==========

[PCAP 패킷 분석 보고서]

과목명: 보안 네트워크 프로그래밍

팀명: 타조

팀원: 이동훈, 정유진, 추민호

PCAP 라이브러리 패킷 분석

==========

[목차]

---

1. PCAP 라이브러리 개요

- PCAP 기본 기능 설명:
PCAP(Packet Capture) 라이브러리를 활용하여 네트워크 인터페이스에서 전달되는 패킷을 실시간으로 수집
Ethernet /IP /TCP / UDP 헤더를 직접 파싱하여 핵심 정보를 추출하는 C 프로그램

wireshark, tcpdump 같은 패킷 분석 도구의 엔진 역할 ====> PCAP
리눅스: libpcap
윈도우: WinPcap /NPcap

PCAP 라이브러리는 운영체제가 제공하는 네트워크 인터페이스로부터 전달되는 패킷을 사용자 공간에서 캡처하고 분석할 수 있도록 해주는 라이브러리이다. 응용 프로그램은 PCAP을 통해 네트워크 카드에서 송수신되는 전송 프레임을 직접 받아 볼 수 있고, 패킷 분석 도구, 침입 탐지 시스템 등을 구현할 수 있다. PCAP은 네트워크 장비 목록 조회, 특정 인터페이스를 캡처 모드로 여는 기능, 캡처할 패킷을 선택하기 위한 필터 설정, 그리고 캡처된 패킷에 대해 콜백 함수를 호출하는 기능 등을 제공한다.

2. 캡쳐 과정 및 설정

1) Npcap 다운로드

2) 컴파일 후 실행 화면

```
PS C:\Users\samsung\pcap-win> ./pcap-win.exe
192.168.216.1:5353 to 224.0.0.251 ports 5353, --------, TTL 1 @16:56:01
192.168.216.1:5353 to 224.0.0.251 ports 5353, --------, TTL 1 @16:56:01
192.168.216.1:5353 to 224.0.0.251 ports 5353, --------, TTL 1 @16:56:01
192.168.216.1:5353 to 224.0.0.251 ports 5353, --------, TTL 1 @16:56:01
192.168.216.1:5353 to 224.0.0.251 ports 5353, --------, TTL 1 @16:56:02
192.168.216.1:5353 to 224.0.0.251 ports 5353, --------, TTL 1 @16:56:02
192.168.216.1:63828 to 224.0.0.252 ports 5355, --------, TTL 1 @16:57:34
192.168.216.1:63828 to 224.0.0.252 ports 5355, --------, TTL 1 @16:57:34
192.168.216.1:5353 to 224.0.0.251 ports 5353, --------, TTL 1 @16:58:02
192.168.216.1:5353 to 224.0.0.251 ports 5353, --------, TTL 1 @16:58:02
192.168.216.1:5353 to 224.0.0.251 ports 5353, --------, TTL 1 @16:58:02
192.168.216.1:5353 to 224.0.0.251 ports 5353, --------, TTL 1 @16:58:02
192.168.216.1:5353 to 224.0.0.251 ports 5353, --------, TTL 1 @16:58:02
192.168.216.1:5353 to 224.0.0.251 ports 5353, --------, TTL 1 @16:58:02
```

	A	B	C	D	E	F	G	H	I
1	timestamp	source	destinatio	ports	tcp_flags	tos	ttl	time_str	
2	1.77E+09	192.168.21	224.0.0.251	5353	--------	0	1	16:56:01	
3	1.77E+09	192.168.21	224.0.0.251	5353	--------	0	1	16:56:01	
4	1.77E+09	192.168.21	224.0.0.251	5353	--------	0	1	16:56:01	
5	1.77E+09	192.168.21	224.0.0.251	5353	--------	0	1	16:56:01	
6	1.77E+09	192.168.21	224.0.0.251	5353	--------	0	1	16:56:02	
7	1.77E+09	192.168.21	224.0.0.251	5353	--------	0	1	16:56:02	
8	1.77E+09	192.168.21	224.0.0.252	5355	--------	0	1	16:57:34	
9	1.77E+09	192.168.21	224.0.0.252	5355	--------	0	1	16:57:34	
10	1.77E+09	192.168.21	224.0.0.251	5353	--------	0	1	16:58:02	

그림 CSV 파일로 저장

패킷이 제대로 캡처가 되지 않아 캡처된 패킷을 받아서 분석

Column(구성):

source: 출발지 IP : port
destination: 목적지 IP
ports: 목적지(=destination port)
tcp_flags: TCP 플래그 if ------일 경우 대부분 UDP
Tos: 서비스 타입(1byte)
TTL: Time to Live 패킷이 통과 가능한 최대 라우터 수
Time_str: 시간(패킷이 캡처된 시각)

source IP :port 목적지 IP 목적지 port Tcp_flags TTL Time_str

```
192.168.216.1:5353 to 224.0.0.251 ports 5353, --------, TTL 1 @17:00:02
192.168.216.1:5353 to 224.0.0.251 ports 5353, --------, TTL 1 @17:00:02
```

