Recent IRR changes

Matsuzaki ‘maz’ Yoshinobu
<maz@iij.ad.jp>

Problem statements

- Object Name Collisions in IRRs
  - IRRs do not enforce global uniqueness of object names. The same object name can exist in different IRRs with conflicting data, leading to inconsistencies and potential routing issues.
- Unrestricted Object Creation in Public IRRs (e.g., RADB)
  - Public IRRs like RADB allow anyone to create route/route6 objects without strict validation. This opens the door to accidental or malicious route hijacks and undermines trust in the routing registry system.

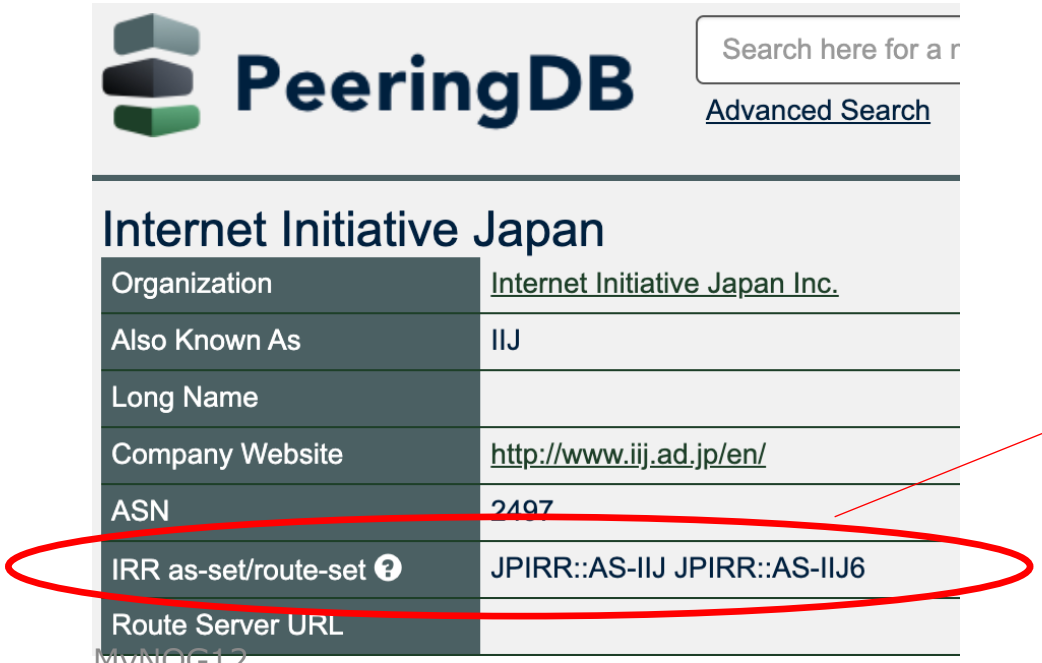
Internet Routing Registry (IRR)

- To publish your own routing intentions
  - route/route6
    - Specify the origin AS for an IP prefix
  - as-set
    - Specify a list of ASes to be advertised
- To construct route filters based on the registered objects

- APNIC IRR and RADB are popular IRR in AP region
  - Some NIRs are also operating its IRR such as JPIRR by JPNIC

IRR as-set

- Publishing your advertisement in advance
- Important technical information exchanged when peering with peers and upstreams




```
as-set:          AS-IIJ
descr:           ASes routed by IIJ
members:         AS-IIJ-T1, AS2497,
                 :
mnt-by:          MAINT-AS2497
source:          JPIRR
```

Automation Tool Example (bgpq4)

