OPSEC Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: October 22, 2018

K. Sriram
D. Montgomery
USA NIST
J. Haas
Juniper Networks, Inc.

April 20, 2018

Enhanced Feasible-Path Unicast Reverse Path Filtering
draft-ietf-opsec-urpf-improvements-00

Abstract

This document identifies a need for improvement of the unicast Reverse Path Filtering techniques (uRPF) [BCP84] for source address validation (SAV) [BCP38]. The strict uRPF is inflexible about directionality, the loose uRPF is oblivious to directionality, and the current feasible-path uRPF attempts to strike a balance between the two [BCP84]. However, as shown in this draft, the existing feasible-path uRPF still has short comings. This document describes an enhanced feasible-path uRPF technique, which aims to be more flexible (in a meaningful way) about directionality than the feasible-path uRPF. It can potentially alleviate ISPs' concerns about the possibility of disrupting service for their customers, and encourage greater deployment of uRPF techniques.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This internet draft identifies a need for improvement of the unicast Reverse Path Filtering (uRPF) techniques [RFC2827] for source address validation (SAV) [RFC3704]. The strict uRPF is inflexible about directionality, the loose uRPF is oblivious to directionality, and the current feasible-path uRPF attempts to strike a balance between the two [RFC3704]. However, as shown in this draft, the existing feasible-path uRPF still has short comings. Even with the feasible-path uRPF, ISPs are often apprehensive that they may be dropping customers' data packets with legitimate source addresses.

This document describes an enhanced feasible-path uRPF technique, which aims to be more flexible (in a meaningful way) about directionality than the feasible-path uRPF. It is based on the principle that if BGP updates for multiple prefixes with the same origin AS were received on different interfaces (at border routers), then incoming data packets with source addresses in any of those prefixes should be accepted on any of those interfaces (presented in Section 3). For some challenging ISP-customer scenarios (see Section 3.3), this document also describes a more relaxed version of the enhanced feasible-path uRPF technique (presented in Section 3.4). Implementation considerations are discussed in Section 3.5.

Note: Definition of Reverse Path Filtering (RPF) list: The list of permissible source address prefixes for incoming data packets on a given interface.

Note: Throughout this document, the routes in consideration are assumed to have been vetted based on prefix filtering [RFC7454] and possibly (in the future) origin validation [RFC6811].

The enhanced feasible-path uRPF methods described here are expected to add greater operational robustness and efficacy to uRPF, while minimizing ISPs' concerns about accidental service disruption for their customers. It is expected that this will encourage more deployment of uRPF to help realize its DDoS prevention benefits network wide.

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. Review of Existing Source Address Validation Techniques

There are various existing techniques for mitigation against DDoS attacks with spoofed addresses [RFC2827] [RFC3704]. There are also some techniques used for mitigating reflection attacks [RRL] [TA14-017A], which are used to amplify the impact in DDoS attacks. Employing a combination of these preventive techniques (as applicable) in enterprise and ISP border routers, broadband and wireless access network, data centers, and DNS servers provides reasonably effective protection against DDoS attacks.

Source address validation (SAV) is performed in network edge devices such as border routers, Cable Modem Termination Systems (CMTS), Digital Subscriber Line Access Multiplexers (DSLAM), and Packet Data Network (PDN) gateways in mobile networks. Ingress Access Control

List (ACL) and unicast Reverse Path Filtering (uRPF) are techniques employed for implementing SAV [RFC2827] [RFC3704] [ISOC].

2.1. SAV using Access Control List

Ingress/egress Access Control Lists (ACLs) are maintained which list acceptable (or alternatively, unacceptable) prefixes for the source addresses in the incoming Internet Protocol (IP) packets. Any packet with a source address that does not match the filter is dropped. The ACLs for the ingress/egress filters need to be maintained to keep them up to date. Updating the ACLs is an operator driven manual process, and hence operationally difficult or infeasible.

Typically, the egress ACLs in access aggregation devices (e.g. CMTS, DSLAM) permit source addresses only from the address spaces (prefixes) that are associated with the interface on which the customer network is connected. Ingress ACLs are typically deployed on border routers, and drop ingress packets when the source address is spoofed (i.e. belongs to obviously disallowed prefix blocks, RFC 1918 prefixes, or provider's own prefixes).

