COMCAST

RPKI ROV

One journey

Tony Tauber

Comcast

2022

Agenda

- ROV Background ←
- Validating
- Publishing

BGP Security Risks

- Misconfiguration
- Malicious Actors
- Traffic doesn’t go to the right place
  - and maybe goes to the wrong place

RPKI Route Origin Validation (ROV)

- **What are these acronyms**
  - RPKI = Resource Public Key Infrastructure
    - The system
  - ROA = Route Origin Authorization
    - The main item of interest
  - ROV – Route Origin Validation
    - How it gets used – the process it enables
- **What does it do?**
  - Provides a method for the ”owner” (registered user) of a prefix to assert which ASN(s) are the correct originator(s) for that prefix
  - Asserts (implicitly) that other originators are not valid

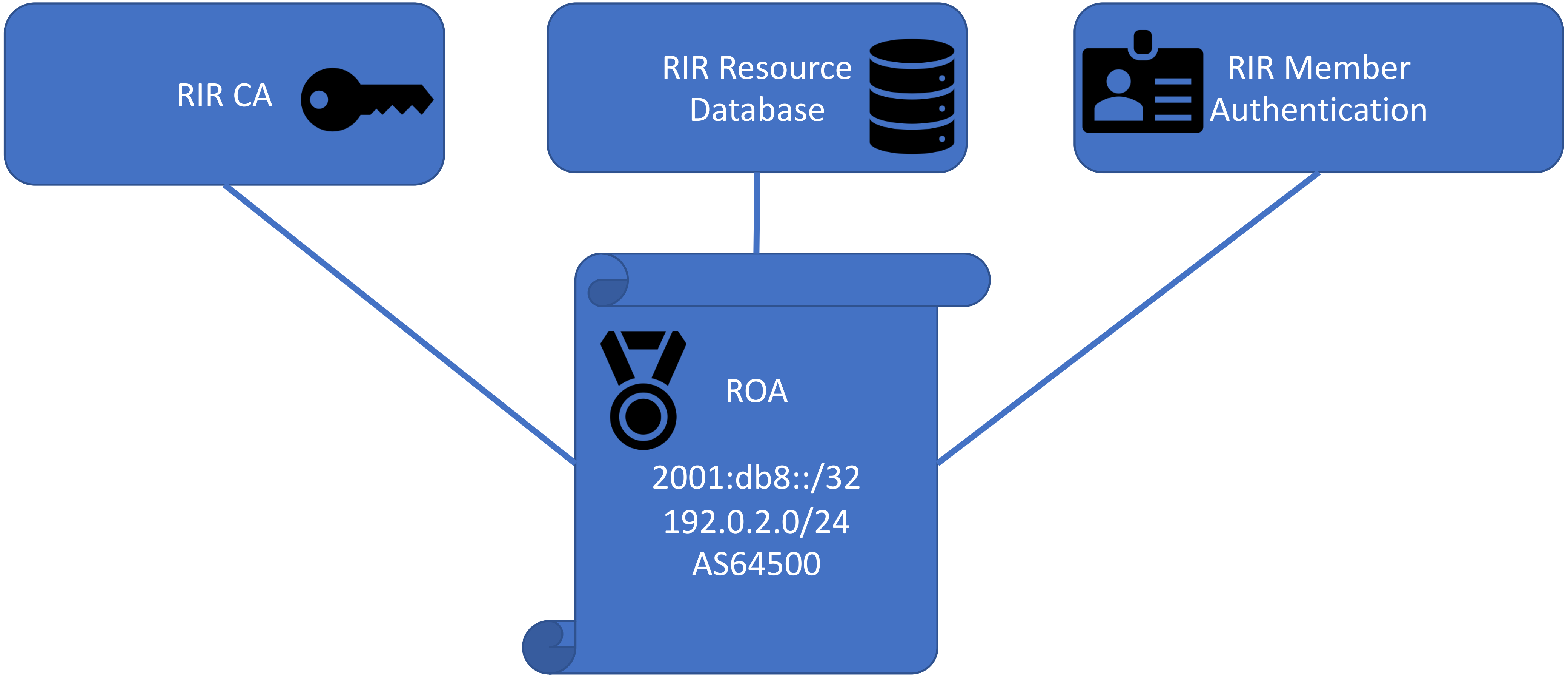
RPKI Route Origin Validation (ROV)

