

Linux Guide

Accessing the command prompt

Before you can access the command prompt, you must login to the system. The administrative user on Linux machines is named **root**. On most Linux distributions, you will be able to enter root and its password on the login screen/prompt. However, there are some Linux distributions that require you to login as a non-administrative user first and then use the **su** – command to switch to the administrative account.

Once logged in, most Linux distributions will be in a graphical environment. To access a command prompt you will need to run a terminal application. In many Linux distributions, a terminal can be opened by right-clicking on the desktop and selecting **Terminal**. Other possible locations are **Applications->System Tools->Terminal** or **Applications->Accessories->Terminal**. If those links are not present then search the menus for anything named “Terminal” or “Console.” A few common terminal applications are **gnome-terminal**, **konsole**, and **xterm**.

Upon logging in, your location on the file system is root’s home directory, /root. Most configuration files in Linux are stored under the /etc directory, and the Quagga files in particular are under /etc/quagga. Before editing these files, it would be best to change your current directory to /etc/quagga using the cd command.

Linux commands

This lab typically requires only a few basic commands, primarily to create the Quagga configuration files and display network information. The following list of commands should aid in accomplishing these tasks:

ls : List files in the current directory.

cd <directory> : Change the current working location on the file system to <directory>.

cp <source> <destination> : Copy the file named <source> to the <destination> file or directory.

rm <filename> : Delete <filename>.

less <filename> : Show the contents of <filename> one page at a time.

ifconfig : List the network interfaces and their configuration (IP address, netmask, status, etc.).

route -n : Display the Linux kernel’s routing table.

ping <destination> : Ping <destination>.

traceroute <destination> : Display the route taken to <destination>.

Editing files

Before editing the Quagga config files, change the current directory to /etc/quagga using the cd command. Since all of the Quagga servers have a similar initial configuration file, it would be easiest to create the file for one server and then copy and edit that file for each other server.

There are several editors available in Linux for both graphical and text-based environments. To edit files in a graphical, notepad-like editor use **gedit <filename>**. Other common graphical editors are **kedit** and **xedit**. If working in a text-based environment then you may use **vim <filename>** (or **vi <filename>**) to edit files. Other text-based editors are **joe**, **jed**, and **pico -w**, but these editors are often not present unless a full or custom install was done.

ITEC451 Network Design & Analysis – Laboratory Guide: Appendix

Although vi/vim is the standard text editor for all distributions of Linux, its interface is not immediately intuitive to most users. To be able to type text in vim, you must place the editor into Insert mode by pressing the “I” key. To exit Insert mode, press **ESC**. To save and quit, type **:wq** and to quit without saving, type **:q!**. Note that on some distributions, vim’s default configuration might make it very difficult to use. If you are unable to use the arrow keys or backspace, try running **vim -N**.

Opening a router console connection using minicom

To connect to the router’s console, you must open a terminal on the serial port. Under Linux, minicom is the standard modem and serial port communications program (similar to Hyper Terminal in Windows). The serial port that would be accessed as COM1 in Windows is **/dev/ttyS0** in Linux. COM2 is **/dev/ttyS1**.

To start minicom, open a separate terminal (minicom is text-based) and run **minicom -w -m**. The **-w** enables line wrapping and **-m** causes Minicom to use ALT as its command key instead of Ctrl-A. Note that the standard gnome terminal also uses ALT as its shortcut key and may interfere with minicom. The terminal’s shortcuts can be disabled under **Edit->Keyboard Shortcuts**. Minicom will attempt to connect to the serial port using any existing settings. If a previous configuration prevents it from starting, run Minicom with the **-c** option to jump directly to the configuration menu.

Once Minicom is running, the configuration menu can be accessed by pressing **ALT-O**. Under “Serial port setup,” set the port settings to 9600 bps, 8N1, hardware flow control and set the modem device if different from the default. Although not absolutely required, the “Init string” and “Reset string” under “Modem and Dialing” can be cleared to prevent minicom from sending these strings to the router when it starts. At the bottom of the setup menu, the configuration can be saved as the default (df1) or given a name. To load a named configuration, run the minicom command with the name of the configuration after the other options.

