Quagga and BGP Secure Routing Extension

– QuaggaSRx –

Version 0.4.2

User's Manual

July 2017

http://bgpsrx.antd.nist.gov

bgpsrx-dev@nist.gov

Oliver Borchert, Kyehwan Lee, Kotikalapudi Sriram, Doug Montgomery

Index

Introduction

Overview

This document describes the integration of the SRx Server API into Quagga. Quagga uses the SRx server by embedding the SRx server proxy. The SRx proxy manages the necessary TCP connection to the server and provides Quagga with a simple to use API.

This version of QuaggaSRx is based on Quagga 0.99.22.

Installation

To install QuaggaSRx, the SRx-server API needs to be installed first. For information on how to install and configure QuaggaSRx, see the text file INSTALL.SRx_client which is located in the source directory of QuaggaSRx. Once the configuration is prepared, the command sequence make; make install; should take care of the rest.

For Red Hat based systems the software can be installed using the 'yum' installer. The repository information is available on the website http://bgpsrx.antd.nist.gov

Quagga SRx configuration

All configuration commands added to Quagga are located within the BGP router configuration. All SRx commands start with the key word **srx** followed by the command and its parameters. The following command sequence is used to enter the configuration section using the telnet terminal (note: while using the configuration file, all commands are located within each router configuration):

1. enable
2. configure terminal
3. router bgp <asn>

SRx Configuration settings

[no] srx display

Turn on/off additional SRx display information for default show commands.

srx-proxy-id <id>

The SRx proxy id MUST be set prior to using the connect command. The srx-server uses the proxy ID to link updates to routers. This is can be either scripted as IPv4 address or plain 4-byte integer decimal value. It is recommended to use the router-id as proxy-id.

srx set-server <host> <0..65535>

Configure the address of the server and its port without connecting.

srx connect [<host> <0..65535>]

Connect the BGP server instance to the SRx server at the given location. The preferred method to connect is using "srx set-server" to configure the srx server connection and calling "srx connect" without any parameters. The Quagga command show/write running-config uses the preferred sequence.

srx disconnect

Disconnect the BGP server instance from the SRx server. For this command the "keep-window" setting is used.

srx keep-window

Specifies the time in seconds the SRx is requested to hold information after it is deleted! This allows a router reboot without losing the validation result information within SRx.

srx extcommunity <0-255> (ibgp_only | include_ebgp)

This setting is introduced with QuaggaSRx 0.3.1 and allows communicating origin validation results to peers. The given integer value in the range of 0-255 is used to identify this extended community value string. It will be used in the field currently marked as 'TBD' [draft-ietf-sidr-origin-validation-signaling]. By default, this enables the communication in send and receive mode for all iBGP peers. In addition to the draft specification QuaggaSRx allows to extend the community into eBGP by adding the parameter 'include_ebgp'. To turn off eBGP reconfigure the router using 'ibgp_only'

no srx extcommunity

Disable the transfer of origin validation results.

srx capability extended [liberal]

This is a setting follows draft-ietf-idr-bgp-extended-messages-21 and allows to increase the regular BGP update message size from 4K to 64K. The optional parameter "liberal" allows a more liberal approach as specified in the specification.

no capability extended [liberal]

Disable the extended message capability.

srx evaluation (origin_only | bgpsec [distributed])

This command enables or disables the policy processing within the decision process as well as activating or deactivating setting of the ignore flag due to an ignore-XXX policy. In addition to enabling the evaluation of validation results this command specifies the mode the evaluation is performed in. In the *origin_only* mode, only origin validation results are used for calculating the validation result. In the *bgpsec* mode, the origin validation result and the path validation result, both are used to determine the final validation result.

To disable the evaluation and learn more about the impact, see the next command "no srx evaluation".

origin_only (default)

Using this setting only origin validation is evaluated. Path validation results will still be requested and notifications from SRx will be processed in regards to maintaining the correct data associated with each update but the results of path validation will not be included in the evaluation of validation results. The following results are possible with origin_only validation processing:

valid	A ROA exists that covers the announced prefix and origin.
notfound	No ROA exists for the announced prefix or a less specific of it.
invalid	A ROA exists that covers the announced prefix or a less specific prefix, but the origin AS does not match.
(*undefined*)	Validation not performed yet.