- The **bgpq4** utility is used to generate configurations (prefix-lists, extended access-lists, policy-statement terms and as-path lists) based on IRR data
  - https://github.com/bgp/bgpq4

```
$ bgpq4 AS-4608
no ip prefix-list NN
ip prefix-list NN permit 103.0.0.0/16
ip prefix-list NN permit 103.138.210.0/24
ip prefix-list NN permit 103.246.136.0/22
 :
```

```
$ bgpq4 -6 AS-4608
no ipv6 prefix-list NN
ipv6 prefix-list NN permit 2001:dc::/35
ipv6 prefix-list NN permit 2001:dc0::/35
ipv6 prefix-list NN permit 2001:dc0:2000::/35
 :
```

Uniqueness of Object Names and Common Issues in IRRs

- Object names are not globally unique
  - Each IRR has its own separate namespace.
  - The same object name may exist in different IRRs.
- Data mirroring between IRRs
  - Multiple IRRs mirror each other’s databases.
  - As a result, a response from a different IRR may contain an object with the same name but different data.
- Common issue: as-set name collisions
  - A frequent problem is the collision of as-set names, where objects with the same name but different content coexist across IRRs.

as-set name collision example

- AS-AMAZON
  - By amazon.com at RADB
  - By someone else at RIPE DB

```
as-set:      AS-AMAZON
descr:       Amazon ASNs
members:     AS-AMAZON-NA, AS-AMAZON-AP,
             AS-AMAZON-EU
admin-c:     AC6-ORG-ARIN
tech-c:      AC6-ORG-ARIN
   :
mnt-by:      MAINT-AS16509
source:      RADB
```

```
as-set:          AS-AMAZON
tech-c:          DUMY-RIPE
admin-c:         DUMY-RIPE
   :
mnt-by:          KATERINA-MNT
source:          RIPE
```

Name collisions might cause trouble

```
$ bgpq4 RADB::AS-AMAZON
no ip prefix-list NN
ip prefix-list NN permit 1.44.96.0/24
ip prefix-list NN permit 1.178.0.0/24
ip prefix-list NN permit 1.178.1.0/24
ip prefix-list NN permit 1.178.4.0/22
ip prefix-list NN permit 1.178.8.0/22
ip prefix-list NN permit 1.178.12.0/22
ip prefix-list NN permit 1.178.16.0/20
ip prefix-list NN permit 1.178.64.0/23
ip prefix-list NN permit 1.178.68.0/22
ip prefix-list NN permit 1.178.72.0/21
 :
(Total 23840 lines)
```

```
$ bgpq4 RIPE::AS-AMAZON
no ip prefix-list NN
! generated prefix-list NN is empty
ip prefix-list NN deny 0.0.0.0/0
```

APNIC IRR and Hierarchical as-set

- prop-151: Restricting non-hierarchical as-set
  - Reached consensus at APNIC 55

- Only hierarchical as-set names (e.g., AS65000:AS-EXAMPLE) can be newly created under this policy
  - Existing non-hierarchical as-sets can still be updated and used as before though

Hierarchical as-set (RFC2622)

- Non-hierarchical as-set example
  - AS-APNIC (This is not an APNIC's as-set)
  - AS-4608 (APNIC's as-set, not in Hierarchical naming scheme)

- Hierarchical as-set example
  - AS24514:AS-MYREN
  - AS16509:AS-AMAZON

- The notation is <AS#>:AS-<as-set name>
  - Only that AS number's maintainer can create the object
  - Resolving Name Collision Issues

