Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: September 27, 2010

J. Scudder
Juniper Networks
E. Chen
Cisco Systems

March 26, 2010

Error Handling for Optional Transitive BGP Attributes
draft-ietf-idr-optional-transitive-02.txt

Abstract

According to the base BGP specification, a BGP speaker that receives an UPDATE message containing a malformed attribute is required to reset the session over which the offending attribute was received. This behavior is undesirable in the case of optional transitive attributes. This document revises BGP's error-handling rules for optional transitive attributes, and provides guidelines for the authors of documents defining new optional transitive attributes. It also introduces a new Path Attribute flag, Neighbor-Complete, to allow more accurate fault-finding. Finally, it revises the error handling procedures for several existing optional transitive attributes.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on September 27, 2010.

Copyright Notice

1. Introduction

According to the base BGP specification [RFC4271], a BGP speaker that receives an UPDATE message containing a malformed attribute is required to reset the session over which the offending attribute was received. This behavior is undesirable in the case of optional transitive attributes whose Partial flag is set; the reason is that such attributes may have been propagated without being checked by intermediate routers that do not recognize the attribute -- in effect the attributes may have been tunneled, and when they do reach a router that recognizes and checks them, the session that is reset may not be associated with the router that is at fault. This document revises BGP's error-handling rules for optional transitive attributes, and provides guidelines for the authors of documents defining new optional transitive attributes. It also revises the error handling procedures for several existing optional transitive attributes. Specifically, the error handling procedures of [RFC4271], [RFC1997], and [RFC4360] are revised.

Error handling procedures are not revised if the error can be imputed to the direct neighbor. A new flag, Neighbor-Complete, is introduced which, when used, allows the direct neighbor's involvement to be determined unequivocally. Imputation of "blame" to the direct neighbor is achieved by checking the Partial flag and the Neighbor-Complete flag. If the Partial flag is clear, or the Neighbor-Complete flag is set, the original error handling procedures remain in force.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Neighbor-Complete Flag Bit

It is desirable to know whether a neighbor recognizes, or does not recognize, a given optional transitive attribute. The Partial Path Attribute flag does not provide exactly this information -- it only enables the determination that a given neighbor did understand such an attribute, if the flag is set to zero. However, if the flag is set to one all that can be concluded is that some BGP speaker in the path did not understand the attribute, it cannot be determined whether the speaker in question was the neighbor or some other speaker.

To remedy this, we introduce a new Path Attribute Flag to those defined in [RFC4271] Section 4.3. The fifth high-order bit (bit 4) of the Attribute Flags octet is the Neighbor-Complete bit. It indicates whether the neighbor that sent the message recognizes the attribute (if set to one) or does not recognize it (if set to zero). The Neighbor-Complete flag only applies to optional transitive attributes. For other types of attributes the flag MUST be sent as zero and ignored when received.

A BGP speaker MUST set the Neighbor-Complete flag to one when sending a recognized, or zero when sending an unrecognized, optional transitive path attribute to its neighbor.

The Neighbor-Complete flag is the equivalent of the Partial flag, with two differences. First, it is reset on a hop-by-hop basis. Second, its "polarity" is reversed, with one instead of zero indicating that a neighbor does recognize the attribute. The reason for this difference is that during the period while this specification is being adopted, some BGP speakers will recognize the Neighbor-Complete flag and some will not. Since the previous definition [RFC4271] of bit 4 required it to be sent as zero, the use of one to mean "attribute recognized" allows the recipient of such a flag to unequivocally determine that a neighbor does recognize the given attribute.

Use of the flag on receipt is discussed in Section 3.

3. Revision to Base Specification

Section 6.3 of [RFC4271] is revised as follows. The paragraphs related to "any recognized attribute" and "an optional attribute" do not apply to optional transitive attributes received with their Partial flag set and Neighbor-Complete flag clear -- an error limited to such an attribute SHALL NOT be responded to by sending a NOTIFICATION message or resetting the BGP session. Instead, when

such an attribute is determined to be malformed, the UPDATE message containing that attribute SHOULD be treated as though all contained routes had been withdrawn just as if they had been listed in the WITHDRAWN ROUTES field of the UPDATE message, thus causing them to be removed from the Adj-RIB-In according to the procedures of [RFC4271]. In the case of an optional transitive attribute which has no effect on route selection or installation, the malformed attribute MAY instead be discarded and the UPDATE message continue to be processed.

