Network Working Group	P.M. Mohapatra, Ed.
Internet-Draft	Cisco Systems
Intended status: Standards Track	J.S. Scudder, Ed.
Expires: May 03, 2012	D.W. Ward, Ed.
	Juniper Networks
	R.B. Bush, Ed.
	Internet Initiative Japan, Inc.
	R.A. Austein, Ed.
	Internet Systems Consortium
	October 31, 2011

BGP Prefix Origin Validation
draft-ietf-sidr-pfx-validate-03

Abstract

To help reduce well-known threats against BGP including prefix mis-announcing and monkey-in-the-middle attacks, one of the security requirements is the ability to validate the origination AS of BGP routes. More specifically, one needs to validate that the AS number claiming to originate an address prefix (as derived from the AS_PATH attribute of the BGP route) is in fact authorized by the prefix holder to do so. This document describes a simple validation mechanism to partially satisfy this requirement.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

A BGP route associates an address prefix with a set of autonomous systems (AS) that identify the interdomain path the prefix has traversed in the form of BGP announcements. This set is represented as the AS_PATH attribute in BGP [RFC4271] and starts with the AS that originated the prefix. To help reduce well-known threats against BGP including prefix mis-announcing and monkey-in-the-middle attacks, one of the security requirements is the ability to validate the origination AS of BGP

routes. More specifically, one needs to validate that the AS number claiming to originate an address prefix (as derived from the AS_PATH attribute of the BGP route) is in fact authorized by the prefix holder to do so. This document describes a simple validation mechanism to partially satisfy this requirement.
The Resource Public Key Infrastructure (RPKI) describes an approach to build a formally verifyable database of IP addresses and AS numbers as resources. The overall architecture of RPKI as defined in [I-D.ietf-sidr-arch] consists of three main components:

*A public key infrastructure (PKI) with the necessary certificate objects,

*Digitally signed routing objects,

*A distributed repository system to hold the objects that would also support periodic retrieval.

The RPKI system is based on resource certificates that define extensions to X.509 to represent IP addresses and AS identifiers [RFC3779], thus the name RPKI. Route Origin Authorizations (ROA) [I-D.ietf-sidr-roa-format] are separate digitally signed objects that define associations between ASes and IP address blocks. Finally the repository system is operated in a distributed fashion through the IANA, RIR hierarchy, and ISPs.
In order to benefit from the RPKI system, it is envisioned that relying parties either at AS or organization level obtain a local copy of the signed object collection, verify the signatures, and process them. The cache must also be refreshed periodically. The exact access mechanism used to retrieve the local cache is beyond the scope of this document.
Individual BGP speakers can utilize the processed data contained in the local cache to validate BGP announcements. The protocol details to retrieve the processed data from the local cache to the BGP speakers is beyond the scope of this document (refer to [I-D.ietf-sidr-rpki-rtr] for such a mechanism). This document proposes a means by which a BGP speaker can make use of the processed data in order to assign a "validity state" to each prefix in a received BGP UPDATE message.
Note that the complete path attestation against the AS_PATH attribute of a route is outside the scope of this document.
Although RPKI provides the context for this draft, it is equally possible to use any other database which is able to map prefixes to their authorized origin ASes. Each distinct database will have its own particular operational and security characteristics; such characteristics are beyond the scope of this document.

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. Prefix-to-AS Mapping Database

The BGP speaker loads validated objects from the cache into local storage. The objects loaded have the content (IP address, prefix length, maximum length, origin AS number). We refer to such a locally stored object colloquially as a "ROA" in the discussion below although we note that this is not a strictly accurate use of the term.
We define several terms in addition to "ROA". Where these terms are used, they are capitalized:

*Prefix: (IP address, prefix length), interpreted as is customary (see [RFC4632]).

*Route: Data derived from a received BGP UPDATE, as defined in [RFC4271], Section 1.1. The Route includes one Prefix and an AS_PATH, among other things.

*ROA Prefix: The Prefix from a ROA.

*ROA ASN: The origin ASN from a ROA.

*Route Prefix: A Prefix derived from a route.

*Route Origin ASN: The origin AS number derived from a Route. The origin AS number is the rightmost AS in the final segment of the AS_PATH attribute in the Route if that segment is of type AS_SEQUENCE, or NONE if the final segment of the AS_PATH attribute is of any type other than AS_SEQUENCE. No ROA can match an origin AS number of "NONE". No Route can match a ROA whose origin AS number is zero.

*Covered: A Route Prefix is said to be Covered by a ROA when the ROA prefix length is less than or equal to the Route prefix length and the ROA prefix address matches the Route prefix address for all bits specified by the ROA prefix length. (This is simply a statement of the well-known concept of determining a prefix match.)

