BGP Tool Hints

Here are some docs about how to build the various BGP tools out there.

Building BGPDUMP

BGPDUMP is a tool and set of libraries maintained by the RIPE NCC which reads MRT (Multi-Threaded Routing Toolkit) files and outputs them in human readable format (ie ASCII).

First off, we clone the git repo locally. Go to the place where you normally do your development work (for example, **/usr/local/src**), and:

```
git clone https://github.com/RIPE-NCC/bgpdump.git
```

Next make sure we have the libraries and packages we need installed:

```
sudo apt install libz-dev libbz2-dev autoheader
```

and then go into the **bgpdump** folder and build the **Makefile**:

```
cd bgpdump
./bootstrap.sh
```

Once this is done we can then build **bgpdump**:

```
make
```

and then install it:

```
sudo make install
```

which will put **bgpdump** into **/usr/local/bin**.

You can now take your MRT files, eg as produce by RouteViews and convert them into plain text for further analysis.

Building a Looking Glass

These instructions cover how to put together the Cougar Looking Glass code. Why choose this one? Well, it's there, it seems to work fine, and installs nicely on Ubuntu 22.04. Even though the code itself hasn't been touched since 2009!

First off, grab the few files you need off the repo - you could clone it, but there are only a few files, and they haven't been changed in a while. Put them all in **/usr/local/src/LG**. All the directions below assume that the working directory is here.

Assuming a basic install of Ubuntu 22.04, we need to add a few packages, including a compiler.

```
sudo apt install apache2 make gcc libexpat1-dev
```

Next we need to install a few extra perl modules to support the looking glass. Assuming we don't have CPAN already set up, we need to do:

```
cpan App::cpanminus
```

This installs the basic CPAN environment to make adding in modules simpler. Once the CPAN environment is in place, you then need to run:

```
cpan install XML::Parser
cpan install Net::Telnet
```

which will install the XML parser and telnet support for Perl. This should all complete successfully - you'll see these two packages and all the supporting modules being automatically installed.

With the Apache2 webserver installed in its default configuration, we now need to create two folders for the Looking Glass components.

```
sudo mkdir -p /var/www/looking-glass/lg
```

And then copy the required components into place.

```
sudo cp -p favicon.ico index.html lg.cgi lg.conf /var/www/looking-glass/lg/
```

Next we need to create the Apache site configuration:

```
cd /etc/apache2/sites-available
```

and create a new config file with the URL of your site, for example, **lg.my.site.conf** with the following contents:

```
<VirtualHost *:80>
  ServerName  lg.my.site

  DocumentRoot /var/www/looking-glass/lg/

  Alias /lg/favicon.ico /var/www/looking-glass/lg/favicon.ico
  Alias /libraries /var/www/looking-glass/libraries
  ScriptAlias /lg /var/www/looking-glass/lg/lg.cgi

  <Directory "/var/www/looking-glass/lg">
    AllowOverride None
    Options +ExecCGI
    AddHandler cgi-script .cgi .pl
    Require all granted
  </Directory>
```

```
</VirtualHost>
```

We need to turn on support for CGI:

```
sudo a2enmod cgid
```

Then we need to enable the site:

```
sudo a2ensite lg.my.site
```

and then restart Apache2 for it to go live:

```
sudo systemctl restart apache2
```

Back to Home page

From:
https://www.bgp4all.com/pfs/ - **Philip Smith's Internet Development Site**

Permanent link:
https://www.bgp4all.com/pfs/hints/misc?rev=1692390779

Last update: **2023/08/18 20:32**