Configuring network settings from the command prompt

If the Linux machine being used does not provide a graphical network configuration tool, all of the network settings can be configured manually from the command prompt. The IP address and netmask can be set with the command: **ifconfig <interface> <ip address> netmask <netmask>**. For example, **ifconfig eth0 192.168.1.1 netmask 255.255.255.0** would set the IP address on the first Ethernet interface.

A default route can be created with the command **route add default gw <router-address>**. A static route to a specific network can be manually added to the kernel’s routing table by using **route add -net <network address> netmask <network mask> gw <next-hop address>**.

To clear an IP address from an interface, use **ifconfig <interface> 0.0.0.0**. To remove a route, simply replace the **add** keyword with **del** in the above route commands. Also note that when Linux is rebooted, all of the interface configuration and static routes will be cleared.

Installing the Quagga RPM package from the command prompt

On Fedora Core, packages can be installed from the command prompt using the yum command. Before running yum for the first time, it may be necessary to import the GPG keys used to verify packages:

```
rpm --import /usr/share/rhn/RPM-GPG-KEY-fedora
rpm --import /usr/share/rhn/RPM-GPG-KEY
yum install quagga
```

ITEC451 Network Design & Analysis – Laboratory Guide: Appendix

To install the RPM binary distribution of Quagga on a Linux distribution that is not Redhat or RPM based, use the command **rpm -i --nodeps quagga-x.x.x.i386.rpm**.

Compiling and installing Quagga from the source distribution

If the version or distribution of Linux being used does not support the pre-compiled binary packages then Quagga will have to be installed from the source distribution. A source install is also necessary to enable developmental support for ISIS. To be able to compile Quagga, the standard development tools and libraries must be installed.

Note that if you are installing Quagga over an existing RPM binary installation to enable ISIS support, you need only perform the configure/make step, copy and chmod the isisd.init file, and optionally insert the service name into /etc/services and run chkconfig for isisd. All other steps can be skipped.

Before installing Quagga, a user and group named quagga should be created. The servers will perform all operations as the quagga user, so every file that the Quagga servers attempt to read or write must be accessible by that user. There are many distribution specific methods for adding users but the following commands should adequately create the user and group on most Linux systems:

```
groupadd quagga
useradd -g quagga -d /var/run/quagga quagga
```

The source package, which is available from www.quagga.net, should first be copied to /usr/local/src and uncompressed with the command **tar xzf quagga-x.x.x.tar.gz**. Note that it is standard practice to place locally installed programs entirely under the /usr/local hierarchy. However, the following commands will install Quagga under /usr to ensure that existing Quagga installations are replaced by this one. If that is not desired, change the prefix, libdir, and sysconfdir paths to /usr/local. From within the Quagga source directory, issue the commands:

```
./configure --prefix=/usr --sysconfdir=/etc/quagga --libdir=/usr/lib/quagga
--localstatedir=/var/run/quagga --enable-isisd
make
make install
```

The Quagga source distribution includes initialization scripts for each server in the redhat directory. Although these scripts are intended for use on Redhat-based systems, it is likely that they will work with other Linux distributions. The script for each server must be copied to /etc/init.d and made executable using:

```
cp ./redhat/<server name>.init /etc/init.d/<server name>
chmod 755 /etc/init.d/<server name>
```

Each init script loads the /etc/sysconfig/quagga file on execution to get the default option switches passed to the server. This file must be copied into place from the redhat directory for the init scripts to function as expected. Once all files are in place, their user and group ownership should be changed to the quagga user.

```
cp ./redhat/quagga.sysconfig /etc/sysconfig/quagga
mkdir /var/log/quagga
chown -r quagga.quagga /etc/quagga /var/run/quagga /var/log/quagga
```

On Redhat-based systems, you may optionally use the **chkconfig --add <server name>** command to allow the server to be managed by the Server Settings->Services applet. This command starts the server immediately and also sets it to run automatically on boot. If there is no configuration file for each server passed to

ITEC451 Network Design & Analysis – Laboratory Guide: Appendix

chkconfig, the command will not succeed. **chkconfig --level 2345 <server name> off** can be used to prevent a server from starting on boot.

