4. TCP NULL Scan



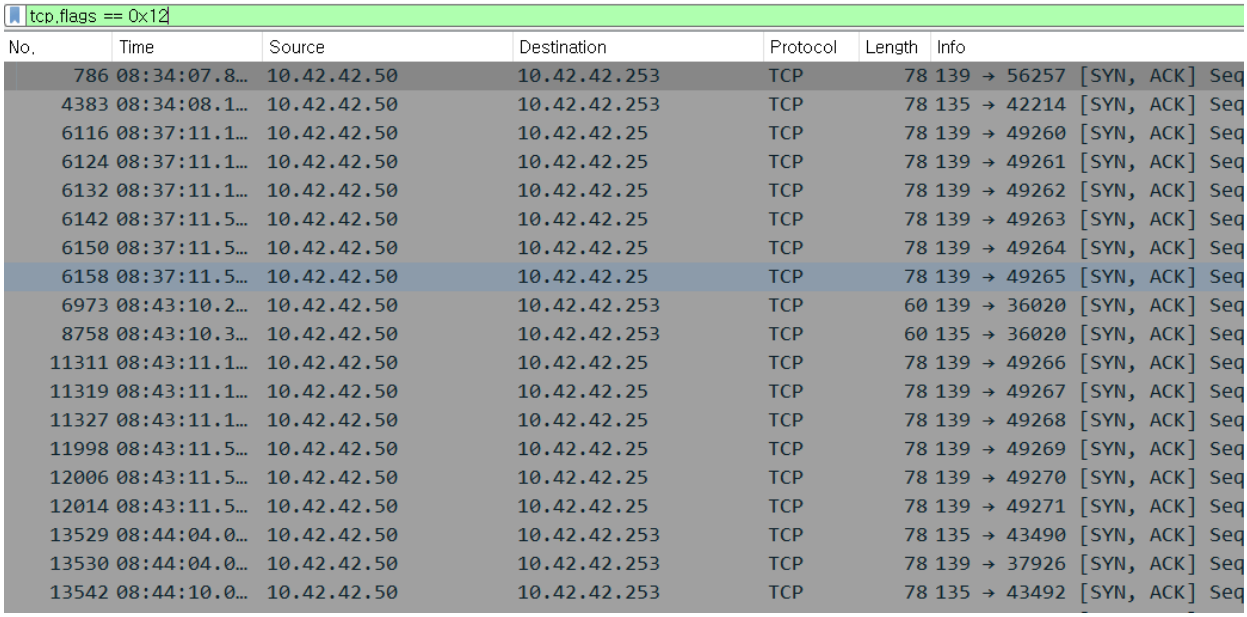
-WireShark를 통한 공격탐지 예시



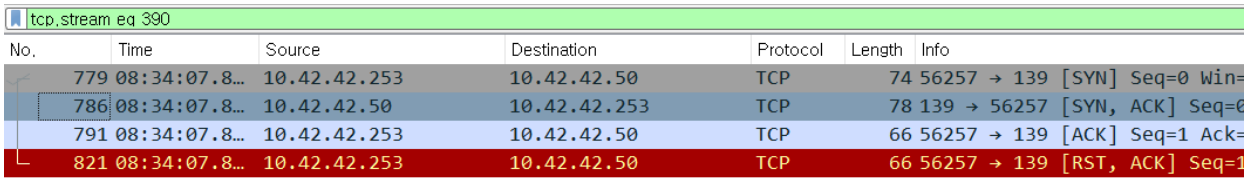
- 패킷에 다량의 [SYN] / [RST, ACK] 패킷이 보인다.
- 대량의 SYN을 보낸 10.42.42.253이 스캐너라는것으로 유추 가능.
- 포트 스캔방식이 SYN을 보내어 왔기 때문에 Open scan 또는 Half Open scan 방식 중 하나이다.

스캔방식 확인을 위한 필터링

- tcp.flags == 0x12



- 253에게 보낸 패킷 하나의 스트림을 확인



- 스캐너는 [ACK]응답을 해줬으니 TCP Connect Scan(Open Scan)으로 진행한걸 확인할 수 있다.

- 어떤 IP 포트가 열려있는지 확인을 위한 작업
- ip.dest eq 10.42.42.253 && tcp.flags eq 0x12

ip.dst eq 10.42.42.253 && tcp.flags eq 0x12

No.	Time	Source	Destination	Protocol	Length	Info
786	08:34:07.8…	10.42.42.50	10.42.42.253	TCP	78	139 → 56257 [SYN, ACK]
4383	08:34:08.1…	10.42.42.50	10.42.42.253	TCP	78	135 → 42214 [SYN, ACK]
6973	08:43:10.2…	10.42.42.50	10.42.42.253	TCP	60	139 → 36020 [SYN, ACK]
8758	08:43:10.3…	10.42.42.50	10.42.42.253	TCP	60	135 → 36020 [SYN, ACK]
13529	08:44:04.0…	10.42.42.50	10.42.42.253	TCP	78	135 → 43490 [SYN, ACK]
13530	08:44:04.0…	10.42.42.50	10.42.42.253	TCP	78	139 → 37926 [SYN, ACK]
13542	08:44:10.0…	10.42.42.50	10.42.42.253	TCP	78	135 → 43492 [SYN, ACK]
13551	08:44:10.1…	10.42.42.50	10.42.42.253	TCP	78	135 → 36119 [SYN, ACK]
13554	08:44:10.2…	10.42.42.50	10.42.42.253	TCP	78	135 → 36120 [SYN, ACK]
13557	08:44:10.3…	10.42.42.50	10.42.42.253	TCP	74	135 → 36121 [SYN, ACK]
13560	08:44:10.4…	10.42.42.50	10.42.42.253	TCP	78	135 → 36122 [SYN, ACK]
13563	08:44:10.5…	10.42.42.50	10.42.42.253	TCP	78	135 → 36123 [SYN, ACK]
13566	08:44:10.6…	10.42.42.50	10.42.42.253	TCP	74	135 → 36124 [SYN, ACK]
13591	08:44:10.7…	10.42.42.50	10.42.42.253	TCP	66	135 → 36131 [SYN, ACK]
13604	08:44:10.7…	10.42.42.50	10.42.42.253	TCP	78	135 → 36134 [SYN, ACK]

