

# TCPDUMP

Presented by  
Jyotsna A  
Vincy Marin Mathew

# Contents

- 1.Introduction
- 2.Product description
- 3.Installation requirements
- 4.Configuration
- 5.Using tcpdump
- 6.tcpdump options
- 7.Imaginatively using tcpdump
- 8.References

# Introduction

- Popular network debugging tool.
- Used to intercept and display packets transmitted/received on a network.
- Filters used to restrict analysis to packets of interest.

# Product description

- Tcpdump, is free, open source software. It is useful for traffic analysis and network monitoring.
- It follows the BSD license.
- It can capture and see the contents of all packets flowing across the network.
- It is a much faster sniffer compared to those using GUI.
- It needs lesser memory and CPU utilization.
- Runs on command line.

# Installation requirements

- Tcpdump works on most Unix-like platforms: Linux, Solaris, BSD, Mac OS X, HP-UX and AIX.
- On Windows, WinDump is used; it's a port of tcpdump to Windows.
- On Unix platforms tcpdump is built upon the libpcap packet capture library and on Windows winpcap.
- On Unix and most other operating systems, a user must have superuser privileges to use tcpdump due to its use of promiscuous mode.

# Configuration

- Build libpcap library.
- ANSI C or GNU C compiler is required to build tcpdump.
- Run ./configure
- After successful completion acquire su permissions and type “make install”
- By default tcpdump is installed with group execute permissions.

# Using tcpdump

- `tcpdump -i ath0`  
listen on interface  
If unspecified tcpdump searches the system interface list for the lowest numbered.
- `tcpdump -w /home/./filename -i ath0`  
write the raw packets to file
- `tcpdump -r /home/./filename`  
read packets from the file(which was created with the -w option)

# tcpdump options

- Command line options are available with tcpdump, can be separated into four categories.
- 1.To control the program operations
- 2.To control how data is displayed
- 3.To control what data is to be displayed
- 4.Filtering commands

# Controlling program behavior

- `tcpdump -c100 -i ath0`  
it will capture 100 packets and then terminate.
- `tcpdump -p -i ath0`  
interface should not be put into promiscuous mode and will capture – traffic to or from the host, multicast traffic, and broadcast traffic.
- `tcpdump -s200 -i ath0`  
-s controls the amount of data captured. Here it will capture the entire packet if its length  $\leq$  200 bytes

# Controlling how information is displayed

- `tcpdump -a -i ath0`
  - a option attempts to force network addresses into names
- `tcpdump -n -i ath0`
  - n option prevents the conversion of addresses into names
- `tcpdump -N -i ath0`
  - N option prevents domain name qualification
- `tcpdump -f -i ath0`
  - f option prevents remote name resolution

# Controlling what's displayed

- `tcpdump -e -i ath0`
  - e option is used to display link-level header information
- `tcpdump -x -i ath0`
  - x option provides a hexadecimal dump of packets, excluding link-level headers
  - The amount of information displayed will depend on how many bytes are collected, as determined by the -s option.

# Filtering

- tcpdump can act as filters
- Filters permit you to specify what traffic you want to capture, allowing you to focus on just what is of interest.
- This can be absolutely essential if you need to extract a small amount of traffic from a massive trace file.
- There are four types of filters.

# 1. Address filtering

- `tcpdump host 205.153.63.30`  
This command captures all traffic to and from the host with the IP address 205.153.63.30
- `tcpdump ether host 0:10:5a:e3:37:c`  
the Ethernet address of a computer to select traffic
- `tcpdump dst 205.153.63.30`  
traffic flows for a single direction  
Either to a host or from a host, using `src` to specify the source of the traffic or `dst` to specify the destination. Multicast or broadcast traffic can be selected by using the keyword `multicast` or `broadcast`, respectively

## 2. Protocol and port filtering

- It is possible to restrict capture to specific protocols and can also restrict capture to services built on top of these protocols -- by using a few specific keywords known by tcpdump, by protocol using the proto keyword, or by service using the port keyword.
- `tcpdump ip`  
restricts the traffic captured to IP traffic
- `tcpdump tcp`

# 3. Packet characteristics

- Filters can also be designed based on packet characteristics such as packet length or the contents of a particular field.
- These filters must include a relational operator. To use length, the keyword less or greater is used.
- Eg: `tcpdump greater 200`  
This command collects packets longer than 200 byte.

## 4. Compound filters

- Compound filters can be constructed in tcpdump using logical operator and, or, and not.
- Negation has the highest precedence.
- tcpdump host 205.153.63.30 and ip  
if we want only the IP packets
- tcpdump host 205.153.63.30 and not ip  
if you want all traffic to the host except IP  
traffic

# Imaginatively Using tcpdump

- Can be used to develop a tcp/ip cutter.
- Due to increased traffic, there exists possibility for the loss of valuable information send from one system to another system. These messages have been sent as packets through network.
  - Using tcpdump we can analyze the source and destination 'IP addresses' of systems involved in the traffic. According to which protocol, these messages are monitored can also be obtained.
- tcpdump is wiretap software that plugs into computer networks. It provides an easy way to analyze the network

# References

- [http://www.unix.org.ua/oreilly/networking\\_2ndEd/tshoot/ch05\\_04.htm](http://www.unix.org.ua/oreilly/networking_2ndEd/tshoot/ch05_04.htm)
- <http://en.wikipedia.org/wiki/Tcpdump>
- <http://www.tcpdump.org>