Configuring Cricket to Monitor SNMP Objects

Contents

Background:

There are two free software packages for monitoring SNMP objects from a web interface.

The most widely known is MRTG, the "Multi Router Traffic Grapher." MRTG is well suited to monitoring network traffic from router interfaces, but has certain limitations in many other situations. Red Hat Linux comes with MRTG as a ready-built software package, and it is very easy to set it up to monitor network traffic. See `http://people.ee.ethz.ch/~oetiker/webtools/mrtg/`.

The newer package is called *Cricket*, and this is what we will install and configure today. Cricket has greater flexibility, and can be extended in more ways. Cricket is available from `http://cricket.sourceforge.net/`.

Both packages are written mostly in Perl. There are links to both of these software packages from the subject web site.

1 Procedure

There are eight steps to install and configure Cricket, as I demonstrated at the last lecture:

- install the necessary Perl modules using CPAN

- create a user called `cricket`

- unpack the Cricket tarball and run the configure script

- copy the sample configuration files into a directory called `cricket-config`

- modify the configuration to point to what you want to monitor

- configure `cron` to run the data collector

- create a `public_html` directory, and set up the programs in that directory

- configure Apache so that it will run the programs

1.1 Open the Documentation

1. Go to the Cricket installation web page at `http://cricket.sourceforge.net/support/doc/beginner.html`.

    You can get there from the main Cricket page by following these links: *support → documentation → Installing Cricket for the Complete Beginner*.

1.2 Installing the Necessary Perl modules

There are several Perl modules that Cricket requires. Some of these are already installed on your computer, some are not. You will install them with the CPAN program, which downloads them from the Internet, compiles, tests, then installs them if they pass all tests.

The modules to install from CPAN include:

- MD5

- DB_File

- Date::Parse

- Time::HiRes

- CGI

- Net::SNMP

1.2.1 Installing the Required Perl Modules from CPAN

1. First, you need the CPAN Perl module. People who cloned their hard disk from mine will find that you need to install it:

```
$ sudo rpm -Uhv \
   /home/nfs/rh-8.0-updated/RedHat/RPMS/perl-CPAN-1.61-55.i386.rpm
```

As usual, use the (tab) key to complete the file name; I don't remember these version numbers myself, of course!

2. Secondly, you need the development software package for the Berkeley Database system required to build DB_File:

```
$ sudo rpm -Uhv \
/home/nfs/rh-8.0-updated/RedHat/RPMS/db4-devel-4.0.14-14.i386.rpm
```

3. Start the CPAN modules' shell method from your own (not cricket's) account:

```
$ sudo perl -MCPAN -e shell
```

4. The shell will begin its interactive configuration process. Accept all defaults by pressing (Enter), **_except for the following_**:

- When prompted "CPAN build and cache directory?" enter: /var/cache/cpan Do not accept the default in your home directory; it will not work, since the root user has no right to write to your NFS mounted home directory.

- When asked to choose a Policy on building prerequisites, choose "follow":

  ```
  Policy on building prerequisites (follow, ask or ignore)? [ask] follow
  ```

    Some Perl modules depend on others. When installing a module that depends on others, a selection of "ask" here means the CPAN shell will prompt you and require you to press (Enter) to install the prerequisite module. The selection "follow" means, "Don't ask me; just go ahead, download and install any modules that are needed to install this one".

- For "Your http_proxy?" enter http://hqproxy.vtc.edu.hk:8080/

- When asked to pick a nearby continent, enter 5 to select (5) North America (Do not select (2) Asia, 2 ftp://ftp.pacific.net.hk/pub/mirror/CPAN/, because like many Hong Kong mirror sites, it is not maintained properly)

- When asked to "Select your country", enter 3 to select (3) United States.

- When asked "Select as many URLs as you like", I suggest just enter the first few numbers (corresponding to the first few URLs); I put "1 2 3 4 5 6 7". The numbers are just separated by spaces.

- Configuration should complete.

