LAB GUIDE

BGP - Internet table injector

VIRTUAL AOS-CX

Important! **For this specific lab guide, it is recommended to run the CX OVA directly on ESXi in order to avoid EVE-NG or GNS3 overhead as the routing process is CPU intensive.**

OBJECTIVE

This lab guide explains how to set-up BGP session to receive the full internet routes on AOS-CX OVA. This allows to practice on BGP understand the sizing requirement for ISP peering.

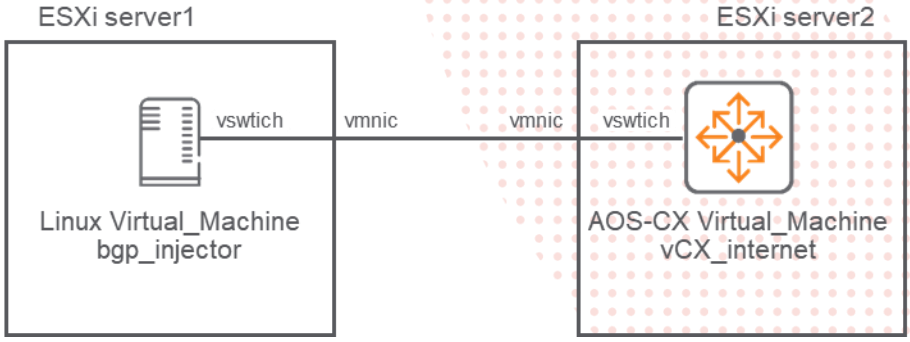
LAB SET-UP

Minimum OVA release: 10.04.3000.

Here is the proposed simple topology.

- Both Virtual Machines are hosted on an ESXi Server. This server could also be Hyper-V or KVM as well. It is strongly recommended to use a server with a powerful CPU (for instance: Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.60GHz)
- BGP_injector should be sized with 2 vCPUs and 4GB RAM.
- vCX_internet OVA should be sized with 4vCPUs and 16GB RAM

- Each VM could also be hosted on separate servers or powerful PCs using other virtualization technologies like VirtualBox. The best performance is expected on ESXi or KVM as overhead are limited.



BGP_INJECTOR SET-UP

Any Linux distribution can be used. Here it is proposed to use Ubuntu (Ubuntu 20.04.1 LTS was used).

The following steps are processed after a fresh installation of Ubuntu server.

Linux Update

```
hpearuba@bgp-injector:~$ sudo su
[sudo] password for hpearuba:
```

- apt upgrade

```
root@bgp-injector:/home/hpearuba# apt upgrade

Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  apport bcache-tools liblzma5 open-vm-tools python3-apport python3-distupgrade python3-problem-report
rsyslog sudo ubuntu-release-upgrader-core unattended-upgrades xz-utils
12 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 2,093 kB of archives.
After this operation, 145 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 rsyslog amd64 8.2001.0-1ubuntu1.1 [427
kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 liblzma5 amd64 5.2.4-1ubuntu1 [91.7 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 sudo amd64 1.8.31-1ubuntu1.1 [513 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 xz-utils amd64 5.2.4-1ubuntu1 [82.5 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 ubuntu-release-upgrader-core all
1:20.04.24 [23.7 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-distupgrade all 1:20.04.24 [103
kB]
```

- Reboot the VM

Install Required pakages

- apt install build-essential zlib1g-dev libbz2-dev autoconf

```
root@bgp-injector:/home/hpearuba# apt install build-essential zlib1g-dev libbz2-dev autoconf

Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu bzip2-doc cpp cpp-9 dpkg-dev fakeroot g++ g++-9 gcc
gcc-9 gcc-9-base
```

```
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1 libbinutils
libc-dev-bin libc6-dev libcc1-0
  libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl libfakeroot libfile-fcntllock-perl libgcc-9-dev libgomp1
libisl22 libitm1 liblsan0 libmpc3
  libquadmath0 libstdc++-9-dev libtsan0 libubsan1 linux-libc-dev make manpages-dev
Suggested packages:
  binutils-doc cpp-doc gcc-9-locales debian-keyring g++-multilib g++-9-multilib gcc-9-doc gcc-multilib
autoconf automake libtool flex bison
  gdb gcc-doc gcc-9-multilib glibc-doc bzr libstdc++-9-doc make-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential bzip2-doc cpp cpp-9 dpkg-dev fakeroot
g++ g++-9 gcc gcc-9 gcc-9-base
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1 libbinutils
libbz2-dev libc-dev-bin libc6-dev
  libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl libfakeroot libfile-fcntllock-perl libgcc-9-dev
libgomp1 libisl22 libitm1 liblsan0
  libmpc3 libquadmath0 libstdc++-9-dev libtsan0 libubsan1 linux-libc-dev make manpages-dev zlib1g-dev
0 upgraded, 44 newly installed, 0 to remove and 0 not upgraded.
Need to get 40.6 MB of archives.
After this operation, 176 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu focal/main amd64 binutils-common amd64 2.34-6ubuntu1 [207 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal/main amd64 libbinutils amd64 2.34-6ubuntu1 [474 kB]
…
[snip]
…
Setting up build-essential (12.8ubuntu1) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for install-info (6.7.0.dfsg.2-5) ...
```

