DASAN Networks, Inc.
6F Fine Venture Building 345-1 Yatap-1Dong
Bundang-gu, Seongnam, Gyeonggi-do, Korea
TEL) 82-221-725-9500
http://www.dasannetworks.com

Document System of DASAN Networks

Tcpdump 사용설명서

Written by TS1 Team
In Jun 2005

Tcpdump란?

Tcpdump는 주어진 조건 식을 만족하는 네트워크 인터페이스를 거치는 패킷들의 헤더들을 출력해 주는 프로그램이다.

Tcpdump의 옵션들

- -a : Network & Broadcast 주소들을 이름들로 바꾼다.
- -c Number : 제시된 수의 패킷을 받은 후 종료한다.
- -d : comile된 packet-matching code를 사람이 읽을 수 있도록 바꾸어 표준 출력으로 출력하고, 종료한다.
- -dd : packet-matching code를 C program의 일부로 출력한다.
- -ddd : packet-matching code를 숫자로 출력한다.
- -e : 출력되는 각각의 행에 대해서 link-level 헤더를 출력한다.
- -f : 외부의 internet address를 가급적 심볼로 출력한다(Sun의 yp server와의 사용은 가급적 피하자).
- -F file : filter 표현의 입력으로 파일을 받아들인다. 커맨드라인에 주어진 추가의 표현들은 모두 무시된다.
- -i device : 어느 인터페이스를 경유하는 패킷들을 잡을지 지정한다. 지정되지 않으면 시스템의 인터페이스 리스트를 뒤져서 가장 낮은 번호를 가진 인터페이스를 선택한다(이 때 loopback은 제외된다).
- -l : 표준 출력으로 나가는 데이터들을 line buffering한다. 다른 프로그램에서 tcpdump로부터 데이터를 받고자 할 때, 유용하다.
- -n : 모든 주소들을 번역하지 않는다(port, host address 등등)
- -N : 호스트 이름을 출력할 때, 도메인을 찍지 않는다.
- -O : packet-matching code optimizer를 실행하지 않는다. 이 옵션은 optimizer에 있는 버그를 찾을 때나 쓰인다.
- -p : 인터페이스를 promiscuous mode로 두지 않는다.
- -q : 프로토콜에 대한 정보를 덜 출력한다. 따라서 출력되는 라인이 좀 더 짧아진다.
- -r file : 패킷들을 '-w'옵션으로 만들어진 파일로부터 읽어 들인다. 파일에 "-" 가 사용되면 표준 입력을 통해서 받아들인다.
- -s length: 패킷들로부터 추출하는 샘플을 default값인 68Byte외의 값으로 설정할 때 사용한다(SunOS의 NIT에서는 최소가 96Byte이다). 68Byte는 IP, ICMP, TCP, UDP등에 적절한 값이지만 Name Server나 NFS 패킷들의 경우에는 프로토콜의 정보들을 Truncation할 우려가 있다. 이 옵션을 수정할 때는 신중해야만 한다. 이유는 샘플 사이즈를 크게 잡으면 곧 패킷 하나하나를 처리하는데 시간이 더 걸릴 뿐만 아니라 패킷 버퍼의 사이즈도 자연히 작아지게 되어 손실되는 패킷들이 발생할 수 있기 때문이다. 또, 작게 잡으면 그만큼의 정보를 잃게 되는 것이다. 따라서 가급적 캡춰하고자 하는 프로토콜의 헤더 사이즈에 가깝게 잡아주어야 한다.
- -T type : 조건 식에 의해 선택된 패킷들을 명시된 형식으로 표시한다. type에는 다음과 같은 것들이 올 수 있다. rpc(Remote Procedure Call), rtp(Real-Time Applications protocol), rtcp(Real-Time Application control protocal), vat(Visual Audio Tool), wb(distributed White Board)
- -S : TCP sequence번호를 상대적인 번호가 아닌 절대적인 번호로 출력한다.
- -t : 출력되는 각각의 라인에 시간을 출력하지 않는다.
- -tt : 출력되는 각각의 라인에 형식이 없는 시간들을 출력한다.
- -v : 좀 더 많은 정보들을 출력한다.
- -vv : '-v'보다 좀 더 많은 정보들을 출력한다.
- -w : 캡춰한 패킷들을 분석해서 출력하는 대신에 그대로 파일에 저장한다.
- -x : 각각의 패킷을 헥사코드로 출력한다.