5. Now, at the cpan> prompt, type:

```
cpan> install MD5
cpan> install Date::Parse
...
```

and so on, installing all the six modules listed above.

6. For more information about the CPAN software, type "help". There is a manual page; type perldoc CPAN, or man CPAN to read it.

1.2.2 Installing the other two Perl Modules

The other two Perl modules are not available from CPAN; see the Cricket documentation for the location. We start here assuming that `ictlab` is mounted on `/home/nfs`. I have set up appropriate *automounter* entries in our LDAP directory server so that when you change to or list this directory, it will mount the network file system by NFS.

Do this in your own account.

Installing SNMP_Session:

1. Unpack the `SNMP_Session` tarball into your (not `cricket`'s) home directory:

   ```
   $ cd
   $ tar xvzf /home/nfs/snmp/SNMP_Session-0.94.tar.gz
   $ cd SNMP_Session-0.94
   $ perl Makefile.PL
   $ make
   $ sudo make install
   ```

   The `SNMP_Session` Perl modules are now installed.

   > You can get the latest copy of this module from `ftp://ftp.switch.ch/software/` `sources/network/snmp/perl/`. The web page is at `http://www.switch.ch/` `misc/leinen/snmp/perl/`.

Installing the Round Robin Database: This is available as two RPM packages, built from source RPMs downloaded from `http://rpmfind.net/linux/falsehope/home/` `gomez/rrdtool/rrdtool-1.0.39-1.7.2.src.rpm`.

1. Simply install using the `rpm` program:

   ```
   $  sudo rpm -Uhv \
   /home/nfs/redhat/contrib/rrdtool-*1.0.39-1.8.0.i386.rpm --nodeps
   ```

   After this, all the required Perl packages are installed.

1.3 Creating the `cricket` account

Here you create a local user. You will install the software into the home directory of this user.

Setting your path

1. Look at your path by typing:

   ```
   $ echo $PATH
   ```

2. If the directories `/sbin` and `/usr/sbin` are not already on your path, add them:

   (a) edit your login script:

      ```
      $  emacs ~/.bash_profile &
      ```

   (b) Add a line like this at the end of your login script:

      ```
      export PATH=$PATH:/sbin:/usr/sbin
      ```

   (c) Copy and paste this line into your terminal window.

Now create the `cricket` account:

1. Make sure that your `useradd` program is configured to create local accounts (see the handout I wrote about this for more information):

   ```
   $ sudo mkdir /home2
   $ sudo useradd -D -b /home2
   ```

2. Create a local user `cricket`:

   ```
   $ sudo useradd -c "Cricket Manager" cricket
   ```

3. Give this user a password:

   ```
   $ sudo passwd cricket
   ```

1.4   Unpacking Cricket and running `configure`

The Cricket tarball is available for download either from the Cricket web site, or you can install it directly from our server:

1. Open a window and become `cricket`:

   ```
   $ xhost +localhost
   $ su - cricket
   ```

   > The first command lets other users besides you (such as `cricket`) display on the local X server.
   >
   > The minus sign '`-`' runs `cricket`'s login scripts.

2. Change to `cricket`'s home directory and unpack the Cricket tarball:

   ```
   $ cd
   $ tar xvzf /home/nfs/snmp/cricket-1.0.3.tar.gz
   ```

3. Now create a symbolic link to the Cricket directory:

   ```
   $ ln -s cricket-1.0.3 cricket
   ```

4. Now run the `configure` script, which puts the path to `perl` in the first line of the Perl programs:

   ```
   $ cd ~/cricket
   $ ./configure
   ```

5. Now copy the file `cricket-conf.pl.sample` to `cricket-conf.pl`:

   ```
   $ cd ~/cricket
   $ cp cricket-conf.pl.sample cricket-conf.pl
   ```

6. Edit the file `~/cricket/cricket-conf.pl`, and change the line:

   ```
   $gCricketHome = "/home/cricket";
   ```

   to

   ```
   $gCricketHome = "/home2/cricket";
   ```

1.5   Copy the Config Tree and Modify it

Cricket uses a directory of configuration files that the authors call a *config tree.* Here we
copy the sample config tree, and modify it.