Install BGP PERL modules from CPAN (Comprehensive Perl Archive Network – cpan.org)

The BGP PERL modules to install are detailed here: https://metacpan.org/release/Net-BGP

- Start cpan

```
root@bgp-injector:/home/hpearuba# cpan

Loading internal logger. Log::Log4perl recommended for better logging

CPAN.pm requires configuration, but most of it can be done automatically.
If you answer 'no' below, you will enter an interactive dialog for each
configuration option instead.

Would you like to configure as much as possible automatically? [yes]


Autoconfiguration complete.

commit: wrote '/root/.cpan/CPAN/MyConfig.pm'

You can re-run configuration any time with 'o conf init' in the CPAN shell
Terminal does not support AddHistory.

To fix enter>  install Term::ReadLine::Perl


cpan shell -- CPAN exploration and modules installation (v2.22)
Enter 'h' for help.

cpan[1]>
```

- Install Net::BGP

```
cpan[1]> install Net::BGP

Fetching with HTTP::Tiny:
http://www.cpan.org/authors/01mailrc.txt.gz
Reading '/root/.cpan/sources/authors/01mailrc.txt.gz'
```

```
.....................................................................DONE
Fetching with HTTP::Tiny:
http://www.cpan.org/modules/02packages.details.txt.gz
Reading '/root/.cpan/sources/modules/02packages.details.txt.gz'
  Database was generated on Tue, 25 Aug 2020 12:55:58 GMT
  HTTP::Date not available
.............
  New CPAN.pm version (v2.28) available.
  [Currently running version is v2.22]
  You might want to try
    install CPAN
    reload cpan
  to both upgrade CPAN.pm and run the new version without leaving
  the current session.
.............................................................DONE
Fetching with HTTP::Tiny:
…
[snip]
…
Installing /usr/local/man/man3/Net::BGP::Notification.3pm
Appending installation info to /usr/local/lib/x86_64-linux-gnu/perl/5.30.0/perllocal.pod
  SSCHECK/Net-BGP-0.17.tar.gz
  /usr/bin/make install  -- OK
```

- Then exit CPAN.

```
cpan[2]> exit
Terminal does not support GetHistory.
Lockfile removed.
root@bgp-injector:/home/hpearuba#
```

Install bgpsimple PERL script

The code is located here: https://github.com/xdel/bgpsimple

- git clone https://github.com/xdel/bgpsimple.git

```
root@bgp-injector:/home/hpearuba# git clone https://github.com/xdel/bgpsimple.git

Cloning into 'bgpsimple'...
remote: Enumerating objects: 35, done.
remote: Total 35 (delta 0), reused 0 (delta 0), pack-reused 35
Unpacking objects: 100% (35/35), 19.24 KiB | 635.00 KiB/s, done.
```

Download, make and install bgpdump

bgpdump is used to convert the BGP routing table raw data from RIPE into a consumable format for bgpsimple script. Information can be retrieved from https://github.com/RIPE-NCC/bgpdump

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

```
root@bgp-injector:/home/hpearuba# git clone https://github.com/RIPE-NCC/bgpdump

Cloning into 'bgpdump'...
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 886 (delta 7), reused 13 (delta 4), pack-reused 862
Receiving objects: 100% (886/886), 102.88 MiB | 1.81 MiB/s, done.
Resolving deltas: 100% (420/420), done.
```

- cd ./bgpdump
- Prepare the makefile: ./bootstrap.sh

```
root@bgp-injector:/home/hpearuba/bgpdump# ./bootstrap.sh

checking for gcc... gcc
checking whether the C compiler works... yes
```

```
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking for ranlib... ranlib
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking arpa/inet.h usability... yes
checking arpa/inet.h presence... yes
checking for arpa/inet.h... yes
checking netinet/in.h usability... yes
checking netinet/in.h presence... yes
checking for netinet/in.h... yes
checking syslog.h usability... yes
checking syslog.h presence... yes
checking for syslog.h... yes
checking whether struct tm is in sys/time.h or time.h... time.h
checking for u_char_t... no
checking for u_int8_t... yes
checking for u_int16_t... yes
checking for u_int32_t... yes
checking for gzopen in -lz... yes
checking for BZ2_bzReadOpen in -lbz2... yes
checking for strlcat... no
checking for inet_ntoa... yes
checking for inet_ntop... yes
configure: creating ./config.status
config.status: creating bgpdump.spec
config.status: creating Makefile
config.status: creating bgpdump-config.h
```