2.2. SAV using Strict Unicast Reverse Path Filtering

In the strict unicast Reverse Path Filtering (uRPF) method, an ingress packet at border router is accepted only if the Forwarding Information Base (FIB) contains a prefix that encompasses the source address, and forwarding information for that prefix points back to the interface over which the packet was received. In other words, the reverse path for routing to the source address (if it were used as a destination address) should use the same interface over which the packet was received. It is well known that this method has limitations when networks are multi-homed and there is asymmetric routing of packets. Asymmetric routing occurs (see Figure 1) when a customer AS announces one prefix (P1) to one transit provider (ISP-a) and a different prefix (P2) to another transit provider (ISP-b), but routes data packets with source addresses in the second prefix (P2) to the first transit provider (ISP-a) or vice versa.

```
              +------------+ ---- P1[AS2 AS1] ---> +------------+
              | AS2(ISP-a) | <----P2[AS3 AS1] ---- |  AS3(ISP-b)|
              +------------+                       +------------+
                       /\                             /\
                        \                             /
                         \                           /
                          \                         /
                    P1[AS1]\                       /P2[AS1]
                            \                     /
                           +-----------------------+
                           |      AS1(customer)    |
                           +-----------------------+
                             P1, P2 (prefixes originated)

             Consider data packets received at AS2
             (1) from AS1 with source address in P2, or
             (2) from AS3 that originated from AS1
                 with source address in P1:
                       * Strict uRPF fails
                       * Feasible-path uRPF fails
                       * Loose uRPF works (but ineffective in IPv4)
                       * Enhanced Feasible-path uRPF works best
```

Figure 1: Scenario 1 for illustration of efficacy of uRPF schemes.

2.3. SAV using Feasible-Path Unicast Reverse Path Filtering

The feasible-path uRPF helps partially overcome the problem identified with the strict uRPF in the multi-homing case. The feasible-path uRPF is similar to the strict uRPF, but in addition to inserting the best-path prefix, additional prefixes from alternative announced routes are also included in the RPF table. This method relies on announcements for the same prefixes (albeit some may be prepended to effect lower preference) propagating to all routers performing feasible-path uRPF checks. Therefore, in the multi-homing scenario, if the customer AS announces routes for both prefixes (P1, P2) to both transit providers (with suitable prepends if needed for traffic engineering), then the feasible-path uRPF method works (see Figure 2). It should be mentioned that the feasible-path uRPF works in this scenario only if customer routes are preferred at AS2 and AS3 over a shorter non-customer route.

```
             +------------+  routes for P1, P2   +-----------+
             | AS2(ISP-a) |<-------------------->| AS3(ISP-b)|
             +------------+        (p2p)         +-----------+
                       /\                            /\
                        \                            /
                  P1[AS1]\                          /P2[AS1]
                          \                        /
            P2[AS1 AS1 AS1]\                      /P1[AS1 AS1 AS1]
                            \                    /
                           +-----------------------+
                           |      AS1(customer)    |
                           +-----------------------+
                             P1, P2 (prefixes originated)

           Consider data packets received at AS2 via AS3
           that originated from AS1 and have source address in P1:
           * Feasible-path uRPF works (if customer route to P1
             is preferred at AS3 over shorter path)
           * Feasible-path uRPF fails (if shorter path to P1
             is preferred at AS3 over customer route)
           * Loose uRPF works (but ineffective in IPv4)
           * Enhanced Feasible-path uRPF works best
```

Figure 2: Scenario 2 for illustration of efficacy of uRPF schemes.

However, the feasible-path uRPF method has limitations as well. One form of limitation naturally occurs when the recommendation of propagating the same prefixes to all routers is not followed. Another form of limitation can be described as follows. In Scenario 2 (described above, illustrated in Figure 2), it is possible that the second transit provider (ISP-b or AS3) does not propagate the prepended route for prefix P1 to the first transit provider (ISP-a or AS2). This is because AS3's decision policy permits giving priority to a shorter route to prefix P1 via a peer (AS2) over a longer route learned directly from the customer (AS1). In such a scenario, AS3 would not send any route announcement for prefix P1 to AS2. Then a data packet with source address in prefix P1 that originates from AS1 and traverses via AS3 to AS2 will get dropped at AS2.

2.4. SAV using Loose Unicast Reverse Path Filtering

In the loose unicast Reverse Path Filtering (uRPF) method, an ingress packet at the border router is accepted only if the FIB has one or more prefixes that encompass the source address. That is, a packet is dropped if no route exists in the FIB for the source address. Loose uRPF sacrifices directionality. This method is not effective for prevention of address spoofing since there is little unrouted address space in IPv4. It only drops packets if the spoofed address

is unreachable in the current FIB (e.g. RFC 1918, unallocated, allocated but currently not routed).