An example of an attribute which has no effect on route selection or installation is the AGGREGATOR attribute.

A document which specifies an optional transitive attribute MUST provide specifics regarding what constitutes an error for that attribute and how that error is to be handled.

Note that the revised error handling only applies when an individual optional transitive attribute is received with its Partial flag set and Neighbor-Complete flag clear and deemed to be erroneous. In the event that an UPDATE message is deemed to be malformed in any other way then the procedures of [RFC4271] MUST be applied. This is likewise the case if an optional transitive attribute is received whose Partial flag is not set or whose Neighbor-Complete flag is set -- this is because the detected error can be imputed to the direct peer.

Examples of errors which would continue to be treated according to the procedures of [RFC4271] include the cases where the Total Attribute Length is inconsistent with the message length, or where there is more than one attribute with a given type code. Also, implicit in the foregoing paragraph is the fact that if due to an error, including those in an optional transitive attribute, the other attributes of the UPDATE message cannot be correctly parsed, then the procedures of [RFC4271] continue to apply.

In the specific case of incorrect path attribute flags -- i.e., a path attribute that is known by its type code to be Optional and Transitive but whose flags are not set accordingly -- the behavior specified by [RFC4271] SHALL be followed. (Consider that in the case of such an error, the "tunneling" argument given above does not apply, by definition.)

Finally, we observe that in order to treat an UPDATE as though all contained routes had been withdrawn as discussed above, the NLRI field and/or MP_REACH and MP_UNREACH [RFC4760] attributes need to be successfully parsed. If this were not possible, the UPDATE would necessarily be malformed in some way beyond the scope of this document and therefore, the procedures of [RFC4271] would continue to

apply.

4. Operational Considerations

Although the "treat as withdraw" error-handling behavior defined in Section 3 makes every effort to preserve BGP's correctness, we note that if an UPDATE received on an IBGP session is subjected to this treatment, inconsistent routing within the affected Autonomous System may result. The consequences of inconsistent routing can include long-lived forwarding loops and black holes. While lamentable, this issue is expected to be rare in practice, and more importantly is seen as less problematic than the session-reset behavior it replaces.

Even if inconsistent routing does not arise, the "treat as withdraw" behavior can cause either complete unreachability or sub-optimal routing for the destinations whose routes are carried in the affected UPDATE message.

Note that "treat as withdraw" is different from discarding an UPDATE message. The latter violates the basic BGP principle of incremental update, and could cause invalid routes to be kept. (See also Appendix A.)

For any malformed attribute which is discarded instead of the containing UPDATE being treated as a withdraw as discussed in Section 3, it is critical to consider the potential impact of doing so. In particular, if the attribute in question has or may have an effect on route selection or installation, the presumption is that discarding it is unsafe, unless careful analysis proves otherwise. The analysis should take into account the tradeoff between preserving connectivity and potential side effects.

Because of these potential issues, a BGP speaker MUST provide debugging facilities to permit issues caused by malformed optional transitive attributes to be diagnosed. At a minimum, such facilities SHOULD include logging an error when such an attribute is detected.

5. Error Handling Procedures for Existing Optional Transitive Attributes

5.1. AGGREGATOR

The error handling of [RFC4271] is revised as follows:

The AGGREGATOR attribute SHALL be considered malformed if any of the following applies:

- Its length is not 6 (when the "4-octet AS number capability" is not advertised to, or not received from the peer [RFC4893]).

- Its length is not 8 (when the "4-octet AS number capability" is both advertised to, and received from the peer).

An UPDATE message with a malformed AGGREGATOR attribute SHALL be handled as follows. If its Partial flag is set and its Neighbor-Complete flag is clear, either the attribute MUST be discarded or the UPDATE containing it treated as a withdraw as discussed in Section 3. Otherwise (i.e. if its Partial flag is clear or its Neighbor-Complete flag is set), the procedures of [RFC4271] MUST be followed with respect to an Optional Attribute Error.

5.2. Community

The error handling of [RFC1997] is revised as follows:

The Community attribute SHALL be considered malformed if its length is not a nonzero multiple of 4.