- make

```
root@bgp-injector:/home/hpearuba/bgpdump# make

gcc -fPIC -g  -Wall -Wsystem-headers -Wno-format-y2k -Wno-sign-compare -Wcast-align -Wmissing-prototypes -
Wpointer-arith -Wreturn-type -Wswitch -Wshadow   -c -o bgpdump_lib.o bgpdump_lib.c
gcc -fPIC -g  -Wall -Wsystem-headers -Wno-format-y2k -Wno-sign-compare -Wcast-align -Wmissing-prototypes -
Wpointer-arith -Wreturn-type -Wswitch -Wshadow   -c -o bgpdump_mstream.o bgpdump_mstream.c
gcc -fPIC -g  -Wall -Wsystem-headers -Wno-format-y2k -Wno-sign-compare -Wcast-align -Wmissing-prototypes -
Wpointer-arith -Wreturn-type -Wswitch -Wshadow   -c -o cfile_tools.o cfile_tools.c
gcc -fPIC -g  -Wall -Wsystem-headers -Wno-format-y2k -Wno-sign-compare -Wcast-align -Wmissing-prototypes -
Wpointer-arith -Wreturn-type -Wswitch -Wshadow   -c -o util.o util.c
gcc -fPIC -g  -Wall -Wsystem-headers -Wno-format-y2k -Wno-sign-compare -Wcast-align -Wmissing-prototypes -
Wpointer-arith -Wreturn-type -Wswitch -Wshadow   -c -o inet_ntop.o inet_ntop.c
ar r libbgpdump.a bgpdump_lib.o bgpdump_mstream.o cfile_tools.o util.o inet_ntop.o
ar: creating libbgpdump.a
ranlib libbgpdump.a
gcc -fPIC -g  -Wall -Wsystem-headers -Wno-format-y2k -Wno-sign-compare -Wcast-align -Wmissing-prototypes -
Wpointer-arith -Wreturn-type -Wswitch -Wshadow   -shared -o libbgpdump.so bgpdump_lib.o bgpdump_mstream.o
cfile_tools.o util.o inet_ntop.o -lbz2 -lz
gcc -fPIC -g  -Wall -Wsystem-headers -Wno-format-y2k -Wno-sign-compare -Wcast-align -Wmissing-prototypes -
Wpointer-arith -Wreturn-type -Wswitch -Wshadow   -o bgpdump bgpdump.c libbgpdump.a -lbz2 -lz
```

- make install

```
root@bgp-injector:/home/hpearuba/bgpdump# make install
```

```
install -d /usr/local/bin /usr/local/include /usr/local/lib
install bgpdump /usr/local/bin
install -m 0644 bgpdump_attr.h bgpdump_formats.h bgpdump_lib.h bgpdump_mstream.h /usr/local/include
install libbgpdump.so libbgpdump.a /usr/local/lib
```

