SIDR W. George
Internet-Draft
Intended status: Standards Track S. Murphy
Expires: June 10, 2017 SPARTA, Inc., a Parsons Company
December 7, 2016

BGPSec Considerations for AS Migration
draft-ietf-sidr-as-migration-06

Abstract

This document discusses considerations and methods for supporting and securing a common method for AS-Migration within the BGPSec protocol.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on June 10, 2017.

Copyright Notice

Table of Contents

1. Introduction

A method of managing a BGP Autonomous System Number (ASN) migration is described in RFC7705 [RFC7705]. Since it concerns the handling of AS_PATH attributes, it is necessary to ensure that the process and features are properly supported in BGPSec [I-D.ietf-sidr-bgpsec-protocol], because BGPSec is explicitly designed to protect against changes in the BGP AS_PATH, whether by choice, by misconfiguration, or by malicious intent. It is critical that the BGPSec protocol framework is able to support this operationally necessary tool without creating an unacceptable security risk or exploit in the process.

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].

1.2. Documentation note

This document uses Autonomous System Numbers (ASNs) from the range reserved for documentation as described in RFC 5398 [RFC5398]. In the examples used here, they are intended to represent Globally Unique ASNs, not ASNs reserved for private use as documented in RFC 1930 [RFC1930] section 10.

2. General Scenario

This document assumes that the reader has read and understood the ASN migration method discussed in RFC7705 [RFC7705] including its examples (see section 2 of the referenced document), as they will be heavily referenced here. The use case being discussed in the referenced document is as follows: For whatever the reason, a provider is in the process of merging two or more ASes, where eventually one subsumes the other(s). BGP AS Confederations RFC 5065 [RFC5065] is not enabled between the ASes, but a mechanism is being used to modify BGP's default behavior and allow the migrating Provider Edge router (PE) to masquerade as the old ASN for the Provider Edge to Customer Edge (PE-CE) eBGP session, or to manipulate the AS_PATH, or both. While BGPSec [I-D.ietf-sidr-bgpsec-protocol] does have a method to handle standard confederation implementations, it is not applicable in this exact case. This migration requires a slightly different solution in BGPSec than for a standard confederation because unlike in a confederation, eBGP peers may not be peering with the "correct" external ASN, and the forward-signed updates are for a public ASN, rather than a private one, so there is no expectation that the BGP speaker would strip the affected signatures before propagating the route to its eBGP neighbors.

In the following examples (section 5.4) (Section 5.4), AS64510 is being subsumed by AS64500, and both ASNs represent a Service Provider (SP) network (see Figures 1 & 2 in RFC7705 [RFC7705]). AS64496 and 64499 represent end customer networks. References to PE, CE, and P routers mirror the diagrams and references in the above cited draft.

3. RPKI Considerations

The methods and implementation discussed in RFC7705 [RFC7705] are widely used during network integrations resulting from mergers and acquisitions, as well as network redesigns, and therefore it is necessary to support this capability on any BGPSec-enabled routers/ASNs. What follows is a discussion of the potential issues to be considered regarding how ASN-migration and BGPSec [I-D.ietf-sidr-bgpsec-protocol] validation might interact.

One of the primary considerations for this document and migration is that service providers (SPs) rarely stop after one merger/acquisition/divestiture, and end up accumulating several legacy ASNs over time. Since they are using methods to migrate that are transparent to and therefore do not require coordination with customers, they do not have a great deal of control over the length of the transition period as they might with something completely under their administrative control (e.g. a key roll). Because they are not forcing a simultaneous migration (i.e. both ends switch to the new ASN at an agreed-upon time), there is no incentive for a given customer to complete the move from the old ASN to the new. This leaves many SPs with multiple legacy ASNs which don't go away very quickly, if at all. As solutions were being proposed for RPKI implementations to solve this transition case, the WG carefully considered operational complexity and hardware scaling issues associated with maintaining multiple legacy ASN keys on routers throughout the combined network. While SPs who choose to remain in this transition phase indefinitely invite added risks because of the operational complexity and scaling considerations associated with maintaining multiple legacy ASN keys on routers throughout the combined network, saying "don't do this" is of limited utility as a solution. As a result, this solution attempts to minimize the additional complexity during the transition period, on the assumption that it will likely be protracted. Note: While this document primarily discusses service provider considerations, it is not solely applicable to SPs, as enterprises often migrate between ASNs using the same functionality. What follows is a discussion of origin and path validation functions and how they interact with ASN migrations.