1. First copy the sample tree as the user `cricket`:

   ```
   $ cd
   $ cp -a cricket/sample-config cricket-config
   ```

2. Now copy the configuration that I wrote for the demonstration in the lecture theatre.
   Here I assume that `ictlab` is mounted on `/home/nfs`.

   ```
   $ cd ~/cricket-config
   $ cp -a /home/nfs/snmp/servers .
   #      This is a dot _____ˆ
   ```

   > There are two files in this directory: `Defaults` and `Targets`. These tell Cricket
   > to monitor the system load on our server `ictlab`, and also the amount of free
   > swap space, and the amount of free memory. You will later need to modify the
   > config tree to monitor other devices.
   >
   > If you copied the two files from `/home/nfs/snmp/servers` into `~/cricket-config`
   > instead of into the directory `~/cricket-config/servers`, then the collector
   > will not work. If you have done this, simply go back and copy the sample
   > configuration files back again.

3. Every time you modify the config tree, you need to run the `compile` script (as the
   user `cricket`):

   ```
   $ ~/cricket/compile
   ```

4. Cricket collects data from data sources using a script called the `collector`. This
   will be run by `cron`. First, check that the config tree has been set up correctly:

   ```
   $ ~/cricket/collector /servers
   ```

   This should run without errors, and five data items should be shown: the three
   values for system load, and the free swap space in kilobytes, and free memory.

1.6   Adding a `cron` entry for Cricket

Cricket collects the information it graphs using `cron`, every five minutes. To use `cron` to
gather the information for Cricket, you need to add an entry to `cricket`'s crontab like
this. First, you need to be `cricket`.

1. The `cron` program collects information about the network periodically using script
   called `collect-subtrees`. This script reads a configuration file ~`cricket/cricket/`
   `subtree-sets`. Edit this file:

   ```
   $ emacs ~/cricket/subtree-sets &
   ```

2. Add a new `set` to it like this:

```
set linuxservers:
        /servers
```

The script `collect-subtrees` takes one parameter, the *set* of data to collect. Here our set is called `linuxservers` so we use that parameter in our `cron` entry.

3. Start editing `cricket`'s crontab:

```
$ crontab -e
```

4. In the editor, you need to add an entry like this:

```
*/5 * * * * $HOME/cricket/collect-subtrees linuxservers
```

which will run the program `$HOME/cricket/collect-subtrees linuxservers` every five minutes.

1.7   Setting up the `public_html` directory in `cricket`'s home directory for Apache

I found it necessary to make a minor change from the instructions for putting the CGI programs in the `public_html` directory. This is what I did, as the user `cricket`:

```
$ cd
$ mkdir -p public_html/cricket
$ cd public_html/cricket
$ ln -s ~/cricket/VERSION
$ ln ~/cricket/grapher.cgi .
$ ln ~/cricket/mini-graph.cgi .
$ ln -s ~/cricket/lib ~/cricket/images .
# Note: this is a dot ------------------^
```

The only difference is that I made a *hard link*, rather than a *symbolic link*, to the CGI programs.

I have also added a link to the documentation, so that you can easily read it online:

```
$ ln -s ~/cricket/doc ~/public_html
```

You can read the Cricket documentation at `http://localhost/~cricket/doc/` after you have set up Apache—see the next section.

1.8   Configuring the Apache Web Server

Now configure Apache to run Cricket:

1. Edit Apache's configuration file:

```
$ sudo -v
$ sudo emacs /etc/httpd/conf/httpd.conf &
```

2. Search for the string "`UserDir`" in your editor

3. Comment out the line

   ```
   UserDir disable
   ```

   by putting a hash "**#**" in front of it:

   ```
   #UserDir disable
   ```

   and *un*comment the line

   ```
   UserDir public_html
   ```

   by *removing* the hash character.