그림 1 패킷 구성 요소

3. 패킷 분석 결과

1764922030	10.30.88.13:54942	13.107.136.10	443	------S-	0	128	17:07:10
1764922030	10.30.88.13:54943	13.107.136.10	443	------S-	0	128	17:07:10
1764922030	13.107.136.10:443	10.30.88.13	54943	---A--S-	0	111	17:07:10
1764922030	13.107.136.10:443	10.30.88.13	54942	---A--S-	0	111	17:07:10
1764922030	10.30.88.13:54943	13.107.136.10	443	---A----	0	128	17:07:10
1764922030	10.30.88.13:54942	13.107.136.10	443	---A----	0	128	17:07:10
1764922030	10.30.88.13:54943	13.107.136.10	443	---AP---	0	128	17:07:10
1764922030	10.30.88.13:54942	13.107.136.10	443	---AP---	0	128	17:07:10
1764922030	13.107.136.10:443	10.30.88.13	54943	---A----	0	63	17:07:10
1764922030	13.107.136.10:443	10.30.88.13	54942	---A----	0	63	17:07:10
1764922030	10.30.88.13:51604	224.0.0.252	5355	--------	0	1	17:07:10
1764922030	13.107.136.10:443	10.30.88.13	54943	---A----	0	112	17:07:10
1764922030	13.107.136.10:443	10.30.88.13	54943	---A----	0	112	17:07:10
1764922030	13.107.136.10:443	10.30.88.13	54943	---A----	0	112	17:07:10

그림 2 PACKET1 file의 패킷 일부

3번째 패킷을 보면

1764922030	13.107.136.10:443	10.30.88.13	54943	---A--S-	0	111	17:07:10

13.107.136.10:443에서 10.30.88.13:54943로 도착했음을 의미한다.

--> 포트 번호에 따라 서비스가 아래 표에 따른 내용이 존재한다
flag가 없을 경우 대부분 UDP이며 만약 문자가 존재한다면 TCP 통신을 의미한다.

포트번호에 따른 서비스

포트번호	protocol	서비스	의미
53	UDP/TCP	DNS	domain 요청 및 응답
443	TCP	HTTPS	암호화 통신 웹사이트
80	TCP	HTTP	웹사이트
22	TCP	SSH	서버 원격 접속
67/68	UDP	DHCP	IP주소 자동 할당

TCP_Flags의 의미

FLAG	문자	의미	사용 예시
SYN	S	연결 시작	TCP 핸드셰이크
ACK	A	패킷 확인 응답	대부분의 패킷
FIN	F	연결 종료 요청	종료
RST	R	연결 강제 종료	비정상
PSH	P	바로 전달	실제 데이터 전송
URG	U	긴급 데이터	urgent 데이터
ECE	E	혼잡 알림	X
CWR	C	혼잡 제어 관련	X

8자리 비트 자리에서 왼쪽부터 CEUAPRSF를 의미 즉 작동하는 것이 무엇인지에 따라 TCP_flags에 뜨는 알파벳이 위치에 맞게 출력된다.

case1) => TCP 핸드쉐이크

1764922030	10.30.88.13:54942	13.107.136.10	443	------S-	0	128	17:07:10
1764922030	10.30.88.13:54943	13.107.136.10	443	------S-	0	128	17:07:10
1764922030	13.107.136.10:443	10.30.88.13	54943	---A--S-	0	111	17:07:10
1764922030	13.107.136.10:443	10.30.88.13	54942	---A--S-	0	111	17:07:10
1764922030	10.30.88.13:54943	13.107.136.10	443	---A----	0	128	17:07:10
1764922030	10.30.88.13:54942	13.107.136.10	443	---A----	0	128	17:07:10

1,2번 패킷

10.30.88.13:54942 -> 13.107.136.10:443 , tcp_flags = ------S- 이므로
클라이언트가 서버에게 SYN 수행
10.30.88.13:54943 -> 13.107.136.10:443 , tcp_flags = ------S- 이므로
클라이언트가 서버에게 SYN 수행

3,4번 패킷

13.107.136.10:443 -> 10.30.88.13:54942와 10.30.88.13:54943 , tcp_flags = ---A--S-이므로 서버가 클라이언트에게 ACK+SYN 수행

5,6번 패킷

10.30.88.13:54943 -> 13.107.136.10:443, tcp_flags = ---A----
10.30.88.13:54942 -> 13.107.136.10:443, tcp_flags = ---A---- 이므로 클라이언트가 서버에게 ACK 수행

이 패킷 과정이 **TCP handshake** 과정임을 보여주고 있다.

case2) => 데이터 송수신

1764922030	10.30.88.13:54943	13.107.136.10	443	---AP---	0	128	17:07:10
1764922030	10.30.88.13:54942	13.107.136.10	443	---AP---	0	128	17:07:10
1764922030	13.107.136.10:443	10.30.88.13	54943	---A----	0	63	17:07:10
1764922030	13.107.136.10:443	10.30.88.13	54942	---A----	0	63	17:07:10

1,2번 패킷
10.30.88.13:54943 -> 13.107.136.10:443 tcp_flags = ---AP---
10.30.88.13:54943 -> 13.107.136.10:442 tcp_flags = ---AP--- 이므로 ACK+PSH 즉 데이터가 들어 있는 패킷이다. 실제로 TLS handshake 메시지나 HTTP 요청 같은게 실려 있을 수도 있다.

