

TCPDUMP Quick Reference

Jialong He

Jialong_he@bigfoot.com

http://www.bigfoot.com/~jialong_he

TCPDUMP

Descriptions

<http://www.tcpdump.org>

Tcpdump prints out the headers of packets on a network interface that match the boolean expression.

```
tcpdump [ -aBdDeflnNOpqRStvxX ] [ -c count ] [ -F file ] [ -i interface ] [ -m module ] [ -r file ]  
[ -s snaplen ] [ -T type ] [ -w file ] [ -E algo:secret ] [ expression ]
```

Selected Options

- F** Use *file* as input for the filter expression. An additional expression given on the command line is ignored.
- I** Listen on interface. If unspecified, tcpdump searches the system interface list for the lowest number.
- p** *Don't* put the interface into promiscuous mode.
- r** Read packets from *file* (which was created with the *-w* option). Standard input is used if *file* is ``-''.
- W** Write the raw packets to *file* rather than parsing and printing them out. They can later be printed with the *-r* option. Standard output is used if *file* is ``-''.

Expression (BPF Packet Filtering)

expression selects which packets will be dumped. If no *expression* is given, all packets on the net will be dumped. Otherwise, only packets for which *expression* is `true' will be dumped.

type: (1) **host**, (2) **net**, (3) **port**

direction: (1) **src**, (2) **dst**, (3) **src or dst**, (4) **src and dst**

protocol: (1) **ether**, (2) **ip**, (3) **tcp**, (4) **udp**, (5) **arp**, (6) **rarp**

logical operator: (1) **and**, (2) **or**, (3) **not**

- dst host host** destination field of the packet is *host*.
- src host host** source field of the packet is *host*.
- host host** **either** source **or** destination of the packet is *host*.
- ether dst ehost** ethernet destination address is *ehost*.
- ether src ehost** ethernet source address is *ehost*.
- ether host ehost** either the ethernet source or destination address is *ehost*.
- gateway host** the packet used *host* as a gateway.
- dst net net** destination address of the packet has a network number of *net*. *Net* may be either a name from /etc/networks or a network number.
- src net net** source address of the packet has a network number of *net*.
- net net** either the source or destination address of the packet has a network number of *net*.
- net net mask mask** the IP address matches *net* with the specific netmask. May be qualified with **src** or **dst**.
- net netlen** the address matches *net* a netmask *len* bits wide. May be qualified with **src** or **dst**.

- dst port port** the packet is ip/tcp, ip/udp and has a destination port value of *port*.
- src port port** the packet has a source port value of *port*.
- port port** either the source or destination port of the packet is *port*.
- tcp src port port** matches only tcp packets whose source port is *port*.
- less length** if the packet has a length less than or equal to length. This is equivalent to: *len* <= length.
- greater length** if the packet has a length greater than or equal to length. This is equivalent to: *len* >= length.
- ip proto protocol** True if the packet is an IP packet (see *ip(4P)*) of protocol type *protocol*. *Protocol* can be a number or one of the names *icmp*, *icmp6*, *igmp*, *igmp*, *pim*, *ah*, *esp*, *udp*, or *tcp*. Note that the identifiers *tcp*, *udp*, and *icmp* are also keywords and must be escaped via backslash (\), which is \\ in the C-shell. Note that this primitive does not chase protocol header chain.
- ether broadcast** the packet is an ethernet broadcast packet.
- ip broadcast** the packet is an IP broadcast packet.
- ether multicast** the packet is an ethernet multicast packet.
- ip multicast** the packet is an IP multicast packet.
- ether proto protocol** if the packet is of ether type *protocol*. *Protocol* can be a number or one of the names *ip*, *ip6*, *arp*, *rarp*, *atalk*, *aarp*, *decnet*, *sca*, *lat*, *mopdl*, *moprc*, or *iso*. Note these identifiers are also keywords and must be escaped via backslash (\).
- expr relop expr** *expr* is an arithmetic expression composed of integer constants (expressed in standard C syntax), the normal binary operators [+ , - , * , / , & , |], a length operator, and special packet data accessors.
relop is one of > , < , >= , <= , = , !=.
- proto [expr : size]** *Proto* is one of **ether**, **fddi**, **tr**, **ip**, **arp**, **rarp**, **tcp**, **udp**, **icmp** or **ip6**. For example, **ether[0] & 1 != 0** catches all multicast traffic.

Primitives may be combined using: A parenthesized group of primitives and operators (parentheses are special to the Shell and must be escaped).

Negation (^! or `not').

Concatenation (^&& or `and').

Alternation (^| or `or').

Example: To print traffic between helios and either hot or ace:

```
tcpdump host helios and ( hot or ace )
```

To print traffic among local net 192.168.1.0

```
tcpdump net 192.168.1.0 mask 255.255.255.0
```