

Workgroup: Network Working Group
Internet-Draft:
draft-ietf-sidrops-aspa-verification-17
Published: 29 February 2024
Intended Status: Standards Track
Expires: 1 September 2024

Authors: A. Azimov E. Bogomazov R. Bush K. Patel
 Yandex Qrator Labs IIJ & Arrcus Arrcus
 J. Snijders K. Sriram
 Fastly USA NIST

**BGP AS_PATH Verification Based on Autonomous System Provider
Authorization (ASPA) Objects**

Abstract

This document describes procedures that make use of Autonomous System Provider Authorization (ASPA) objects in the Resource Public Key Infrastructure (RPKI) to verify the Border Gateway Protocol (BGP) AS_PATH attribute of advertised routes. This type of AS_PATH verification provides detection and mitigation of route leaks and improbable AS paths. It also provides protection, to some degree, against prefix hijacks with forged-origin or forged-path-segment.

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 <https://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 1 September 2024.

Copyright Notice

Copyright (c) 2024 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

- [1. Introduction](#)
 - [1.1. Anomaly Propagation](#)
 - [1.2. Terminology](#)
 - [1.3. Requirements Language](#)
- [2. BGP Roles](#)
- [3. Autonomous System Provider Authorization](#)
- [4. ASPA Registration Recommendations](#)
- [5. Hop-Check Function](#)
- [6. AS PATH Verification](#)
 - [6.1. Algorithm for Upstream Paths](#)
 - [6.2. Algorithm for Downstream Paths](#)
 - [6.2.1. Principles for Determination of Invalid, Valid, and Unknown in Downstream Path Verification \(for \$N \geq 3\$ \)](#)
 - [6.2.2. Formal Procedure for Verification of Downstream Paths](#)
- [7. AS PATH Verification and Anomaly Mitigation Recommendations](#)
 - [7.1. Verification and Mitigation at Ingress eBGP Router](#)
 - [7.2. Only to Customer \(OTC\) Attribute](#)
- [8. Properties of ASPA-based Path Verification](#)
- [9. Operational Considerations](#)
 - [9.1. 4-Byte AS Number Requirement](#)
 - [9.2. Correctness of the ASPA](#)
 - [9.3. Make Before Break](#)
 - [9.4. DoS/DDoS Mitigation Service Provider](#)
- [10. Comparison to Other Technologies](#)
 - [10.1. BGPsec](#)
 - [10.2. Peerlock](#)
- [11. IANA Considerations](#)
- [12. Security Considerations](#)
- [13. Implementation Status](#)
- [14. References](#)
 - [14.1. Normative References](#)
 - [14.2. Informative References](#)
- [Appendix A. Acknowledgments](#)
- [Contributors](#)
- [Authors' Addresses](#)

1. Introduction

The Border Gateway Protocol (BGP) as originally designed is known to be vulnerable to prefix (or route) hijacks and BGP route leaks [[RFC7908](#)]. Some existing BGP extensions are able to partially solve

these problems. For example, Resource Public Key Infrastructure (RPKI) based route origin validation (RPKI-ROV) [[RFC6480](#)] [[RFC6482](#)] [[RFC6811](#)] [[RFC9319](#)] can be used to detect and filter accidental mis-originations. [[RFC9234](#)] or [[I-D.ietf-grow-route-leak-detection-mitigation](#)] can be used to detect and mitigate accidental route leaks. While RPKI-ROV can prevent accidental prefix hijacks, malicious forged-origin prefix hijacks can still occur [[RFC9319](#)]. RFC9319 includes some recommendations for reducing the attack surface for forged-origin prefix hijacks.

This document describes procedures that make use of Autonomous System Provider Authorization (ASPA) objects [[I-D.ietf-sidrops-aspa-profile](#)] in the RPKI to verify properties of the BGP AS_PATH attribute of advertised routes. ASPA-based AS_PATH verification provides detection and mitigation of route leaks and improbable AS paths. It also provides protection, to some degree, against prefix hijacks with forged-origin or forged-path-segment ([Section 8](#)). These new ASPA-based procedures automatically detect such anomalous AS_PATHs in announcements that are received from customers, lateral peers (defined in [[RFC7908](#)]), transit providers, IXP Route Servers (RS), RS-clients, and mutual-transits. The protections provided by these procedures (together with RPKI-ROV) are based on cryptographic techniques, and they are effective against many accidental and malicious actions.

ASPA objects are cryptographically signed registrations of customer-to-provider relationships and stored in a distributed database [[I-D.ietf-sidrops-aspa-profile](#)]. ASPA-based path verification is an incrementally deployable technique and provides benefits to early adopters in the context of limited deployment.

The procedures described in this document are applicable only for BGP routes with {AFI, SAFI} combinations {AFI 1 (IPv4), SAFI 1} and {AFI 2 (IPv6), SAFI 1} [[IANA-AF](#)]. SAFI 1 represents NLRI used for unicast forwarding [[IANA-SAF](#)].

1.1. Anomaly Propagation

Both route leaks and hijacks have similar effects on ISP operations - they redirect traffic and can result in denial of service (DoS), eavesdropping, increased latency, and packet loss. The level of risk, however, depends significantly on the extent of propagation of the anomalies. For example, a route leak or hijack that is propagated only to customers may cause bottlenecking within a particular ISP's customer cone, but if the anomaly propagates through lateral (i.e., non-transit) peers and transit providers, or reaches global distribution through transit-free networks, then the

ill effects will likely be amplified and experienced across continents.