3.1. Origin Validation

Route Origin Validation as defined by RFC 6480 [RFC6480] does not modification to enable AS migration, as the existing protocol and procedure allows for a solution. In the scenario discussed in RFC 7705 [RFC7705], AS64510 is being replaced by AS64500. If there are any existing routes originated by AS64510 on the router being moved into the new ASN, this simply requires generating new Route Origination Authorizations (ROAs) for the routes with the new ASN and treating them as new routes to be added to AS64500. However, we also need to consider the situation where one or more other PEs are still in AS64510, and are originating one or more routes that may be distinct from any that the router under migration is originating. PE1 (which is now a part of AS64500 and instructed to use Replace Old AS as defined in RFC 7705 [RFC7705] to remove AS64510 from the path) needs to be able to properly handle routes originated from AS64510. If the route now shows up as originating from AS64500, any downstream peers' validation check will fail unless a ROA is *also* available for AS64500 as the origin ASN. In addition to generating a ROA for

65400 for any prefixes originated by the router being moved, it may be necessary to generate ROAs for 65400 for prefixes that are originating on routers still in 65410, since the AS replacement function will change the origin AS in some cases. This means that there will be multiple ROAs showing different ASes authorized to orignate the same prefixes until all routers originating prefixes from AS64510 are migrated to AS64500. Multiple ROAs of this type are permissible per RFC 6480 [RFC6480] section 3.2, and so managing origin validation during a migration like this is merely applying the defined case where a set of prefixes are originated from more than one ASN. Therefore, for each ROA that authorizes the old ASN (e.g. AS64510) to originate a prefix, a new ROA MUST also be created that authorizes the replacing ASN (e.g. AS64500) to originate the same prefix.

3.2. Path Validation

BGPSec Path Validation requires that each router in the AS Path cryptographically sign its update to assert that "Every AS on the path of ASes listed in the update message has explicitly authorized the advertisement of the route to the subsequent AS in the path." (see intro of [I-D.ietf-sidr-bgpsec-protocol]) Since the referenced

AS migration technique is explicitly modifying the AS_PATH between two eBGP peers who are not coordinating with one another (are not in the same administrative domain), no level of trust can be assumed, and therefore it may be difficult to identify legitimate manipulation of the AS_PATH for migration activities when compared to manipulation due to misconfiguration or malicious intent.

3.2.1. Outbound announcements (PE-->CE)

When PE1 is moved from AS64510 to AS64500, it will be provisioned with the appropriate keys for AS64500 to allow it to forward-sign routes using AS64500. However, there is no guidance in the BGPSec protocol specification [I-D.ietf-sidr-bgpsec-protocol] on whether or not the forward-signed ASN value is required to match the configured remote AS to validate properly. That is, if CE1's BGP session is configured as "remote AS 64510", the presence of "local AS 64510" on PE1 will ensure that there is no ASN mismatch on the BGP session itself, but if CE1 receives updates from its remote neighbor (PE1) forward-signed from AS64500, there is no guidance as to whether the BGPSec validator on CE1 still considers those valid by default. RFC4271 [RFC4271] section 6.3 mentions this match between the ASN of the peer and the AS_PATH data, but it is listed as an optional validation, rather than a requirement. We cannot assume that this mismatch will be allowed by vendor implementations and thus using it as a means to solve this migration case is likely to be problematic.