QuaggaSRx introduces a fourth validation result type called "*undefined"*. This result type allows distinguishing between an actual validation result and the status when no connection from the Quagga Router to SRx exists, or not enough information is at available to make the final decision on the validation result for the update. As soon as QuaggaSRx can determine the outcome of the validation, then the validation result is set to the specific validation state.

bgpsec [distributed]

This evaluation mode activates origin validation and path validation. QuaggaSRx uses the validation results of origin validation and path validation to compute the final BGPSEC validation result (valid|invalid|*undefined*). SRx reports prefix-origin validation and path validation independently as soon as they are available. Note that the SRx path validation refers only to the validation of the path signatures, NOT including the origin validation. QuaggaSRx merges the independent results of origin and path validation into one final BGPSEC validation.

To fully activate BGPsec path processing see the next section for "Configure BGPsec path validation"

The option **distributed** specifies the location where the BGPsec path validation will take place. If provided, the BGPsec path validation will be done by SRx-Server, if omitted, the BGPsec path validation will be performed locally within the router itself. In the latter case, all keys must be pre-installed in the router. To use the validation cache for retrieving the router keys (public keys), the validation must be performed by the SRx-Server.

The following table illustrates the possible BGPSEC validation results:

BGPSEC		Path Validation		
		Valid	**Invalid**	**Undefined**
Prefix – Origin Validation	**Valid**	**V**	**I**	**?**
	NotFound	**I**	**I**	**I**
	Invalid	**I**	**I**	**I**
	Undefined	**?**	**I**	**?**

V=Valid, I=Invalid, ?=Undefined

no srx evaluation

Disable policy processing of the validation result. In this mode QuaggaSRx performs normal BGP processing. Regardless to this setting, the QuaggaSRx will send validation requests to SRx-server and process its notifications but does not act upon it. To disable SRx communication disconnect from SRx.

srx set-origin-value [value]

Set the default value for origin validation. This value is used until the SRx-server provides the real validation value. Accepted values are "*valid", "notfound", "invalid"*, and "*undefined".*

srx set-path-value [value]

Set the default value for path validation. This value is used until the SRx-server provides the real validation value. Accepted values are "*valid", "invalid"*, and "*undefined".*

srx set-path-value [value]

Set the default value for path validation. This value is used until the SRx-server provides the real validation

Configure BGPsec path validation

To fully enable BGPsec path validation the "**srx evaluation**" mode needs to be configured for "**bgpsec**" as described in the previous chapter. In addition, the private key needs to be set and the neighbor needs to be enabled. QuaggaSRx uses the SRxCryptoAPI to perform all BGPsec related operations. This API will provide tools for key generation etc.

srx bgpsec ski (0|1) <1..255> [key-ski]

With version 0.4.2 QuaggaSRx allows to have 2 separate private keys installed. In addition, each key can be assigned its separate algorithm suite identifier. The key ski is the hex representation of the 20-octet long ski (40 HEX ASCII characters) that specifies the private key. The private key will be loaded using the SRxCryptoAPI.

srx key (0|1) active
This command specifies which of the up to two keys is currently activate. At this point, only one key can be active at a time.

neighbor A.B.C.D bgpsec (send|receive|both)
This command extends the default "neighbor" configuration of the underlying Quagga engine. Currently BGPsec is enabled for IPv4 peers only.

For information on configuring the SRxCryptoAPI please see its respective user manual.

SRx Policy Configuration

QuaggaSRx provides three different policy types,

- Ignore updates with selected validation result
- Modify the local preference of updates depending on the validation result
- Prefer updates whose validation result is "*valid"*

By default, QuaggaSRx does not enable any policies except for ignore-undefined. Policies do influence the BGP decision process in the following order:

1. Ignore Policies:
   These policies prevent updates with a certain validation result from entering the decision process. They are stored in the RIB in but will not be considered for route selection.
2. Local Preference Modification policies:
   These policies allow a dynamic/fixed modification of each update's local preference value in accordance with its validation result. The dynamic method allows combining other local preference policies with the validation result policies. In case a dynamic local preference policy reduces the local preference to a value less than zero "0" (underflow), the local preference will be adjusted to zero.
3. Prefer Valid:
   This policy prefers updates with the validation state "*valid"* to updates whose values are different from "*valid"*. The policy "Prefer Valid" is executed directly after local preference policies.

