A simple guide to routing security

A. Robachevsky <robachevsky@isoc.org>

Context: MANRS Actions

Filtering
Ensure the correctness of your own announcements and of announcements from your customers to adjacent networks with prefix and AS-path granularity

Anti-Spoofing
Enable source address validation for at least single-homed stub customer networks, your own end-users, and infrastructure

Coordination
Maintain globally accessible up-to-date contact information

Routing Information
Publish your data, so others can validate routing information on a global scale

Tools
Provide monitoring and debugging tools to help others

Promotion
Actively encourage MANRS adoption among peers, customers, and partners

Context: MANRS Actions

Filtering
Ensure the correctness of your own announcements and of announcements from your customers to adjacent networks with prefix and AS-path granularity

Anti-Spoofing
Enable source address validation for at least single-homed stub customer networks, your own end-users, and infrastructure

Coordination
Maintain globally accessible up-to-date contact information

Routing Information
Publish your data, so others can validate routing information on a global scale

Tools
Provide monitoring and debugging tools to help others

Promotion
Actively encourage MANRS adoption among peers, customers, and partners

Too many places, too many options

- BGP Operations and Security, RFC7454
- MANRS Implementation Guide
- BCP38
- BCP84
- Peer-lock
- FullBogons
- …

What is the workflow for filtering?

What is the workflow for route filtering?

How and what the route servers filters
The DE-CIX filters are updated every 6 hours. Don't forget to register your IP prefixes in the IRR database well in advance (at least 24h before announcing the first time).

Bogon and Martian filtering
Please make sure not to announce routes that

- are > /24 (IPv4) and > /48 (IPv6) (RFC7454)
- have a different BGP next-hop to the IP of your own router
- are bogons/martians (private and reserved IP prefixes as defined by RFC1918, RFC2544, RFC3927, RFC 5735, RFC5737, RFC6598 and RFC6890)
- are a DE-CIX peering LAN (please also do not announce any of our peering LANs in the DFZ!)
- contain bogon ASNs in the BGP AS path (private and reserved ASN numbers as defined by RFC7607, RFC6793, RFC5398, RFC6996, RFC7300)
- differ in the leftmost ASN in the AS path from your own ASN
- have an AS path length > 32
- are < /8 (IPv4) and < /16 (IPv6) (RFC7454)

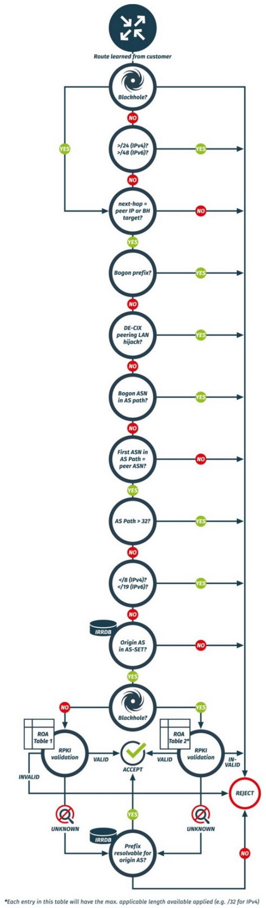
We will drop these kinds of routes.

Check the status of your routes
You can check the status of your announced routes to us in the DE-CIX Looking Glass – the reason why a route is filtered is also shown, as is a hint on how to fix the issue.

You can find more info on how to use the DE-CIX Looking Glass here.

IRR and RPKI validation
Any routes you announce will also be RPKI (RFC6811, RFC7115) validated and checked against Internet Routing Registry (IRR) data. The AS-SET you provide to us will be recursively resolved. Then filtering is executed as follows:

- The origin ASN needs to be in the customer cone (make sure that your AS-SET is well maintained and that all your downstreams are included)
- Is the route a blackhole (RFC7999)?
  - **If not, the route undergoes strict RPKI validation filtering** (both origin and maxLength):
    - If the result is RPKI Valid, the route is accepted (a missing route object will have no implication in this case).
    - If the result is RPKI Invalid, the route is rejected.
    - If the result is RPKI NotFound/Unknown, we check if the route is resolvable for its origin ASN (this will be the case if a proper route object exists) and it might get accepted or rejected depending on the result.**
  - **If it is, the route undergoes loose RPKI validation filtering** (origin only):
    - If the result is RPKI Valid, the route is accepted.
    - if the result is RPKI Invalid, the route is rejected.
    - If the result is RPKI NotFound/Unknown, we check if the route is resolvable for its origin



*Each entry in this table will have the max. applicable length available applied (e.g. /32 for IPv4)

https://www.de-cix.net/en/locations/frankfurt/route-server-guide

How to implement anti-spoofing?

How to implement anti-spoofing?

Guiding Principles for Anti-Spoofing Architectures

To be as effective as possible anti-spoofing techniques should be applied as close to the source as possible. In enterprise networks, the source addresses used by every device are often controlled and enforced so that security audits can pinpoint exactly which device sent which packet.

For a successful implementation of MANRS, such fine granularity at the device level is not necessary as MANRS focuses on routing security and anti-spoofing on a network level. Therefore common anti-spoofing architectures focus on making sure that customers don't send packets with the wrong sou

Enforcing the use of valid source addresses on a customer level has the benefit that custo each other's addresses, which prevents them from causing problems for each other that

If for some reason it is not possible to enforce source address usage per customer, then

https://github.com/manrs-tools/manrs-docs/blob/main/pdf/MANRS-Network-Implementation-Guide.pdf

RIPE Anti-Spoofing Task Force HOW-TO

Publication date: 09 May 2008

Introduction

This document presents practical recommendations for the implementation of anti-spoofing mechanisms at the critical points of the network infrastructure of carriers and/or ISPs.

These practical recommendations are based on the experience of the editors and collaborators and on previous existing work, like existing best common practices [1].

https://www.ripe.net/publications/docs/ripe-431

Scenario 2 Anti-spoofing

Creating filters based on prefix lists:

```
IOS-XR:
Under interface configuration:
RP/0/0/CPU0:R5(config-if)#ipv4 verify unicast source reachable-via ?
  any  Source is reachable via any interface
  rx   Source is reachable via interface on which packet was received

IOS-XE:
Under interface configuration
ip verify unicast source reachable-via {rx | any} [allow-default] [allow-self-ping] [lis
Where list is a list of ACLs.
```

Implementing source address validation using access lists:

```
IOS-XE provides for a list of ACLs in the ip verify unicast command. Both IOS-XE and IOS
```

Applicability:

```
Anti-spoofing is implemented as unicast reverse path filtering. See
https://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k-r6-6/ip-addresses/c
```

https://www.manrs.org/participant/93/

More questions

- How to expand the AS-SET from a customer?
- How to migrate to hierarchical AS-SET names?
- Strict and loose filtering and where is the threshold?
- Which IRR to use?

Proposal

- A series of concise BCOPs, each addressing a specific aspect
  - Or a more consolidated one, e.g. integrating filtering and anti-spoofing?
- Broad community review and good visibility
- Use cases with concrete recommendations/instructions
  - “That is an example of how you can do that”
- Easy to manage and update
- Easy to reference
- MANRS NetOps Actions could provide a structure