3.2.2. Inbound announcements (CE-->PE)

Inbound is more complicated, because the CE doesn't know that PE1 has changed ASNs, so it is forward-signing all of its routes with AS64510, not AS64500. The BGPSec speaker cannot manipulate previous signatures, and therefore cannot manipulate the previous AS Path without causing a mismatch that will invalidate the route. If the updates are simply left intact, the ISP would still need to publish and maintain valid and active public-keys for AS 64510 if it is to appear in the BGPSec_Path_Signature in order that receivers can validate the BGPSEC_Path_Signature arrived intact/whole. However, if the updates are left intact, this will cause the AS Path length to be increased, which is unacceptable as discussed in RFC7705 [RFC7705].

4. Requirements

In order to be deployable, any solution to the described problem needs to consider the following requirements, listed in no particular order. BGPSec:

- MUST support AS Migration for both inbound and outbound route announcements (see Section 3.2.1 and 3.2.2) without reducing BGPSec's protections for route path
- MUST NOT require any reconfiguration on the remote eBGP neighbor (CE)
- SHOULD NOT require global (i.e. network-wide) configuration changes to support migration. The goal is to limit required configuration changes to the devices (PEs) being migrated.
- MUST NOT lengthen AS Path during migration
- MUST operate within existing trust boundaries e.g. can't expect remote side to accept pCount=0 (see Section 4.2 of [I-D.ietf-sidr-bgpsec-protocol]) from untrusted/non-confed neighbor

5. Solution

As noted in [I-D.ietf-sidr-bgpsec-protocol], section 4.2, BGPSec already has a solution for hiding ASNs where increasing the AS Path length is undesirable. So a simple solution would be to retain the keys for AS64510 on PE1, and forward-sign towards CE1 with AS64510 and pCount=0. However, this would mean passing a pCount=0 between two ASNs that are in different administrative and trust domains such that it could represent a significant attack vector to manipulate BGPSec-signed paths. The expectation for legitimate instances of

pCount=0 (to make a route-server that is not part of the transit path invisible) is that there is some sort of existing trust relationship between the operators of the route-server and the downstream peers such that the peers could be explicitly configured by policy to accept pCount=0 announcements only on the sessions where they are expected. For the same reason that things like "Local AS" [RFC7705] are used for ASN migration without end customer coordination, it is unrealistic to assume any sort of coordination between the SP and the

administrators of CE1 to ensure that they will by policy accept pCount=0 signatures during the transition period, and therefore this is not a workable solution.

A better solution presents itself when considering how to handle routes coming from the CE toward the PE, where the routes are forward-signed to AS64510, but will eventually need to show AS64500 in the outbound route announcement. Because both AS64500 and AS64510 are in the same administrative domain, a signature from AS64510 forward-signed to AS64500 with pCount=0 would be acceptable as it would be within the appropriate trust boundary so that each BGP speaker could be explicitly configured to accept pCount=0 where appropriate between the two ASNs. At the very simplest, this could potentially be used at the eBGP boundary between the two ASNs during migration. Since the AS_PATH manipulation described above usually happens at the PE router on a per-session basis, and does not happen network-wide simultaneously, it is not generally appropriate to apply this AS hiding technique across all routes exchanged between the two ASNs, as it may result in routing loops and other undesirable behavior. Therefore the most appropriate place to implement this is on the local PE that still has eBGP sessions with peers expecting to peer with AS64510 (using the transition mechanisms detailed in RFC7705 [RFC7705]). Since that PE has been moved to AS64500, it is not possible for it to forward-sign AS64510 with pCount=0 without some minor changes to the BGPSec behavior to address this use case.

AS migration is using AS_PATH and remote AS manipulation to act as if a PE under migration exists simultaneously in both ASNs even though it is only configured with one global ASN. This document describes applying a similar technique to the BGPSec signatures generated for routing updates processed through this migration machinery. Each routing update that is received from or destined to an eBGP neighbor that is still using the old ASN (64510) will be signed twice, once with the ASN to be hidden and once with the ASN that will remain visible. In essence, we are treating the update as if the PE had an internal BGP hop and the update was passed across an eBGP session between AS64500 and AS64510, configured to use and accept pCount=0, while eliminating the processing and storage overhead of creating an actual eBGP session between the two ASNs within the PE router. This will result in a properly secured AS Path in the affected route