The ability to constrain the propagation of BGP anomalies to transit providers and lateral peers - without requiring support from the source of the anomaly (which is critical if the source has malicious intent) - should significantly improve the robustness of the global inter-domain routing system.

1.2. Terminology

The use of the term "route is ineligible" in this document has the same meaning as in [[RFC4271](#)], i.e., "route is ineligible to be installed in Loc-RIB and will be excluded from the next phase of route selection."

For brevity, the term "provider" is often used instead of "transit provider" in this document; they mean the same.

1.3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [[RFC2119](#)] [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

2. BGP Roles

For path verification purposes in this document, the BGP roles an AS can have in relation to a neighbor AS are customer, provider, lateral peer, Route Server (RS), RS-client, and mutual-transit. These relationships, except mutual-transit, are defined in [[RFC9234](#)]. Mutual-transit ASes MAY export everything (both customer and non-customer routes) to each other, i.e., consider each other as a customer. For mutual-transit ASes, the customer-to-provider relationship applies in each direction.

All roles are configured locally and used for the registration of ASPA objects ([Section 3](#), [Section 4](#)) and/or for path verification ([Section 6](#)). The procedures for local BGP Role announcement in the BGP OPEN message and neighbor role cross-check specified in [[RFC9234](#)] are RECOMMENDED. Those procedures are not applied for cross-checking a mutual-transit role since this role is not specified in [[RFC9234](#)].

3. Autonomous System Provider Authorization

An ASPA record is a digitally signed object that binds a set of Provider AS numbers to a Customer AS (CAS) number (in terms of BGP

announcements) and is signed by the CAS [[I-D.ietf-sidrops-aspa-profile](#)]. The ASPA attests that the CAS indicated a Set of Provider ASes (SPAS), which applies only to the IPv4 and IPv6 address families (i.e., AFI = 1 and AFI = 2) and only to Network Layer Reachability Information used for unicast forwarding (SAFI = 1). The definition of Provider AS is given in Section 1 of the ASPA profile object document [[I-D.ietf-sidrops-aspa-profile](#)]. A function of a Provider AS is to propagate a CAS's route announcements onward, i.e., to the Provider's upstream providers, lateral peers, or customers. Another function is to offer outbound (customer to Internet) data traffic connectivity to the CAS.

The notation (AS x, {AS y1, AS y2, ...}), is used to represent an ASPA object for a CAS denoted as AS x. In this notation, the set {AS y1, AS y2, ...} represents the Set of Provider ASes (SPAS) of the CAS (AS x). A CAS is expected to register a single ASPA listing all its Provider ASes (see [Section 4](#)). If a CAS has a single ASPA, then the SPAS for the CAS is the set of Provider ASes listed in that ASPA. In case a CAS has multiple ASPAs, then the SPAS is the union of the Provider ASes listed in all ASPAs of the CAS.

Verified ASPA Payload (VAP) refers to the payload in a cryptographically verified (i.e., X.509 valid [[RFC3779](#)] [[RFC5280](#)]) ASPA object. In the procedures for the AS path verification described in this document ([Section 5](#), [Section 6](#)), VAP-SPAS refers to the set of provider ASes derived from the VAP(s) of the CAS in consideration.

4. ASPA Registration Recommendations

An ASPA object showing only AS 0 as a provider AS is referred to as an AS0 ASPA. A non-transparent Route Server AS (RS AS) is one that includes its AS number in the AS_PATH. Registering as AS0 ASPA is a statement by the registering AS that it has no transit providers, and it is also not an RS-client at a non-transparent RS AS. If that statement is true, then the AS MUST register an AS 0 ASPA.

Normally, the Provider ASes of a CAS would be congruent for the address family combinations {AFI 1 (IPv4), SAFI 1} and {AFI 2 (IPv6), SAFI 1}. Exceptions to this are expected to be rare. In any case, the CAS MUST list the union of all Provider ASes applicable to the address family combinations stated above in the SPAS and MUST also include any non-transparent RS AS(es) at which it is an RS-client. In the procedures for the AS path verification described in this document ([Section 5](#), [Section 6](#)), the SPAS is always considered to be uniformly applicable to {AFI 1 (IPv4), SAFI 1} and {AFI 2 (IPv6), SAFI 1}.

A compliant AS, including a Route Server AS (RS AS), MUST have an ASPA. An AS SHOULD NOT have more than one ASPA. An RS AS SHOULD register an AS 0 ASPA.

As mentioned in [Section 3](#), the set of provider ASes contained in the VAP(s) is referred to as the VAP-SPAS of the AS registering the ASPA(s). Normally, a VAP-SPAS is not expected to contain both an AS 0 and other Provider ASes, but an unexpected presence of AS 0 has no influence on the AS path verification procedures (see [Section 5](#), [Section 6](#)).

Each of the two ASes in a mutual-transit pair MUST register its ASPA including the other AS in its SPAS. If one of the ASes in the pair does this registration but the other does not, it increases the risk of incorrect AS path verification results for routes that include the pair.

The ASes on the boundary of an AS Confederation MUST register ASPAs using the Confederation's global ASN as the CAS.

As specified earlier, a compliant AS should maintain a single ASPA object that includes all its provider ASes, including any non-transparent RS ASes. Such a practice helps prevent race conditions during ASPA updates that might affect prefix propagation. The software that provides hosting for ASPA records SHOULD support enforcement of this practice. During a transition process between different certificate authority (CA) registries, the ASPA records SHOULD be kept identical in all relevant registries.