- **What is in a ROA?**
  - A signed statement consisting of:
    - **prefix**
    - **maximum prefix length**
    - **originating ASN**
- **RPKI also has other types of objects to make it work**

RPKI Route Origin Validation (ROV)

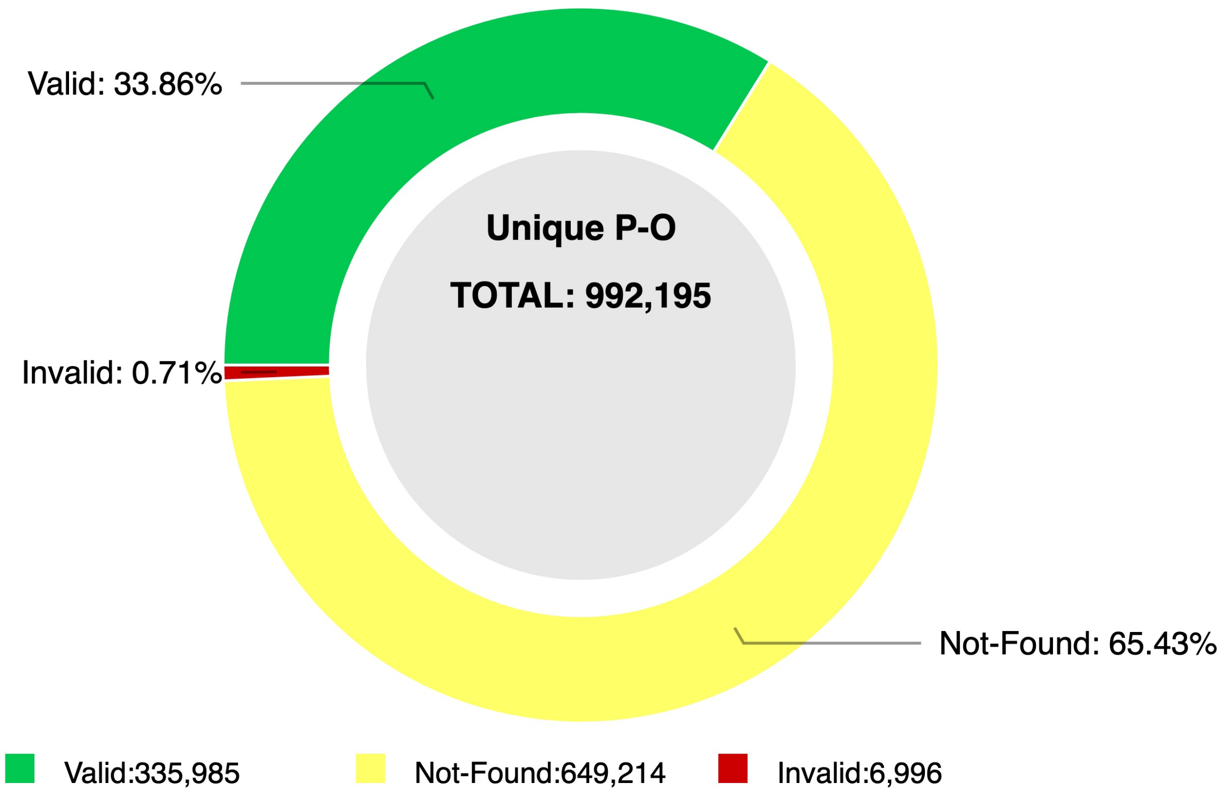
- **How does it work?**
  - The “root” assigner of all IP space (v4+v6) is IANA
  - Delegated to 5 RIRs (Regional Internet Registries)
    - ARIN, RIPE NCC, APNIC, LACNIC, AFRINIC
  - They assign further to
    - LIRs (Local Internet Registries)
    - Service Providers
    - Enterprises
  - RIR portals for address holders to generate ROAs
  - ROAs are published out by the RIR so that anyone can view them

ROAs


RIR CA
RIR Resource Database
RIR Member Authentication
ROA
2001:db8::/32
192.0.2.0/24
AS64500

Global ROV coverage

RPKI-ROV Analysis of Unique Prefix-Origin Pairs (IPv4)



