BGP Prefix Origin Validation Extension for Quagga Manual

Michael Mester
mester@zedat.fu-berlin.de

June 14, 2013

Abstract

This manual describes how the BGP prefix origin validation extension feature for the Quagga Routing Suite can be installed and configured. The purpose of this extension is to add BGP prefix origin validation to the BGP routing daemon. For further information on prefix origin validation, see RFC 6811.

Functional Features

- The BGP router can connect to one or more RPKI cache servers and receive validated prefix data from them. The servers can be ordered into groups with different preference values.
- The router will check if incoming routes can be found in the validated prefix table. The routes will be marked with one of three validation states: VALID, NOT FOUND and INVALID.
- Per default the router will prefer valid routes over not found routes and it will neither accept nor distribute invalid routes. Statically configured routes which are invalid will also not be announced to neighbors. But the router can be confgured to use invalid routes.
- If no connection to an RPKI cache server can be established the router will go on without prefix origin validation, but in the background it will keep on trying to connect.
- When the router receives an update from the cache all routes in the local routing table will be re-evaluated. Route maps however will not be applied to those routes again. (Maybe this will be possible in the future.)

Installation

1. Download and install RTRlib, the RPKI RTR Client C Library from `http://rpki.realmv6.org/wiki/Download`. For testing purposes it is a good idea to compile with debug symbols and activate debug messages: `cmake -D CMAKE_BUILD_TYPE=Debug .`

   Do not forget to call `ldconfig` after the installation to update the shared library cache.

2. (a) Checkout the latest version, which includes the Quagga master branch patched with the RPKI extension:
   `git clone git://github.com/rtrlib/quagga-rtrlib.git`

   (b) Alternatively, you can patch the Quagga source of a specific version. Copy the patchfile from `http://rpki.realmv6.org/wiki/Download` into the Quagga root directy and apply it by running:
   `patch -p1 < quagga_rpki.patch`

   We recommend step (a). Run the script `update-autotools` and `autoreconf` afterwards.

3. Configure Quagga with the configure script as usual and add `--enable-rpki` to the command line options. Run `make` and `make install` afterwards.

Configuration

The RPKI extension adds the following configuration commands to Quagga.

enable-rpki

Syntax: `enable-rpki`

Available in: Configuration mode

Parameters: None

Description: This command activates the rpki configuration mode. All other commands which start with `rpki` can only be used after the `enable-rpki` command. When it is used in a telnet session, leaving of this mode will cause a restart of the rpki session with the current settings. This means that any changes of the rpki configuration will not come into effect before the rpki configuration mode is exited. Please note: this command alone does not activate prefix validation. You need to add at least one cache group with at least one reachable cache in it.

rpki group

Syntax:
```
rpki group PREFERENCE
no rpki group
```

Available in: rpki configuration mode

Parameters: `PREFERENCE` Optional: NO Range: 0 – 4,294,967,296

Define a preference value for this group. The group with the lowest value will be chosen first. If no cache in a group is reachable, the group with the next higher value is chosen.

Description: Begin a new cache group. A group can contain several rpki cache servers. Every cache command which follows a group command will belong to this group. If you are configuring via virtual terminal then you can also select an already existing group with this command. After you selected a group you can add or remove caches from it. The no command deletes the whole group.

rpki cache

Syntax:
```
rpki cache (A.B.C.D|WORD) PORT [SSH_USERNAME]
[SSH_PRIVKEY_PATH] [SSH_PUBKEY_PATH] [KNOWN_HOSTS_PATH]
no rpki cache (A.B.C.D|WORD) [PORT]
```

Available in: rpki configuration mode

Parameters: `(A.B.C.D|WORD)` Optional: NO

Set the ip address or hostname of the cache server.

`PORT` Optional: NO

Set the port number for this cache. In the no variant of this command the port may be omitted. It is only useful in case two caches are running on the same host.

`[SSH_UNAME] [SSH_PRIVKEY_PATH] [SSH_PUBKEY_PATH]` Optional: YES

If you want the router to use an ssh connection instead of an unprotected tcp connection then you have to provide the necessary client authentication information: user name, private and public key. The parameters SSH_PRIVKEY_PATH and SSH_PUBKEY_PATH must be the correct paths to the key files on the local filesystem.

