Installing and Configuring BGPmon:
Quickstart Guide Version 1.2

The BGP monitoring tool (BGPmon) collects BGP data by 1) peering with BGP routers, 2) indirectly gathering data from other collectors using the MRT format, and 3) chaining to share data between BGPmon instances. All resulting data is provided in real-time via simple TCP connections. The latest version of BGPmon can be downloaded from:
http://bgpmon.netsec.colostate.edu/index.php/download

All BGPmon configuration is done via a command line interface. The interface is similar to that found on widely used routers. If you are familiar with the commands for Cisco or Juniper, generally the same commands should work on BGPmon. For example, the commands you use to configure a BGP peering session on a Cisco router should also work for BGPmon.

This guide will help you install and start BGPmon (Sections 1 and 2), login, change the default passwords, and save a configuration file (Section3), and configure BGPmon to collect data (Section 4). Once BGPmon is configured, you can check the resulting data and provide access to users (Section 5). Finally, you can optionally change the default port settings and configure access control lists (Sections 6 and 7). This guide was tested on Ubuntu 11.04 and Fedora 14.0.

A more comprehensive guide, the Administrator's Reference Manual, lists all commands and can be found in the BGPmon download package in 'bgpmon/doc/arm/'. The ARM can also be downloaded from:
http://bgpmon.netsec.colostate.edu/index.php/documentation.

1 Install BGPmon

```
> ./configure
> make
> sudo make install
```

=> BGPmon requires the XML dev library:
– Ubuntu: apt-get install libxml2-dev
– Fedora: yum install libxml2-devel

2 Launch BGPmon

```
> sudo service bgpmon start
```

By default BGPmon logs messages to /var/log/messages. For information on starting the BGPmon server directly and command line arguments, refer to ARM section 2.5.1.

3 Login and Configure Command Line Access

By default, BGPmon listens for command line login on the loopback address, port 50,000. You will login and be prompted for a password. Follow the steps below to login, change the default passwords, and save your configuration file.

BGPmon supports three modes of operation; *Unprivileged (or User) Mode*, *Privileged Mode*, and *Configuration Mode*. Available commands will vary depending on the mode. To see the list of available BGPmon commands, enter "?" for a summary of available commands. Enter "??" for an expanded list of commands. Command arguments can also be displayed by entering "?" after an available command.

3.1 Login and Enter Unprivileged Mode

```
> telnet localhost 50000
Password:
```

'BGPmon' is the default password to enter *unprivileged mode*.

Unprivileged mode allows you to view statistics, show routing tables, and generally view (but not change) configuration parameters. In order to change the configuration settings, switch to *privileged mode* and then enter *configuration mode*.

3.2 Enter privileged mode

```
host> enable
enable password:
```

'BGPmon' is the default password to enter *privileged mode*.

From *privileged mode* you have access to additional data and can change settings by entering *configuration mode*.

3.3 Enter configuration mode

```
host# configure
host(config)#
```

3.4 Change unprivileged mode and privileged mode passwords

```
host(config)# password <new password>
host(config)# enable password <new password>
```

Changes unprivileged mode and privileged mode passwords, respectively.

3.5 Save changes

```
host(config)# exit
host# copy running-config startup-config
```

Any configuration changes are made to the running configuration stored in memory and will be lost after a restart. To save your changes, copy the running configuration to the startup configuration.
Your passwords have now been changed. If you forget your new passwords, refer to BGPmon login recovery in Section 3.1.2 of the Administrators Reference Manual (ARM).

4 Configure Data Input

BGPmon collects data by peering with BGP routers (Section 4.2), chaining to share data between BGPmon instances (Section 4.3), and indirectly gathering data from other collectors using the MRT format (Section 4.4). These can be used simultaneously and in any combination.

4.1 Peer with IPv4 router

```
host(config)# router bgp <AS number>
host(config-router)# neighbor <IPv4 address> remote as <Remote-AS-number>
host(config-router)# exit
host(config)# exit
host# copy running-config startup-config
```

=> Note: you must be *configuration mode* to change your configuration. See Section 3.
This configuration enables BGP peering session with an IPv4 router. You must also configure the router to peer with BGPmon. See your router documentation and configure BGPmon as you would configure any other peer on the router.
BGPmon supports a full range of BGP capabilities and peering parameters, refer to ARM sections 3.6.3 and 3.6.4. BGPmon also supports the creation of peer-groups, refer to ARM section 3.6.2.

4.2 Peer with IPv6 router

=> Note: this is a workaround for a known issue in version 7.2.2. The procedure for peering with an IPv6 router will change dramatically in the future.

```
host(config)# router bgp <AS number>
host(config-router)# neighbor <IPv6 address> remote as <Remote-AS-number>
host(config-router)# neighbor <IPv6 address> local bgpid <Router-BGPID>
host(config-router)# exit
host(config)# exit
host# copy running-config startup-config
host# exit
host> exit
```

Shutdown BGPmon server:

```
> sudo service bgpmon stop
```

Open */usr/local/etc/bgpmon_config.txt* with your favourite text editor. Under the <*PEER*> tag that contains your IPv6 router information, add the following line:

```
<MONITOR_IP_ADDR>IPv6_ANY</MONITOR_IP_ADDR>
```

Save configuration and restart BGPmon server:

```
> sudo service bgpmon start
```

This configuration enables BGP peering session with an IPv6 router.