The QuaggaSRx implementation changes the default decision process in the following manner:

1. Weight (kept as is)
2. Local Preference:
   After determining the local preference, the srx policy will apply changes to the local preference according to the validation result
3. Prefer "valid" updates over updates with a different validation state.
4. … (kept as is)

It is possible that in certain circumstances updates of different validation states are compared where none of the updates is valid. In this case, no ranking is performed because this situation is transient and can occur during the introduction of a new ROA where certain updates are already re-evaluated but others are not re-evaluated yet.

IMPORTANT: Policies should not be modified during operations. Changes in policies do not trigger a re-evaluation of already installed routes. Only changes due to the validation itself of update announcements or withdrawals trigger the decision process for re-evaluation of the validation result!

[no] srx policy (ignore-notfound | ignore-invalid | ignore-undefined)

Activates or deactivates this set of policies that specify which update has to be ignored.

ignore-notfound

Updates with the validation result "*notfound"* will be flagged as ignored and will not be processed further in the decision process.

ignore-invalid

Updates with the validation result "*invalid"* will be flagged as ignored and will not be processed further in the decision process

ignore-undefined (default)

Updates that have not yet been validated by the SRx server are considered undefined. This is an intermediate state, and as soon as the SRx server processes the validation for the update, it will receive one of the final validation results (valid, notfound, invalid).

[no] srx policy local-preference (valid | notfound | invalid) <value> [add | subtract]

Local preference modification policies are only applied to updates with the validation results "valid", "invalid", or "notfound". The value is a positive integer value that is used as a fixed local preference value overwriting a pre-existing value or modifying the pre-existing value. The later one is specified by adding the keywords "**add"** or "**subtract"** to the policy configuration.

Modifying the local preference by adding or subtracting allows combining other policies with origin validation / path validation. An example could be a policy where an operator wants to configure the router in such way that all routes of peer A are preferred over routes by peer B except if peer B has a "valid" route while peer A has only an "invalid" one for the same prefix. Also, routes of peer A with the validation status "notfound" or "undefined" are still preferred over "valid" routes of peer B. The following configuration would allow such a setup:

Configure default local preference for peer A: 106
Configure default local preference for peer B: 100

srx policy local-preference valid 5 add
srx policy local-preference invalid 5 subtract

1: U(A): v | n | ? → LP >= 106, U(B): n | i | ? → LP <= 100, selected update: U(A)
2: U(A): v | n | ? → LP >= 106, U(B): v → LP = 105, selected update: U(A)
3: U(A): i → LP = 101, U(B): i | n | ? → LP <= 100, selected update: U(A)
4: U(A): i → LP = 101, U(B): v → LP = 105, selected update: **U(B)**

v = valid, n = notfound, i = invalid,? = undefined

As illustrated above in scenario #4 with U(B) = "valid" and U(A) = "invalid", the "valid" route of Peer B will be chosen over the "invalid" route of peer A.

To deactivate a policy the keyword "**no**" must be placed before the policy definition.

[no] srx policy prefer-valid

This policy indicates that the tiebreaker between two BGP updates is the validation state "**valid"**. This means updates that are "valid" are selected over updates whose validation state differs from "valid". In case both updates are either "valid" or both are other than "valid", other tiebreakers such as shortest path, MED, router id etc. will be used to determine the route selection.

Display commands

For the display, QuaggaSRx seamlessly integrates validation information into the standard **show [ip] bgp** commands. The additional information must be enabled or disabled within using the srx display command as described above.

SRx Configuration Display

To display the SRx configuration within QuaggaSRx it is necessary to maneuver to the bgp configuration level. This level will be entered by entering the "enabled" mode, "configure terminal" – "router bgp <ASN>". Once in this level (same as the one used for configuring policies etc.), the console command **show srx-config** displays all configuration settings for the SRx connection, configured policies as well as the status of the SRx connection.

```
bgpd(config-router)# show srx-config
SRx configuration settings:
  server.........: localhost
  port...........: 17900
  proxy-id.......: 0x0A040000 (10.4.0.0 - 168034304)
  keep-window....: 900
  evaluation.....: origin_only (prefix-origin processing)
  default value..: (origin)   ? = undefined
  default value..: ( path )   ? = undefined
  policy.........: prefer-valid
  connected......: true
bgpd(config-router)#
```

QuaggaSRx: Proxy configuration display

Furthermore, the SRx extensions are also included in the stock commands such as show running-config as well as write running-config.