updates, because the PE router will be provisioned with valid keys

for both AS64500 and AS64510. An important distinction here is that while AS migration under standard BGP4 is manipulating the AS_PATH attribute, BGPSec uses an attribute called the Secure_Path (see Section 3.1 of [I-D.ietf-sidr-bgpsec-protocol]), and BGPSec capable neighbors do not exchange AS_PATH information in their route announcements. However, a BGPSec neighbor peering with a non-BGPSec-capable neighbor will use the information found in Secure_Path to reconstruct a standard AS_PATH for updates sent to that neighbor. Unlike in Secure_Path where the ASN to be hidden is still present, but ignored when considering AS Path (due to pCount=0), when reconstructing an AS_PATH for a non-BGPSec neighbor, the pCount=0 ASNs will not appear in the AS_PATH at all (see section 4.4 of the [I-D.ietf-sidr-bgpsec-protocol]). This document is not changing existing AS_PATH reconstruction behavior, merely highlighting it for clarity.

The procedure to support AS Migration in BGPSec is slightly different depending on whether the PE under migration is receiving the routes from one of its eBGP peers ("inbound" as in section 3.2.2) or destined toward the eBGP peers ("outbound" as in section 3.2.1).

5.1. Outbound (PE->CE)

When a PE router receives an update destined for an eBGP neighbor that is locally configured with AS-migration mechanisms as discussed in RFC7705 [RFC7705], it MUST generate a valid BGPSec signature as defined in [I-D.ietf-sidr-bgpsec-protocol] for _both_ configured ASNs. It MUST generate a signature from the new (global) ASN forward signing to the old (local) ASN with pCount=0, and then it MUST generate a forward signature from the old (local) ASN to the target eBGP ASN with pCount=1 as normal.

5.2. Inbound (CE->PE)

When a PE router receives an update from an eBGP neighbor that is locally configured with AS-migration mechanisms (i.e. the opposite direction of the previous route flow), it MUST generate a signature from the old (local) ASN forward signing to the new (global) ASN with pCount=0. It is not necessary to generate the second signature from the new (global) ASN because the Autonomous System Border Router (ASBR) will generate that when it forward signs towards its eBGP peers as defined in normal BGPSec operation. Note that a signature is not normally added when a routing update is sent across an iBGP session. The requirement to sign updates in iBGP represents a change to the normal behavior for this specific AS-migration scenario only.

5.3. Other considerations

In this case, the PE is adding BGPSec attributes to routes received from or destined to an iBGP neighbor, and using pCount=0 to mask them. While this is not prohibited by BGPSec [I-D.ietf-sidr-bgpsec-protocol], BGPSec-capable routers that receive updates from BGPSec-enabled iBGP neighbors MUST accept updates with new (properly-formed) BGPSec attributes, including the presence of pCount=0 on a previous signature, or they will interfere with this method. In similar fashion, any BGPSec-capable route-reflectors in the path of these updates MUST reflect them transparently to their BGPSec-capable clients.

In order to secure this set of signatures, the PE router MUST be provisioned with valid keys for _both_ configured ASNs (old and new), and the key for the old ASN MUST be kept valid until all eBGP sessions are migrated to the new ASN. Downstream neighbors will see this as a valid BGPSec path, as they will simply trust that their upstream neighbor accepted pCount=0 because it was explicitly configured to do so based on a trust relationship and business relationship between the upstream and its neighbor (the old and new ASNs).

Additionally, section 4 of RFC7705 [RFC7705] discusses methods in which AS migrations can be completed for iBGP peers such that a session between two routers will be treated as iBGP even if the neighbor ASN is not the same ASN on each peer's global configuration. As far as BGPSec is concerned, this requires the same procedure as when the routers migrating are applying AS migration mechanisms to eBGP peers, but the router functioning as the "ASBR" between old and new ASN is different. In eBGP, the router being migrated has direct eBGP sessions to the old ASN and signs from old ASN to new with pCount=0 before passing the update along to additional routers in its global (new) ASN. In iBGP, the router being migrated is receiving updates (that may have originated either from eBGP neighbors or other iBGP neighbors) from its downstream neighbors in the old ASN, and MUST sign those updates from old ASN to new with pCount=0 before sending them on to other peers.