5. Hop-Check Function

Let AS(i) and AS(j) represent adjacent unique ASes in an AS_PATH, and thus (AS(i), AS(j)) represents an AS hop. A hop-check function, hop(AS(i), AS(j)), checks if the ordered pair of ASNs, (AS(i), AS(j)), has the property that AS(j) is an attested provider of AS(i) per VAP-SPAS of AS(i). The VAP-SPAS table is assumed to be organized in such a way that it can be queried to check (1) if a specified CAS = AS(i) has an entry (i.e., SPAS listed), or (2) if for a given (AS(i), AS(j)) tuple, AS(j) is listed in the VAP-SPAS as a provider associated with CAS = AS(i). A provider AS ID included in the SPAS can correspond to a Provider, a non-transparent RS, or a mutual-transit neighbor. A non-transparent RS is effectively a Provider to its RS-client. Mutual-transit neighbors regard each other as a Provider (see [Section 4](#)). The term "Provider+" in the definition of the hop-check function is meant to encompass all three possibilities: Provider, non-transparent RS, or mutual-transit neighbor. This function is specified as follows:

```

hop(AS(i), AS(j)) = /
                    | "No Attestation" if there is no entry
                    |   in VAP-SPAS table for CAS = AS(i)
                    |
                    / Else, "Provider+" if the VAP-SPAS entry
                    \   for CAS = AS(i) includes AS(j)
                    |
                    | Else, "Not Provider+"
                    \

```

Figure 1: Hop-check function.

To be clear, this function checks if AS(j) is included in the VAP-SPAS of AS(i), and in doing so it does not need to distinguish between Provider, non-transparent RS, and mutual-transit neighbor.

The "No Attestation" result is returned only when the CAS = AS(i) has no entry in the VAP-SPAS table, which occurs when no ASPA is registered for the CAS or none of its ASPAs are cryptographically valid. The hop-check function is used in the ASPA-based AS_PATH verification algorithms described in [Section 6.1](#) and [Section 6.2](#).

6. AS_PATH Verification

The procedures described in this document are applicable only to four-octet AS number compatible BGP speakers [[RFC6793](#)]. If such a BGP speaker receives both AS_PATH and AS4_PATH attributes in an UPDATE, then the procedures are applied on the reconstructed AS path (Section 4.2.3 of [[RFC6793](#)]). So, the term AS_PATH is used in this document to refer to the usual AS_PATH [[RFC4271](#)] as well as the reconstructed AS path.

If an attacker creates a route leak intentionally, they may try to strip their AS from the AS_PATH. To partly guard against that, a check is necessary to match the most recently added AS in the AS_PATH to the BGP neighbor's ASN. This check MUST be performed as specified in Section 6.3 of [[RFC4271](#)]. If the check fails, then the AS_PATH is considered a Malformed AS_PATH and the UPDATE is considered to be in error (Section 6.3 of [[RFC4271](#)]). The case of transparent RS MUST also be appropriately taken care of (e.g., by suspending the neighbor ASN check). The check fails also when the AS_PATH is empty (zero length) and such UPDATES will also be considered to be in error.

[[I-D.ietf-idr-deprecate-as-set-confed-set](#)] specifies that "treat-as-withdraw" error handling [[RFC7606](#)] SHOULD be applied to routes with AS_SET in the AS_PATH. In the current document, routes with AS_SET

are given Invalid evaluation in the AS_PATH verification procedures ([Section 6.1](#) and [Section 6.2](#)). See [Section 7](#) for how routes with Invalid AS_PATH are handled.

In [Section 6.1](#) and [Section 6.2](#) below, the terms "upstream path" and "downstream path" generally refer to AS paths received in the upstream direction (from a customer or a lateral peer) and in the downstream direction (from a provider or a mutual-transit neighbor), respectively. An RS-client receiving a route from its RS is a special case where the algorithm for upstream paths is applied ([Section 6.1](#)).

6.1. Algorithm for Upstream Paths

The upstream verification algorithm described here is applied when a route is received from a customer or lateral peer, or is received by an RS from an RS-client, or is received by an RS-client from an RS. In all these cases, the receiving/validating eBGP router expects the AS_PATH to consist of only customer-to-provider hops successively from the origin AS to the neighbor AS (most recently added).

The basic principles of the upstream verification algorithm are stated here. Let the sequence $\{AS(N), AS(N-1), \dots, AS(2), AS(1)\}$ represent the AS_PATH in terms of unique ASNs, where $AS(1)$ is the origin AS and $AS(N)$ is the most recently added AS and neighbor of the receiving/validating AS. For each hop $AS(i-1)$ to $AS(i)$ in this sequence, the hop-check function, $hop(AS(i-1), AS(i))$, must equal "Provider+" ([Section 5](#)) for the AS_PATH to be Valid. If the hop-check function for at least one of those hops is "Not Provider+", then the AS_PATH is deemed Invalid. If the AS_PATH verification outcome is neither Valid nor Invalid (per the above principles), then it is evaluated as Unknown.

The upstream path verification procedure is specified as follows:

1. If the AS_PATH has an AS_SET, then the procedure halts with the outcome "Invalid".
2. Collapse prepends in the AS_SEQUENCE(s) in the AS_PATH (i.e., keep only the unique AS numbers). Let the resulting ordered sequence be represented by $\{AS(N), AS(N-1), \dots, AS(2), AS(1)\}$, where $AS(1)$ is the first-added (i.e., origin) AS and $AS(N)$ is the last-added AS and neighbor to the receiving/validating AS.
3. If $N = 1$, then the procedure halts with the outcome "Valid". Else, continue.
4. At this step, $N \geq 2$. If there is an i such that $2 \leq i \leq N$ and $hop(AS(i-1), AS(i)) = \text{"Not Provider+"}$, then the procedure halts with the outcome "Invalid". Else, continue.