Now you can use safer as-set

```
$ bgpq4 AS16509:AS-AMAZON
no ip prefix-list NN
ip prefix-list NN permit 1.44.96.0/24
ip prefix-list NN permit 1.178.0.0/24
ip prefix-list NN permit 1.178.1.0/24
ip prefix-list NN permit 1.178.4.0/22
ip prefix-list NN permit 1.178.8.0/22
ip prefix-list NN permit 1.178.12.0/22
ip prefix-list NN permit 1.178.16.0/20
ip prefix-list NN permit 1.178.64.0/23
ip prefix-list NN permit 1.178.68.0/22
ip prefix-list NN permit 1.178.72.0/21
 :
```

```
$ bgpq4 -6 AS16509:AS-AMAZON
no ipv6 prefix-list NN
ipv6 prefix-list NN permit 2001:300:fffb::/48
ipv6 prefix-list NN permit 2001:300:fffc::/48
ipv6 prefix-list NN permit 2001:300:fffd::/48
ipv6 prefix-list NN permit 2001:470:3b4::/48
ipv6 prefix-list NN permit 2001:470:426::/48
ipv6 prefix-list NN permit 2001:4f8:2::/48
ipv6 prefix-list NN permit 2001:4f8:b::/48
ipv6 prefix-list NN permit 2001:4f8:11::/48
ipv6 prefix-list NN permit 2001:559:0:2::/64
ipv6 prefix-list NN permit 2001:559:0:3::/64
 :
```

Migration to Hierarchical as-set is Recommended

- To align with best practices and ensure long-term stability, it is recommended to gradually migrate existing *as-sets* to the hierarchical format
- This transition reduces future risks of policy conflicts and improves clarity in routing policies.

Safe Transition Plan

- Step 1: Create a new Hierarchical as-set
  - Register a new as-set in Hierarchical naming scheme
    - Let’s say it’s AS2497:AS-IIJ
  - This has the same members as the existing as-set (AS-IIJ)
- Step 2: Make the old as-set a reference to the new one
  - Modify the member of the existing as-set as AS2497:AS-IIJ
  - Thereafter, only AS2497:AS-IIJ needs to be updated
- Step 3: Notify peers
  - Update the peeringdb information
  - Ask peers and upstreams to refer to the new as-set
- Step 4: Delete the old as-set
  - Delete the old non-hierarchical as-set after this transition completes

Our Current AS-SET (Non-hierarchical as-set)

```
as-set:     AS-IIJ
descr:      ASes routed by IIJ
members:    AS-IIJ-T1, AS2497,
              :
mnt-by:     MAINT-AS2497
source:     JPIRR
```

New AS-SET (Hierarchical as-set)

```
as-set:    AS2497:AS-IIJ
descr:     ASes routed by IIJ
members:   AS2497:AS-IIJ-T1,
           AS2497,
             :
mnt-by:    MAINT-AS2497
source:    JPIRR
```

For backward compatibility (Non-hierarchical)
To be deleted after migration

```
as-set:    AS-IIJ
descr:     Please refer AS2497:AS-IIJ
members:   AS2497:AS-IIJ
mnt-by:    MAINT-AS2497
source:    JPIRR
```