NIST RPKI Monitor: RPKI-ROV Analysis **Protocol:** IPv4 **RIR:** All **Date:** 2022-01-22 06:00

Courtesy: NIST (National Institute of Standards and Technologies
https://rpki-monitor.antd.nist.gov/ROV

Considerations

- Fail-open model
  - Given that most of the prefixes are still not covered (i.e., “not-found”)
  - Hence absence of a covering ROA will still allow for route propagation
  - Same for complete loss of RTR connections/data at router level (more later)
- Already some large ISPs doing ROV
  - Hence invalid announcements are already getting dropped

Components

- Repositories
    - ROAs are published on servers operated by RIRs and their delegates
- Validating Caches (VC) running Relying Party (RP) software
    - Servers running validator software which fetches ROAs and other data using
        - rsync – TCP protocol for synchronizing files on servers (TCP port 873)
        - RRDP – RPKI Repository Delta Protocol which uses HTTPS as transport (newer, preferred)
    - Run cryptographic integrity checks to produce VRPs (Validated ROA Payload)
    - RPKI-to-Router (RPKI-RTR) protocol (TCP port 323 or 8323)
        - Allows for fetching VRP data by routers
        - Routers cache the data locally and refresh at intervals
            - Retain local cached data for a configurable time in case connection to cache is lost

Motivation

- Low barrier of entry
    - No new gear (features on existing routers)
    - Some VMs running freely available open-source software
- Risk of doing nothing
    - Vulnerability of mis-origination by others
- Risks of doing something
    - Collateral damage, increased complexity, new troubleshooting
- Management buy-in
    - Can’t make the case on my own just in case something goes wrong
        - “Whose idea was this anyway?”
    - Different recent incidents in the trade press helped the case

Reading (Validating) / Writing (Publishing)

- Can do one without doing the other, not necessary to do together
- Need to work as incrementally as possible
  - Can’t do things globally
  - Always have a backout plan of each (sub-)step
- Publishing
  - Hosted model: RIR publishes the data that members enter in the portal
    - e.g., ARIN Online
  - Delegated model: RIR delegates to LIR (Local Internet Registry)
    - Run own CA (Certificate Authority) and PP (Publication Point) servers
- Validating… (covered in later slides)

RPKI ROV High Level Plan

- Reading – Route Origin Validation using published ROAs
  - Add inbound route-policy to ”drop invalid” after dropping bogons
  - Field trial with subset of interconnection partners in August 2020
  - Broader rollout through remainder of 2020 and early 2021
- Writing – Publishing ROAs for our own address space
  - Start with one or small number of prefixes
  - Gradually expand

Environment

- Validation – Cisco/Juniper edge routers
    - Incremental rollout
- Publication – ROA generation
    - 100 + prefixes
    - Two dozen internal ASNs
    - Thousands of more-specifics

Agenda

- ROV Background
- Validating ←
- Publishing

ROV – Route Origin Validation

- Easier to do with small risk
  - Luckily, it “fails open” – in absence of a ROA, BGP route is accepted
- Only external eBGP sessions
  - Not on sessions among our different regional ASes for instance
  - No iBGP (doesn’t even make sense)
  - Key reason: we carry many more-specifics internally
- Config per router, per neighbor
  - Easier to see if something goes wrong and back out if necessary
- Pairwise coordination with all partners is not the goal, notification is

