TCPDUMP 기본 사용법 #01

TCPDUMP 사용법 및 예제

TCPDUMP 란?

명령줄 인터페이스에서 실행하는 네트워크 인터페이스의 패킷을 가로채는 (스니핑) 프로그램이다
TCP/IP 뿐만이 아닌 다른 여러 패킷의 정보를 출력할 수 있다.

기본 사용법

패킷을 수집하여 파일로 저장한 후 Wireshark 등의 패킷 분석 프로그램을 사용한다.
수많은 부가 옵션이 있으나 주로 쓰이는 몇 가지 옵션만 기술하였다.

패킷 수집 패턴 커맨드 (Basic / Expression / Header Flag)

Basic (필수 옵션)

기본적인 옵션값

-i : 네트워크 인터페이스가 여러 개일 경우 지정할 수 있다
-c : 스니핑 할 패킷의 개수를 정할 수 있다.
-v : 패킷 헤더의 정보를 조금 더 자세하게 보여준다 -vv 하나 더 붙이면 더욱 자세해진다
-n : 알려진 도메인 문자열을 IP 로 보이게 해준다. -nn 하나 더 붙이면 Port 도 숫자로 표시한다.
-w : 스니핑 한 패킷을 .pcap 타입 바이너리로 저장한다. -r 옵션으로 TXT 타입으로 로드할 수 있다.

 # 특정 인터페이스(-i) 의 모든 패킷을 100개만(-c) .pcap 파일 형식으로 저장(-w) 하기
해당 형식은 txt 가 아닌 Binary 타입이며 -r 옵션으로 읽거나 wireshark 에서 로드할 수 있다.

```
tcpdump  -i ens33 -vv -c 100 -w ttt.pcap
tcpdump: listening on ens33, link-type EN10MB (Ethernet), capture size 262144 bytes
100 packets captured
100 packets received by filter
```

```
0 packets dropped by kernel
```

 # 특정 인터페이스(-i) 의 모든 패킷을 5개만(-c) DNS/PORT 번역하지 않고 IP 그대로 출력(-nn) 한다.

```
tcpdump  -i br0 -n -c 5
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on br0, link-type EN10MB (Ethernet), capture size 262144 bytes
14:08:14.531606 ARP, Request who-has 8.8.8.8 tell 10.10.10.200, length 28
14:08:15.533556 ARP, Request who-has 8.8.8.8 tell 10.10.10.200, length 28
14:08:16.535615 ARP, Request who-has 8.8.8.8 tell 10.10.10.200, length 28
14:08:18.531679 ARP, Request who-has 8.8.8.8 tell 10.10.10.200, length 28
14:08:19.533632 ARP, Request who-has 8.8.8.8 tell 10.10.10.200, length 28
5 packets captured
5 packets received by filter
0 packets dropped by kernel
```

 # 더 자세한 패킷 정보를 출력 (-v , -vv) 한다.

```
tcpdump  -i br0 -n -vv -c 5
tcpdump: listening on br0, link-type EN10MB (Ethernet), capture size 262144 bytes
14:10:46.548613 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 8.8.8.8 tell 10.10.10.200, length 28
14:10:47.549640 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 8.8.8.8 tell 10.10.10.200, length 28
14:10:48.551557 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 8.8.8.8 tell 10.10.10.200, length 28
14:10:50.548614 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 8.8.8.8 tell 10.10.10.200, length 28
14:10:51.549621 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 8.8.8.8 tell 10.10.10.200, length 28
5 packets captured
5 packets received by filter
0 packets dropped by kernel
```

 # 특정 인터페이스에서(-i) 특정 IP 가 포함된 패킷(-nn, grep) 을 보여준다
tcpdump -i ens33 -nn | grep "1.2.3.4"
```
tcpdump: verbose output suppressed, use -v or -vv for full
protocol decode
listening on ens33, link-type EN10MB (Ethernet), capture size
262144 bytes
5 packets captured
6 packets received by filter
0 packets dropped by kernel
14:28:31.524052 IP 1.2.3.4.22 > 5.6.7.8.53707: Flags [P.], seq
4138122029:4138122237, ack 1068364849, win 1432, length 208
14:28:31.525940 IP 5.6.7.8.53707 > 1.2.3.4.22: Flags [.], ack
208, win 509, length 0
```

Expression (조건식)

조건에 부합하는 패킷 정보만을 보여준다. 조건의 가장 큰 구분은 다음과 같다. 조합할 수 있다.

1) Type (패킷의 종류 조건)
- host , net , port

2) Dir (패킷의 방향 조건)
- src, dst , src or dst, src and dst

3) Proto (패킷의 프토토콜 타입 조건)
- ether, fddi, ip, arp, rarp, decnet, lat, sca, moprc, mopdl, tcp, udp

4) 그 외
- 산술 연산자 ('and(&&)', 'or(||)', 'not(!)')
- gateway, broadcast, less, greater

