Cleaning up the RIPE-NONAUTH dataset
Policy Proposal 2018-06

Erik Bais (A2B Internet)
Martin Levy (Cloudflare)
Job Snijders (NTT Communications)
Current situation: post-NWI-5 world

Fantastic work – a large loophole is closed
No new out-of-region objects can be created in the RIPE IRR

https://www.ripe.net/manage-ips-and-asns/db/impact-analysis-for-nwi-5-implementation
Current situation: post-NWI-5 world

• The NWI-5 project split the RIPE IRR into two datasets
  • **RIPE** (exclusively contains data that was created with the consent of the resource holder)
  • **RIPE-NONAUTH** (contains data for which we can’t know if consent was given, pile of garbage)
• We (as community) purposefully left cleaning up **RIPE-NONAUTH** as out of scope for NWI-5 to increase the chances of NWI-5’s successful execution
How do we clean up RIPE-NONAUTH?

• We can leverage a different data source to scrub the RIPE-NONAUTH dataset: RPKI

• **RPKI ROAs** as published by the five RIRs are always created with the full consent of the resource owner

• Data in **RIPE-NONAUTH** is unvalidated the resource owner may not even be aware the objects exist
Proposal: Let RPKI “drown out” conflicting IRR

• RPKI can be used for *BGP Origin Validation* – but also for other things!
• What about applying the RFC 6811 “Origin Validation procedure” to IRR data?
• Treat IRR data objects as if they are BGP announcements?
Example:

route: 129.250.15.0/24
origin: AS60068
descr: AS60068 route object
descr: this is a test of hijack possibilities with current state of RIPE/RADB security setup - this records covers IP address used for rr.ntt.net service
descr: please note this is just a demonstrative object, with no real harmful intention
mnt-by: DATACAMP-MNT
created: 2018-02-10T16:57:07Z
last-modified: 2018-09-04T19:07:32Z
source: RIPE-NONAUTH
hanna:~ job$ whois -h whois.bgpmon.net 129.250.15.0/24
% This is the BGPmon.net whois Service
% You can use this whois gateway to retrieve information
% about an IP address or prefix
% We support both IPv4 and IPv6 address.
%
% For more information visit:
% https://portal.bgpmon.net/bgpmonapi.php

Prefix: 129.250.0.0/16
Prefix description: NTT Communications backbone
Country code: US
Origin AS: 2914
Origin AS Name: NTT America, Inc.
RPKI status: ROA validation successful
First seen: 2011-10-19
Last seen: 2018-10-14
Seen by #peers: 87
Understanding what transpired

• If a network deploys RPKI based BGP Origin Validation with a “invalid == reject” routing policy
• an announcement where **129.250.15.0/24** is originated by AS60068 would be rejected
  • Because 129.250.15.0/24 conflicts with the RPKI ROA
• The IRR object describes a state of the network which cannot exist – it is in conflict with the published routing intentions of NTT
• Everyone generating a BGP prefix list filter for AS 60068 now has a hole punched for 129.250.15.0/24
• NTT has no method to delete the 129.250.15.0/24AS60068 object!
Process

1. A RIPE NCC script fetches all RPKI ROAs
2. If a ROA covers (part of a) route object in RIPE-NONAUTH, check if any of the ROA origin ASNs matches with the origin ASN listed in RIPE-NONAUTH
   - If yes: don’t delete – don’t do anything
   - If no ROA: don’t delete – don’t do anything
   - If invalid: delete the RIPE-NONAUTH IRR route object

No need to integrate this in the WHOIS software, can be separate script that runs every few minutes.
result = NOT_DELETE;

// Iterate through all the Covering entries in the local VRP database, pfx_validate_table.
entry = next_lookup_result(pfx_validate_table, route_prefix);

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

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

// If one or more VRP entries Covered the route prefix, but none Matched, return "Invalid" validation state.
if (prefix_exists == TRUE) {
    result = DELETE_IRR_OBJECT;
}

return (result);
Other industry developments

• **Use RPKI ROAs for provisioning BGP prefix-filters**

• Extending IRRd so that when IRR information is in direct conflict with a RPKI ROA – the conflicting information is suppressed ([Github](https://github.com/))
  • whois.radb.net
  • rr.ntt.net
  • ... others?

• Come to Open Source working group for more news about IRRd v4!
RPKI suppressing conflicting IRR advantages

• Industry-wide common method to get rid of stale proxy route objects – by creating a ROA you hide old garbage in IRRs

• By creating a ROA – you will significantly decrease the chances of people being able to use IRR to hijack your resource
Questions / Comments?

• PDP process takes place in Routing WG