4. Search for the "Directory" section for `public_html`.

5. If the section is commented out, copy it and uncomment the copy.

6. You need to add the option `ExecCGI`, and should end up with something like this:

   ```
   <Directory /home2/*/public_html>
       Options MultiViews Indexes SymLinksIfOwnerMatch ExecCGI
   </Directory>
   ```

   > Note that the manual for Apache is included on your hard disk. When Apache
   > is running, you will find it at this location: `http://localhost/manual/`. The
   > manual is very complete, and explains every possible configuration option of
   > Apache.

7. You also need to uncomment the line in Apache's configuration file:

   ```
   AddHandler cgi-script .cgi
   ```

8. Note too that `cricket`'s home directory, and ∼`cricket/public_html`, and ∼`cricket`
   `/public_html/cricket` all need to have the group write permission removed, but
   the execute permission added for group and others. This is to satisfy the require-
   ments of `suexec`, which is the mechanism by which Apache executes Cricket's pro-
   grams as the user `cricket`.

   > The complete documentation for `suexec` is included with the online `Apache`
   > documentation on your hard disk.
   >
   > Okay, it seems nobody understands that, so here's the recipe:
   >
   > ```
   > $ chmod go=x ∼
   > $ chmod go=rx ∼/cricket ∼/public_html ∼/public_html/cricket
   > ```
   > The permissions should look something like this:
   > ```
   > $ ls -ldL ∼ ∼/cricket ∼/public_html ∼/public_html/cricket
   > drwx--x--x   7 cricket  cricket      4096 Dec 18 13:44 /home2/cricket
   > drwxr-xr-x   7 cricket  cricket      4096 Dec 11 17:21 /home2/cricket/cricket
   > drwxr-xr-x   2 cricket  cricket      4096 Dec 11 17:55 /home2/cricket/public_html
   > drwxr-xr-x   2 cricket  cricket      4096 Dec 11 17:55 /home2/cricket/public_html/cricket
   > ```
   > You could remove write permission for group from all the files like this:

```
$ chmod -R g-w ~
```

which will remove the write permission for group of all files below ~/public_html.

9. You can use the `serviceconf` program to start Apache's `httpd` service in runlevels 3, 4 and 5.

   Alternatively you can use the same method to start and maintain the Apache service as with `nscd` given on page 15.

   ```
   $ sudo service httpd start
   $ sudo chkconfig --level 345 httpd on
   ```

Testing, and using the log files:

1. Now open a web browser, and go to `http://localhost/~cricket/cricket/grapher.cgi`

2. If you see a server error message:

   (a) View the log files for Apache to see what is happening. I suggest change the group ownership of the log file directory to you, so that you can see the files without being root:

   ```
   $ ls -ld /var/log/httpd
   drwx------    2 root     root             4096 Oct 31 17:52 /var/log/httpd
   $ sudo chgrp nicku /var/log/httpd
   $ sudo chmod g+rx /var/log/httpd
   $ ls -ld /var/log/httpd
   drwxr-x---    2 root     nicku            4096 Oct 31 17:52 /var/log/httpd
   ```

   Now change to the web server's log directory and examine the logs:

   ```
   $ cd /var/log/httpd
   $ ls -ltr
   $ tail -f error_log
   ```

   (b) You may also want to open two more windows, and leave them open, running `tail -f` on `suexec_log` and `access_log`. In particular, monitoring `suexec_log` will let you know about permission problems when accessing the programs.

   (c) After modifying Apache's configuration, tell Apache to reload it:

   ```
   $ sudo service httpd graceful
   ```

3. If you see the Cricket page, click on servers, then load, and you should see the graphs of system load there.

2   Adding a new set of Graphs and Targets for Cricket to Monitor

Now this is where you use your knowledge of SNMP. Refer to the page *Setting up New Devices in Cricket*, at `http://cricket.sourceforge.net/support/doc/new-devices`.

`html`, or `http://localhost/~cricket/doc/new-devices.html` if you make the link I described at the end of section 1.7 on page 7. There is a useful set of configurations in `http://www.gnac.com/techinfo/cricket_contrib/index.html`. The reference for the configuration of Cricket is at `http://cricket.sourceforge.net/support/doc/reference.html`.