Although not required to operate, service-name to port-number mappings can be added to /etc/services to allow the server's name to be used in place of the port on telnet commands. This can be done by inserting the contents of the **SERVICES** file in the Quagga source directory into **/etc/services**. Note that Quagga uses the ports 2601-2608 for its servers. These ports may already be assigned in some /etc/services files. If that is the case, the original mappings can be commented out and replaced by the Quagga mappings or an alias can be created for each server by placing its name after the corresponding TCP port number of the existing mappings.

Manually starting and stopping Quagga services

On Redhat-based Linux distributions, the commands **service <server> start** and **service <server> stop** can be used to start and stop the Quagga servers (zebra, ripd, ospfd, bgpd). If your Linux system does not have the service command, try **/etc/init.d/<server> start** and **stop**. For example, “/etc/init.d/zebra start” will start the Zebra server on many systems. If that also fails then check /etc/init.d and /etc/rc.d for any file relating to Quagga or its servers and attempt to run that file with the start and stop options.

If all else fails, each Quagga server can be started by running the server directly. However, this should be the last method tried. The command that follows assumes that a quagga user exists and the Quagga files are in their standard locations. If either of these is not the case, you will have to edit the command to match your installation. For more information on each option, read the server's man page (man <server>) or the Quagga online documentation. As the root user, enter a command similar to **/usr/sbin/<server> -d -u quagga -f /etc/<server>.conf -i /var/run/quagga/<server>.pid**

Be aware that when this command was tested on our servers, we had to insert the commands **line vty** and **no exec-timeout** into each server's configuration file to be able to establish a telnet connection to that server. The reason for this is unknown. To stop a server started in this manner, use **kill** on the PID listed in the <server>.pid file. The shortcut **killall <server>** can also be used.

Running a TFTP server and upgrading the Cisco IOS image

When upgrading the IOS image on a Cisco router, a TFTP server is typically used as the source of the image. Although there are free TFTP servers available for Microsoft Windows platforms, most do not support multiple simultaneous connections. Using the standard Linux TFTP server is often the most convenient way to upgrade the IOS image on multiple routers. Note that both the Linux TFTP server and the Cisco router must have IP addresses configured and have network connectivity to each other to be able to copy an image.

The tftp-server package is rarely installed by default in most Linux distributions unless a full install was done. To install the TFTP server on Redhat-based machines run **yum install tftp-server** as root. Other distributions will most likely have a similar tftp-server package.

The TFTP service is normally handled by xinetd. The service can be activated by issuing the **chkconfig tftp on** command. If your distribution does not have chkconfig, edit **/etc/xinetd.d/tftp**, set “disabled” equal to “no” and restart xinetd with **killall -HUP xinetd**. Note that the default TFTP directory can also be changed in /etc/xinetd.d/tftp. On many Linux systems **/tftpboot** is the default location from which the server will share files. Copy the Cisco IOS image into this directory to make it available to the routers.

ITEC451 Network Design & Analysis – Laboratory Guide: Appendix

Before copying a new image onto the router it may be necessary to erase the flash. If there is sufficient room on the flash device it is possible to store two IOS images and use the **boot system flash <IOS image>** to boot the second image. If there is not enough space, the old image must be cleared with **erase flash**. Before erasing the flash, a copy of the existing image should be placed on the TFTP server with **copy flash tftp**.

To upgrade the IOS image, first verify that the router can ping the TFTP server. Note that an initial ping should not be skipped even if network connectivity was previously verified. It is not unusual for the first packet sent from the router to be dropped. After connectivity has been confirmed, issue the command **copy tftp flash** and fill in the server's IP address and the IOS image name when prompted.