5. If there is an i such that $2 \leq i \leq N$ and $\text{hop}(\text{AS}(i-1), \text{AS}(i)) = \text{"No Attestation"}$, then the procedure halts with the outcome "Unknown". Else, the procedure halts with the outcome "Valid".

6.2. Algorithm for Downstream Paths

The downstream verification algorithm described here is applied when a route is received from a transit provider or mutual-transit neighbor. As described in [Section 4](#), a sending mutual-transit AS acts towards its receiving mutual-transit AS in a manner similar to that of a provider towards its customer.

It is not essential, but the reader may take a look at the illustrations and formal proof in [\[sriram1\]](#) to develop a clearer understanding of the algorithm described here.

Here again (as in [Section 6.1](#)), let the AS_PATH be simplified and represented by the ordered sequence of unique ASNs as $\{\text{AS}(N), \text{AS}(N-1), \dots, \text{AS}(2), \text{AS}(1)\}$.

If $1 \leq N \leq 2$, then the AS_PATH is trivially Valid.

[Section 6.2.1](#) below assumes that the AS_PATH contains 3 or more unique ASNs ($N \geq 3$).

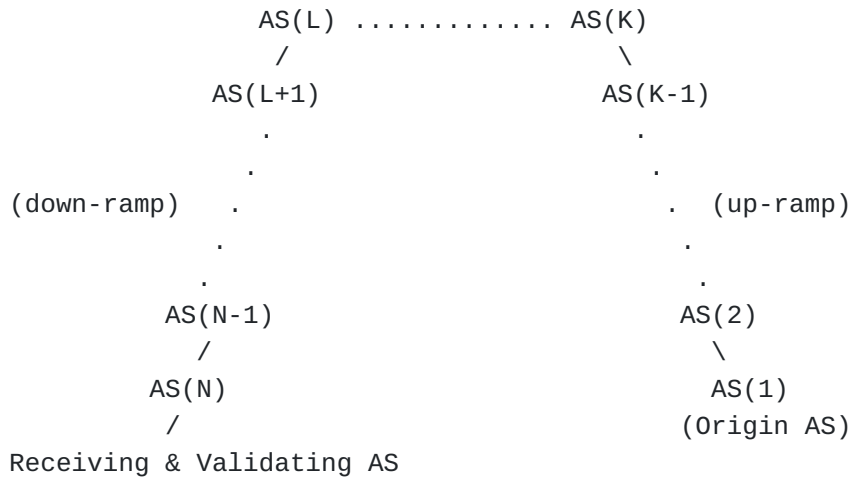
6.2.1. Principles for Determination of Invalid, Valid, and Unknown in Downstream Path Verification (for $N \geq 3$)

Determination of Invalid AS_PATH :

Given the above-mentioned ordered sequence, if there exist indices u and v such that (1) $u \leq v$, (2) $\text{hop}(\text{AS}(u-1), \text{AS}(u)) = \text{"Not Provider+"}$, and (3) $\text{hop}(\text{AS}(v+1), \text{AS}(v)) = \text{"Not Provider+"}$, then the AS_PATH is Invalid.

Determination of Valid AS_PATH :

As shown in [Figure 2](#), assume that the ASes in the AS_PATH are in the same physical (locational) order as in the sequence representation $\{\text{AS}(N), \text{AS}(N-1), \dots, \text{AS}(2), \text{AS}(1)\}$, i.e., $\text{AS}(N)$ is the left-most and $\text{AS}(1)$ the right-most.



Each ramp has consecutive ASPA-attested
customer-to-provider hops in the bottom-to-top direction

Figure 2: Illustration of up-ramp and down-ramp.

Looking at [Figure 2](#), the UPDATE is received from a provider or a mutual-transit neighbor (i.e., AS(N) has that role in relation to the receiver). The AS_PATH may have both an up-ramp (on the right starting at AS(1)) and a down-ramp (on the left starting at AS(N)). The ramps are described as a sequence of ASes that consists of consecutive customer-to-provider hops. The up-ramp starts at AS(1) and each AS hop, (AS(i), AS(i+1)), in it has the property that $\text{hop}(\text{AS}(i), \text{AS}(i+1)) = \text{"Provider+"}$ for $i = 1, 2, \dots, K-1$. If such a K does not exist, then K is set to 1. The up-ramp ends (reaches its apex) at AS(K) because $\text{hop}(\text{AS}(K), \text{AS}(K+1)) = \text{"Not Provider+"}$ or "No Attestation". The down-ramp runs backward from AS(N) to AS(L). Each AS hop, (AS(j), AS(j-1)), in it has the property that $\text{hop}(\text{AS}(j), \text{AS}(j-1)) = \text{"Provider+"}$ for $j = N, N-1, \dots, L+1$. If such an L does not exist, then L is set to N. The down-ramp ends at AS(L) because $\text{hop}(\text{AS}(L), \text{AS}(L-1)) = \text{"Not Provider+"}$ or "No Attestation". Thus, the apex of the down-ramp is AS(L).