5.4. Example

The following example will illustrate the method being used above. As with previous examples, PE1 is the router being migrated, AS64510 is the old ASN, which is being subsumed by AS64500, the ASN to be permanently retained. 64505 is another external peer, used to demonstrate what the announcements will look like to a third party

peer that is not part of the migration. Some additional notation is used to delineate the details of each signature as follows:

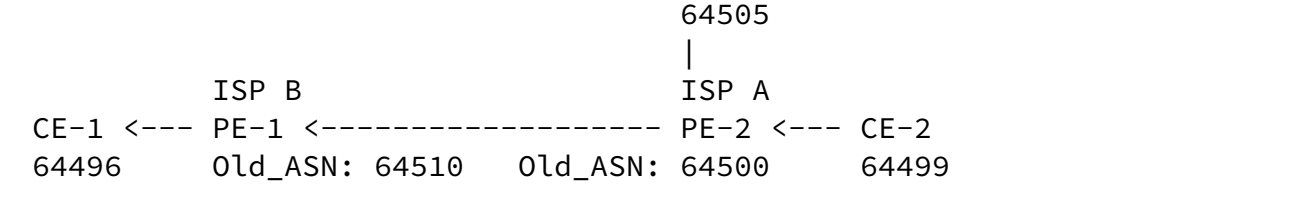
The origin BGPSEC signature attribute takes the form: sig(<Target ASN>, Origin ASN, pCount, NLRI Prefix) key

Intermediate BGPSEC signature attributes take the form: sig(<Target ASN>, Signer ASN, pCount, <most recent sig field>) key

Equivalent AS_PATH refers to what the AS_PATH would look like if it was reconstructed to be sent to a non-BGPSec peer, while Secure_Path shows the AS Path as represented between BGPSec peers.

Note: The representation of signature attribute generation is being simplified here somewhat for the sake of brevity; the actual details of the signing process are as described Sections 4.1 and 4.2 in [I-D.ietf-sidr-bgpsec-protocol]. For example, what is covered by the signature also includes Flags, Algorithm Suite ID, NLRI length, etc. Also, the key is not carried in the update, instead the SKI is carried.

Before Merger



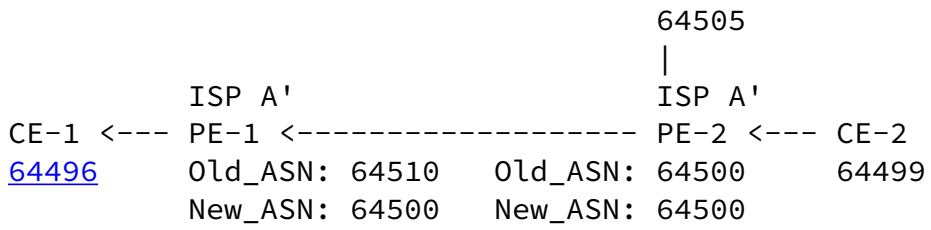
CE-2 to PE-2: sig(<64500>, O=64499, pCount=1, N)K_64499-CE2 [sig1]
Equivalent AS_PATH=(64499)
Secure_Path=(64499)
length=sum(pCount)=1

PE-2 to 64505: sig(<64505>, 64500, pCount=1, <sig1>)K_64500-PE2 [sig2]
sig(<64500>, 64499, pCount=1, N)K_64499-CE2 [sig1]
Equivalent AS_PATH=(64500,64499)
Secure_Path=(64500,64499)
length=sum(pCount)=2

PE-2 to PE-1: sig(<64510>, 64500, pCount=1, <sig1>)K_64500-PE2 [sig3]
sig(<64500>, 64499, pCount=1, N)K_64499-CE2 [sig1]
Equivalent AS_PATH=(64500,64499)
Secure_Path=(64500,64499)
length=sum(pCount)=2

