IDR Interim, Flowspec v2

June 7, 2021

Agenda

The current state of things:

- Motivations for Flowspec v2
- Incremental deployment of new Flowspec features
- Interactions with Flowspec v1

Discussion:

- How do we deal with the incremental deployment issues?
- Once we understand incremental deployment, what do we do about the protocol?

Motivations for a Flowspec v2

- Original proposal draft-hares-idr-flowspec-v2-00
- Flowspec v1 routes were not safely "opaque"
  - RFC 8955 removed this wording
  - Move to a full TLV format
    (PCEP did this when they borrowed Flowspec encodings)
  - Introduces a new error modality: *Well-formed, but invalid*
- Several Flowspec use cases popular at the time required explicit ordering of rules.
  - Default sorting rules does well for DDoS
  - Default sorting rules problematic for some firewall optimizers

Motivations for a Flowspec v2 (2)

- Flowspec actions becoming more sophisticated with deeper interactions.
  - Draft for v2 hand-waves “throw it into a Wide Community”.
  - Stronger need for a way to remove ambiguity of behaviors when there’s different rules.
  - Flowspec interfaceset feature provides an example of sometimes match criteria end up outside NLRI
  - Conflicting types of redirection actions, for example.

Motivations for a Flowspec v2 (3)

- Encoding challenges for first two points potentially addressable in Flowspec v1
  - draft-haas-flowspec-capability-bits covers case for whether a given Flowspec speaker is capable of decoding a feature (component)
  - draft-haas-flowspec-term-order permits rule ordering in a simple way using existing encodings
- Do we possibly want to get rid of customer-injected eBGP Flow as a use case to remove route validation complexity?

Incremental deployment of new Flowspec features

IDR discussion related to the draft-haas features has helped quantify some of the incremental deployment issues:

1. Parsing the component, including new error handling requirements
2. Implementing the component in a filter
3. Rules with components that aren’t understood may result in “holes” in the Flowspec term set. If the rules aren’t “independent”, mis-filtering may happen.

The conditions above highlight that for incremental deployment purposes Flowspec domains may be segmented based on what components are supported within that domain. How does that impact the use of the mechanism?

Incremental deployment of new Flowspec features (2)

Not Discussed, but Important:

- Component ID for new components can be critical for default sorting of rules

Interactions with Flowspec v1

- Operationally, the question is motivated by Flowspec v1 permitting multiple BGP speakers in a Flowspec domain to inject routes.
  - This was one of the problematic scenarios for term ordering
- A given Flowspec speaker may have both Flowspec v1 (RFC 8955) routes, and routes with extensions. Those routes with extensions could be a different SAFI or same SAFI with different format.
  - Should the implemented filter be permitted to have contributors from both types?
  - If the filters from the types are intended to be segregated, how should they interact?