 # 특정 인터페이스(-i) 에서 tcp 패킷이며 Destination Port 가 22인 패킷을 IP 로(-nn) 출력하기
tcpdump -i ens33 tcp dst port 22 -nn
```
tcpdump: verbose output suppressed, use -v or -vv for full
protocol decode
listening on ens33, link-type EN10MB (Ethernet), capture size
262144 bytes
16:50:25.964928 IP 211.115.223.215.53707 > 211.239.150.48.22:
Flags [.], ack 4194846013, win 511, length 0
```

```
16:50:26.016014 IP 211.115.223.215.53707 > 211.239.150.48.22:
Flags [.], ack 161, win 511, length 0
16:50:26.060903 IP 211.115.223.215.53707 > 211.239.150.48.22:
Flags [.], ack 321, win 510, length 0
```

 # 특정 인터페이스(-i) 에서 tcp 패킷"이며" Destination Port 가 80 "이거나" 443 인 패킷을 IP 로(-nn) 출력하기

```
tcpdump -i ens33 -nn tcp and dst port 80 or dst port 443
16:59:12.840778 IP 211.239.150.48.49514 > 8.8.8.8.443: Flags
[F.], seq 5, ack 9, win 229, options [nop,nop,TS val
3924566459 ecr 3515313672], length 0
16:59:14.863982 IP 211.239.150.48.47286 > 8.8.8.8.80: Flags
[S], seq 873533324, win 29200, options [mss 1460,sackOK,TS val
3924568482 ecr 0,nop,wscale 7], length 0
16:59:15.865581 IP 211.239.150.48.47286 > 8.8.8.8.80: Flags
[S], seq 873533324, win 29200, options [mss 1460,sackOK,TS val
3924569484 ecr 0,nop,wscale 7], length 0
```

Header Flag

패킷 해더의 정의된 플래그 타입을 조건으로 한 결과값을 출력할 수 있다.

 SYN : Client 가 데이터를 보내기 전 Server 에 악수(동기)를 요청하는 가장 첫 단계
이 플래그를 통해 IN/OUT Port 통신을 체크할 수 있어서 자주 활용된다.

TCP Header

Interesting Parts	TCP header
tcp[0:2]	source port
tcp[2:2]	destination port
tcp[4:4]	sequence number
tcp[8:4]	acknowledgement number
tcp[12]	header length
tcp[13]	tcp flags
tcp[14:2]	window size
tcp[16:2]	checksum
tcp[18:2]	urgent pointer
tcp[20..60]	options or data

UDP Header

Interesting Parts	UDP Header
udp[0:2]	source port
udp[2:2]	destination port
udp[4:2]	datagram length
udp[6:2]	UDP checksum

Flags

Flags	Numerically	Meaning
---- --S- 0000 0010	0x02	normal syn
---A --S- 0001 0010	0x12	normal syn-ack
---A ---- 0001 0000	0x10	normal ack
--UA P--- 0011 1000	0x38	psh-urg-ack.
---A -R-- 0001 0100	0x14	rst-ack
---- --SF 0000 0011	0x03	syn-fin scan
--U- P--F 0010 1001	0x29	urg-psh-fin
-Y-- ---- 0100 0000	0x40	anything >= 0x40
XY-- ---- 1100 0000	0xC0	both reserved bits set
XYUA PRSF 1111 1111	0xFF	FULL_XMAS scan

 # SYN 패킷만 수집 하기

= TCP Flag(tcp[13]) 중에 SYN (0x02) 인 IP (-nn) 패킷 10개를(-c) txt 타입 result.log 로 저장(>) 하기

```
tcpdump -i ens33 -c 10 "tcp[13] == 0x02" -n -nn > result.log
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), capture size 262144 bytes
10 packets captured
10 packets received by filter
0 packets dropped by kernel
```

 # cat resule.log

```
16:16:15.308872 IP 211.239.150.48.43192 > 61.100.13.230.10051: Flags [S], seq 3596242163, win 29200, options [mss 1460,sackOK,TS val 3921988927 ecr 0,nop,wscale 7], length 0
16:16:17.048128 IP 121.40.192.14.41265 > 211.239.150.48.4244: Flags [S], seq 3887030721, win 1024, length 0
```

[S] = SYN Flag
수집한 부분을 확인할 수 있으며
json 정렬이나, 조건식을 통해 다듬을 수도 있다.

```
Unskilled Attackers Pester Real Security Folks
==============================================
                      TCPDUMP FLAGS
Unskilled =  URG  =  (Not Displayed in Flag Field, Displayed elsewhere)
Attackers =  ACK  =  (Not Displayed in Flag Field, Displayed elsewhere)
Pester    =  PSH  =  [P] (Push Data)
Real      =  RST  =  [R] (Reset Connection)
Security  =  SYN  =  [S] (Start Connection)
Folks     =  FIN  =  [F] (Finish Connection)
          SYN-ACK =  [S.] (SynAcK Packet)
                     [.] (No Flag Set)
```

 # JSON 을 사용한 덤프파일 정렬 예시

result.log
서버 IP: 211.239.150.48

JSON:

```
[ { ipaddr1: '221.239.150.48',
    ipaddr2: '61.100.13.230',
    type: 'Outbound',
    port: '10051' },
  { ipaddr1: '121.40.192.14',
    ipaddr2: '221.239.150.48',
    type: 'Inbound',
    port: '4244' } ]
```

```
Table:
ipaddr1         ipaddr2         type      port
--------------  --------------  --------  -----
221.239.150.48  61.100.13.230   Outbound  10051
121.40.192.14   211.239.150.48  Inbound   4244
```

참고 자료

TCP 3 way handshark : https://websecurity.tistory.com/93