If there is an up-ramp that runs across all ASes in the AS_PATH (i.e., $K = N$), then clearly the AS_PATH is Valid. Similarly, if there is a down-ramp that runs across all ASes in the AS_PATH (i.e., $L = 1$), then also the AS_PATH is Valid. However, if both ramps exist in an AS_PATH with $K < N$ and $L > 1$, then the AS_PATH is Valid if and only if $L-K \leq 1$. Note that K could be greater than L (i.e., L-K has a negative value), which means that the up-ramp and down-ramp overlap, and that could happen when some adjacent ASes in the AS_PATH have mutual-transit relationship between them (i.e., include each other in their respective SPAS) (see [Section 4](#)). If $L-K = 0$, it

means that the apexes of the up-ramp and down-ramp are at the same AS. If $L-K = 1$, it means that the apexes are at adjacent ASes. In summary, the AS_PATH is Valid if $L-K$ is 0 or 1 or has a negative value.

Determination of Unknown AS_PATH:

If $L-K \geq 2$, then the AS_PATH is either Invalid (route leak) or Unknown (see illustrations and proof in [[sriram1](#)]). However, if $L-K \geq 2$ and an Invalid outcome was not found by the process described earlier in this section, then the AS_PATH is determined to be Unknown.

6.2.2. Formal Procedure for Verification of Downstream Paths

The downstream path verification procedure is formally specified as follows:

1. If the AS_PATH has an AS_SET, then the procedure halts with the outcome "Invalid".
2. Collapse prepends in the AS_SEQUENCE(s) in the AS_PATH (i.e., keep only the unique AS numbers). Let the resulting ordered sequence be represented by $\{AS(N), AS(N-1), \dots, AS(2), AS(1)\}$, where $AS(1)$ is the first-added (i.e., origin) AS and $AS(N)$ is the last-added AS and neighbor to the receiving/validating AS.
3. If $1 \leq N \leq 2$, then the procedure halts with the outcome "Valid". Else, continue.
4. At this step, $N \geq 3$. Given the above-mentioned ordered sequence, find the lowest value of u ($2 \leq u \leq N$) for which $\text{hop}(AS(u-1), AS(u)) = \text{"Not Provider+"}$. Call it u_{\min} . If no such u_{\min} exists, set $u_{\min} = N+1$. Find the highest value of v ($N-1 \geq v \geq 1$) for which $\text{hop}(AS(v+1), AS(v)) = \text{"Not Provider+"}$. Call it v_{\max} . If no such v_{\max} exists, then set $v_{\max} = 0$. If $u_{\min} \leq v_{\max}$, then the procedure halts with the outcome "Invalid". Else, continue.
5. Up-ramp: For $2 \leq i \leq N$, determine the largest K such that $\text{hop}(AS(i-1), AS(i)) = \text{"Provider+"}$ for each i in the range $2 \leq i \leq K$. If such a largest K does not exist, then set $K = 1$.
6. Down-ramp: For $N-1 \geq j \geq 1$, determine the smallest L such that $\text{hop}(AS(j+1), AS(j)) = \text{"Provider+"}$ for each j in the range $N-1 \geq j \geq L$. If such smallest L does not exist, then set $L = N$.

7. If $L-K \leq 1$, then the procedure halts with the outcome "Valid".
Else, the procedure halts with the outcome "Unknown".

In the above procedure, the computations in Steps 4, 5, and 6 can be done at the same time.

7. AS_PATH Verification and Anomaly Mitigation Recommendations

AS_PATH verification and anomaly mitigation recommendations for eBGP routers are specified in this section. The recommendations apply to eBGP routers in general, including those on the boundary of an AS Confederation facing external ASes. However, the procedures for ASPA-based AS_PATH verification in this document are NOT RECOMMENDED for use on eBGP links internal to the Confederation.

The verification procedures described in this document MUST be applied to BGP routes with {AFI, SAFI} combinations {AFI 1 (IPv4), SAFI 1} and {AFI 2 (IPv6), SAFI 1} [[IANA-AF](#)]. The procedures MUST NOT be applied to other address families by default.

7.1. Verification and Mitigation at Ingress eBGP Router

Verification: Conforming implementations of this specification are not required to implement the AS_PATH verification procedures (step-wise lists) exactly as described in [Section 6.1](#) and [Section 6.2](#) but MUST provide functionality equivalent to the external behavior resulting from those procedures. In other words, the algorithms used in a specific implementation may differ, for example, for computational efficiency purposes, but the AS_PATH verification outcomes MUST be identical to those obtained by the procedures described in [Section 6.1](#) and [Section 6.2](#).

Mitigation: Mitigation recommendations are provided here with the understanding that the deployed mitigation policy is set by network operator discretion. If the AS_PATH is determined to be Invalid, then the route SHOULD be considered ineligible for route selection (see [Section 1.2](#)) and MUST be kept in the Adj-RIB-In for potential future re-evaluation (see [[RFC9324](#)]). Also, for any route with an Invalid AS_PATH, the cause of the Invalid state SHOULD be logged for monitoring and diagnostic purposes. The cause of the Invalid state can be in the form of listing the AS hops which were evaluated by the hop-check function to be "Not Provider+". For any route with an Unknown AS_PATH, the cause of the Unknown state SHOULD be logged for monitoring and diagnostic purposes. The cause of the Unknown state can be in the form of listing the AS hops which were evaluated by the hop-check function to be "No Attestation" or "Not Provider+".

7.2. Only to Customer (OTC) Attribute

While the ASPA-based AS_PATH verification method ([Section 7.1](#)) detects and mitigates route leaks that were created by preceding ASes listed in the AS_PATH, it lacks the ability to prevent route leaks from occurring at the local AS. The use of the Only to Customer (OTC) Attribute [[RFC9234](#)] fills in that gap. The procedures utilizing the OTC Attribute set out in [[RFC9234](#)] complement those described in this document. Implementation of those procedures in addition to ASPA-based AS_PATH verification is encouraged.

