tcpdump is a common packet analyzer that runs under the command line. It allows the user to intercept and display TCP/IP and other packets being transmitted or received over a network to which the computer is attached.

Man Page

Also check out these cool tcpdump usage examples here

1) analyze traffic remotely over ssh w/ wireshark

ssh root@example.com tcpdump -w - 'port !22' | wireshark -k -i -

This captures traffic on a remote machine with tcpdump, sends the raw pcap data over the ssh link, and displays it in wireshark. Hitting ctrl+C will stop the capture and unfortunately close your wireshark window.

2) Get Cisco network information

tcpdump -nn -v -i eth0 -s 1500 -c 1 'ether[20:2] == 0x2000'

This gives you lots of nifty Cisco network information like VLAN tag, port and switch information.

3) Remotely sniff traffic and pass to snort

ssh root@pyramid \ "tcpdump -nn -i eth1 -w -" | snort -c /etc/snort/snort.conf -r –

I have a small embedded linux device that I wanted to use for sniffing my external network, but I didn't want to recompile/cross-compile snort for the embedded platform. So I used tcpdump over ssh to pass all the traffic as pcap data to a "normal" Linux system that then takes the pcap data and passes it to snort for processing.

4) Sniffing network (gui)

tcpdump -v -i <INTERFACE> -s 0 -w /tmp/sniff.pcap port <PORT> # On the remote side

Then hit ^C to stop, get the file by scp, and you can now use wireshark like this :

wireshark /tmp/sniff.pcap

If you have tshark on remote host, you could use that :

wireshark -k -i <(ssh -l root <REMOTE HOST> tshark -w - not tcp port 22)

The last snippet comes fromhttp://wiki.wireshark.org/CaptureSetup/Pipes

5) Getting started with tcpdump

tcpdump -nli eth0; tcpdump -nli eth0 src or dst w.x.y.z;

tcpdump -nli eth0 port 80; tcpdump -nli eth0 proto udp

At some point you want to know what packets are flowing on your network. Use tcpdump for this. The man page is obtuse, to say the least, so here are some simple commands to get you started.

-n means show IP numbers and don't try to translate them to names.

-l means write a line as soon as it is ready.

-i eth0 means trace the packets flowing through the first ethernet interface.

src or dst w.x.y.z traces only packets going to or from IP address w.x.y.z.

port 80 traces only packets for HTTP.

proto udp traces only packets for UDP protocol.

Once you are happy with each option combine them with 'and' 'or' 'not' to get the effects you want.

6) Capture data in ASCII. 1500 bytes

tcpdump -i eth0 -n tcp port 80 -A -s1500

Sniffing traffic on port 80 only the first 1500 bytes

7) See entire packet payload using tcpdump.

tcpdump -nnvvXSs 1514 -i <device> <filters>

This command will show you the entire payload of a packet.

The final “s” increases the snaplength, grabbing the whole packet.

8) view http traffic

tcpdump -i eth0 port 80 -w - | hd

9) ignore all ssh traffic

tcpdump -i eth0 -n 'port ! 22'

and That’s how you take a Tcpdump ;)