- 10.42.42.50의 135,139번 포트가 열려있다는 사실 확인.
- 135,139번 포트는 smb를 사용하는 포트로 버전에 따라 취약점이 존재

- 타겟에 대한 정보 및 OS 유추

No.	Time	Source	Destination	Protocol	Length	Info
13567	08:44:10.6…	10.42.42.253	10.42.42.50	TCP	60	36124 → 135 [RST] Seq=1 Win=0 Len=0
13568	08:44:10.6…	10.42.42.253	10.42.42.56	ICMP	162	Echo (ping) request id=0x9c52, seq=295/9985, ttl=45 (reply in 13570)
13569	08:44:10.6…	10.42.42.253	10.42.42.25	ICMP	162	Echo (ping) request id=0x9c52, seq=295/9985, ttl=48 (reply in 13571)
13570	08:44:10.6…	10.42.42.56	10.42.42.253	ICMP	162	Echo (ping) reply id=0x9c52, seq=295/9985, ttl=64 (request in 13568)
13571	08:44:10.6…	10.42.42.25	10.42.42.253	ICMP	162	Echo (ping) reply id=0x9c52, seq=295/9985, ttl=64 (request in 13569)
13572	08:44:10.6…	10.42.42.253	10.42.42.50	ICMP	162	Echo (ping) request id=0x9c52, seq=295/9985, ttl=46 (no response found!)
13573	08:44:10.6…	10.42.42.50	10.42.42.253	ICMP	162	Echo (ping) reply id=0x9c52, seq=295/9985, ttl=128
13574	08:44:10.6…	10.42.42.253	10.42.42.56	ICMP	192	Echo (ping) request id=0x9c53, seq=296/10241, ttl=45 (reply in 13576)
13575	08:44:10.6…	10.42.42.253	10.42.42.25	ICMP	192	Echo (ping) request id=0x9c53, seq=296/10241, ttl=37 (reply in 13577)
13576	08:44:10.6…	10.42.42.56	10.42.42.253	ICMP	192	Echo (ping) reply id=0x9c53, seq=296/10241, ttl=64 (request in 13574)
13577	08:44:10.6…	10.42.42.25	10.42.42.253	ICMP	192	Echo (ping) reply id=0x9c53, seq=296/10241, ttl=64 (request in 13575)
13578	08:44:10.6…	10.42.42.253	10.42.42.50	ICMP	192	Echo (ping) request id=0x9c53, seq=296/10241, ttl=49 (reply in 13579)
13579	08:44:10.6…	10.42.42.50	10.42.42.253	ICMP	192	Echo (ping) reply id=0x9c53, seq=296/10241, ttl=128 (request in 13578)

- ICMP Ping scan을 진행한 패킷들이다.
- TTL값을 통해 OS 정보를 유추 할 수 있는데 Window의 TTL 초기값은 128, Linux 계열은 64로 설정된다.
- 또한 Apple의 MAC OS는 서버의 MAC 주소로 알 수 있는음

```
> Frame 13569: 162 bytes on wire (1296 bits), 162 bytes captured (1296 bits)
v Ethernet II, Src: QuantaCo_82:1f:4a (00:23:8b:82:1f:4a), Dst: Apple_92:6e:dc (00:16:cb:92:6e:dc)
    > Destination: Apple_92:6e:dc (00:16:cb:92:6e:dc)
    > Source: QuantaCo_82:1f:4a (00:23:8b:82:1f:4a)
      Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 10.42.42.253, Dst: 10.42.42.25
> Internet Control Message Protocol
```

- 따라서 10.42.42.253의 ip를 이용하여 TCP Connect Scan 및 Ping Scan을 진행 하였고 10.42.42 네트워크에서 25,50,56 호스트를 발견.
- 25의 OS는 MAC, 50의 OS는 Window인 것과 135,139 포트가 열려있다는 것을 탐지했다는걸 확인할 수 있었다.

-요약

1. Wireshark를 통해 포트 스캐닝 공격 기법이 무엇인지, 패킷을 통해 진행 방법과 탐지하는 법을 알아 보았다.
2. 또한 인터넷에 돌아다니는 많은 패킷 중 암호화되지 않은 패킷들에 대한 정보를 패킷분석 도구를 통해 쉽게 알아낼 수 있어 암호화되지 않은 패킷의 취약성까지 확인할 수 있었다.

-참고자료
1.무작정 보안 공부 : https://taesam.tistory.com/25