3. SAV using Enhanced Feasible-Path uRPF

3.1. Description of the Method

Enhanced feasible-path uRPF adds greater operational robustness and efficacy to existing uRPF methods discussed in Section 2. The technique is based on the principle that if BGP updates for multiple prefixes with the same origin AS were received on different interfaces (at border routers), then incoming data packets with source addresses in any of those prefixes should be accepted on any of those interfaces. It can be best explained with an example as follows:

Let us say, a border router of ISP-A has in its Adj-RIB-Ins [RFC4271]. the set of prefixes {Q1, Q2, Q3} each of which has AS-x as its origin and AS-x is in ISP-A's customer cone. Further, the border router received a route for prefix Q1 over a customer facing interface, while it learned routes for prefixes Q2 and Q3 from a lateral peer and an upstream transit provider, respectively. In this example scenario, the enhanced feasible-path uRPF method requires Q1, Q2, and Q3 be included in the RPF list for the customer interface in consideration. Loose uRPF (see Section 2.4) is recommended to be applied to the peer and provider interfaces in consideration.

Thus, enhanced feasible-path uRPF defines feasible paths for customer interfaces in a more generalized but precise way (as compared to feasible-path uRPF).

Looking back at Scenarios 1 and 2 (Figure 1 and Figure 2), the enhanced feasible-path uRPF provides comparable or better performance than the other uRPF methods. Scenario 3 (Figure 3) further illustrates the enhanced feasible-path uRPF method with a more concrete example. In this scenario, the focus is on operation of the feasible-path uRPF at ISP4 (AS4). ISP4 learns a route for prefix P1 via a customer-to-provider (C2P) interface from customer ISP2 (AS2). This route for P1 has origin AS1. ISP4 also learns a route for P2 via another C2P interface from customer ISP3 (AS3). Additionally, AS4 learns a route for P3 via a peer-to-peer (p2p) interface from ISP5 (AS5). Routes for all three prefixes have the same origin AS (i.e. AS1). Using the enhanced feasible-path uRPF scheme, given the commonality of the origin AS across the routes for P1, P2 and P3, AS4 includes all of these prefixes to the RPF list for the customer interfaces (from AS2 and AS3).

```
                    +----------+   P3[AS5 AS1]  +------------+
                    | AS4(ISP4)|<---------------|  AS5(ISP5) |
                    +----------+      (p2p)     +------------+
                        /\   /\                        /\
                        /     \                        /
            P1[AS2 AS1]/       \P2[AS3 AS1]           /
                 (C2P)/         \(C2P)               /
                     /           \                  /
              +----------+    +----------+         /
              | AS2(ISP2)|    | AS3(ISP3)|        /
              +----------+    +----------+       /
                       /\           /\          /
                        \           /          /
                  P1[AS1]\         /P2[AS1]   /P3[AS1]
                     (C2P)\       /(C2P)     /(C2P)
                           \     /          /
                        +----------------+ /
                        |  AS1(customer) |/
                        +----------------+
                             P1, P2, P3 (prefixes originated)

            Consider that data packets (sourced from AS1)
            may be received at AS4 with source address
            in P1, P2 or P3 via any of the neighbors (AS2, AS3, AS5):
            * Feasible-path uRPF fails
            * Loose uRPF works (but not desirable)
            * Enhanced Feasible-path uRPF works best
```

Figure 3: Scenario 3 for illustration of efficacy of uRPF schemes.

3.1.1. Algorithm A: Enhanced Feasible-Path uRPF

The underlying algorithm in the solution method described above can be specified as follows (to be implemented in a transit AS):

1. Create the list of unique origin ASes considering only the routes in the Adj-RIB-Ins of customer interfaces. Call it Set A = {AS1, AS2, ..., ASn}.

2. Considering all routes in Adj-RIB-Ins for all interfaces (customer, lateral peer, and provider), form the set of unique prefixes that have a common origin AS1. Call it Set X1.

3. Include set X1 in Reverse Path Filter (RPF) list on all customer interfaces on which one or more of the prefixes in set X1 were received.

4. Repeat Steps 2 and 3 for each of the remaining ASes in Set A (i.e., for ASi, where i = 2, ..., n).

3.2. Operational Recommendations

The following operational recommendations will make the operation of the enhanced feasible-path uRPF robust:

For multi-homed stub AS:

- A multi-homed stub AS SHOULD announce at least one of the prefixes it originates to each of its transit provider ASes.

For non-stub AS:

- A non-stub AS SHOULD also announce at least one of the prefixes it originates to each of its transit provider ASes.