8. Properties of ASPA-based Path Verification

The ASPA-based path verification procedures are able to check routes received from customers, lateral peers, transit providers, RSeS, RS-clients, and mutual-transits. These procedures combined with BGP Roles and the OTC Attribute [[RFC9234](#)] and RPKI-ROV [[RFC6811](#)] [[RFC9319](#)] can provide a fully automated solution to detect and filter many of the ordinary prefix hijacks, route leaks, and prefix hijacks with forged-origin or forged-path-segment (see Property 3 below).

The ASPA-based path verification at ingress eBGP routers ([Section 6](#), [Section 7.1](#)) has the following properties (detection capabilities):

Property 1: Let AS A and AS B be any two ASes in the Internet doing ASPA (registration and path verification) and no assumption is made about the ASPA deployment status of other ASes. Consider a route propagated from AS A to a customer or lateral peer. The route is subsequently leaked by an offending AS in the AS path before being received at AS B on a customer or lateral peer interface. The ASPA-based path verification at AS B always detects such a route leak though it may not be able to identify the AS that originated the leak. This assertion is true even when the sender AS A (or receiver AS B) is an RS AS and the neighbor AS that AS A sent to (or AS B received from) is an RS-client.

Property 2: Again, let AS A and AS B be any two ASes in the Internet doing ASPA (registration and path verification) and no assumption is made about the ASPA deployment status of other ASes. Consider a route received at AS B on a customer or lateral peer interface that is a forged-origin prefix hijack involving AS A as the forged-origin. The ASPA-based path verification at AS B always detects such a forged-origin prefix hijack.

Property 3: This is an extension of Property 2 above to the case of prefix hijacking with a forged-path-segment. Such hijacking refers to the forging of multiple contiguous ASes in an AS path beginning with the origin AS. Again, let AS A and AS B be any two

ASes in the Internet doing ASPA (registration and path verification). Let AS A's providers, AS P and AS Q, also be registering ASPA. No assumption is made about the ASPA deployment status of any other ASes in the Internet. Consider a route received at AS B on a customer or lateral peer interface that is a prefix hijack with a forged-path-segment {AS P, AS A} or {AS Q, AS A}. That is, the hijacker attaches this path-segment at the beginning of their route announcement. The ASPA-based path verification at AS B always detects such a forged-path-segment prefix hijack. For a chance to be successful (remain undetected by AS B), the hijacker may resort to a forged-path-segment with three ASes including a provider AS of AS P (or AS Q). But even that can be foiled (detected) if the providers of AS P and AS Q also register ASPA. Having to use a longer forged-path-segment to avoid detection by AS B diminishes the ability of the hijacked route to compete with the corresponding legitimate route in route selection.

Property 4: Let AS A, AS B, and AS C be any three ASes in the Internet doing ASPA (registration and path verification). Consider a route propagated from AS A in any direction (i.e., to a neighbor AS with any of the BGP roles described in [Section 2](#)). Let the route be received at AS B from any direction and detected to be a route leak (facilitated due to a sufficient set of ASes doing ASPA in the AS path from AS A to AS B). Assume that AS B's local policy is such that it only lowers the route's LOCAL_PREF [[RFC4271](#)]. Let such a route, selected and forwarded by AS B, be subsequently received at AS C. No assumption is made about the ASPA compliance of the ASes in the intervening path from AS B to AS C. The ASPA-based path verification at AS C always detects such received route as a leak regardless of the direction (type of peer) it was received from.

In the description of the properties listed above, the term "customer" can be replaced with "RS-client".

An observation that follows from Property #1 above is that if any two ISP ASes register ASPAs and implement the detection and mitigation procedures, then any route received from one of them and leaked to the other by a common customer AS (ASPA compliant or not) will be automatically detected and mitigated. In effect, if most major ISPs are compliant, the propagation of route leaks in the Internet will be severely limited.

The above properties show that ASPA-based path verification offers significant benefits to early adopters. Limitations of the method with regard to some forms of malicious AS path manipulations are discussed in [Section 12](#).

9. Operational Considerations

9.1. 4-Byte AS Number Requirement

The procedures specified in this document are compatible only with BGP implementations that support 4-byte ASNs in the AS_PATH. This limitation should not have a real effect on operations since legacy BGP routers are rare, and it is highly unlikely that they support integration with the RPKI.

9.2. Correctness of the ASPA

ASPA issuers should be aware of the implications of ASPA-based AS path verification. Network operators must keep their ASPA objects correct and up to date. Otherwise, for example, if a provider AS is left out of the Set of Provider ASes (SPAS) in the ASPA, then routes containing the CAS (in the ASPA) and said provider AS may be incorrectly labeled as route leaks and considered ineligible for route selection (see [Section 7.1](#)).

9.3. Make Before Break

ASPA issuers SHOULD apply the make-before-break principle while updating an ASPA registration. For example, when adding new Provider AS(es) in the SPAS, if the new ASPA is meant to replace a previously created ASPA, the latter SHOULD be decommissioned only after allowing sufficient time for the new ASPA to propagate to Relying Parties (RP) through the global RPKI system.

9.4. DoS/DDoS Mitigation Service Provider

An AS may have a mitigation service provider (MSP) for protection from Denial of Service (DoS)/Distributed DoS (DDoS) attacks targeting servers with IP addresses in the prefixes the AS originates. Such an AS MAY include the MSP's AS in the SPAS of its ASPA. With such an ASPA in place, in the event of an attack, the AS (customer of the MSP) can announce more specific prefixes (over a BGP session) to the MSP's AS for mitigation purposes, and such announcements would be able to pass the ASPA-based path verification. It is assumed that appropriate ROAs are registered in advance so that the announcements can pass RPKI-ROV as well.