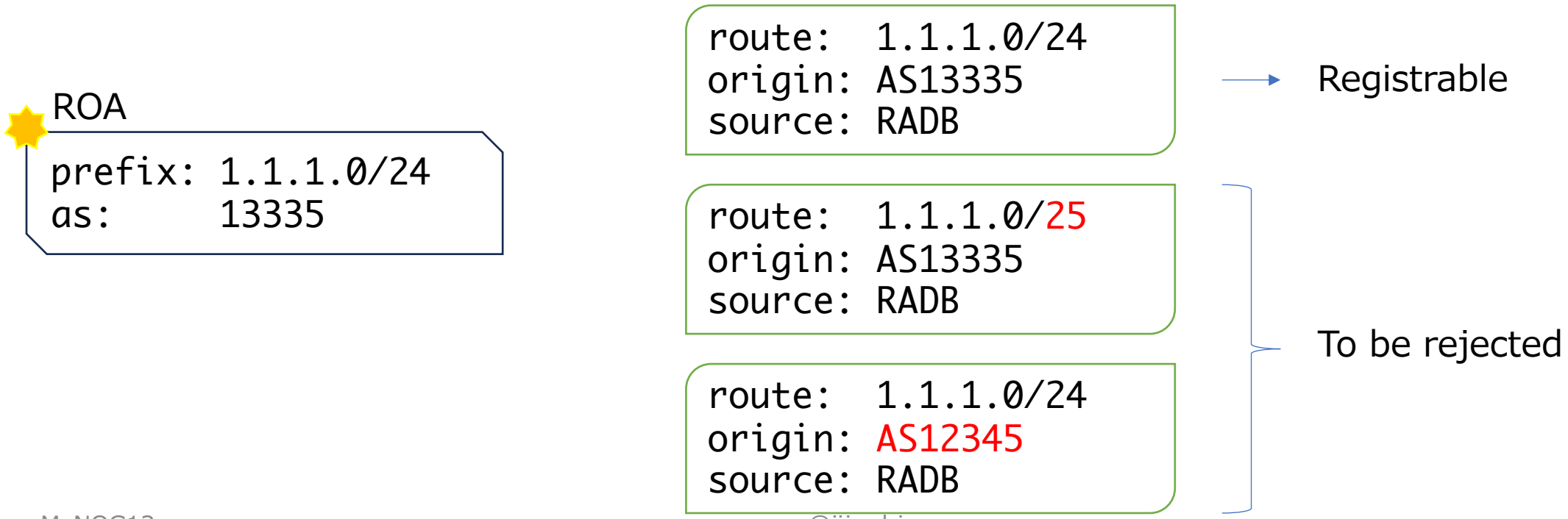
RADB and RPKI

- RADB migrated to IRRDv4 on November 13th, 2023
  - New features related to RPKI have been implemented

- route/route6 objects falling under RPKI Invalid
  - RPKI Invalid objects will no longer visible in a query
  - Objects that have not_found or valid RPKI will not be effected
  - Any new objects that are Invalid will be rejected and any modification of an existing Invalid object will be rejected as well

