Network Working Group
Request for Comments: 3704
Updates: 2827
BCP: 84
Category: Best Current Practice

F. Baker
Cisco Systems
P. Savola
CSC/FUNET

March 2004

Ingress Filtering for Multihomed Networks

Status of this Memo

Copyright Notice

Abstract

BCP 38, RFC 2827, is designed to limit the impact of distributed denial of service attacks, by denying traffic with spoofed addresses access to the network, and to help ensure that traffic is traceable to its correct source network. As a side effect of protecting the Internet against such attacks, the network implementing the solution also protects itself from this and other attacks, such as spoofed management access to networking equipment. There are cases when this may create problems, e.g., with multihoming. This document describes the current ingress filtering operational mechanisms, examines generic issues related to ingress filtering, and delves into the effects on multihoming in particular. This memo updates RFC 2827.

Table of Contents

1. Introduction

BCP 38, RFC 2827 [1], is designed to limit the impact of distributed denial of service attacks, by denying traffic with spoofed addresses access to the network, and to help ensure that traffic is traceable to its correct source network. As a side effect of protecting the Internet against such attacks, the network implementing the solution also protects itself from this and other attacks, such as spoofed management access to networking equipment. There are cases when this may create problems, e.g., with multihoming. This document describes the current ingress filtering operational mechanisms, examines generic issues related to ingress filtering and delves into the effects on multihoming in particular.

RFC 2827 recommends that ISPs police their customers' traffic by dropping traffic entering their networks that is coming from a source address not legitimately in use by the customer network. The filtering includes but is in no way limited to the traffic whose source address is a so-called "Martian Address" - an address that is reserved [3], including any address within 0.0.0.0/8, 10.0.0.0/8, 127.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, 224.0.0.0/4, or 240.0.0.0/4.

The reasoning behind the ingress filtering procedure is that Distributed Denial of Service Attacks frequently spoof other systems' source addresses, placing a random number in the field. In some attacks, this random number is deterministically within the target network, simultaneously attacking one or more machines and causing those machines to attack others with ICMP messages or other traffic; in this case, the attacked sites can protect themselves by proper filtering, by verifying that their prefixes are not used in the source addresses in packets received from the Internet. In other attacks, the source address is literally a random 32 bit number, resulting in the source of the attack being difficult to trace. If the traffic leaving an edge network and entering an ISP can be limited to traffic it is legitimately sending, attacks can be somewhat mitigated: traffic with random or improper source addresses can be suppressed before it does significant damage, and attacks can be readily traced back to at least their source networks.

This document is aimed at ISP and edge network operators who 1) would like to learn more of ingress filtering methods in general, or 2) are already using ingress filtering to some degree but who would like to expand its use and want to avoid the pitfalls of ingress filtering in the multihomed/asymmetric scenarios.

In section 2, several different ways to implement ingress filtering are described and examined in the generic context. In section 3, some clarifications on the applicability of ingress filtering methods are made. In section 4, ingress filtering is analyzed in detail from the multihoming perspective. In section 5, conclusions and potential future work items are identified.

2. Different Ways to Implement Ingress Filtering

This section serves as an introduction to different operational techniques used to implement ingress filtering as of writing this memo. The mechanisms are described and analyzed in general terms, and multihoming-specific issues are described in Section 4.

There are at least five ways one can implement RFC 2827, with varying impacts. These include (the names are in relatively common usage):

- Ingress Access Lists
- Strict Reverse Path Forwarding
- Feasible Path Reverse Path Forwarding
- Loose Reverse Path Forwarding
- Loose Reverse Path Forwarding ignoring default routes

Other mechanisms are also possible, and indeed, there are a number of techniques that might profit from further study, specification, implementation, and/or deployment; see Section 6. However, these are out of scope.

2.1. Ingress Access Lists

An Ingress Access List is a filter that checks the source address of every message received on a network interface against a list of acceptable prefixes, dropping any packet that does not match the filter. While this is by no means the only way to implement an ingress filter, it is the one proposed by RFC 2827 [1], and in some sense the most deterministic one.

However, Ingress Access Lists are typically maintained manually; for example, forgetting to have the list updated at the ISPs if the set of prefixes changes (e.g., as a result of multihoming) might lead to discarding the packets if they do not pass the ingress filter.