3,4번째 패킷
10.30.88.13:54943 -> 13.107.136.10:443, tcp_flags = ---A----
10.30.88.13:54942 -> 13.107.136.10:443, tcp_flags = ---A----
순수 ACK 패킷으로 데이터는 없지만 전에 준 데이터에 대한 수신 확인을 나타낸다.

case3) => DNS 요청

1764932481	10.30.88.13:50258	164.124.101.2	53	--------	0	128	20:01:21

내 컴퓨터 즉 10.30.88.13:50258 -> 164.124.101.2:53 로 DNS 서버로 도메인 이름를 문의하는 요청 패킷이다. 이 패킷은 웹 접속 등 애플리케이션 동작을 위해 도메인 이름을 IP 주소로 변환하는 과정으로 해석되며 출발지 포트는 대부분 5만 번대의 에페메럴 포트였고 목적지 port는 반드시 53이여야한다.

case4) => DHCP (IP 자동 할당 요청)

1764932483	0.0.0.0:68	255.255.255.255	67	--------	10	64	20:01:23
1764932493	10.30.90.1:68	255.255.255.255	67	--------	0	128	20:01:33

1번 패킷을 보면 0.0.0.0:68 -> 255.255.255.255: 67로 IP 없는 client가 브로드 캐스트로 IP 주소를 달라 요청하는 패킷

2번 패킷은 다른 장비가 DHCP 요청을 보내는 패킷

case5) => UDP/TCP 통신

1764932497	10.30.88.13:55710	211.243.0.167	56934	--------	0	128	20:01:37
1764932497	10.30.88.13:55715	74.125.247.128	3478	--------	0	128	20:01:37
1764932497	10.222.92.95:49795	255.255.255.255	21478	--------	0	128	20:01:37
1764932497	10.30.88.13:55709	10.15.0.30	21064	--------	0	128	20:01:37
1764932497	10.30.88.13:55710	10.15.0.30	21064	--------	0	128	20:01:37
1764932498	10.30.88.13:55715	74.125.247.128	3478	--------	0	128	20:01:38

이 같은 패킷이 700개 넘게 반복되는데 플래그가 --------이면 대부분 UDP 통신이다. 특히 패킷 분석 결과 211.243.0.167:56934의 UDP 트래픽이 74.125.247.128:3478의 TCP 트래픽이 많이 관찰되었는데 이는 음성/영상 통신이나 업에디트 등 장기간 데이터를 전송하는 애플리케이션 패킷으로 생각하고 있다.

case6) => 비정상 종료

1764932494	10.30.88.13:51821	142.250.197.174	443	---A-R--	0	128	20:01:34

그림과 같이 flag가 ---A-R--인 경우 연결 **강제 종료**로 해당 세션을 즉시 끊어버린다.

case7) => 연결 종료 단계

1764932492	23.203.133.172:80	10.30.88.13	62483	=---A---F	0	49	20:01:32
1764932492	10.30.88.13:62483	23.203.133.172	80	---A----	0	128	20:01:32
1764932492	52.107.254.217:443	10.30.88.13	62484	---AP---	0	112	20:01:32
1764932492	10.30.88.13:62483	23.203.133.172	80	=---A---F	0	128	20:01:32
1764932492	23.203.133.172:80	10.30.88.13	62483	---A----	0	63	20:01:32

1번 패킷에서 23.203.133.172:80 -> 10.30.88.13:62483 tcp_flags: ---A---F로 서버가 클라이언트한테 FIN + ACK 연결 종료 요청한다.
2번 패킷에서 10.30.88.13:62483 -> 23.203.133.172:80 로 flags를 A를 보내면서 FIN으로 연결 종료 요청을 수락한다.
4번 패킷에서 클라이언트도 서버에게 ---A---F로 클라이언트도 연결 종료 요청을 보내면서
5번 패킷에서 서버도 이에 A flags를 보내면서 연결 종료 요청을 수락한다.

2. 결론

이번 프로젝트에서 PCAP 라이브러리를 이용하여 실제 네트워크 환경에서 흐르는 패킷을 캡처하고 CSV 형태로 저장된 데이터를 기반으로 통신 과정을 분석하였으며, 이를 기반으로 보안 관제, 침입 탐지, 트래픽 모니터링 등에서 기초를 마련할 수 있었다. Wireshark가 자동으로 해석해 주는 결과를 포트번호와 TCP 플래그, TTL 값으로 이해하는 과정이 큰 도움이 되었다. 추후 SYN Flooding 공격 등을 탐지에 기반이 될 것이다.