- Check files: ls- l

```
root@bgp-injector:/home/hpearuba/bgpdump# ls -l

total 1080
drwxr-xr-x 2 root root   4096 Aug 25 22:45 autom4te.cache
-rwxr-xr-x 1 root root 161600 Aug 25 22:46 bgpdump
-rw-r--r-- 1 root root   6044 Aug 25 22:32 bgpdump_attr.h
-rw-r--r-- 1 root root  84118 Aug 25 22:32 bgpdump.c
-rw-r--r-- 1 root root   2691 Aug 25 22:45 bgpdump-config.h
-rw-r--r-- 1 root root   2443 Aug 25 22:45 bgpdump-config.h.in
-rw-r--r-- 1 root root   9871 Aug 25 22:32 bgpdump_formats.h
-rw-r--r-- 1 root root  57772 Aug 25 22:32 bgpdump_lib.c
-rw-r--r-- 1 root root   2279 Aug 25 22:32 bgpdump_lib.h
-rw-r--r-- 1 root root  84032 Aug 25 22:46 bgpdump_lib.o
-rw-r--r-- 1 root root   3075 Aug 25 22:32 bgpdump_mstream.c
-rw-r--r-- 1 root root   2005 Aug 25 22:32 bgpdump_mstream.h
-rw-r--r-- 1 root root  10752 Aug 25 22:46 bgpdump_mstream.o
-rw-r--r-- 1 root root   1460 Aug 25 22:45 bgpdump.spec
-rw-r--r-- 1 root root   1492 Aug 25 22:32 bgpdump.spec.in
-rwxr-xr-x 1 root root     43 Aug 25 22:32 bootstrap.sh
-rw-r--r-- 1 root root  13683 Aug 25 22:32 cfile_tools.c
-rw-r--r-- 1 root root   2256 Aug 25 22:32 cfile_tools.h
-rw-r--r-- 1 root root  22440 Aug 25 22:46 cfile_tools.o
-rw-r--r-- 1 root root  10276 Aug 25 22:32 ChangeLog
-rw-r--r-- 1 root root  27032 Aug 25 22:45 config.log
-rwxr-xr-x 1 root root  27575 Aug 25 22:45 config.status
-rwxr-xr-x 1 root root 145852 Aug 25 22:45 configure
-rw-r--r-- 1 root root   1704 Aug 25 22:32 configure.in
-rw-r--r-- 1 root root  16538 Aug 25 22:32 COPYING
-rw-r--r-- 1 root root  15547 Aug 25 22:32 example.c
-rw-r--r-- 1 root root   9207 Aug 25 22:32 inet_ntop.c
-rw-r--r-- 1 root root  15392 Aug 25 22:46 inet_ntop.o
-rw-r--r-- 1 root root 147732 Aug 25 22:46 libbgpdump.a
-rwxr-xr-x 1 root root 102816 Aug 25 22:46 libbgpdump.so
-rw-r--r-- 1 root root   1801 Aug 25 22:45 Makefile
-rw-r--r-- 1 root root   1670 Aug 25 22:32 Makefile.in
-rw-r--r-- 1 root root    180 Aug 25 22:32 README
drwxr-xr-x 2 root root   4096 Aug 25 22:32 test_data
drwxr-xr-x 2 root root   4096 Aug 25 22:32 test_expect
-rwxr-xr-x 1 root root   1097 Aug 25 22:32 test.sh
-rw-r--r-- 1 root root   2630 Aug 25 22:32 util.c
-rw-r--r-- 1 root root   1702 Aug 25 22:32 util.h
-rw-r--r-- 1 root root  14000 Aug 25 22:46 util.o
```

- cd ..
- Test bgdpdump:

```
root@bgp-injector:/home/hpearuba# bgpdump

2020-08-26 08:56:42 [info] logging to syslog
bgpdump version 1.6.2
Usage: bgpdump [-m|-M] [-t dump|-t change] [-O <output-file>] <input-file>
bgpdump translates binary MRT files (possibly compressed) into readable output
Output mode:
    -H          multi-line, human-readable (the default)
    -m          one-line per entry with unix timestamps
    -M          one-line per entry with human readable timestamps
    (there are other differences between -m and -M)

Common options:
    -O <file>   output to <file> instead of STDOUT
    -s          log to syslog (the default)
    -v          log to STDERR
    -q          quiet

Options for -m and -M modes:
```

```
    -t dump     timestamps for RIB dumps reflect the time of the dump (the default)
    -t change   timestamps for RIB dumps reflect the last route modification
    -p          show packet index at second position
    -l          show large communities field after regular communities
    -u          show unassigned attributes in one-line mode

Special options:
    -T          run unit tests and exit


root@bgp-injector:/home/hpearuba# bgpdump -T

fe80:: -> fe80:: [ok]
2001:db8::1 -> 2001:db8::1 [ok]
::ffff:192.168.2.1 -> ::ffff:192.168.2.1 [ok]
::192.168.1.2 -> ::192.168.1.2 [ok]
2001:7f8:30::2:1:0:8447 -> 2001:7f8:30::2:1:0:8447 [ok]
0 =?= 0 (1)
99999 =?= 99999 (5)
4294967295 =?= 4294967295 (10)
```

At this stage you may remove the bgpdump directory (if needed) or keep it.

Download a dump of BGP routing table from RIPE

- cd to home/user directory
- Create a directory to start bgp_simple from and store the BGP dumped routing table: mkdir internet_routing