조건 식(expression)

옵션의 제일 마지막인 조건 식은 어떤 패킷들을 출력할지를 선택하는데 쓰인다. 조건식이 주어지지 않는 다면 모든 패킷들이 그 대상이 될 것이다. 일단 주어지면, 아무리 패킷들이 많아도 조 건식에 부합하는 패킷만을 출력한다.

조건 식들은 하나 또는 몇 개의 primitive들로 구성되어 있다. primitive들은 보통 하나 혹은 몇 개의 qualifier들 다음에 오는 하나의 값으로 이루어진다. Qualifier들은 모두 3 종류이며 다음과 같다.

- type : 주어진 값의 종류가 무엇인지를 나타낸다. 가능한 type들은 'host', 'net', 'port'가 있다. type이 없는 값들은 type을 host라 가정한다.
- dir : id로부터의 어떤 특정한 전송 방향을 나타낸다. 가능한 방향은 'src', 'dst', 'src or dst', 'src and dst'이다. 만약 방향이 정해지지 않았다면, src or dst라 가정한다. "For `null' link layers (i.e. point to point protocols such as slip) the inb ound and out bound qualifiers can be used to specify a desired direction."
- proto : 매칭을 특정 프로토콜에 한해서 수행한다. 가능한 프로토콜들은 ether, fddi, ip, arp, rarp, decnet, lat, sca, moprc, mopdl, tcp, udp이다. 만약 프로토콜이 명시되지 않았다면, 해당하는 값의 type에 관련된 모든 프로토콜들이 그 대상이 된다.

이 밖에도 위의 패턴을 따르지 않는 Primitive들이 존재한다(gateway, broadcst, less, greater, 산술식).

좀 더 정교한 조건 식들을 사용하려면, 'and(&&)', 'or(||)', 'not(!)'들을 사용하여 여러 primitive들을 연결하면 된다. 같은 표현들은 생략될 수 있다.

사용 가능한 Primitive들

- dst host HOST : packet의 IP destination 항목이 HOST일 때 참이 된다.
- src host HOST : packet의 IP source 항목이 HOST일 때 참이 된다.
- host HOST : IP source, IP destination 항목 중 어느 하나라도 HOST이면 참이다.
- ether dst ehost : ethernet destination 주소가 ehost일 때 참이다.
- ether src ehost : ethernet source 주소가 ehost일 때 참이다.
- ether host ehost : ethernet source, destination 항목들 중 어느 하나라도 ehost이면 참이다.
- gateway host : 패킷이 host를 게이트웨이로 사용하면 참이다. 이 말의 의미는 ethernet sour ce나 destination 항목은 host이지만, IP source와 destination은 host가 아닐 때를 말한다.
- dst net NET : 패킷의 IP destination 주소가 NET의 network number를 가지고 있을 때 참이 다.
- src net NET : 패킷의 IP source 주소가 NET의 network number를 가지고 있을 때 참이다.
- net NET : 패킷의 IP source 주소 혹은 destination 주소가 NET의 network number를 가 지고 있을 때 참이다.
- net netmask mask : IP 어드레스가 지정된 netmask를 통해서 net과 매칭되면 참이다.
- net net/len : IP 어드레스가 netmask와 len 비트만큼 매치되면 참이다.
- dst port PORT : 패킷이 ip/tcp, ip/udp 프로토콜의 패킷이고 destination port의 값이 PORT일 때 참이다. port는 /etc/services에 명시된 이름일 수도 있고 그냥 숫자일 수도 있다. 만약 이름이 사용됐다면 port 번호와 프로토콜이 같이 체크될 것이다. 만약 숫자나 불 확실한 이름이 사용됐을 경우에는 port 번호만이 체크될 것이다.
- src port PORT : 패킷의 source port의 값으로 PORT를 가지면 참이다.
- port PORT : 패킷의 source, destination port 중에 하나라도 PORT이면 참이다.
- less length : 패킷이 length보다 짧거나 같으면 참이다.(len <= length)
- greater length : 패킷이 length보다 짧거나 같으면 참이다.(len >= length)
- ip proto protocol : 패킷이 지정된 종류의 프로토콜의 ip패킷이면 참이다. Protocol은 icmp, igrp, udp, nd, tcp 중의 하나 혹은 몇 개가 될 수 있다. 주의할 점은 tcp, udp, icmp들은 '\'로 escape되어야 한다.