Cricket has a very active mailing list for users, about ten posts per day. I had an perfect solution to my problem in less than two hours. The subscription information is available at `http://cricket.sourceforge.net/support/`.

2.1 Cricket's Configuration Tree

The directory $\sim$/`cricket-config` contains a directory tree. This directory tree contains the configuration of cricket. It is called a "config tree" because the configuration files at the higher levels provide default values, and some of these values are overridden by the configuration files at the lower levels.

Every directory in $\sim$/`cricket-config` contains a file called `Defaults`.

The top level configuration file, $\sim$/`cricket-config/Defaults`, contains a very large number of useful definitions. In particular, it is set up so that all you need to plot network traffic is in place. The OIDs are provided for all the useful items in the Mib-2 Interfaces table. Look at this file; you will see that the default graphs are graphs of network traffic in and out of a network interface.

Each chunk of the file begins with a word such as "`target`" or "OID". After that word, they are different, but you will see many things such as
`rrd-datafile = %dataDir%/%auto-target-name%.rrd`.
The percent signs indicate a variable that is expanded somewhere else in the configuration tree. Many of these variables have reasonable defaults that are defined at the top level of the tree. You only need to fill in the values for some (such as the hostname of the target) further down in the config tree.

2.1.1 A Closer Look at the Config Tree

The configuration is divided into *dictionaries*. Below some dictionaries, there are *tags* and *values*, arranged like this:

$dictionary - name$
    $tag1$ = $value1$
    $tag2$ = $value2$
    $\ldots$

These are the dictionaries available:

target A device that we monitor is a target. There is one RRD file for each target.

datasource A Data source is one line on a graph. One RRD can have many datasources in it. All datasources in one RRD need to be updated by the collector at once.

targetType Determines what kinds of data sources there are for the target with this targetType. The collector uses the targetType to decide what data to fetch, and how to fetch it.

graph Determines how individual data sources are graphed.

color Maps colour names to HTML colours. Probably no need to change this.

oid Maps an OID to a name

html determines what HTML goes into each page

event draws vertical bars on the graph to indicate events, such as the end of one day and the start of the next. No need to change this.

rra Holds parameters used by Cricket when creating a new RRD file. No need to change this.

2.2 The servers subtree

Let us consider how the configuration for `/servers` works

I decided to monitor the system load average and the available free memory.

There are five *data sources* here, three for the load averages, one for available swap, and one for free memory. These are listed under the `targetType` dictionary name.

2.2.1 Variable Expansion

When the collector gets the system load data from `ictlab`, it uses an SNMP URL something like this:

`snmp://public@nicku.org:161:2.0:5:1.0:1/1.3.6.1.4.1.2021.10.1.3.1`

But where did all that come from?

Let's look in the Target dictionary in the top level ~/`cricket-config/Defaults` file:

```
Target    --default--
    snmp-host           = %auto-target-name%
    snmp-community      = public
    snmp-port           = 161
    snmp-timeout        = 2.0
    snmp-retries        = 5
    snmp-backoff        = 1.0
    snmp-version        = 1
    snmp                = %snmp-community%@%snmp-host%:%snmp-port%:
%snmp-timeout%:%snmp-retries%:%snmp-backoff%:%snmp-version%
```

So what about `%auto-target-name%`? This is automatically set by Cricket to the name of the target. So let's see part of the file ~/`cricket-config/servers/Targets`:

```
target  nicku.org
    target-type =   linux-machine
```

Finally, let's see how the OID and the data source is defined in ~/`cricket-config` `/servers/Defaults`:

```
OID     laLoad1min  1.3.6.1.4.1.2021.10.1.3.1
datasource laLoad1min  ds-source   =   snmp://%snmp%/laLoad1min
```

So default values mostly come from the top level of the configuration tree and are inherited below. You can override any default defined at the top level with values you define further down.