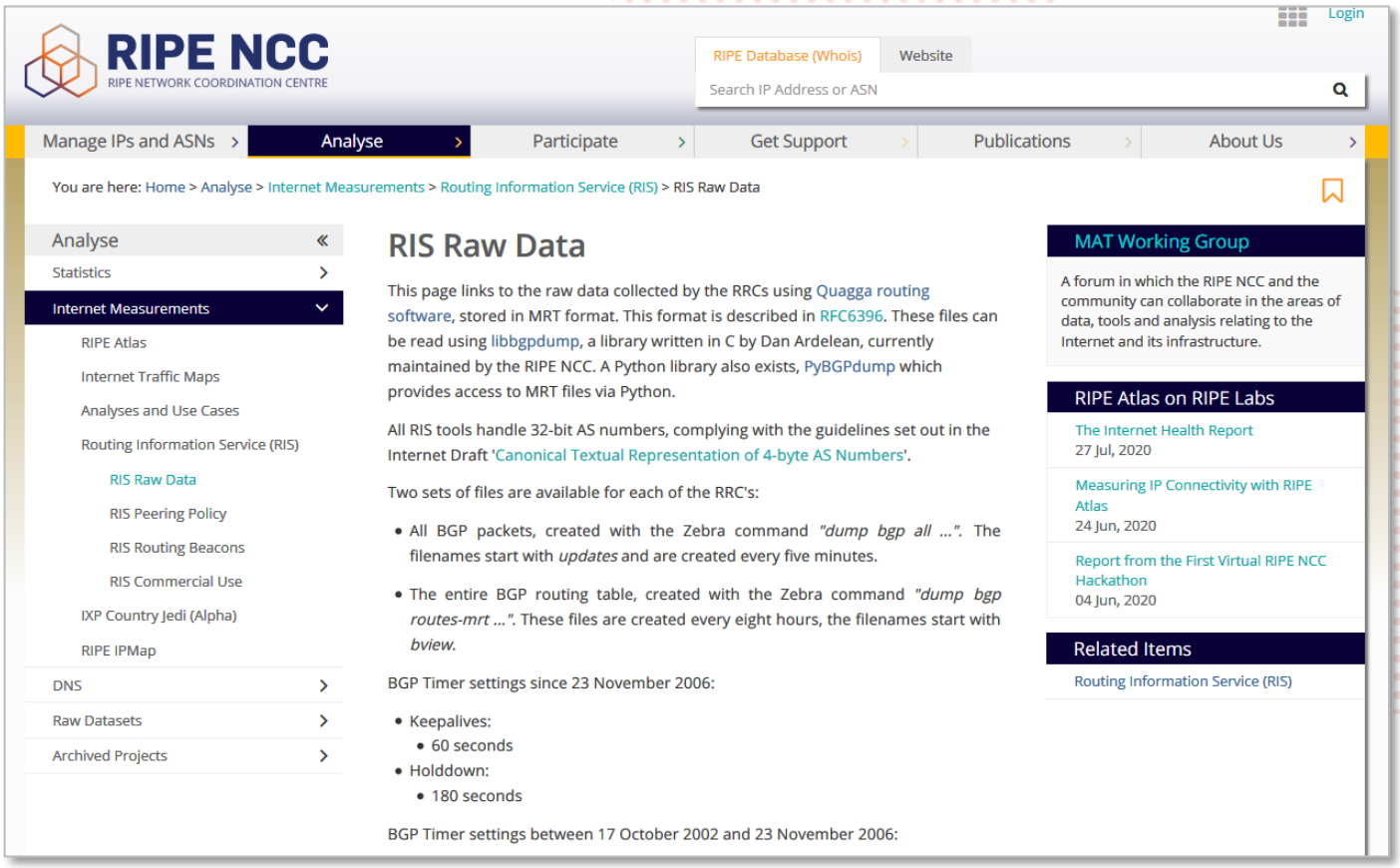
```
root@bgp-injector:/home/hpearuba# mkdir internet_routing
```

- Copy bgp_simple to the created folder

```
root@bgp-injector:/home/hpearuba# cp bgpsimple/bgp_simple.pl ./internet_routing/
```

- cd internet_routing

- The BGP routing table data are accessible from RIPE: http://www.ripe.net/projects/ris/rawdata.html



An alternative site is: http://www.routeviews.org. (not covered in this lab).

- Select one of the available collectors: rcc00, rcc01, rrc02…
  Please consider that the raw data size is not the same among all these mirroring sites. This may vary from ~100MB to 900MB. It is recommended to start with lower size to check if the performance of your server is good enough to sustain higher size. (RCC0 provides the largest file – August 2020: ~800MB)
- Here we use rcc14.ripe.net: http://data.ris.ripe.net/rrc14/
  wget data.ris.ripe.net/rrc14/latest-bview.gz

```
root@bgp-injector:/home/hpearuba/internet_routing# wget data.ris.ripe.net/rrc14/latest-bview.gz

--2020-08-26 06:56:52--  http://data.ris.ripe.net/rrc14/latest-bview.gz
Resolving data.ris.ripe.net (data.ris.ripe.net)... 193.0.6.132, 2001:67c:2e8:22::c100:684
Connecting to data.ris.ripe.net (data.ris.ripe.net)|193.0.6.132|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 72226825 (69M) [application/x-gzip]
Saving to: 'latest-bview.gz.1'

latest-bview.gz.1
100%[=====================================================================>]  68.88M  5.73MB/s    in 11s

2020-08-26 06:57:07 (6.19 MB/s) - 'latest-bview.gz.1' saved [72226825/72226825]
```