- Additionally, from the routes it has learned from customers, a non-stub AS SHOULD announce at least one route per origin AS to each of its transit provider ASes.

(Note: It is worth noting that in the above recommendations if "at least one" is replaced with "all", then even traditional feasible-path uRPF will work as effectively.)

3.3. A Challenging Scenario

It should be observed that in the absence of ASes adhering the above recommendations, the following example scenario may be constructed which poses a challenge for the enhanced feasible-path uRPF (as well as for traditional feasible-path uRPF). In the scenario illustrated in Figure 4, since routes for neither P1 nor P2 are propagated on the AS2-AS4 interface, the enhanced feasible-path uRPF at AS4 will reject data packets received on that interface with source addresses in P1 or P2.

```
                    +----------+
                    | AS4(ISP4)|
                    +----------+
                        /\   /\
                        /     \  P1[AS3 AS1]
         P1 and P2 not /       \ P2[AS3 AS1]
           propagated /         \ (C2P)
             (C2P)   /           \
              +----------+    +----------+
              | AS2(ISP2)|    | AS3(ISP3)|
              +----------+    +----------+
                       /\           /\
                        \           / P1[AS1]
       P1[AS1] NO_EXPORT \         / P2[AS1]
       P2[AS1] NO_EXPORT  \       / (C2P)
                    (C2P)  \     /
                        +----------------+
                        |  AS1(customer) |
                        +----------------+
                             P1, P2 (prefixes originated)
```

Figure 4: Illustration of a challenging scenario.

3.4. Algorithm B: Enhanced Feasible-Path uRPF with Additional Flexibility Across Customer Cone

Adding further flexibility to the enhanced feasible-path uRPF method can help address the potential limitation identified above using the scenario in Figure 4 (Section 3.3). In the following, "route" refers to a route currently existing in the Adj-RIB-in. Including the additional degree of flexibility, the modified algorithm (implemented in a transit AS) can be described as follows (we call this Algorithm B):

1. Create the set of all directly-connected customer interfaces. Call it Set I = {I1, I2, ..., Ik}.

2. Create the set of all unique prefixes for which routes exist in Adj-RIB-Ins for the interfaces in Set I. Call it Set P = {P1, P2, ..., Pm}.

3. Create the set of all unique origin ASes seen in the routes that exist in Adj-RIB-Ins for the interfaces in Set I. Call it Set A = {AS1, AS2, ..., ASn}.

4. Create the set of all unique prefixes for which routes exist in Adj-RIB-Ins of all lateral peer and provider interfaces such that

each of the routes has its origin AS belonging in Set A. Call it Set Q = {Q1, Q2, ..., Qj}.

5. Then, Set Z = Union(P,Q) represents the RPF list for every customer interface in Set I.

6. Apply loose uRPF method for SAV on all peer and provider interfaces.

When Algorithm B (which is more flexible than Algorithm A) is employed, the type of limitation identified in Figure 4 (Section 3.3) goes away.

3.5. Implementation Considerations

The existing RPF checks in edge routers take advantage of existing line card implementations to perform the RPF functions. For implementation of the enhanced feasible-path uRPF, the general necessary feature would be to extend the line cards to take arbitrary RPF lists that are not necessarily the same as the existing FIB contents. In the algorithms (Section 3.1.1 and Section 3.4) described here, the RPF lists are constructed by applying a set of rules to all received BGP routes (not just those selected as best path and installed in FIB).

3.5.1. Impact on FIB Memory Size Requirement

The techniques described here require that there should be FIB memory (i.e., TCAM) available to store the RPF lists in line cards. For an ISP's AS, the RPF list size for each line card will roughly and conservatively equal the total number of prefixes in its customer cone (assuming the algorithm in Section 3.4 is used). (Note: Most ISP customer cone scenarios would not require the algorithm in Section 3.4, but instead be served best by the algorithm in Section 3.1.1, which requires much less FIB memory.) The following table shows the measured customer cone sizes for various types of ISPs [sriram-ripe63]:

Type of ISP	Measured Customer Cone Size in # Prefixes (in turn this is an estimate for RPF list size on line card)
Very Large Global ISP	32392
Very Large Global ISP	29528
Large Global ISP	20038
Mid-size Global ISP	8661
Regional ISP (in Asia)	1101

Table 1: Customer cone sizes (# prefixes) for various types of ISPs.

For some super large global ISPs that are at the core of the Internet, the customer cone size (# prefixes) can be as high as a few hundred thousand [CAIDA]. But uRPF is most effective when deployed at ASes at the edges of the Internet where the customer cone sizes are smaller as shown in Table 1.