SRx Related BGP Display

The following described information is visible only if the **srx display** is configured. The command **[no] srx display** is used to configure the SRx display. By default, the command is activated and SRx related information is added to the standard show commands:

Command: **show ip bgp**

```
bgpd# show ip bgp
BGP table version is 0, local router ID is 10.4.0.0
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Validation:    v - valid, u - unknown, i - invalid, ? - undefined
SRx Status:    I - route ignored, D - SRx evaluation deactivated
SRxVal Format: validation result (origin validation, path validation)
Origin codes: i - IGP, e - EGP, ? - incomplete

   Ident    SRxVal SRxLP Status Network          Next Hop            Metric  LocPrf Weight Path
*  5CD7FB96 v(v,-)              10.1.0.0/16      10.43.0.3                            0 3 2 1 i
*> 5CD7FB96 v(v,-)                            10.43.0.2                            0 3 2 1 i
*  25980063 i(i,-)                            10.64.0.1               0            0 6 i
*  D83F5805 n(n,-)              10.2.0.0/16      10.43.0.3                            0 3 2 i
*> D83F5805 n(n,-)                            10.43.0.2                            0 3 2 i
*  29761735 n(n,-)              10.3.0.0/16      10.43.0.3               0            0 3 i
*> 29761735 n(n,-)                            10.43.0.2               0            0 3 i
*> 3E0D0376 n(n,-)              10.3.0.0/17      10.43.0.2               0            0 3 i
*> 4D1E9C39 n(n,-)              10.3.128.0/17    10.43.0.3               0            0 3 i
*> -------- ?(?,-)              10.4.0.0/16      0.0.0.0                 0          32768 i
*> 779FF6C1 n(n,-)              10.5.0.0/16      10.45.0.2               0            0 5 i
*> 58EB063B n(n,-)              10.6.0.0/16      10.64.0.1               0            0 6 i

Total number of prefixes 8
bgpd#
```

Command: **show ip bgp <network>**

```
bgpd# show ip bgp 10.1.0.0/16
BGP routing table entry for 10.1.0.0/16
Paths: (3 available, best #2, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  10.43.0.3 10.45.0.2 10.64.0.1
  3 2 1
    SRx Information:
      Update ID: 0x5CD7FB96
      Validation:
        prefix-origin: valid
        path processing disabled!
    10.43.0.3 from 10.43.0.3 (10.3.1.0)
      Origin IGP, localpref 100, valid, external
      Last update: Wed Dec 31 19:10:25 1969

  3 2 1
    SRx Information:
      Update ID: 0x5CD7FB96
      Validation:
        prefix-origin: valid
        path processing disabled!
    10.43.0.2 from 10.43.0.2 (10.3.0.0)
      Origin IGP, localpref 100, valid, external, best
      Last update: Wed Dec 31 19:10:20 1969

  6
    SRx Information:
      Update ID: 0x25980063
      Validation:
        prefix-origin: invalid
        path processing disabled!
    10.64.0.1 from 10.64.0.1 (10.6.0.0)
      Origin IGP, metric 0, localpref 100, valid, external
      Last update: Wed Dec 31 18:46:32 1969

bgpd#
```

Support

Before contacting us, please verify that QuaggaSRx and SRx server are connected and properly communicating. We provide tools such as Wireshark plugins that allow analyzing the traffic in a human readable manner. Also, check firewall settings. For result inquiries of updates look up the update id and query the update information at SRx server. If nothing helps (not even a reboot), please contact us, and we will try to help. In case of crashes, please provide a description on how to reproduce the crash and if possible a core dump.

To be informed of bug fixes or ask questions to the community, subscribe to the users email list by sending an email to bgpsrx-users-request@nist.gov with subscribe in the subject.

Questions to the developers and general contact information:

Email: bgpsrx-dev@nist.gov
Web: https://bgpsrx.antd.nist.gov

Developers:

Oliver Borchert oliver.borchert@nist.gov
Kyehwan Lee

Previous Developers:

Patrick Gleichmann (V0.1.0 only)