`[SERVER_PUBKEY_PATH]` Optional: YES

With this parameter the known hosts file name can be provided. It is optional even if you want an ssh connection to be established. If the value is NULL, the directory is set to the default known hosts file, normally ~/.ssh/known_hosts. The known hosts file is used to certify remote hosts are genuine. It may include "%s" which will be replaced by the user home directory.

Description: Define a new cache server or delete it with the no variant. The cache connection can be established via unprotected tcp or ssh. This command can only be used if a rpki group has been defined before. The cache will be added to the latest created or selected group.

rpki polling period

Syntax:
```
rpki polling_period SECONDS
no rpki polling_period
```

Available in: rpki configuration mode

Parameters: `SECONDS` Optional: NO Range: 1 - 3600

Polling period in seconds.

Description: Set the polling period which is the time the router will wait until it requests the next rpki update from the cache server. The no command sets the value back to default which is 300.

rpki timeout

Syntax:
```
rpki timeout SECONDS
no rpki timeout
```

Available in: rpki configuration mode

Parameters: `SECONDS` Optional: NO Range: 1 – 4,294,967,296

Timeout in seconds.

Description: Define a timeout value. Received pfx_records are deleted if the client was unable to refresh data for this time period. The no command sets the value back to default which is 600.

rpki initial-synchronisation-timeout

Syntax:
```
rpki initial-synchronisation-timeout TIMEOUT
no rpki initial-synchronisation-timeout
```

Available in: rpki configuration mode

Parameters: `TIMEOUT` Optional: NO Range: 1 – 4,294,967,296

Description: Set a timeout for the initial synchronisation of prefix validation data. The router waits at most the given time in seconds until a connection to at least one rpki cache is synchronised, i.e. the router got some validated prefix data. If the timeout expires, bgp routing goes on without rpki. But it will keep on trying to establish the connection in the background. The no command sets the value back to default which is 30 seconds.

debug rpki

Syntax:
```
debug rpki
no debug rpki
```

Available in: configuration mode and enable mode

Parameters: `None`

Description: Enable or disable debugging output for rpki.

bgp bestpath prefix-validate

Syntax:
```
bgp bestpath prefix-validate allow-invalid
no bgp bestpath prefix-validate allow-invalid

bgp bestpath prefix-validate disable
no bgp bestpath prefix-validate disable
```

Available in: bgp mode

Parameters: None

Description: The `allow-invalid` command allows the router to use routes even if their origin has been marked as invalid by rpki. It is a good advice to use this command if a route map is used together with `catch rpki invalid`. (See Example)

The `prefix-validate disable` command disables the prefix validation completely. The router does still try to connect to the configured rpki caches and get validated prefix data. This command is useful for configuration testing.

Information display on virtual terminal

Show rpki prefix-table

Syntax: `show rpki prefix-table`

Available in: view and enable mode

Parameters: None

Description: Prints a table to stdout which contains all the validated prefixes which have been received from the caches.

Show rpki cache-connection

Syntax: `show rpki cache-connection`

Available in: view and enable mode

Parameters: None

Description: Shows to which caches the router is connected to

Example configuration file

```
hostname bgpd1
password zebra
!
log stdout
debug bgp updates
debug bgp keepalives
debug rpki
!
enable-rpki
  rpki polling_period 1000
  rpki timeout 10
  rpki group 1
    ! SSH Example:
    rpki cache rpki.realmv6.org 22 rtr-ssh ./ssh_key/id_rsa ./ssh_key/id_rsa.pub
    ! TCP Example:
    rpki cache rpki.realmv6.org 42420
!
router bgp 60001
 bgp router-id 141.22.28.223
 bgp bestpath prefix-validate allow-invalid
 network 192.168.0.0/16
 neighbor 123.123.123.0 remote-as 60002
 neighbor 123.123.123.0 route-map rpki in
!
 address-family ipv6
  neighbor 123.123.123.0 activate
  neighbor 123.123.123.0 route-map rpki in
 exit-address-family
!
route-map rpki permit 10
 match rpki invalid
 set local-preference 10
!
route-map rpki permit 20
 match rpki notfound
 set local-preference 20
!
route-map rpki permit 30
 match rpki valid
 set local-preference 30
!
route-map rpki permit 40
!
```