Naturally, this problem is not limited to Ingress Access Lists -- it is inherent to Ingress Filtering when the ingress filter is not complete. However, usually Ingress Access Lists are more difficult to maintain than the other mechanisms, and having an outdated list can prevent legitimate access.

2.2. Strict Reverse Path Forwarding

Strict Reverse Path Forwarding (Strict RPF) is a simple way to implement an ingress filter. It is conceptually identical to using access lists for ingress filtering, with the exception that the access list is dynamic. This may also be used to avoid duplicate configuration (e.g., maintaining both static routes or BGP prefix-list filters and interface access-lists). The procedure is that the source address is looked up in the Forwarding Information Base (FIB) - and if the packet is received on the interface which would be used to forward the traffic to the source of the packet, it passes the check.

Strict Reverse Path Forwarding is a very reasonable approach in front of any kind of edge network; in particular, it is far superior to Ingress Access Lists when the network edge is advertising multiple prefixes using BGP. It makes for a simple, cheap, fast, and dynamic filter.

But Strict Reverse Path Forwarding has some problems of its own. First, the test is only applicable in places where routing is symmetrical - where IP datagrams in one direction and responses from the other deterministically follow the same path. While this is common at edge network interfaces to their ISP, it is in no sense common between ISPs, which normally use asymmetrical "hot potato" routing. Also, if BGP is carrying prefixes and some legitimate prefixes are not being advertised or not being accepted by the ISP under its policy, the effect is the same as ingress filtering using an incomplete access list: some legitimate traffic is filtered for lack of a route in the filtering router’s Forwarding Information Base.

There are operational techniques, especially with BGP but somewhat applicable to other routing protocols as well, to make strict RPF work better in the case of asymmetric or multihomed traffic. The ISP assigns a better metric which is not propagated outside of the router, either a vendor-specific "weight" or a protocol distance to prefer the directly received routes. With BGP and sufficient machinery in place, setting the preferences could even be automated, using BGP Communities [2]. That way, the route will always be the best one in the FIB, even in the scenarios where only the primary connectivity would be used and typically no packets would pass

through the interface. This method assumes that there is no strict RPF filtering between the primary and secondary edge routers; in particular, when applied to multihoming to different ISPs, this assumption may fail.

2.3. Feasible Path Reverse Path Forwarding

Feasible Path Reverse Path Forwarding (Feasible RPF) is an extension of Strict RPF. The source address is still looked up in the FIB (or an equivalent, RPF-specific table) but instead of just inserting one best route there, the alternative paths (if any) have been added as well, and are valid for consideration. The list is populated using routing-protocol specific methods, for example by including all or N (where N is less than all) feasible BGP paths in the Routing Information Base (RIB). Sometimes this method has been implemented as part of a Strict RPF implementation.

In the case of asymmetric routing and/or multihoming at the edge of the network, this approach provides a way to relatively easily address the biggest problems of Strict RPF.

It is critical to understand the context in which Feasible RPF operates. The mechanism relies on consistent route advertisements (i.e., the same prefix(es), through all the paths) propagating to all the routers performing Feasible RPF checking. For example, this may not hold e.g., in the case where a secondary ISP does not propagate the BGP advertisement to the primary ISP e.g., due to route-maps or other routing policies not being up-to-date. The failure modes are typically similar to "operationally enhanced Strict RPF", as described above.

As a general guideline, if an advertisement is filtered, the packets will be filtered as well.

In consequence, properly defined, Feasible RPF is a very powerful tool in certain kinds of asymmetric routing scenarios, but it is important to understand its operational role and applicability better.

2.4. Loose Reverse Path Forwarding

Loose Reverse Path Forwarding (Loose RPF) is algorithmically similar to strict RPF, but differs in that it checks only for the existence of a route (even a default route, if applicable), not where the route points to. Practically, this could be considered as a "route presence check" ("loose RPF is a misnomer in a sense because there is no "reverse path" check in the first place).

The questionable benefit of Loose RPF is found in asymmetric routing situations: a packet is dropped if there is no route at all, such as to "Martian addresses" or addresses that are not currently routed, but is not dropped if a route exists.

Loose Reverse Path Forwarding has problems, however. Since it sacrifices directionality, it loses the ability to limit an edge network's traffic to traffic legitimately sourced from that network, in most cases, rendering the mechanism useless as an ingress filtering mechanism.

