Tools For Development And Exploration

This document describes the development tools available on the TLAB machines, as well as tools that you can use to explore the local network and the Internet. Your code will be expected to compile with these tools and work on the TLAB machines. If you decide to do development somewhere else, make sure you use the same versions of the tools. This is especially important in the case of GCC, especially if you are using C++. All of these tools are included with most Linux distributions or are quite easy to install.

Paths

It is important to note that two versions of certain tools are available on the TLAB machines, a default Red Hat 6.2 version in /usr/bin and an upgraded version in /usr/local/bin. Please make sure that /usr/local/bin precedes /usr/bin on your path. On a sh-style shell, you can set your PATH in the following way:

```
$ PATH=/usr/local/bin:$PATH; export PATH
```

while on a csh-style shell, you would run

```
% setenv PATH /usr/local/bin:$PATH
```

Editors

An editor such as XEmacs, GNU Emacs, vi, or others is essential. All are installed on the TLAB machines. Both Emacs editors integrate with make, GCC, and GDB to form a relatively powerful integrated development environment.

GNU Make

Make is the core tool for build management in Minet and many other programs. Make is covered in a separate document, and the GNU Make manual is available from the web page.

CVS

We will use CVS for version management during the class. Each of you will have read access to the Minet repository. If and when we release bug fixes, we'll ask you to use CVS to update your local copy of Minet. The CVS commands you will have to understand are "checkout" and "update". A pointer to the very helpful CVS Bubbles web site is on the web page.

GCC Compiler System

The TLAB machines include GCC 2.95.2, which is a considerable improvement over the old version of EGCS that is included with Red Hat 6.2 Linux. If you use another machine, please be sure that your gcc's version is the following:

```
$ gcc -v
Reading specs from /usr/local/lib/gcc-lib/i686-pc-linux-
gnu/2.95.2/specs
gcc version 2.95.2 19991024 (release)
```

Other versions of GCC may also work, but we will compile and test your code on the tlab machines.

GDB Debugger

The TLAB machines include GDB 5.0, which is an improvement over the 4.x release included with Red Hat. If you are having strange problems in debugging C++ code, make sure you are running the 5.0 version or later.

Promiscuous mode and the Berkeley Packet Filter

Your kernel must be able to run the Ethernet card in promiscuous mode and it must have Berkeley Packet Filter support enabled. The default Red Hat 6.2 kernel permits this if you are root or you are running a binary that is suid root. On the tlab machines, we will supply suid binaries that will give you appropriate access.

Libpcap

Libpcap is an interface that program can use to access an Ethernet card running in promiscuous mode. Make sure you have version 0.4-19, which is the version that ships with Red Hat 6.2. To find out your version, you can use rpm. You should see the following output:

```
$ rpm -q -a | grep libpcap
libpcap-0.4-19
```

Libnet

Libnet is an interface for injecting raw Ethernet packets into the network. You should have version 1.0.1b.

Tcpdump and Ethereal

Tcpdump is a program for printing the traffic that your Ethernet card sees in a human readable form. If you are running the Ethernet card in promiscuous mode, then you can see all of the traffic that passes by. Running tcpdump without arguments will spew all the traffic that can be seen. You can supply tcpdump with a packet filter to limit what is printed to what you are specifically interested in. For example,

```
$ /usr/sbin/tcpdump src scratchy and tcp
```

will show you TCP traffic originating from the host "scratchy." If /usr/sbin/tcpdump does not work for you, use /usr/local/sbin/tcpdump. /usr/local/sbin/ethereal is a graphical version of tcpdump. You will find these programs essential to debugging your code.

Ifconfig

/sbin/ifconfig –a gives you information about the interfaces in a machine.

Route

/sbin/route will show you the routing table of the machine

Ping

Ping is a program (and a duck) that you can use to see if the TCP/IP stack on a remote machine is actually functioning, provided that the stack supports ICMP. Minet is pingable.

Traceroute

Traceroute lets you discover the path that an IP packet takes from your local machine to some remote host, provided the intervening routers support ICMP. For example, here is the route from one Northwestern machine to www.wisc.edu:

```
$ /usr/sbin/traceroute www.wisc.edu
traceroute to www.wisc.edu (144.92.104.37), 30 hops max, 38 byte packets
 1  birl-idf-eth-3.nwu.edu (129.105.100.170)  0.818 ms  0.715 ms  0.729 ms
 2  lev-mdf-5-gig-3-0-1.nwu.edu (129.105.253.53)  0.958 ms  0.874 ms  0.795 ms
 3  lev-mdf-rtr-1.nwu.edu (129.105.253.238)  1.556 ms  1.290 ms  1.281 ms
 4  lev-mdf-rtr-2.nwu.edu (199.249.169.65)  1.941 ms  1.984 ms  2.804 ms
 5  206.220.243.35 (206.220.243.35)  8.204 ms  7.681 ms  7.440 ms
 6  r-macc.net.wisc.edu (144.92.128.129)  7.977 ms  8.865 ms  8.772 ms
 7  gopher.adp.wisc.edu (144.92.104.37)  9.245 ms  8.048 ms  9.038 ms
```

Netcat

Netcat, or "nc", is a tool for sending or receiving data using UDP or TCP. It is very handy for testing servers and things like the Minet protocol stack.

Nslookup, Dig, and Whois

Nslookup is a command-line and interactive interface to the DNS system. You can use it to find mappings of hostnames to IP address. Dig is similar in principle, but it presents much more detailed information from DNS. It is very useful in figuring out how various name server tricks are played. For fun, you can use dig to figure out how Akamai caching works. Whois allows you to query for detailed information on top-level domains. For example, you can use whois to find out who owns flibbertygibbet.com.