10. Comparison to Other Technologies

10.1. BGPsec

BGPsec [[RFC8205](#)] was designed to solve the problem of AS_PATH verification by including cryptographic signatures in BGP Update messages. It offers protection against unauthorized path modifications and assures that the BGPsec Update actually traveled

the path shown in the BGPsec_PATH Attribute. However, it does not detect route leaks (valley-free violations). In comparison, the ASPA-based path verification described in this document detects if the AS path is improbable and focuses on detecting route leaks (including malicious cases) and forged-origin hijacks.

BGPsec and ASPA are complementary technologies.

10.2. Peerlock

The Peerlock mechanism [[Peerlock](#)] [[Flexsealing](#)] has a similar objective as the APSA-based route leak protection mechanism described in this document. It is commonly deployed by large Internet carriers to protect each other from route leaks. Peerlock depends on a laborious manual process in which operators coordinate the distribution of unstructured Provider Authorizations through out-of-band means in a many-to-many fashion. On the other hand, ASPA's use of the RPKI allows for automated, scalable, and ubiquitous deployment, making the protection mechanism available to a wider range of network operators.

The ASPA mechanism implemented in router code (in contrast to Peerlock's AS_PATH regular expressions) also provides a way to detect anomalies propagated from transit providers and IX route servers. ASPA is intended to be a complete solution and replacement for existing Peerlock deployments.

11. IANA Considerations

This document includes no request to IANA.

12. Security Considerations

While the ASPA-based mechanism is able to detect and mitigate the majority of mistakes and malicious activity affecting routes, it might fail to detect some malicious path modifications, especially for routes that are received from transit providers.

Since an upstream provider becomes a trusted point, in theory, it might be able to propagate some instances of hijacked prefixes with forged-origin or forged-path-segment or even routes with manipulated AS_PATHs, and such attacks might go undetected by its customers. This can be illustrated with some examples. In [Figure 3](#), normally the receiving/validating AS located at the lower left side should receive a route with AS_PATH {AS(5), AS(4), AS(3), AS(2), AS(1)} and it would be Valid ([Section 6.2](#)) given all the ASPAs that are shown in the figure. However, if AS(5) which is a transit provider to the validating AS acts maliciously and sends the route with a shortened AS_PATH such as {AS(5), AS(3), AS(2), AS(1)} or {AS(5), AS(2), AS(1)}, such path manipulation would be undetectable (i.e., the

be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

*A BGP implementation [OpenBGPD](#) [bgpd] (version 7.8 and higher), written in C, was provided by Claudio Jeker, Theo Buehler, and Job Snijders.

*The implementation NIST-BGP-SRx [BGP-SRx] is a software suite that provides a validation engine (BGP-SRx) and a Quagga-based BGP router (Quagga-SRx). It includes unit test cases for testing the ASPA-based path verification. It was provided by Oliver Borchert, Kyehwan Lee, and their colleagues at US NIST. It requires some additional work to incorporate the latest changes in the draft specifications related to IXP RS AS and RS-client.

14. References

14.1. Normative References

[I-D.ietf-sidrops-aspa-profile]

Azimov, A., Uskov, E., Bush, R., Snijders, J., Housley, R., and B. Maddison, "A Profile for Autonomous System Provider Authorization", Work in Progress, Internet-Draft, draft-ietf-sidrops-aspa-profile-17, 7 November 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-sidrops-aspa-profile-17>>.

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

[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, <<https://www.rfc-editor.org/info/rfc4271>>.

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

[RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, DOI 10.17487/

RFC6482, February 2012, <<https://www.rfc-editor.org/info/rfc6482>>.

- [RFC6793] Vohra, Q. and E. Chen, "BGP Support for Four-Octet Autonomous System (AS) Number Space", RFC 6793, DOI 10.17487/RFC6793, December 2012, <<https://www.rfc-editor.org/info/rfc6793>>.
- [RFC6811] Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", RFC 6811, DOI 10.17487/RFC6811, January 2013, <<https://www.rfc-editor.org/info/rfc6811>>.
- [RFC7606] Chen, E., Ed., Scudder, J., Ed., Mohapatra, P., and K. Patel, "Revised Error Handling for BGP UPDATE Messages", RFC 7606, DOI 10.17487/RFC7606, August 2015, <<https://www.rfc-editor.org/info/rfc7606>>.
- [RFC7908] Sriram, K., Montgomery, D., McPherson, D., Osterweil, E., and B. Dickson, "Problem Definition and Classification of BGP Route Leaks", RFC 7908, DOI 10.17487/RFC7908, June 2016, <<https://www.rfc-editor.org/info/rfc7908>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8481] Bush, R., "Clarifications to BGP Origin Validation Based on Resource Public Key Infrastructure (RPKI)", RFC 8481, DOI 10.17487/RFC8481, September 2018, <<https://www.rfc-editor.org/info/rfc8481>>.
- [RFC8893] Bush, R., Volk, R., and J. Heitz, "Resource Public Key Infrastructure (RPKI) Origin Validation for BGP Export", RFC 8893, DOI 10.17487/RFC8893, September 2020, <<https://www.rfc-editor.org/info/rfc8893>>.
- [RFC9234] Azimov, A., Bogomazov, E., Bush, R., Patel, K., and K. Sriram, "Route Leak Prevention and Detection Using Roles in UPDATE and OPEN Messages", RFC 9234, DOI 10.17487/RFC9234, May 2022, <<https://www.rfc-editor.org/info/rfc9234>>.
- [RFC9324] Bush, R., Patel, K., Smith, P., and M. Tinka, "Policy Based on the Resource Public Key Infrastructure (RPKI) without Route Refresh", RFC 9324, DOI 10.17487/RFC9324, December 2022, <<https://www.rfc-editor.org/info/rfc9324>>.