2.3   Tables and Instance Numbers

We examined the Interfaces table of Mib-2 in some detail. Each row in the table corresponds to one network interface. How do we use SNMP to `get` information on a particular network interface? Let's look at `ifInOctets`, in the MIB file `/usr/share/snmp/mibs` `/RFC1213-MIB.txt`. The definition of `ifInOctets` is:

```
ifInOctets OBJECT-TYPE
    SYNTAX   Counter
    ACCESS   read-only
    STATUS   mandatory
    DESCRIPTION
            "The total number of octets received on the
            interface, including framing characters."
    ::=   ifEntry 10
```

Let's get the OID of this table entry:

```
$ snmptranslate -IR -Of ifInOctets
.1.3.6.1.2.1.2.2.1.10
```

Good, now let's see the value on `ictlab`:

```
$ snmpwalk -c public ictlab .1.3.6.1.2.1.2.2.1.10
interfaces.ifTable.ifEntry.ifInOctets.1 = Counter32: 752366405
interfaces.ifTable.ifEntry.ifInOctets.2 = Counter32: 791891118
```

Hey, there are two of them! What are they? Well, Mib-2 provides a description of the interface in `ifDescr`; let's see what that says:

```
$ snmpwalk -c public ictlab ifDescr
interfaces.ifTable.ifEntry.ifDescr.1 = lo
interfaces.ifTable.ifEntry.ifDescr.2 = eth0
```

Okay, so the entries in the `ifTable` with a `.1` at the end on `ictlab` relate to the loopback interface, while the entries with a `.2` at the end are about the Ethernet interface.

This number at the end of the OID is called an *instance number*, and is the way that SNMP selects items from a table. Notice that the instance numbers begin at 1 for table entries. What about for scalars?

```
$ snmpwalk -c public ictlab sysUpTime
system.sysUpTime.0 = Timeticks: (103614809) 11 days, 23:49:08.09
```

The instance number for a scalar is always zero.

Typically, the loopback interface is the first entry in the table, and other interfaces follow. However, SNMP does not prevent a router from changing the order of the interfaces. This could be a real problem; after rebooting a router, you may find that the graph for your subnet is now collecting data for a different subnet! Luckily, Cricket has a solution to this called *instance mapping*, described in `http://cricket.sourceforge.net/support/` `doc/inst-mapping.html`.

It is also possible to directly provide a value for the instance number for a target to graph, instead of using instance mapping. For example, you could set

```
target hostname-of-server-or-router
    inst = 2
```

to draw graphs of interface two of the machine called *hostname-of-server-or-router*.

2.3.1 The `listInterfaces` Utility

A very nice Cricket utility is provided to automatically find the names of all the interfaces on a server or router, then map each name to an instance number. This is described in `http://cricket.sourceforge.net/support/doc/beginner.html`. The name of the utility is `listInterfaces`. If `main-router` is the hostname of a server or router that you want to plot statistics of the network interfaces from, you can do:

```
$ ~/cricket/util/listInterfaces main-router > interfaces
```

The file `interfaces` can then be used in the `~/cricket/cricket-config/router-interfaces` directory to specify how to plot each interface. All that is left is simply to put an entry in `subtree-sets` so that the collector will collect the data from the server or router, and then run the `compile` script. Cricket provides enormous flexibility once it is set up.

2.4 Monitoring High-End Cisco Switches: the `genCatConfig` Utility

To monitor high-end Cisco Catalyst™ switches, such as our 6509 (but not Cisco Catalyst 3500XL switches), there is a tool available for automatically generating the configuration for Cricket. It is currently available from `http://www.certaintysolutions.com/tech-advice/cricket-contrib/`; you can download the program from that page. Note that all these programs are available from our own server `ictlab`, by many protocols, including `http://nicku.org/ftp/snmp/cricket-contrib`, and by NFS from `ictlab:/var/ftp/pub/snmp`.

Read the file `~/cricket/util/README.genCatConfig` for the full documentation; this is just a short summary. To install it,

- become `cricket`, and

- untar the package into `/tmp`. Then

- copy the directory `/tmp/sample-config/catalysts` and its contents into `~/cricket-config`.