*Matched: A Route Prefix is said to be Matched by a ROA when the Route Prefix is Covered by that ROA and in addition, the Route prefix length is less than or equal to the ROA maximum length and the Route Origin ASN is equal to the ROA ASN, keeping in mind that a ROA ASN of zero can never be matched, nor can a route origin AS number of "NONE".

Given these definitions, any given BGP Route learned from an EBGP peer will be found to have one of the following "validation states":

*Not found: No ROA Covers the Route Prefix.

*Valid: At least one ROA Matches the Route Prefix.

*Invalid: At least one ROA Covers the Route Prefix, but no ROA Matches it.

When a BGP speaker receives an UPDATE from one of its EBGP peers, it SHOULD perform a lookup as described above for each of the Routes in the UPDATE message. The "validation state" of the Route SHOULD be set to reflect the result of the lookup. Note that the validation state of the Route does not determine whether the Route is stored in the local BGP speaker's Adj-RIB-In. This procedure SHOULD NOT be performed for Routes learned from peers of types other than EBGP. (Any of these MAY be overridden by configuration.)
Use of the validation state is discussed in Section 3 and Section 5.
We observe that a Route can be Matched or Covered by more than one ROA. This procedure does not mandate an order in which ROAs must be visited; however, the "validation state" output is fully determined.

2.1. Pseudo-Code

```
//Input are the variables derived from a BGP UPDATE message
//that need to be validated.
//
//The input prefix is comprised of prefix.address and
//prefix.length.
//
//origin_as is the rightmost AS in the final segment of the
//AS_PATH attribute in the UPDATE message if that segment is
//AS_SEQUENCE.  If the final segment of AS_PATH is not an
//AS_SEQUENCE, origin_as is NONE.
//
//Collectively, the prefix and origin_as correspond to the
//Route defined in the preceding section.
input = {prefix, origin_as};

//Initialize result to "not found" state
result = BGP_PFXV_STATE_NOT_FOUND;

//pfx_validate_table organizes all the ROA entries retrieved
//from the RPKI cache based on the IP address and the prefix
//length field. There can be multiple such entries that match
//the input. Iterate through all of them.
entry = next_lookup_result(pfx_validate_table, input.prefix);

while (entry != NULL) {
    prefix_exists = TRUE;

    if (input.prefix.length <= entry->max_length) {
        if (input.origin_as != NONE
            && entry->origin_as != 0
            && input.origin_as == entry->origin_as) {
            result = BGP_PFXV_STATE_VALID;
            return (result);
        }
    }
    entry = next_lookup_result(pfx_validate_table, input.prefix);
}

//If pfx_validate_table contains one or more prefixes that
//match the input, but none of them resulted in a "valid"
//outcome since the origin_as did not match, return the
//result state as "invalid". Else the initialized state of
//"not found" applies to this validation operation.
if (prefix_exists == TRUE) {
    result = BGP_PFXV_STATE_INVALID;
}

return (result);
```

The following pseudo-code illustrates the procedure above. In case of ambiguity, the procedure above, rather than the pseudo-code, should be taken as authoritative.

3. Policy Control

An implementation MUST provide the ability to match and set the validation state of routes as part of its route policy filtering function. Use of validation state in route policy is elaborated in Section 5. For more details on operational policy considerations, see [I-D.ietf-sidr-origin-ops].

4. Interaction with Local Cache

Each BGP speaker supporting prefix validation as described in this document is expected to communicate with one or multiple local caches that store a database of RPKI signed objects. The protocol mechanisms used to fetch the data and store them locally at the BGP speaker is beyond the scope of this document (please refer [I-D.ietf-sidr-rpki-rtr]). Irrespective of the protocol, the prefix validation algorithm as outlined in this document is expected to function correctly in the event of failures and other timing conditions that may result in an empty and/or partial prefix-to-AS mapping database. Indeed, if the (in-PoP) cache is not available and the mapping database is empty on the BGP speaker, all the lookups will result in "not found" state and the prefixes will be advertised to rest of the network (unless restricted by policy configuration). Similarly, if BGP UPDATEs arrive at the speaker while the fetch operation from the cache is in progress, some prefix lookups will also result in "not found" state. The implementation is expected to handle these timing conditions and MUST re-validate affected prefixes once the fetch operation is complete. The same applies during any subsequent incremental updates of the validation database.
In the event that connectivity to the cache is lost, the router should make a reasonable effort to fetch a new validation database (either from the same, or a different cache), and SHOULD wait until the new validation database has been fetched before purging the previous one. A configurable timer MUST be provided to bound the length of time the router will wait before purging the previous validation database.