PE-1 to CE-1: sig(<64496>, 64510, pCount=1, <sig3>)K_64510-PE1 [sig4]
sig(<64510>, 64500, pCount=1, <sig1>)K_64500-PE2 [sig3]
sig(<64500>, 64499, pCount=1, N)K_64499-CE2 [sig1]
Equivalent AS_PATH= (64510,64500,64499)
Secure_Path=(64510,64500,64499)
length=sum(pCount)=3

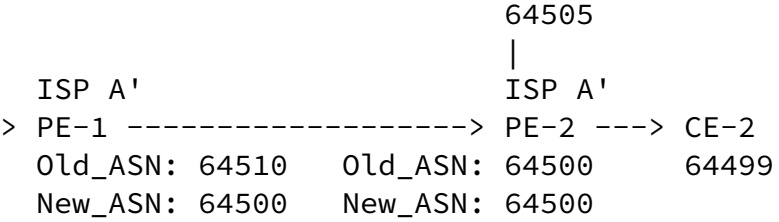
Migrating, route flow outbound PE-1 to CE-1




```
CE-2 to PE-2:  sig(<64500>, 64499, pCount=1, N)K_64499-CE2  [sig11]
               Equivalent AS_PATH=(64499)
               Secure_Path=(64499)
               length=sum(pCount)=1

PE-2 to 64505: sig(<64505>, 64500, pCount=1, <sig11>)K_64500-PE2  [sig12]
               sig(<64500>, 64499, pCount=1, N)K_64499-CE2  [sig11]
               Equivalent AS_PATH=(64500,64499)
               Secure_Path=(64500,64499)
               length=sum(pCount)=2

PE-2 to PE-1:  sig(<64500>, 64499, pCount=1, N)K_64499-CE2  [sig11]
               Equivalent AS_PATH=(64499)
               Secure_Path=(64499)
               length=sum(pCount)=1
```

```
#PE-2 sends to PE-1 (in iBGP) the exact same update
#as received from AS64499.


PE-1 to CE-1:  sig(<64496>, 64510, pCount=1, <sig13>)K_64510-PE1  [sig14]
               sig(<64510>, 64500, pCount=0, <sig11>)K_64500-PE2  [sig13]
               sig(<64500>, 64499, pCount=1, N)K_64499-CE2  [sig11]
               Equivalent AS_PATH=(64510,64499)
               Secure_Path=(64510, 64500(pCount=0),64499)
               length=sum(pCount)=2 (length is NOT 3)
#PE1 adds [sig13] acting as AS64500
#PE1 accepts [sig13] with pCount=0 acting as AS64510,
#as it would if it received sig13 from an eBGP peer
```


Migrating, route flow inbound CE-1 to PE-1




```
CE-1 to PE-1:  sig(<64510>, 64496, pCount=1, N)K_64496-CE1   [sig21]
               Equivalent AS_PATH=(64496)
               Secure_Path=(64496)
               length=sum(pCount)=1

PE-1 to PE-2:  sig(<64500>, 64510, pCount=0, <sig21>)K_64510-PE1  [sig22]
               sig(<64510>, 64496, pCount=1, N)K_64496-CE1   [sig21]
               Equivalent AS_PATH=(64496)
```

```
                Secure_Path=(64510 (pCount=0),64496)
                length=sum(pCount)=1 (length is NOT 2)
#PE1 adds [sig22] acting as AS64510
#PE1 accepts [sig22] with pCount=0 acting as AS64500,
#as it would if it received sig22 from an eBGP peer

PE-2 to 64505: sig(<64505>, 64500, pCount=1, <sig22>)K_64500-PE2  [sig23]
                sig(<64500>, 64510, pCount=0, <sig21>)K_64510-PE1  [sig22]
                sig(<64510>, 64496, pCount=1, N)K_64496-CE1   [sig21]
                Equivalent AS_PATH=(64500,64496)
                Secure_Path=(64500,64510 (pCount=0), 64496)
                length=sum(pCount)=2 (length is NOT 3)

PE-2 to CE-2:  sig(<64499>, 64500, pCount=1, <sig22>)K_64500-PE2  [sig24]
                sig(<64500>, 64510, pCount=0, <sig21>)K_64510-PE1  [sig22]
                sig(<64510>, 64496, pCount=1, N)K_64496-CE1   [sig21]
                Equivalent AS_PATH=(64500,64496)
                Secure_Path=(64500, 64510 (pCount=0), 64496)
                length=sum(pCount)=2 (length is NOT 3)
```