An UPDATE message with a malformed Community attribute SHALL be handled as follows. If its Partial flag is set and its Neighbor-Complete flag is clear, the update containing it MUST be treated as a withdraw as discussed in Section 3. Otherwise (i.e. if its Partial flag is clear or its Neighbor-Complete flag is set), the procedures of [RFC4271] MUST be followed with respect to an Optional Attribute Error.

5.3. Extended Community

The error handling of [RFC4360] is revised as follows:

The Extended Community attribute SHALL be considered malformed if its length is not a nonzero multiple of 8.

An UPDATE message with a malformed Extended Community attribute SHALL be handled as follows. If its Partial flag is set and its Neighbor-Complete flag is clear, the update containing it MUST be treated as a withdraw as discussed in Section 3. Otherwise (i.e. if its Partial flag is clear or its Neighbor-Complete flag is set), the procedures of [RFC4271] MUST be followed with respect to an Optional Attribute Error.

Note that a BGP speaker MUST NOT treat an unrecognized Extended Community Type or Sub-Type as an error.

6. Security Considerations

This specification addresses the vulnerability of a BGP speaker to a potential attack whereby a distant attacker can generate a malformed optional transitive attribute that is not recognized by intervening routers (which thus propagate the attribute unchecked) but that causes session resets when it reaches routers that do recognize the given attribute type.

In other respects, this specification does not change BGP's security characteristics.

7. Acknowledgements

The authors wish to thank Ron Bonica, Andy Davidson, Dong Jie, Rex Fernando, Joel Halpern, Akira Kato, Miya Kohno, Alton Lo, Shin Miyakawa, Jonathan Oddy, Robert Raszuk, Yakov Rekhter, Rob Shakir, Ananth Suryanarayana, and Kaliraj Vairavakkalai for their observations and discussion of this topic. The Neighbor-Complete flag was introduced as the result of helpful discussion with Jie Dong and Mach Chen.

8. IANA Considerations

IANA is requested to establish and maintain a registry of BGP Path Attribute Flags. Flags one through four are defined in [RFC4271]. Flag five is defined in Section 2 of this document. Future allocations are to be made according to the IETF Standards Action policy [RFC5226].

9. References

9.1. Normative References

[RFC1997] Chandrasekeran, R., Traina, P., and T. Li, "BGP Communities Attribute", RFC 1997, August 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC4360] Sangli, S., Tappan, D., and Y. Rekhter, "BGP Extended Communities Attribute", RFC 4360, February 2006.

[RFC4893] Vohra, Q. and E. Chen, "BGP Support for Four-octet AS Number Space", RFC 4893, May 2007.

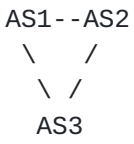
[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

9.2. Informative References

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, January 2007.

Appendix A. Why not discard UPDATES?

A commonly asked question is "why not simply discard the UPDATE message instead of treating it like a withdraw? Isn't that safer and easier?" The answer is that it might be easier, but it would compromise BGP's correctness so is unsafe. Consider the following example of what might happen if UPDATE messages carrying bad attributes were simply discarded:



- AS1 prefers to reach AS3 directly, and advertises its route to AS2.
- AS2 prefers to reach AS3 directly, and advertises its route to AS1.
- Connections AS3-AS1 and AS3-AS2 fail simultaneously.
- AS1 switches to prefer AS2's route, and sends an update message which includes a withdraw of its previous announcement. The withdraw is bundled with some advertisements. It includes a bad attribute. As a result, AS2 ignores the message.
- AS2 switches to prefer AS1's route, and sends an update message which includes a withdraw of its previous announcement. The withdraw is bundled with some advertisements. It includes a bad attribute. As a result, AS1 ignores the message.

The end result is that AS1 forwards traffic for AS3 towards AS2, and AS2 forwards traffic for AS3 towards AS1. This is a permanent (until

corrected) forwarding loop.

Although the example above discusses route withdraws, we observe that in BGP the announcement of a route also withdraws the route previously advertised. The implicit withdraw can be converted into a real withdraw in a number of ways; for example, the previously-announced route might have been accepted by policy, but the new announcement might be rejected by policy. For this reason, the same concerns apply even if explicit withdraws are removed from consideration.

Authors' Addresses

John G. Scudder
Juniper Networks

Email: jgs@juniper.net

Enke Chen
Cisco Systems

Email: enkechen@cisco.com