- ehter broadcast : 패킷이 ethernet broadcast 패킷이라면 참이다. ehter는 생략 가능하다.
- ip broadcast : 패킷이 IP broadcast 패킷이라면 참이다.
- ether multicast : 패킷이 IP multicast 패킷이라면 참이다.
- ether proto protocol : 패킷이 ether type의 protocol이라면 참이다. protocol은 ip, arp, rarp 중에 하나 혹은 몇 개가 될 수 있다. ip proto protocol에서와 마찬가지로 ip, arp, rarp는 escape 되어야 한다.
- decnet src host : 만약 DECNET의 source address가 host이면 참이다. 이 어드레스는 '10.123'이 나 DECNET의 host name일 수 있다. DECNET host name은 DECNET에서 돌아가도록 설정된 Ultrix 시스템에서만 사용 가능하다.
- decnet dst host : DECNET destination address가 host이면 참이다.
- decnet host HOST : DECNET source, destination address중의 하나라도 HOST이면 참이다.
- ip, arp, rarp, decent : ether proto [ip|arp|rarp|decnet]의 약어
- lat, moprc, mop이 : ether proto [lat|moprc|mopdl]의 약어
- tcp, udp, icmp : ip proto [tcp|udp|icmp]의 약어
- expr relop expr : EXPR
  proto [expr:size]의 형식을 띤다. proto, expr, size에 올 수 있는 것들은 다음과 같다.
- proto : ether, fddi, ip, arp, rarp, tcp, udp, icmp
- expr : indicate Byte offset of packet of proto
- size : optional. indicate the size of bytes in field of interest
- default is one, and can be two or four
- RELOP
  !=, =, <=, >=, etc.
- 이 조건 식을 사용하기 위해서는 먼저 해당하는 Protocol(proto)의 헤더에 관련된 것들을 자세히 알아야만 한다. proto에는 대상이 될 프로토콜을 지정한다. expr에는 프로토콜 헤더의 처음부터의 Byte Offset을 지정하는 식이 들어가게 된다. Size는 Option이며 지정이 안 되어 있을 경우에는 자동으로 1byte를 지칭한다. 따라서 이 조건 식을 사용하게 되면 헤더에 포함된 정보를 Bitmask를 사용하여 직 접 원하는 패킷인지를 가려낼 수 있기 때문에, 보다 정밀한 사용이 가능하게 된다.

Tcpdump의 사용 예제들

- br1 interface에서 10.1.1.1의 호스트로부터 Input, Output 패킷들을 출력
  # tcpdump –i br1 host 10.1.1.1
- br1 interface에서 10.1.1.1에서 100.1.1.1을 제외한 모든 호스트로 날아다니는 IP 패킷들을 출력
  # tcpdump –i br1 ip host 10.1.1.1 and not 100.1.1.1
- br1 interface 거치는 ftp에 관련된 패킷들을 출력
  # tcpdump –i br1 port 21 or 20
- System에 Telnet 접속 후 25번 Uplink Port에서 TCP 23번을 제외하고 ICMP Packet을 출력
  # tcpdump –i eth25 not port 23 and icmp
- 가입자 1번 Port에서 들어오는 Packet중 Ether Broadcast만 출력

  # tcpdump –i eth01 ether broadcast
- 가입자 1번 Port에서 들어오는 Packet중 IP Broadcast만 출력

  # tcpdump –i eth01 ip broadcast
- br1 interface 에서 들어오는 모든 Packet을 출력
- # tcpdump –i br1

3방향 핸드쉐이크는 다음과 같이 진행

- 클라이언트는 서버에 TCP연결 요구 신호로 SYN 보냄
- 새로운 SYN에 의해 신호화된 연결 요구, 요청에 대한 ACK보냄(서버)
- SYN과ACK를 받고 계속 연결 유지 시, 마지막 하나의 ACK를 서버에 보냄

예제)TCPdump를 사용한 3방향 핸드쉐이크

tclient.net.39904 >telnet.com.23: S 733381829(0) win 8760 <mss 1460>(DF)
telnet.com.23 >tclient.net.39904: S 1192930639:1192930639(0) ack 733381830 win -> 1024 <mss 1460>(DF)
tclient.net.39904>telnet.com.23: . ack 1 win 8760