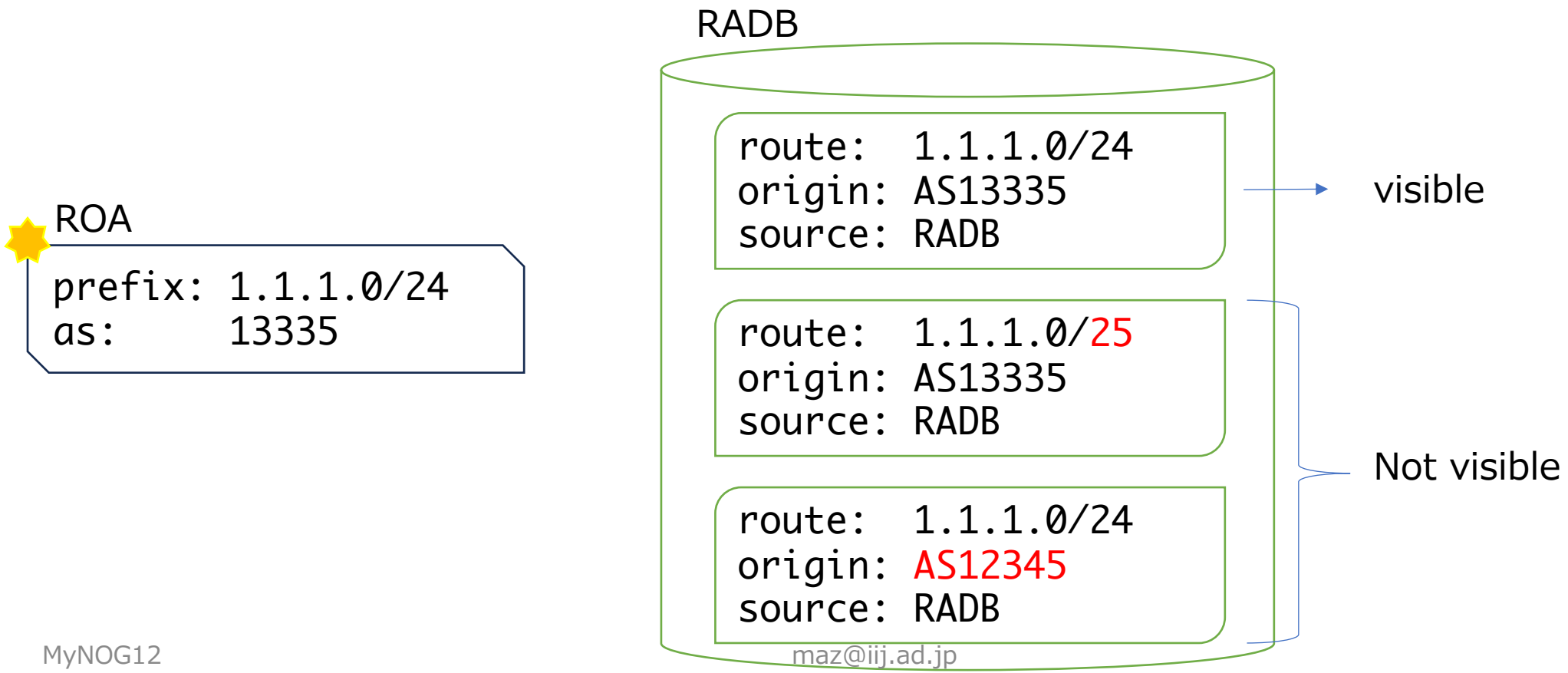
ROA and RPKI ROV Invalid

- A route object registration that is inconsistent with the corresponding ROA will be rejected by RADB



ROA and RPKI ROV Invalid

- RPKI Invalid objects will no longer visible in a query



Creating a minimal ROA

- Assume we have the following ROA and route object

ROA

```
prefix: 2001:db8::/32
as:     64512
```

```
route:  2001:db8::/32
origin: AS64512
source: RADB
```

- If for some reason I want to create a route object with a sub prefix (/34), how should I modify/issue ROA?

idea 1: max-length

ROA

```
prefix:     2001:db8::/32
as:         64512
max-length:34
```

idea 2: additional ROA

ROA

```
prefix: 2001:db8::/32
as:     64512
```

ROA

```
prefix: 2001:db8::/34
as:     64512
```

Route Hijack Problem – Real Example

- Background
  - A malicious actor somehow created a fake route object in RADB without authorization (could be a proxy registered)

```
route:      xxx.xxx.96.0/19
origin:     AS214860
descr:      Customer Prefix
```

  - This allowed the hijacker to announce the prefix and achieve global reachability
- Impact
  - Despite being unauthorized, the route appeared valid to many networks relying solely on IRR data

The Response – Creating a ROA

- The legitimate resource holder issued an AS0 ROA
  - They were not announcing the prefix to the Internet

```
Prefix:     xxx.xxx.0.0/16
Max Length: 16
AS:         0
```

- RPKI ROV (Route Origin Validation)
  - The hijacked route (origin AS214860) became invalid and was filtered by ROV-enforcing networks
- IRR Impact
  - The invalid ROA effectively neutralized the unauthorized RADB route object, making it invisible to IRR-based filtering system

Why ROA is a Powerful Defense Tool

- Immediate Mitigation
  - Once the ROA is published, hijacked routes become invalid nearly instantly in RPKI ROV-enabled networks
- IRR Alignment
  - RADB stops serving route objects that fail RPKI validation, reducing visibility of unauthorized objects
- Lessons Learned
  - Relying only on IRR data (e.g., RADB) is risky due to lack of strict authentication
  - Combining IRR + RPKI ensures stronger, multi-layered protection

Summary

- APNIC account holders are only permitted to create hierarchical as-sets at APNIC IRR
- For transition, Safe Transition Plan (P.12) is available
- RADB is rejecting RPKI ROA Invalid
  - This protects you because no one can register a route object that is inconsistent with the ROA!
- Keep your ROA minimal as possible to protect your network