Convert BGP routing table from RIPE to the format required for bgpsimple

- From home/user/internet_routing directory/
- get instructions in bgpsimple README: cat /home/user/bgpsimple/README

```
"The original intention of the script was to feed as many routes to a peer as possible for testing
purposes. To get large amounts of real world prefixes, dumps from a DFZ attached peers must be retrieved.
Luckily, there are some ressources that provide such dumps, for example RIPE RIS
(http://www.ripe.net/projects/ris/rawdata.html). Simply fetch a full dump from
http://data.ris.ripe.net/rrc00/ (bview.*).

The script cannot work with this binary data, some conversion has to be done. To do that, get the bgpdump
library/utility (http://www.ris.ripe.net/source/). Compile bgpdump, then feed the dump through it. Make
sure to specify the -m option to produce the condensed format:

  zcat bview.20081221.0759.gz | bgpdump -m - > myroutes"
```

- Convert the raw data: zcat latest-bview.gz | bgpdump –m –O internet_routes -

```
root@bgp-injector:/home/hpearuba/internet_routing# zcat latest-bview.gz | bgpdump -m -O internet_routes -

redirecting output to 'internet_routes'
2020-08-25 16:35:37 [info] logging to syslog
```

- Check the size of the internet_routes file:

```
root@bgp-injector:/home/hpearuba/internet_routing# ls -l
total 10734988
-rwxr-xr-x 1 root root      21175 Aug 25 14:53 bgp_simple.pl
-rw-r--r-- 1 root root 1229511640 Aug 26 07:00 internet_routes
-rw-r--r-- 1 root root   72226825 Aug 26 00:30 latest-bview.gz
```

The BGP-injector VM is now ready for sending BGP internet routes to any BGP listener (AOS-CX VM for instance).

AOS-CX VM SETUP

Configure the AOS-CX VM before starting the BGP test. In this configuration, only one interface is actually used, as the focus is to explain how-to inject the internet routing table. Other labs focusing on dual ISP connectivity and routing design might require much more interfaces.

Proposed Configuration

The configuration is simple. The only caveat is to set-up long BGP timers. Indeed, the bgp_simple process being probably monothreaded, if standard timers (like 30s/90s) are taken, you may experience BGP session drop. Please make sure that the hold-timer timer is longer than the duration to load the entire BGP RIB (at least 30minutes).

```
vCX-BGP# show run

Current configuration:
!
!Version ArubaOS-CX Virtual.10.05.0010
!export-password: default
hostname vCX-BGP
user admin group administrators password ciphertext …
led locator on
!
ssh server vrf default
ssh server vrf mgmt
vlan 1
interface mgmt
    no shutdown
    ip dhcp
interface 1/1/1
    no shutdown
    ip address 10.136.40.220/24
ip route 0.0.0.0/0 10.136.40.6
!
router bgp 64999
    bgp router-id 192.168.9.9
    trap-enable
```

```
    bgp fast-external-fallover
    bgp log-neighbor-changes
    neighbor internet peer-group
    neighbor internet remote-as 64512
    neighbor internet description peering with internet simulator
    neighbor internet timers 1200 3600
    neighbor 10.136.40.46 peer-group internet
    address-family ipv4 unicast
        neighbor 10.136.40.46 activate
    exit-address-family
!
https-server vrf default
https-server vrf mgmt
```

