Introduction to MRTG

Brad Marshall <brad.marshall@member.sage-au.org.au>

May 10, 2004

1 Introduction

MRTG stands for Multi Router Traffic Grapher, and can be used to graph much, much more than traffic from routers. This article will cover how to configure MRTG to graph data locally from a machine, as well as how to obtain data remotely over the network using SSH and graph it. It will not cover how to configure SNMP, or any of the details behind how MRTG works.

2 Installation

To install MRTG, download the source from http://www.mrtg.org/ and untar it. You will require gcc, perl, gd, libpng and zlib installed before compiling it.

Once all the dependancies are installed, cd into the untarred source directory, and run something like following. To see more options run configure with a –help argument.

```
$ ./configure --prefix=/usr/local/mrtg
```

Once the configuration has been done, run the following:

```
$ make
$ sudo make install
```

This should install MRTG into the location you specified in the prefix.

3 Scripting

The form that is needed by MRTG to graph from looks something like:

```
input
output
uptime
hostname
```

So any values that you can provide in this form can be graphed. Note that MRTG can only graph integers, so it may be required to multiple the values by a factor.

The following perl script shows how to graph both used and free memory on a Linux machine.

```
#!/usr/bin/perl

$machine = ‘/bin/hostname‘;
chomp($machine);
$mem = ‘/usr/bin/free | grep Mem‘;
$uptime = ‘/usr/bin/uptime‘;

if ($mem =˜ /ˆMem:\s*(\d*)\s*(\d*)\s*(\d*)/) {
        $tot = $1;
        $used = $2;
        $free = $3;
}

if ($uptime =˜ /up (.*),  \d* users?,/) {
        $up = $1;
}
print "$used\n";
print "$free\n";
print "$up\n";
print "$machine\n";
```

4 Configuration

There are only a few important global configuration settings for MRTG, which are as follows:

```
Htmldir: /path/to/mrtg/html
Imagedir: /path/to/mrtg/html/images
Logdir: /path/to/mrtg/html/logs
```

4.1 Local Monitoring

The next step is to add a section to mrtg.cfg, as follows:

```
Target[mem]: ‘/path/to/mrtg/scripts/mem.pl‘
Options[mem]: gauge,noinfo, nopercent, growright, nobanner
Title[mem]: Memory Usage
MaxBytes[mem]: 256000
YLegend[mem]: Memory Usage
ShortLegend[mem]: Memory Usage
```

```
LegendO[mem]: Memory Free:
Legend2[mem]: Memory Free
LegendI[mem]: Memory Used:
Legend1[mem]: Memory Used
PageTop[mem]: Memory Usage
```

It is also possible to graph one value by setting the script to return 0 (or anything else) and configuring the section in mrtg like the following. Notice the missing LegendI and Legend1 as well as the addition of noi in the options section.

```
Target[uptime]: ‘/path/to/mrtg/scripts/uptime.pl‘
Options[uptime]: gauge,noinfo, nopercent, \
                growright, nobanner, noi
Title[uptime]: Load
MaxBytes[uptime]: 1000
YLegend[uptime]: load
ShortLegend[uptime]: load
LegendO[uptime]: Uptime:
Legend2[uptime]: Load
PageTop[uptime]: Load
```

4.2 Remote Monitoring

Before setting up the configuration for remote monitoring, it is important to configure the connection method. This example shows how to use ssh with a passphrase-less key to allow the connection.

On the server running MRTG, create a key as the user who is running the processes as follows:

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/srcuser/.ssh/id\_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/srcuser/.ssh/id\_rsa.
Your public key has been saved in /home/srcuser/.ssh/id\_rsa.pub.
The key fingerprint is:
a1:b8:35:0d:80:1b:ab:eb:ca:00:f3:78:bb:83:ae:cd srcuser@host
```

Then copy the contents of srcuser1/.ssh/id_rsa.pub and append it on the remote host in destuser/.ssh/authorized_keys. Once this is has been done, it should then be possible to ssh from the local server to the remote one as the specified user without a passphrase. It is important to do this manually at least once and accept the ssh host key.

Next, create the scripts on the remote host as per usual, and place them in the desired directory. Then create the configuration on the local server as usual, with the following change:

```
Target[dest-conn]: ‘ssh destuser@host2 \
                    /path/to/mrtg/bin/script.pl‘
```

This specifies that the script to run will ssh to the remote host and run the script and return the information to MRTG on the local server.

5 Cron Setup

Once monitoring is setup for all the values desired, set up a cronjob like the following that runs at the desired interval - in this case, every 5 minutes.

```
*/5 * * * *  /path/to/mrtg/bin/mrtg \
                /path/to/mrtg/etc/mrtg.cfg \
                --logging /var/log/mrtg.log
```

6 Conclusion

Using the technique described, it is possible to graph a wide variety of properties of a server without the complexity of setting up and maintaining SNMP. It is important to monitor hosts to get a grasp of what the machine is doing as well as to gauge when upgrades are required.