14.2. Informative References

[BGP-SRx]

NIST, "BGP Secure Routing Extension (BGP-SRx) Software Suite", NIST Open-Source Software , <<https://www.nist.gov/services-resources/software/bgp-secure-routing-extension-bgp-srx-software-suite>>.

[bgpd]

Jeker, C., "OpenBGPD", <<http://www.openbgpd.org/>>.

[Flexsealing]

McDaniel, T., Smith, J., and M. Schuchard, "Flexsealing BGP Against Route Leaks: Peerlock Active Measurement and Analysis", November 2020, <<https://arxiv.org/pdf/2006.06576.pdf>>.

[I-D.ietf-grow-route-leak-detection-mitigation] Sriram, K. and A. Azimov, "Methods for Detection and Mitigation of BGP Route Leaks", Work in Progress, Internet-Draft, draft-ietf-grow-route-leak-detection-mitigation-10, 8 January 2024, <<https://datatracker.ietf.org/doc/html/draft-ietf-grow-route-leak-detection-mitigation-10>>.

[I-D.ietf-idr-deprecate-as-set-confed-set]

Kumari, W. A., Sriram, K., Hannachi, L., and J. Haas, "Deprecation of AS_SET and AS_CONFED_SET in BGP", Work in Progress, Internet-Draft, draft-ietf-idr-deprecate-as-set-confed-set-12, 10 January 2024, <<https://datatracker.ietf.org/doc/html/draft-ietf-idr-deprecate-as-set-confed-set-12>>.

[IANA-AF]

IANA, "Address Family Numbers", <<https://www.iana.org/assignments/address-family-numbers/address-family-numbers.xhtml>>.

[IANA-SAF]

IANA, "Subsequent Address Family Identifiers (SAFI) Parameters", <<https://www.iana.org/assignments/safi-namespace/safi-namespace.xhtml>>.

[Peerlock]

Snijders, J., "Peerlock", June 2016, <https://www.nanog.org/sites/default/files/Snijders_Everyday_Practical_Bgp.pdf>.

[RFC3779]

Lynn, C., Kent, S., and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, DOI 10.17487/RFC3779, June 2004, <<https://www.rfc-editor.org/info/rfc3779>>.

[RFC5280]

Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <<https://www.rfc-editor.org/info/rfc5280>>.

[RFC7942]

Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.

[RFC8205]

Lepinski, M., Ed. and K. Sriram, Ed., "BGPsec Protocol Specification", RFC 8205, DOI 10.17487/RFC8205, September 2017, <<https://www.rfc-editor.org/info/rfc8205>>.

[RFC9319]

Gilad, Y., Goldberg, S., Sriram, K., Snijders, J., and B. Maddison, "The Use of maxLength in the Resource Public Key Infrastructure (RPKI)", BCP 185, RFC 9319, DOI 10.17487/RFC9319, October 2022, <<https://www.rfc-editor.org/info/rfc9319>>.

[sriram1]

Sriram, K. and J. Heitz, "On the Accuracy of Algorithms for ASPA Based Route Leak Detection", IETF SIDROPS Meeting, Proceedings of the IETF 110, March 2021, <<https://datatracker.ietf.org/meeting/110/materials/slides-110-sidrops-sriram-aspa-alg-accuracy-01>>.

Appendix A. Acknowledgments

The authors wish to thank Jakob Heitz, Amir Herzberg, Igor Lubashev, Ben Maddison, Russ Housley, Jeff Haas, Nan Geng, Nick Hilliard, Shunwan Zhuang, Yangyang Wang, Martin Hoffmann, Jay Borkenhagen, Amreesh Phokeer, Aftab Siddiqui, Dai Zhibin, Doug Montgomery, Rich Compton, Andrei Robachevsky, and Iljitsch van Beijnum for comments, suggestions, and discussion on the path verification procedures or the text in the document. For the implementation and testing of the procedures in the document, the authors wish to thank Claudio Jeker and Theo Buehler [[bgpd](#)] as well as Kyehwan Lee and Oliver Borchert [[BGP-SRx](#)].

Contributors

The following people made significant contributions to this document and should be considered co-authors:

Claudio Jeker
OpenBSD
Email: cjeker@diehard.n-r-g.com

Authors' Addresses

Alexander Azimov
Yandex
Ulitsa Lva Tolstogo 16
Moscow

119021

Russian Federation

Email: a.e.azimov@gmail.com

Eugene Bogomazov

Qrator Labs

1-y Magistralnyy tupik 5A

Moscow

123290

Russian Federation

Email: eb@qrator.net

Randy Bush

Internet Initiative Japan & Arrcus, Inc.

5147 Crystal Springs

Bainbridge Island, Washington 98110

United States of America

Email: randy@psg.com

Keyur Patel

Arrcus

2077 Gateway Place

Suite #400

San Jose, CA 95119

United States of America

Email: keyur@arrcus.com

Job Snijders

Fastly

Amsterdam

Netherlands

Email: job@fastly.com

Kotikalapudi Sriram

USA National Institute of Standards and Technology

100 Bureau Drive

Gaithersburg, MD 20899

United States of America

Email: ksriram@nist.gov