세션을 종료 방법

1. 정상적인 방법
   - TCP세션을 종료하구 싶을 경우, FIN신호를 보냄
   - 수신 호스트는 ACK로 다시 신호를 보냄(half close)
2. 정상적인 방법의 예
   - 클라이언트는 FIN으로 종료하고, 서버가ACK를 실행
     tclient.net.39904 > telnet.com.23: F 14:14(0) ack win 8760(DF)
     telnet.com.23 > tclient.net.39904: . Ack 15 win 1024(DF)
   - 서버는 FIN으로 종료하고, 클라이언트가 ACK를 다음과 같이 실행
     telnet.com.23 > tclient.net.39904: F 186:186(0) ack win 1024(DF)
     tclient.net.39904 > telnet.com.23: . Ack 187 8760 (DF)

3. 비정상적인 종료 방법
   - 갑작스럽고, 예기치 못한 비정상적인 종료
   - 하나의 호스트가 다른 쪽에 RESET을 보냄으로 종료됨

4. 비정상적인 종료 방법 예
     tclient.net.39904 > telnet.com.telnet: R 28:28(0) ack 1 win 8760(DF)

데이터 전송

- TCP가 연결을 시작하는 이유는 어떠한 데이터를 전송하기 위해 연결

- 3방향 핸드쉐이크 후와 종료 전에, tclient.net과 telnet.com 사이의 예
  tclient.net.39904 > telnet.com.23 : P 1:28(27) ack 1 win 8760
  telnet.com.23 > tclient.net.39904: P 1:14(13) ack 1 win 1024
  telnet.com.23 > tclient.net.39904: P 14:23(9) ack 28 win 1024

TCP Flag

정상적인 TCP 연결은 하나 혹은 그 이상의 플래그를 가짐(연결구분)

Tcp 플래그	플래그 표현	플래그 의미
SYN	“S”	tcp 연결의l 첫 번째 부분인 세션 연결 요청
ACK	“ack”	송신자로부터 데이터 영수증으로 받음을 알림 다른 플래그와 함께 결합하여 사용, 부가적 가능
FIN	“F”	수신호스트로의 연결을 정상적으로 마칠 때 사용
RESET	“R”	수신 호스트와의 연결을 즉시 종료하기 위해 사용
PUSH	“P”	응용 소프트웨어 데이터를 송신호스토로 “전송” 데이터 수신 시에 버퍼가 채워지기를 기다리지 않고, 어플리케이션으로 전달(telnet) 대역폭의 효율보다는 응답성에 초점
URGENT	“urg”	“긴급한” 데이터가 다른 데이터에 우선함 ftp 다운로드 시 멈출 때(Ctrl +C)
Placeholder	“.”	연결 과정이 SYN,FIN,RESET이나 PUSH 플래그를 가지고 있지 않다면, 마침표(.)가 목적지 포트 후에 표시

Tcpdump의 일반적인 포맷

09:32:43:910000 nmap.edu.1173 > dns.net.21: S 62697789:62697789(0) win 512

- TCPdump 결과는 유일하게 구별,플래그 필드와 시퀀스 숫자가 TCP의 특징을 보여줌
  - > 는 방향을 나타낸다. 현 상태는 왼쪽은 source 오른쪽은 destination이다.
  - Nmap.edu : 출발지 호스트 명
  - 1173 : 출발지 포트 번호거나 포트 서비스
  - dns.net : 목적지 호스트 번호
  - 21 : 목적지 포트 번호
  - S : TCP 플래그. S는 TCP연결을 시작하기 위해 요청하는 SYN 플래그를 나타냄
    - F(finish) : connection을 끊을 때
    - S(synch) : 통신 전후에 Synch을 맞출 때
    - P(push) : data를 이동시킬 때
    - R(reset) : connection을 reset시킬 때 사용한다.
    - “.”(no flag)
  - 62697789:62697789(0) TCP 시작 시퀀스 번호 : TCP 종료 시퀀스 번호

src > dst: flags data-seqno ack window urgent options

- Data-seqno : 패킷이 쪼개진 데이터에서의 패킷의 sequence number
- Ack : 다음 받길 희망하는 sequence number
- Window : 이 connection에서 다른 방향으로의 받을 수 있는 window byte의 남는 space
- Urg : tcp에서의 urgent flag
- Options : option 사항 기술