Also, many ISPs use default routes for various purposes such as collecting illegitimate traffic at so-called "Honey Pot" systems or discarding any traffic they do not have a "real" route to, and smaller ISPs may well purchase transit capabilities and use a default route from a larger provider. At least some implementations of Loose RPF check where the default route points to. If the route points to the interface where Loose RPF is enabled, any packet is allowed from that interface; if it points nowhere or to some other interface, the packets with bogus source addresses will be discarded at the Loose RPF interface even in the presence of a default route. If such fine-grained checking is not implemented, presence of a default route nullifies the effect of Loose RPF completely.

One case where Loose RPF might fit well could be an ISP filtering packets from its upstream providers, to get rid of packets with "Martian" or other non-routed addresses.

If other approaches are unsuitable, loose RPF could be used as a form of contract verification: the other network is presumably certifying that it has provided appropriate ingress filtering rules, so the network doing the filtering need only verify the fact and react if any packets which would show a breach in the contract are detected. Of course, this mechanism would only show if the source addresses used are "martian" or other unrouted addresses -- not if they are from someone else's address space.

2.5. Loose Reverse Path Forwarding Ignoring Default Routes

The fifth implementation technique may be characterized as Loose RPF ignoring default routes, i.e., an "explicit route presence check". In this approach, the router looks up the source address in the route table, and preserves the packet if a route is found. However, in the lookup, default routes are excluded. Therefore, the technique is mostly usable in scenarios where default routes are used only to catch traffic with bogus source addresses, with an extensive (or even full) list of explicit routes to cover legitimate traffic.

Like Loose RPF, this is useful in places where asymmetric routing is found, such as on inter-ISP links. However, like Loose RPF, since it sacrifices directionality, it loses the ability to limit an edge network's traffic to traffic legitimately sourced from that network.

3. Clarifying the Applicability of Ingress Filtering

What may not be readily apparent is that ingress filtering is not applied only at the "last-mile" interface between the ISP and the end user. It's perfectly fine, and recommended, to also perform ingress filtering at the edges of ISPs where appropriate, at the routers connecting LANs to an enterprise network, etc. -- this increases the defense in depth.

3.1. Ingress Filtering at Multiple Levels

Because of wider deployment of ingress filtering, the issue is recursive. Ingress filtering has to work everywhere where it's used, not just between the first two parties. That is, if a user negotiates a special ingress filtering arrangement with his ISP, he should also ensure (or make sure the ISP ensures) that the same arrangements also apply to the ISP's upstream and peering links, if ingress filtering is being used there -- or will get used, at some point in the future; similarly with the upstream ISPs and peers.

In consequence, manual models which do not automatically propagate the information to every party where the packets would go and where ingress filtering might be applied have only limited generic usefulness.

3.2. Ingress Filtering to Protect Your Own Infrastructure

Another feature stemming from wider deployment of ingress filtering may not be readily apparent. The routers and other ISP infrastructure are vulnerable to several kinds of attacks. The threat is typically mitigated by restricting who can access these systems.

However, unless ingress filtering (or at least, a limited subset of it) has been deployed at every border (towards the customers, peers and upstreams) -- blocking the use of your own addresses as source addresses -- the attackers may be able to circumvent the protections of the infrastructure gear.

Therefore, by deploying ingress filtering, one does not just help the Internet as a whole, but protects against several classes of threats to your own infrastructure as well.

3.3. Ingress Filtering on Peering Links

Ingress filtering on peering links, whether by ISPs or by end-sites, is not really that much different from the more typical "downstream" or "upstream" ingress filtering.

However, it's important to note that with mixed upstream/downstream and peering links, the different links may have different properties (e.g., relating to contracts, trust, viability of the ingress filtering mechanisms, etc.). In the most typical case, just using an ingress filtering mechanism towards a peer (e.g., Strict RPF) works just fine as long as the routing between the peers is kept reasonably symmetric. It might even be considered useful to be able to filter out source addresses coming from an upstream link which should have come over a peering link (implying something like Strict RPF is used towards the upstream) -- but this is a more complex topic and considered out of scope; see Section 6.

4. Solutions to Ingress Filtering with Multihoming

First, one must ask why a site multihomes; for example, the edge network might:

- use two ISPs for backing up the Internet connectivity to ensure robustness,
- use whichever ISP is offering the fastest TCP service at the moment,
- need several points of access to the Internet in places where no one ISP offers service, or
- be changing ISPs (and therefore multihoming only temporarily).

