gkrellm

I've been using gkrellm for quite a while now to have a contantly updated view of my system. By placing it in the upper right hand corner of my desktop, I can occasionally glance over there and see if there's a spike in network activity, disk usage, processor usage, or many others. To the left, there's a snapshot of a running gkrellm, courtesy of the gkrellm site.

At first glance, it might seem fun, but useless. Actually, it's a terribly useful monitoring tool; once you've watched it for a while, you get a very good sense of what load is normal for your system. When you see spikes outside of the normal range, it might be because you're under attack, getting hit by a runaway process, being

used a spam relay, or someone 5 cubicles down is running nmap on a remote network without permission. (Boy, was he surprised when I showed up asking what was going on within 30 seconds of when he started the nmap run :−).

As you no doubt remember from my SSH Techniques article, I can run an X application over the network. By ssh'ing to a machine I wish to monitor and typing

```
gkrelmm &
```

in the ssh session, I can run the monitoring tool on the remote box so it collects statistics on that box, but pull the display back to my laptop.

There's a problem with this approach, though − bandwidth. :−(I'm still working on a modem line, and despite a bunch of networking tricks to load balance over multiple modems, compress traffic, route interactive traffic before bulk, and so on, I'm still working off a slow modem line or two. Modem lines carry X windows displays very slowly − so slow that even running a single gkrellm over ssh would take a number of seconds to display even a single image, and completly tie up the line.

Gkrellm is now at version 2.0, which addresses this problem, and quite well. In addition to the old mode of running "gkrellm "which displays the status of the machine on which it's running, there's now a client−server approach. Here's how it works.

The server

On all the machines I want to monitor, I install the gkrellm−server package (you may need to get the glib2 package from your distribution to complete the install) and run:

```
nohup gkrellmd −u 3 −P 19150 −m 2 −a 127.0.0.1 −a 12.13.14.15 &
```

This tells the server to provide 3 updates a second, listen on port 19150 for incoming connections, allow up to 2 connections on that port from people running the gkrellm display, and only allow connctions from localhost or 12.13.14.15 (the IP address assigned to the ethernet card). I only need to run this once when the system starts up, so this might be a good candidate for running in /etc/rc.d/rc.local; after running

```
adduser gkrellmd
```

once, add this to /etc/rc.d/rc.local and run it once by hand:

```
nohup su gkrellmd −c '/bin/nice /usr/bin/gkrellmd −u 3 −m 2 −P 19150 −a 127.0.0.1 −a 12.13.14.15
```

The gkrellm server just sits there, quietly collecting system statistics until someone starts up the display portion of the tool on an X windows display. First, lets encrypt the system statistics so nobody can see them on the Internet:

Encrypting the traffic

```
ssh −L 19151:12.13.14.15:19150 12.13.14.15
```

We'll use port forwarding with an ssh session so that whenever the gkrellm display connects to locahost port 19151, the connection will be sent to 19150 on the machine I want to monitor, where the waiting gkrellm server will have lots of statistics to send back. Lets do that, in a new terminal (if you haven't installed it yet on

the display system, pull it down and install it, possibly along with other support libraries from your distribution):

The display

```
gkrellm −f −s 127.0.0.1 −P 19151 &
```

My gkrellm display starts on my laptop, connects to the gkrellm server through ssh (so nobody can sniff my system stats), and shows a constant update of what's going on on the remote system.

You can do this for as many machines as you'd like; each one requires a gkrellmd running on the machine to be monitored, an ssh connection to tunnel the traffic, and a gkrellm display on the local machine to show the statistics. Luckily, if you're monitoring multiple machines on the same remote network, you can use a single ssh connection to carry their traffic if you don't mind the stats flying by unencrypted on that remote network cable (they're still encrypted over the Internet):

```
ssh −L 19151:12.13.14.15:19150 −L 19152:12.13.14.16:19150 −L 19153:12.13.14.17:19150 12.13.14.1
```

Here, I ssh to the gateway machine on that lan (12.13.14.1) and ask it to send traffic to .15, .16, and .17 when I connect to localhost ports 19151, 19152, and 19153, respectively. Assuming I have already started gkrellmd's with:

```
nohup gkrellmd −u 3 −P 19150 −m 2 −a 127.0.0.1 −a 12.13.14.1 &
```

(note the new IP address we're allowing to connect; you may need to try both the outside and inside IP addresses of the gateway), I can now start monitoring all three machines at once, and start one to watch my local machine too:

```
gkrellm −f −s 127.0.0.1 −P 19151 &gkrellm −f −s 127.0.0.1 −P 19152 &gkrellm −f −s 127.0.0.1 −P 19
```

Customization

You may have no interest in seeing some of the statistics shown by gkrellm, or may wish to customize the output; no problem. Right click on the system name of the machine at the top. You'll be placed in a menu system where you can make all kinds of customzation choices. Those choices are stored individually for each system, so you can focus on particular services for different machines.

Plugins

In addition to the monitors included in the base package, gkrellm allows other authors to provide plugins to watch and/or control other components of the system. Take a look at these too; people get very innovative on what they want to watch.

William is an Open−Source developer, enthusiast, and advocate from New Hampshire, USA. His day job at SANS pays him to work on network security and Linux projects.