draft-sidr-bgpsec-protocol-05
Open Issues

Overview

- I received many helpful reviews:
  - Thanks Rob, Sandy, Sean, Randy, and Wes
- Most issues are minor stuff that I am in the process of fixing
- This presentation is mostly normative changes, with a couple non-normative things where I would like working group input

Part 1: GENERAL STUFF

1.1 Nomenclature for “AS Path”

- We don't have an AS_PATH
- What do we call the sequence of ASes through which an update passes?
  - Surely there is something better than “The sequence of ASes through which an update passes”
  - However, “AS path” might be confusing?

1.2 Mandate for Transport Security

- Do we want to specify a specific mechanism for transport security (either MUST or SHOULD)?
  - E.g., TCP-AO
- Does it make since to mandate transport security without specifying a specific mechanism?

1.3 FYI: Editor's Notes

- The -05 version is the last version with Editor's Notes.
- My plan is that I completely remove them from the -06 version

Part 2: CAPABILITY NEGOTIATION

2.1. Negotiation Errors

- Is failure to negotiate the BGPSEC capability is it an error that prompts sending of a NOTIFICATION message?
- I believe the answer is “No”, but I am not certain
- RFC 5492 essentially says whether you produce an error and close the session depends on the capability that you fail to negotiate

2.2. SAFI of 0 vs Null SAFI

- Do we need a “SAFI present” flag?
- That is, does setting the SAFI to zero in the capability advertisement have a different semantics than “SAFI not present”?

2.3 Negotiation of 4-Byte AS

- Do BGPSEC speakers also include the 4-Byte AS number capability?
- If so, what happens when a speaker does not? Do we close the session?

2.4 Both “No send” and “No receive”

- Is it an error to send announce support for BGPSEC, but be unwilling to either send or receive BGPSEC update messages?

Part 3: THE ATTRIBUTE

3.1 Name of the attribute

- Anyone have a problem with BGPSEC_PATH
  - Replacing BGPSEC_PATH_SIGNATURES

3.2 Router IDs

- Draft-ietf-sidr-bgpsec-pki-profiles RECOMMENDS that routers have a subject name that includes a router-id identifying a router (or set of routers)
- Do we want to add router-id (or perhaps certificate subject name) as an unsigned field right next to SKI in the BGPSEC attribute?
- If we don't, there are potentially a large number of certs matching a given AS that must be sifted through be the validation code

3.3 Additional Info – Is it useful?

- Additional Info was put into the document as a signed origin-added blob of bits, back when we took out Expire_Time.
- At the time some members of the working group believed we may in the future want to add an Expire_Time or Signing_Time or something to support “better” protection against
- Is this blob of bits still useful?
- If so, is the current specification good enough?

3.4 Non-BGPSEC stuff in your AS

- The attribute is currently “non-transitive”
- This could be a problem if your route reflectors don't support BGPSEC
- Related issue: If you have both BGPSEC and non-BGPSEC edge routers, will you have issues with consistency of decisions?

3.5 Silly “Length” Question

- Does the value in the Length field include the octets used to express the Length field?
- Example: If I have 2 octets of Length field and 7 octets of other stuff, is the value in the Length field 9 or 7?

Part 4: SENDER PROCESSING

4.1 Originating a BGPSEC Update

- A router internal to your AS speaks BGPSEC. It originates an update message and sends it to your edge router. Does the update message it sends have an empty AS_PATH or an empty BGPSEC_PATH attribute?

Part 5: VALIDATION

5.1 Validation State Names

- Does anyone have a problem with “Valid” and “Invalid”?
- Do I need to add text explaining why there are two states?

5.2 Error Handling

- Currently the text says “Log that an error occurred and drop the update”
- Is this the right text? Is there something I should cite?
- If we chain back to RFC 4271, we get “close the session”, which probably isn't what we want

5.3 Deferring Validation

- Currently: “During exceptional conditions (e.g., the BGPSEC speaker receives an incredibly large number of update messages at once) a BGPSEC speaker MAY defer validation of incoming BGPSEC update messages. The treatment of such BGPSEC update messages, whose validation has been deferred, is a matter of local policy.”

5.3 Deferring Validation (continued)

- Currently: “Implementations that support such deferment of validation MUST perform validation of these messages as soon as possible (i.e., as soon as resources are available to perform validation) and MUST re-run best path selection once the validation status of such update”
- Is this text reasonable?
- Do we need additional guidance?

5.4 Edge Validation

- Currently: “BGPSEC validation needs only be performed at eBGP edge. The validation status of a BGP signed/unsigned update MAY be conveyed via iBGP from an ingress edge router to an egress edge router. Local policy in the AS determines the specific means for conveying the validation status through various pre-existing mechanisms (e.g., modifying some attribute). Based entirely on local policy settings, an egress router MAY trust the validation status conveyed by an ingress router or it MAY perform its own validation.”

5.5 Use of Validation State

- Do we want a mandate that you cannot pick non-BGPSEC for best path if you have BGPSEC valid? What about BGPSEC invalid?
  - I believe the answer is “No Mandate”
- Do we want to say anything at all about SHOULD prefer valid over valid?
- What about non-BGPSEC vs invalid?

5.6 Re-Running Validation

- Does there need to be text related to Re-Running the validation algorithm?
- Should implementations re-run validation occasionally because RPKI state may have changed?
- Should implementations re-run validation EVERY time RPKI state changes?
- Clearly, if you re-run validation and validity changes then you should re-run best path

5.7 draft-ymbk-rpki-rtr-keys

- Shall I include an informative reference to draft-ymbk-rpki-rtr-keys?
- Randy assures me that it will soon
    - I believe that this an extension to the rtr protocol to support bgpsec keys