One can imagine a number of approaches to working around the limitations of ingress filters for multihomed networks. Options include:

1. Do not multihome.
2. Do not use ingress filters.
3. Accept that service will be incomplete.
4. On some interfaces, weaken ingress filtering by using an appropriate form of loose RPF check, as described in Section 4.1.

5. Ensure, by BGP or by contract, that each ISP's ingress filter is complete, as described in Section 4.2.

6. Ensure that edge networks only deliver traffic to their ISPs that will in fact pass the ingress filter, as described in Section 4.3.

The first three of these are obviously mentioned for completeness; they are not and cannot be viable positions; the final three are considered below.

The fourth and the fifth must be ensured in the upstream ISPs as well, as described in Section 3.1.

Next, we now look at the viable ways for dealing with the side-effects of ingress filters.

4.1. Use Loose RPF When Appropriate

Where asymmetric routing is preferred or is unavoidable, ingress filtering may be difficult to deploy using a mechanism such as strict RPF which requires the paths to be symmetrical. In many cases, using operational methods or feasible RPF may ensure the ingress filter is complete, like described below. Failing that, the only real options are to not perform ingress filtering, use a manual access-list (possibly in addition to some other mechanisms), or to using some form of Loose RPF check.

Failing to provide any ingress filter at all essentially trusts the downstream network to behave itself, which is not the wisest course of action. However, especially in the case of very large networks of even hundreds or thousands of prefixes, maintaining manual access-lists may be too much to ask.

The use of Loose RPF does not seem like a good choice between the edge network and the ISP, since it loses the directionality of the test. This argues in favor of either using a complete filter in the upstream network or ensuring in the downstream network that packets the upstream network will reject will never reach it.

Therefore, the use of Loose RPF cannot be recommended, except as a way to measure whether "martian" or other unrouted addresses are being used.

4.2. Ensure That Each ISP's Ingress Filter Is Complete

For the edge network, if multihoming is being used for robustness or to change routing from time to time depending on measured ISP behavior, the simplest approach will be to ensure that its ISPs in fact carry its addresses in routing. This will often require the edge network to use provider-independent prefixes and exchange routes with its ISPs with BGP, to ensure that its prefix is carried upstream to the major transit ISPs. Of necessity, this implies that the edge network will be of a size and technical competence to qualify for a separate address assignment and an autonomous system number from its RIR.

There are a number of techniques which make it easier to ensure the ISP's ingress filter is complete. Feasible RPF and Strict RPF with operational techniques both work quite well for multihomed or asymmetric scenarios between the ISP and an edge network.

When a routing protocol is not being used, but rather the customer information is generated from databases such as Radius, TACACS, or Diameter, the ingress filtering can be the most easily ensured and kept up-to-date with Strict RPF or Ingress Access Lists generated automatically from such databases.

4.3. Send Traffic Using a Provider Prefix Only to That Provider

For smaller edge networks that use provider-based addressing and whose ISPs implement ingress filters (which they should do), the third option is to route traffic being sourced from a given provider's address space to that provider.

This is not a complicated procedure, but requires careful planning and configuration. For robustness, the edge network may choose to connect to each of its ISPs through two or more different Points of Presence (POPs), so that if one POP or line experiences an outage, another link to the same ISP can be used. Alternatively, a set of tunnels could be configured instead of multiple connections to the same ISP [4][5]. This way the edge routers are configured to first inspect the source address of a packet destined to an ISP and shunt it into the appropriate tunnel or interface toward the ISP.

If such a scenario is applied exhaustively, so that an exit router is chosen in the edge network for every prefix the network uses, traffic originating from any other prefix can be summarily discarded instead of sending it to an ISP.

5. Security Considerations

Ingress filtering is typically performed to ensure that traffic arriving on one network interface legitimately comes from a computer residing on a network reachable through that interface.

The closer to the actual source ingress filtering is performed, the more effective it is. One could wish that the first hop router would ensure that traffic being sourced from its neighboring end system was correctly addressed; a router further away can only ensure that it is possible that there is such a system within the indicated prefix. Therefore, ingress filtering should be done at multiple levels, with different level of granularity.

It bears to keep in mind that while one goal of ingress filtering is to make attacks traceable, it is impossible to know whether the particular attacker "somewhere in the Internet" is being ingress filtered or not. Therefore, one can only guess whether the source addresses have been spoofed or not: in any case, getting a possible lead -- e.g., to contact a potential source to ask whether they're observing an attack or not -- is still valuable, and more so when the ingress filtering gets more and more widely deployed.