5. Deployment Considerations

Once a route is received from an EBGP peer it is categorized according the procedure given in Section 2. Subsequently, routing policy as discussed in Section 3 can be used to take action based on the validation state.
Policies which could be implemented include filtering routes based on validation state (for example, rejecting all "invalid" routes) or adjusting a route's degree of preference in the selection algorithm based on its validation state. The latter could be accomplished by adjusting the value of such attributes as LOCAL_PREF. Considering invalid routes for BGP decision process is a pure local policy matter and should be done with utmost care.

In some cases (particularly when the selection algorithm is influenced by the adjustment of a route property that is not propagated into IBGP) it could be necessary for routing correctness to propagate the validation state to the IBGP peer. This can be accomplished on the sending side by setting a community or extended community based on the validation state, and on the receiving side by matching the (extended) community and setting the validation state.

6. Contributors

*Rex Fernando rex@cisco.com

*Keyur Patel keyupate@cisco.com

*Cisco Systems

*Miya Kohno mkohno@juniper.net

*Juniper Networks

*Shin Miyakawa miyakawa@nttv6.jp

*Taka Mizuguchi

*Tomoya Yoshida

*NTT Communications

*Russ Housley housley@vigilsec.com

*Vigil Security

*Junaid Israr jisra052@uottawa.ca

*Mouhcine Guennoun mguennou@uottawa.ca

*Hussein Mouftah mouftah@site.uottawa.ca

*University of Ottawa School of Information Technology and Engineering(SITE) 800 King Edward Avenue, Ottawa, Ontario, Canada, K1N 6N5

7. Acknowledgements

Junaid Israr's contribution to this specification is part of his PhD research work and thesis at University of Ottawa, Canada. Hannes Gredler provided valuable feedback.

8. IANA Considerations

9. Security Considerations

Although this specification discusses one portion of a system to validate BGP routes, it should be noted that it relies on a database (RPKI or other) to provide validation information. As such, the security properties of that database must be considered in order to determine the security provided by the overall solution. If "invalid" routes are blocked as this specification suggests, the overall system provides a possible denial-of-service vector, for example if an attacker is able to inject one or more spoofed records into the validation database which lead a good route to be declared invalid. In addition, this system is only able to provide limited protection against a determined attacker -- the attacker need only prepend the "valid" source AS to a forged BGP route announcement in order to defeat the protection provided by this system. This mechanism does not protect against "AS in the middle attacks" or provide any path validation. It only attempts to verify the origin. In general, this system should be thought of more as a protection against misconfiguration than as true "security" in the strong sense.

10. References

10.1. Normative References

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
[RFC3779]	Lynn, C., Kent, S. and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, June 2004.
[RFC4271]	Rekhter, Y., Li, T. and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.
[RFC4632]	Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, August 2006.
[I-D.ietf-sidr-roa-format]	Lepinski, M, Kent, S and D Kong, "A Profile for Route Origin Authorizations (ROAs)", Internet-Draft draft-ietf-sidr-roa-format-12, May 2011.

10.2. Informational References

[I-D.ietf-sidr-arch]	Lepinski, M and S Kent, "An Infrastructure to Support Secure Internet Routing", Internet-Draft draft-ietf-sidr-arch-13, May 2011.
[I-D.ietf-sidr-rpki-rtr]	Bush, R and R Austein, "The RPKI/Router Protocol", Internet-Draft draft-ietf-sidr-rpki-rtr-19, October 2011.

[I-D.ietf-sidr-origin-ops]	Bush, R, "RPKI-Based Origin Validation Operation", Internet-Draft draft-ietf-sidr-origin-ops-12, October 2011.

Authors' Addresses

Pradosh Mohapatra editor Mohapatra Cisco Systems 170 W. Tasman Drive San Jose, CA 95134 USA EMail: pmohapat@cisco.com

John Scudder editor Scudder Juniper Networks 1194 N. Mathilda Ave Sunnyvale, CA 94089 USA EMail: jgs@juniper.net

David Ward editor Ward Juniper Networks 1194 N. Mathilda Ave Sunnyvale, CA 94089 USA EMail: dward@juniper.net

Randy Bush editor Bush Internet Initiative Japan, Inc. 5147 Crystral Springs Bainbridge Island, Washington 98110 USA EMail: randy@psg.com

Rob Austein editor Austein Internet Systems Consortium 950 Charter Street Redwood City, CA 94063 USA EMail: sra@isc.org