Critical Consideration of AS-Dot Notation

(aka “"The relative importance of backward compatibility for router configurations, compared with the shinyness of the new AS-dot notation")

Paul Jakma

2007-10-25

Overview

- AS-dot has subtle and significant near-term compatibility problems
- AS-dot introduces **permanent** awkwardness
- Any benefits AS-dot confers may be transient
- The above need to be weighed carefully, before we see serious AS4 deployment..

As-dot Motivations

“.. the larger values in the four-octet AS number set when using asplain notation introduces the increased risk of transcription error with these numbers.”

-- draft-michaelson-4byte-as-representation-04

Motivational Decomposition

- The rationale perhaps is speculative and subjective
- Assumes one form is easier to remember than the other
- Is 72277 really harder to cut&paste than 1.6741?
- What about 169482837 vs 2586.6741?
- As the numbers get bigger, they appear to get harder to remember, regardless of the dot
  - Further, the dot is of numerical significance..

Other conceptual questions

- Why the hybrid-number, using decimal numbers to describe words with non-decimal alignment, in a *flat* number-space?
  - Getting the dot wrong changes the number
- Aside: Hex would have been more compact, hence easier to remember, so perhaps would better have satisfied the motivation
  - 2586.6741 -> A1A.1A55
  - Saves another digit! ;)
  - Further, the dot potentially can be dropped without changing the meaning...

As-Dot Problems

'.' unfortunately has special meaning to regular expressions:

“*When querying with regular expressions for 4-byte ASNs, please don't forget to escape the dot. For example, '100.5' will match both AS100.5 and AS10015, while '100\.5' will only match AS100.5.*”

-- RIS LG

AS-Dot Problems (2)

“You may enter asnumbers in any format you like, as long as the context is unambigous.”

“... you potentially have to work on your regular expressions, i.e. if you used _[0-9]+_ in the asn16 world you have now to cope with _[0-9\.]+_ or similar.”

Juergen Kammer (AS4 Quagga site)

Regex Filters Compatibility

- BGP implementations commonly provide means to apply routing policy by filtering the AS_PATH, COMMUNITY and EXT_COMMUNITY attributes
  - Select paths with a regex against a string representation of some attribute
- Till now, AS numbers have been represented as their string of decimal digits, matched by [0-9]+
  - (but there are other ways to do it..)
- Under AS-dot, we'll need [0-9\.]+ to match an ASN

Regex Filters Compatibility (2)

- Existing filters shouldn't break against sub-65536 ASNs, even with AS-dot
- So any policy breakage due to AS-dot support may easily go unnoticed, initially.
- As more >65535 ASNs appear in attributes, the chance of breakage being hit increases
- Breakage may be subtle. E.g. TE policy breakage rather than lack of connectivity..

So What..

- “So what? It's just a dot. Deal with it..”
- The number of ASes with critical, regex-specified routing-policy may be insignificant (?)
  - Though likely far more significant than the current deployment of AS-dot (?)
- Let those affected just update their configurations

RPSL Compatibility

- RPSL specifies various ways to express routing policy (in forms similar to those common among BGP implementations). It defines an ASN as:
  - “*<as-number> An AS number x is represented as the string "ASx".*” -- RFC2622
- This is quite compatible with AS4, and indeed ASNs of any magnitude!
  - Existing implementations may or may not be buggy, as will be true with as-dot

RPSL Compatibility

- AS-dot is not compatible with existing RPSL though
- draft-uijterwaal-rpsl-4byteas-ext adds AS-Dot support
- Incompatible with RFC2622, in same way that BGP configuration regex filters are not forward-compatible across AS-dot
- It's not just RPSL parsing tools which must be updated, but RPSL policy ***data*** is rendered invalid..
  - Impact is unknown, of similar or lesser impact as with router configurations, presumably (?)

Comparison to “AS-Plain”

- “AS-plain” (i.e. normal numbers) is incompatible with:
  - ...
- Forward and backward compatible for:
  - RPSL
  - AS_PATH and (EXT_)?COMMUNITY filters
- Avoiding needless change means “stuff keeps working”

Policy & Consensus

- The accepted, draft RIPE Policy document 2005-12 sets out the initial allocation policy for 4-byte Ases
  - It assumes, but does not specify, the use of AS-dot for RIPE
- There is no wider consensus for adopting AS-dot notation
  - The draft failed to reach consensus at IETF
  - Reaction appeared mostly negative on NANOG

Summary

- The move to AS-dot is not quite free
  - BGP “vendors” can not offer both as-dot and configuration forward-compatibility
  - RPSL can't offer forward compatibility for policy data
- The impact is unknown
- However, there simply is no need to move to AS-dot
- No “rough consensus” in favour of AS-dot apparent to date

Discussion

Some suggestions:

“*The use of AS-dot by RIPE NCC, despite its problems and lack of industry or RIPE policy support, should be affirmed by the members, or the use should cease..*”

“*BGP implementations should deal with the configuration forward-compatibility problems by ...*”