- Copy the contents of `/tmp/util` into `~/cricket/util`.

- Edit the two executables and change the first line in each file from

  ```
  #!/usr/local/bin/perl
  ```

  to

  ```
  #! /usr/bin/perl
  ```

- Make a host table entry for the router in `/etc/hosts`.

- Change directory to `~/cricket-config/catalysts`

- Then run

  ```
  $ ~/cricket/util/genCatConfig -2 -v -C ⟨community-string⟩ ⟨router-hostname⟩
  ```

replacing ⟨*router-hostname*⟩ with the host name of the Catalyst router that you put into the `/etc/hosts` file, and ⟨*community-string*⟩ with its read-only community string.

- Add `/catalysts/`***router-hostname*** to Cricket's `subtree-sets` file.

- Run ∼`/cricket/compile`. It should now collect data. Test it using the collector.

2.5   Monitoring 35xx Cisco Switches and Routers

According to the genCatConfig documentation:

> Some Catalysts (e.g. 2900XL and 3500XL switches) run router IOS and respond to a different set of MIBs. They should be handled as routers, not switches. (see `genRtrConfig`).

So download `genRtrConfig` from the same place as `genCatConfig` above. The installation process is similar, but not identical. In particular, the Cisco 3500XL series does not support the 64 bit counters, so we have to use SNMP version 1! Here is how to do it:

1. as `cricket`, untar the package into `/tmp`:

   ```
   $ cd /tmp
   $ tar xvzf /home/nfs/snmp/genRtrConfig-1.4.tar.gz
   ```

2. Copy the directory `/tmp/sample-config/cisco-routers` and its contents to ∼`/cricket-config`:

   ```
   $ cp -a /tmp/sample-config/cisco-routers ~/cricket-config
   ```

3. Copy the contents of `/tmp/util` to ∼`/cricket/util`:

   ```
   $ cp -a /tmp/util/* ~/cricket/util
   ```

4. Edit ∼`/cricket/util/genRtrConfig` and change the first line from `#!/usr/local/bin/perl` to `#!/usr/bin/perl`

5. Make a host table entry for the switch in `/etc/hosts`.

6. Change directory to ∼`/cricket-config/cisco-routers` and run the program `genRtrConfig` like this:

   ```
   $ ~/cricket/util/genRtrConfig -C ⟨community-string⟩ --chassis \
   --ciscoint -v ⟨hostname-of-switch⟩
   ```

7. Add `/cisco-routers/`⟨*switch-hostname*⟩ to Cricket's `subtree-sets` file.

8. Run ∼`/cricket/compile`. It should now collect data. Test it using the collector.

2.6 Running the Name Service Caching Daemon

In the past, some people experienced problems unless the name service caching daemon
(`nscd`) is running on their machines. This service is very important to reduce the load
on our LDAP server, and will improve the performance of your machine also, and should
always be turned on. To turn on `nscd`:

1. First, start it now:

   ```
   $ sudo service nscd start
   ```

2. Now make sure that it always starts as a service when the computer boots:

   ```
   $ sudo chkconfig --level 345 nscd on
   ```

   This turns the `nscd` service on in runlevels 3, 4 and 5.

3. Finally, verify that the service is configured correctly:

   ```
   $ chkconfig nscd --list
   nscd              0:off   1:off   2:off   3:on    4:on    5:on    6:off
   ```

   If you do not see "`on`" for runlevels 3, 4 and 5, something is wrong (perhaps a typing
   mistake?)

2.7 Installing the Red Hat Updates

You will also benefit by installing the Red Hat Linux updates. These are in the di-
rectory `/home/nfs/redhat-8.0/updates` on `ictlab`, assuming `ictlab:/var/ftp/pub` is
mounted on `/home/nfs`. You can install them like this:

```
$ cd /home/nfs/redhat-8.0/updates/i686
$ sudo rpm -Fhv *.rpm
$ cd ../i386
$ sudo rpm -Fhv *.rpm ../noarch/*.rpm
```