4.3 Setup a chain

```
host(config)# chain <address> <[rib:port] [update:port]>
host(config)# no chain <address> <[rib:port] [update:port]>
host(config)# exit
host# copy running-config startup-config
```

=> Note you must be *configuration mode* to change your configuration. See Section 3.
A chain is a connection between BGPmon instances that allows one BGPmon to receive the data output of another BGPmon. The *chain* command instructs BGPmon to retrieve data from the BGPmon instance at the configured address and port. The *no chain* command deletes a chain. No configuration is required on the BGPmon providing the data. You can configuration access control lists (Section 7) to restrict who can establish chains to your system.
For more chaining information, refer to ARM section 3.5.

4.4 Peer with MRT collector

```
host(config)# mrt-listener enable
or
host(config)# mrt-listener disable
host(config)# exit
host# copy running-config startup-config
```

=> Note you must be *configuration mode* to change your configuration. See Section 3.
By default, BGPmon has enabled an MRT listener on port 50,003. A number of existing BGP data collectors such as RIPE RIS provide data in MRT format. A collector can simply send MRT data to BGPmon on port 50,003 and the data will incorporated into the BGPmon output. Access control lists (ACLs) can be used to restrict who can feed MRT data into BGPmon. Section 7 discusses how to configure access control lists.

If you don't plan to supply MRT data, it is recommend you disable the MRT listener.

5 Configuration Results and Data Output

5.1 Showing Data From the Command Line

```
host> show bgp neighbor
host> show chains
host> show mrt clients
host> show bgp routes
```

Once you have configured BGPmon, you can view the current peers, chains, MRT clients, and resulting routes using the *show* commands above. The show command is available in both *unprivileged* and *privileged* modes.

5.2 Providing Data to Users

```
> telnet <bgpmonhost> 50001
or
> telnet <bgpmonhost> 50002
```

All BGP updates received from a peer, chain, or MRT collector are converted into an XML format and made available on port 50001. Users can access the BGP update data by simply opening a TCP connection to your BGPmon host on port 50001. You can configuration access control lists (Section 7) to restrict who can obtain BGP updates from your system.

BGPmon maintains a RIB table for each directly connected peer and each indirect MRT peer. The RIB tables are announced periodically on port 50002. BGPmon does not store routing tables from chains, but does pass through the RIB tables periodically reported by the downstream chain. Users can access the routing table data by simply opening a TCP connection to your BGPmon host on port 5000w. You can configuration access control lists (Section 7) to restrict who can obtain routing tables from your system.

6 Optional: Change Default Ports

```
host(config)# client-listener update port <new-port>
host(config)# client-listener rib port <new-port>
host(config)# login-listener port <new-port>
host(config)# mrt-listener port <new-port>
host(config)# exit
host# copy running-config startup-config
```

=> Note you must be *configuration mode* to change your configuration. See Section 3.
By default, BGPmon listens for login access on port 50000, sends BGP update messages on port 50001, sends BGP RIB tables on port 50002, and listens for MRT connections on port 50003. The commands listed above allow the administrator to change any of these port settings.

7 Optional: Configuring Access Control Lists

```
host(config)# acl [aclname]
Created ACL, now editing: <aclname>
host(config-acl)# permit <address> <subnet mask> <index num>
host(config-acl)# deny <address> <subnet mask> <index num>
host(config-acl)# show acl
host(config-acl)# no <index num>
host(config-acl)# exit
host(config)# exit
host# copy running-config startup-config
```

=> Note you must be *configuration mode* to change your configuration. See Section 3.
An administrator can limit access to the system via Access Control Lists (ACLs). The above commands create an ACL called aclname. The permit and deny commands add to the ACL. The show command displays the resulting ACL. Finally, the no command removes rule index num from the ACL.

7.1 Assigning an ACL to an Interface

```
host# configure
host(config)# login-listener acl <login-acl>
host(config)# client-listener update acl <client-up-acl>
host(config)# client-listener rib acl <client-acl>
host(config)# mrt-listener acl <mrt-acl>
host(config)# exit
host# copy running-config startup-config
```

ACLs can be applied to the command line interface, update clients, RIB clients, and MRT. The above commands assign admin-acl, client-up-acl, client-rib-acl, and mrt-acl to the command line interface, BGP update output, rib table output, and MRT listener, respectively.

Note that to a BGPmon instance, a downstream client and downstream chain are indistinguishable. Thus assigning an ACL to the update and RIB table output also limits who can obtain a chain from this BGPmon instance.

8 Shutting Down BGPmon

```
> telnet localhost 50000
Password:
host> enable
Password:
host# shutdown
```

BGPmon can be shut down from the command-line interface. Recall that BGPmon stores the configuration in memory. If you want your configuration changes to be saved, be sure to copy the running configuration into the saved configuration as discussed in the earlier sections.