6. Acknowledgements

Thanks to Kotikalapudi Sriram, Shane Amante, Warren Kumari, Terry Manderson, Keyur Patel, Alia Atlas, and Alvaro Retana for their review comments.

Additionally, the solution presented in this document is an amalgam of several SIDR interim meeting discussions plus a discussion at IETF85, collected and articulated thanks to Sandy Murphy.

7. IANA Considerations

This memo includes no request to IANA.

8. Note for RFC Editor

This section can be removed prior to publication.

RFC Editor - this document updates draft-ietf-sidr-bgpsec-protocol, but the normal Updates= metadata method cannot be used until an RFC number is assigned to the document being updated. Please ensure that

the metadata is corrected when the bgpsec-protocol document has been assigned an RFC number.

9. Security Considerations

RFC7705 [RFC7705] discusses a process by which one ASN is migrated into and subsumed by another. Because this process involves manipulating the AS_Path in a BGP route to make it deviate from the actual path that it took through the network, this migration process is attempting to do exactly what BGPSec is working to prevent. BGPSec MUST be able to manage this legitimate use of AS_Path manipulation without generating a vulnerability in the RPKI route security infrastructure, and this document was written to define the method by which the protocol can meet this need.

The solution discussed above is considered to be reasonably secure from exploitation by a malicious actor because it requires both signatures to be secured as if they were forward-signed between two eBGP neighbors. This requires any router using this solution to be provisioned with valid keys for both the migrated and subsumed ASN so that it can generate valid signatures for each of the two ASNs it is adding to the path. If the AS's keys are compromised, or zero-length keys are permitted, this does potentially enable an AS_PATH shortening attack, but these are existing security risks for BGPSec.

10. References

10.1. Normative References

[I-D.ietf-sidr-bgpsec-protocol]
Lepinski, M. and K. Sriram, "BGPsec Protocol Specification", draft-ietf-sidr-bgpsec-protocol-20 (work in progress), December 2016.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC7705] George, W. and S. Amante, "Autonomous System Migration Mechanisms and Their Effects on the BGP AS_PATH Attribute", RFC 7705, DOI 10.17487/RFC7705, November 2015, <http://www.rfc-editor.org/info/rfc7705>.

10.2. Informative References

[RFC1930] Hawkinson, J. and T. Bates, "Guidelines for creation, selection, and registration of an Autonomous System (AS)", BCP 6, RFC 1930, DOI 10.17487/RFC1930, March 1996, <http://www.rfc-editor.org/info/rfc1930>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <http://www.rfc-editor.org/info/rfc4271>.

[RFC5065] Traina, P., McPherson, D., and J. Scudder, "Autonomous System Confederations for BGP", RFC 5065, DOI 10.17487/RFC5065, August 2007, <http://www.rfc-editor.org/info/rfc5065>.

[RFC5398] Huston, G., "Autonomous System (AS) Number Reservation for Documentation Use", RFC 5398, DOI 10.17487/RFC5398, December 2008, <http://www.rfc-editor.org/info/rfc5398>.

[RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, DOI 10.17487/RFC6480, February 2012, <http://www.rfc-editor.org/info/rfc6480>.

Authors' Addresses

Wesley George

Email: wesgeorge@puck.nether.net

Sandy Murphy
SPARTA, Inc., a Parsons Company
7110 Samuel Morse Drive
Columbia, MD 21046
US

Phone: +1 443-430-8000
Email: sandy@tislabs.com