TESTING

Start bgpsimple on BGP-injector

- cd to home/user/internet_routing
- Reminder of bgpsimple options:

```
root@bgp-injector:/home/hpearuba/internet_routing# ./bgp_simple.pl

Please provide -myas, -myip, -peerip and -peeras!

bgp_simple.pl: Simple BGP peering and route injection script.
Version v0.12, 22-Jan-2011.

usage:
bgp_simple.pl:
              -myas           ASNUMBER        # (mandatory) our AS number
              -myip           IP address      # (mandatory) our IP address to source the sesion from
              -peerip         IP address      # (mandatory) peer IP address
              -peeras         ASNUMBER        # (mandatory) peer AS number
              [-holdtime]     Seconds         # (optional) BGP hold time duration in seconds (default 60s)
              [-keepalive]    Seconds         # (optional) BGP KeepAlive timer duration in seconds (default 20s)
              [-nolisten]                     # (optional) dont listen at $myip, tcp/179
              [-v]                            # (optional) provide verbose output to STDOUT, use twice to get debugs
              [-p file]                       # (optional) prefixes to advertise (bgpdump formatted)
              [-o file]                       # (optional) write all sent and received UPDATE messages to file
              [-m number]                     # (optional) maximum number of prefixes to advertise
              [-n IP address]                 # (optional) next hop self, overrides original value
              [-l number]                     # (optional) set default value for LOCAL_PREF
              [-dry]                          # (optional) dry run; dont build adjacency, but check prefix file (requires -p)
              [-f KEY=REGEX]                  # (optional) filter on input prefixes (requires -p), repeat for multiple filters
                                                     KEY is one of the following attributes (CaSE insensitive):

                                                     NEIG            originating neighbor
                                                     NLRI            NLRI/prefix(es)
                                                     ASPT            AS_PATH
                                                     ORIG            ORIGIN
                                                     NXHP            NEXT_HOP
                                                     LOCP            LOCAL_PREF
                                                     MED             MULTI_EXIT_DISC
                                                     COMM            COMMUNITY
                                                     ATOM            ATOMIC_AGGREGATE
                                                     AGG             AGGREGATOR

                                                     REGEX is a perl regular expression to be expected in a
                                                     match statement (m/REGEX/)

Without any prefix file to import, only an adjacency is established and the received NLRIs, including their
attributes, are logged.
```

Notes:

1. The Autonomous System values have to match on both BGP-injector and vCX-BGP.
2. The Source (myip) and destination (peerip) IP addresses have to match the corresponding IP addresses from the interconnectivity subnet between BGP-injector and vCX-BGP.
3. You can limit the number of routes as a first trial with –m option (ex: -m 1000 for the first 1000 routes). There is no limit

being set in the below example.

4. It is recommended to use the –n option to set the next-hop as being the AOS-CX VM. This will be useful in other labs to avoid reachability dependency of unknown next-hops.
5. The output of this command is very verbose. It is recommended to redirect the output to dev/null. An alternative is to comment from line 640 to line 649 the debug section of bgp_simple script.
6. BGP timers should match timers on vCX-BGP VM.
7. The file to provide for prefixes to be sent is the output of the bgpdump conversion. Here: internet_routes file.
8. Use "&" to make the script run in the background so that stopping the SSH session to BGP-injector has no effect on the established BGP peering. In order to stop the peering, you can either put the process in the foreground (fg) and CTRL+C; or you may kill the process (kill -9 <process-id_provided_at_output_of_script_start>).

- ./bgp_simple.pl -myas 64512 -myip 10.136.40.46 -peerip 10.136.40.220 -peeras 64999 -keepalive 1200 -holdtime 3600 -p internet_routes -n > /dev/null &

```
root@bgp-injector:/home/hpearuba/internet_routing# ./bgp_simple.pl -myas 64512 -myip 10.136.40.46 -peerip
10.136.40.220 -peeras 64999 -keepalive 1200 -holdtime 3600 -p internet_routes -n > /dev/null &
[1] 15098
```

Verification

BGP-injector

Check that process is running as expected.

```
root@bgp-injector:/home/hpearuba/internet_routing# ps -efl | grep bgp

4 S root       39935   39621  7  80   0 -  4505 poll_s 07:02 pts/1    00:13:35 /usr/bin/perl
./bgp_simple.pl -myas 64512 -myip 10.136.40.46 -peerip 10.136.40.220 -peeras 64999 -keepalive 1200 -
holdtime 3600 -p internet_routes-opt -n
0 S root       44472   39621  0  80   0 -  1575 pipe_w 10:02 pts/1    00:00:00 grep --color=auto bgp
```

vCX-BGP

- Check that BGP-injector is reachable.

```
vCX-BGP# ping 10.136.40.46
PING 10.136.40.46 (10.136.40.46) 100(128) bytes of data.
108 bytes from 10.136.40.46: icmp_seq=1 ttl=64 time=1.32 ms
108 bytes from 10.136.40.46: icmp_seq=2 ttl=64 time=1.18 ms
^C
--- 10.136.40.46 ping statistics ---

2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 1.187/1.257/1.327/0.070 ms
```

- Check that BGP session is established and check the BGP messages being received. This should increase up to the completion of the load of internet routes. You have to repeat the command. It is expected that the show bgp command is getting less responsive.