A very large global ISP's router line card is likely to have a FIB size large enough to accommodate 2 to 6 million routes [cisco1]. Similarly, the line cards in routers corresponding to a large global ISP, a mid-size global ISP, and a regional ISP are likely to have FIB sizes large enough to accommodate about 1 million, 0.5 million, and 100K routes, respectively [cisco2]. Comparing these FIB size numbers with the corresponding RPF list size numbers in Table 1, it can be surmised that the conservatively estimated RPF list size is only a small fraction of the anticipated FIB memory size under relevant ISP scenarios.

3.6. Summary of Recommendations

Depending on the scenario, an ISP or enterprise AS operator should follow one of the following recommendations concerning uRPF/SAV:

1. For directly connected networks, i.e., subnets directly connected to the AS and not multi-homed, the AS in consideration SHOULD perform ACL-based SAV.

2. For a directly connected single-homed stub AS (customer), the AS in consideration SHOULD perform SAV based on the strict uRPF method.

3. For all other scenarios:

   - If the scenario does not involve complexity such as NO_EXPORT of routes (see Section 3.3, Figure 4), then the enhanced feasible-path uRPF method in Algorithm A (see Section 3.1.1) SHOULD be applied.

   - Else, if the scenario involves the aforementioned complexity, then the enhanced feasible-path uRPF method in Algorithm B (see Section 3.4) SHOULD be applied.

4. Security Considerations

The security considerations in BCP 38 [RFC2827] and BCP 84 [RFC3704] apply for this document as well. In addition, AS operator should apply the uRPF method that performs best (i.e., with zero or insignificant possibility of dropping legitimate data packets) for the type of peer (customer, provider, etc.) and the nature of customer cone scenario that apply (see Section 3.1.1 and Section 3.4).

5. IANA Considerations

This document does not request new capabilities or attributes. It does not create any new IANA registries.

6. Acknowledgements

The authors would like to thank Job Snijders, Marco Marzetti, Marco d'Itri, Nick Hilliard, Gert Doering, Igor Gashinsky, Barry Greene, and Joel Jaeggli for comments and suggestions.

7. Informative References

[CAIDA] "Information for AS 174 (COGENT-174)", CAIDA Spoofer Project , <https://spoofer.caida.org/as.php?asn=174>.

[cisco1] "Internet Routing Table Growth Causes ROUTING-FIB-4-RSRC_LOW Message on Trident-Based Line Cards", Cisco Trouble-shooting Tech-notes , January 2014, <https://www.cisco.com/c/en/us/support/docs/routers/asr-9000-series-aggregation-services-routers/116999-problem-line-card-00.html>.

[cisco2] "Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 5.x (Chapter: Managing the Unicast RIB and FIB)", Cisco Configuration Guides , June 2018, <https://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/5_x/nx-os/unicast/configuration/guide/l3_cli_nxos/l3_manage-routes.html#22859>.

[ISOC] Vixie (Ed.), P., "Addressing the challenge of IP spoofing", ISOC report , September 2015, <https://www.us-cert.gov/ncas/alerts/TA14-017A>.

[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>.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <https://www.rfc-editor.org/info/rfc2827>.

[RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, DOI 10.17487/RFC3704, March 2004, <https://www.rfc-editor.org/info/rfc3704>.

[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>.

[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>.

[RFC7454] Durand, J., Pepelnjak, I., and G. Doering, "BGP Operations and Security", BCP 194, RFC 7454, DOI 10.17487/RFC7454, February 2015, <https://www.rfc-editor.org/info/rfc7454>.

[RRL] "Response Rate Limiting in the Domain Name System", Redbarn blog , <http://www.redbarn.org/dns/ratelimits>.

[sriram-ripe63]
Sriram, K. and R. Bush, "Estimating CPU Cost of BGPSEC on a Router", Presented at RIPE-63; also at IETF-83 SIDR WG Meeting, March 2012, <http://www.ietf.org/proceedings/83/slides/slides-83-sidr-7.pdf>.

[TA14-017A]
"UDP-Based Amplification Attacks", US-CERT alert TA14-017A , January 2014, <https://www.us-cert.gov/ncas/alerts/TA14-017A>.

Authors' Addresses

Kotikalapudi Sriram
USA National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg MD 20899
USA

Email: ksriram@nist.gov

Doug Montgomery
USA National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg MD 20899
USA

Email: dougm@nist.gov

Jeffrey Haas
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale CA 94089
USA

Email: jhaas@juniper.net