In consequence, every administrative domain should try to ensure a sufficient level of ingress filtering on its borders.

Security properties and applicability of different ingress filtering types differ a lot.

- Ingress Access Lists require typically manual maintenance, but are the most bulletproof when done properly; typically, ingress access lists are best fit between the edge and the ISP when the configuration is not too dynamic if strict RPF is not an option, between ISPs if the number of used prefixes is low, or as an additional layer of protection.

- Strict RPF check is a very easy and sure way to implement ingress filtering. It is typically fit between the edge network and the ISP. In many cases, a simple strict RPF can be augmented by operational procedures in the case of asymmetric traffic patterns, or the feasible RPF technique to also account for other alternative paths.

- Feasible Path RPF check is an extension of Strict RPF. It is suitable in all the scenarios where Strict RPF is, but multihomed or asymmetric scenarios in particular. However, one must remember that Feasible RPF assumes the consistent origination and

propagation of routing information to work; the implications of this must be understood especially if a prefix advertisement passes through third parties.

- Loose RPF primarily filters out unrouted prefixes such as Martian addresses. It can be applied in the upstream interfaces to reduce the size of DoS attacks with unrouted source addresses. In the downstream interfaces it can only be used as a contract verification, that the other network has performed at least some ingress filtering.

When weighing the tradeoffs of different ingress filtering mechanisms, the security properties of a more relaxed approach should be carefully considered before applying it. Especially when applied by an ISP towards an edge network, there don't seem to be many reasons why a stricter form of ingress filtering would not be appropriate.

6. Conclusions and Future Work

This memo describes ingress filtering techniques in general and the options for multihomed networks in particular.

It is important for ISPs to implement ingress filtering to prevent spoofed addresses being used, both to curtail DoS attacks and to make them more traceable, and to protect their own infrastructure. This memo describes mechanisms that could be used to achieve that effect, and the tradeoffs of those mechanisms.

To summarize:

- Ingress filtering should always be done between the ISP and a single-homed edge network.

- Ingress filtering with Feasible RPF or similar Strict RPF techniques could almost always be applied between the ISP and multi-homed edge networks as well.

- Both the ISPs and edge networks should verify that their own addresses are not being used in source addresses in the packets coming from outside their network.

- Some form of ingress filtering is also reasonable between ISPs, especially if the number of prefixes is low.

This memo will lower the bar for the adoption of ingress filtering especially in the scenarios like asymmetric/multihomed networks where the general belief has been that ingress filtering is difficult to implement.

One can identify multiple areas where additional work would be useful:

- Specify the mechanisms in more detail: there is some variance between implementations e.g., on whether traffic to multicast destination addresses will always pass the Strict RPF filter or not. By formally specifying the mechanisms the implementations might get harmonized.

- Study and specify Routing Information Base (RIB) -based RPF mechanisms, e.g., Feasible Path RPF, in more detail. In particular, consider under which assumptions these mechanisms work as intended and where they don't.

- Write a more generic note on the ingress filtering mechanisms than this memo, after the taxonomy and the details or the mechanisms (points above) have been fleshed out.

- Consider the more complex case where a network has connectivity with different properties (e.g., peers and upstreams), and wants to ensure that traffic sourced with a peer's address should not be accepted from the upstream.

7. Acknowledgements

Rob Austein, Barry Greene, Christoph Reichert, Daniel Senie, Pedro Roque, and Iljitsch van Beijnum reviewed this document and helped in improving it. Thomas Narten, Ted Hardie, and Russ Housley provided good feedback which boosted the document in its final stages.

8. References

8.1. Normative References

[1] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

8.2. Informative References

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

[3] IANA, "Special-Use IPv4 Addresses", RFC 3330, September 2002.

[4] Bates, T. and Y. Rekhter, "Scalable Support for Multi-homed Multi-provider Connectivity", RFC 2260, January 1998.

[5] Hagino, J. and H. Snyder, "IPv6 Multihoming Support at Site Exit Routers", RFC 3178, October 2001.

9. Authors' Addresses

Fred Baker
Cisco Systems
Santa Barbara, CA 93117
US

EMail: fred@cisco.com

Pekka Savola
CSC/FUNET
Espoo
Finland

EMail: psavola@funet.fi

10. Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.