It can take up 20 to 30 minutes to load the routes !

```
vCX-BGP# show bgp ipv4 unicast summary
VRF : default
BGP Summary
-----------
 Local AS               : 64999        BGP Router Identifier  : 192.168.9.9
 Peers                  : 1            Log Neighbor Changes   : Yes
 Cfg. Hold Time         : 180          Cfg. Keep Alive        : 60

 Neighbor        Remote-AS MsgRcvd MsgSent   Up/Down Time State        AdminStatus
 10.136.40.46    64512       4540048 13      03h:05m:07s  Established   Up
```

- Check the routing table. The number of routes from BGP RIB should increase. Again it may take long time to get a fully populated routing table with these internet routes.

```
vCX-BGP# show ip route summary

 IPv4 Route Table Summary

 VRF name :  default
  Protocol       Active Routes
 -------------- -------------
  connected       1
  local           1
  static          1
  bgp             586532
```

- Show the routing table to identify a given prefix.

This command can take a long time (1minute).

```
vCX-BGP# show ip route

Displaying ipv4 routes selected for forwarding

'[x/y]' denotes [distance/metric]

0.0.0.0/0, vrf default
        via  10.136.40.6,  [1/0],  static
1.0.0.0/24, vrf default
        via  10.136.40.46,  [20/100],  bgp
1.0.4.0/22, vrf default
        via  10.136.40.46,  [20/100],  bgp
1.0.4.0/24, vrf default
        via  10.136.40.46,  [20/100],  bgp
1.0.5.0/24, vrf default
        via  10.136.40.46,  [20/100],  bgp
1.0.6.0/24, vrf default
        via  10.136.40.46,  [20/100],  bgp
1.0.7.0/24, vrf default
        via  10.136.40.46,  [20/100],  bgp
1.0.16.0/24, vrf default
        via  10.136.40.46,  [20/100],  bgp
1.0.64.0/18, vrf default
        via  10.136.40.46,  [20/100],  bgp
1.0.128.0/17, vrf default
        via  10.136.40.46,  [20/100],  bgp
1.0.128.0/18, vrf default
        via  10.136.40.46,  [20/100],  bgp
1.0.128.0/19, vrf default
        via  10.136.40.46,  [20/100],  bgp
1.0.128.0/24, vrf default
        via  10.136.40.46,  [20/100],  bgp
1.0.129.0/24, vrf default
        via  10.136.40.46,  [20/100],  bgp
1.0.130.0/24, vrf default
        via  10.136.40.46,  [20/100],  bgp
1.0.131.0/24, vrf default
        via  10.136.40.46,  [20/100],  bgp
1.0.132.0/24, vrf default
        via  10.136.40.46,  [20/100],  bgp
1.0.133.0/24, vrf default
        via  10.136.40.46,  [20/100],  bgp
1.0.134.0/24, vrf default
        via  10.136.40.46,  [20/100],  bgp
1.0.135.0/24, vrf default
        via  10.136.40.46,  [20/100],  bgp
1.0.136.0/24, vrf default
        via  10.136.40.46,  [20/100],  bgp
1.0.137.0/24, vrf default
        via  10.136.40.46,  [20/100],  bgp
1.0.138.0/24, vrf default
```

```
        via  10.136.40.46,  [20/100],  bgp
 -- MORE --, next page: Space, next line: Enter, quit: q
```

- Show the BGP route information for a given prefix.

This command can take a long time (1minute).

```
vCX-BGP# show bgp ipv4 unicast 1.0.128.0/18

VRF : default
BGP Local AS 64999       BGP Router-id 192.168.9.9

   Network           : 1.0.128.0/18             Nexthop           : 10.136.40.46
   Peer              : 10.136.40.46             Origin            : IGP
   Metric            : 100                      Local Pref        : 100
   Weight            : 0                        Calc. Local Pref  : 100
   Best              : Yes                      Valid             : Yes
   Type              : external                 Stale             : No
   Originator ID     : 0.0.0.0
   Aggregator ID     :
   Aggregator AS     :
   Atomic Aggregate  :
   RFD Flaps         : 0                        RFD Penalty       : 0

   AS-Path           :  64512 6762 38040 23969

   Cluster List      :
   Communities       : 6762:1,6762:33,6762:40,6762:16500
     Ext-Communities :
```

Please note the following:

1. The Nexthop is set as the BGP-injector VM itself. This is expected due to next-hop-self option used on bgp_simple script. It is desired so that there is no issue of reachability for internet next-hop IP addresses.
2. Consider the richness of this test compared to IXIA testing: there is diversity of AS Paths and of BGP communities. This set-up can be used to validate routing design for internet peering at scale.

There is code optimization being developed to reduce the time to get the show bgp command output. It is currently not recommended to run the command: show bgp ipv4 unicast, in order to display the full BGP routing table as it may take >10 minutes.

a Hewlett Packard Enterprise company

www.arubanetworks.com

3333 Scott Blvd. Santa Clara, CA 95054
1.844.472.2782 | T: 1.408.227.4500 | FAX: 1.408.227.4550 | info@arubanetworks.com