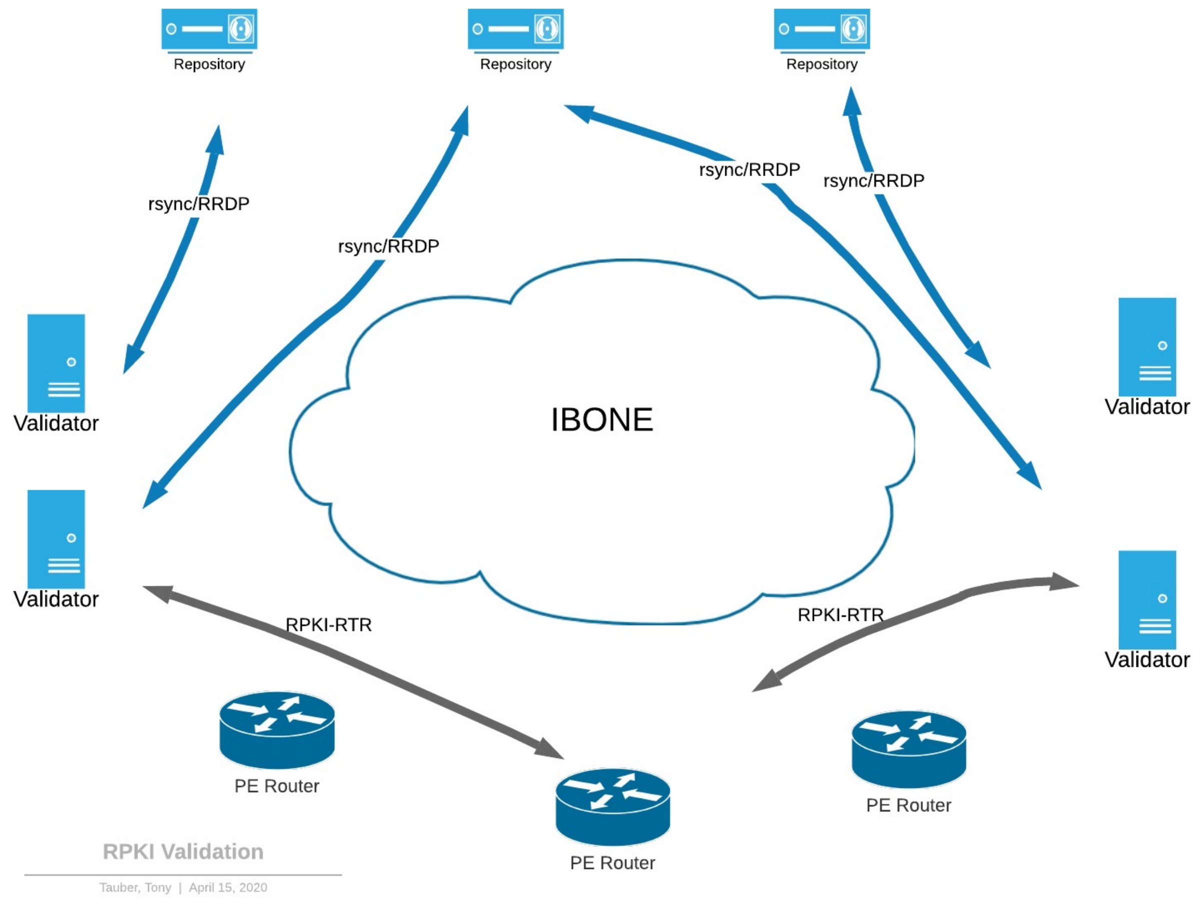
Validating Cache Relying Party Software

- Geographic diversity
  - Deploy to two different data centers in case one has an outage
- Software diversity
  - Deploy two different codebases in case one has a problem
- Several freely available open-source options

Validating Cache Relying Party Software

- Initial choices
  - RIPE RPKI Validator v.3 – RIPE NCC –written in Java language
    - Later replaced with rpki-client (with web wrapper) and StayRTR
  - Routinator – NLnet Labs – written in Rust programming language
- Hence, each router will have 4 different RTR servers configured
  - Deployed and managed by our DNS staff
- All have packages now, easy to install and keep updated
- Can produce metrics also for consumption

RPKI ROV infrastructure design



RPKI Validation

Tauber, Tony | April 15, 2020

ROV – Bugs?

- Cisco and Juniper both had some bugs
  - Made sure to patch to the recommended versions
- RP Software has had some bugs
  - Mostly bounds-checking and the like
  - Installed fixed packages as they were released

Agenda

- ROV Background
- Validating
- Publishing ←

Signing and Publishing ROAs – Hosted

- Via RIR portals
  - Varying degrees of ease and integration
  - For example, publish ROAs to match existing BGP announcements
- APIs available
  - ARIN API script – Rich Compton from Charter Communications
  - Not polished but wouldn’t be possible without it!

Signing and Publishing ROAs – Delegated

- Address-issuing authority delegates to you
  - RIRs in our case, could be more layers down
  - Issues a Certificate which is used to sign ROAs and other artifacts
  - Hosts a record with URL to Publication Point (PP)
- Certificate Authority (CA) and Publisher Software:
  - Krill – NLnet Labs
  - rpki.net – Dragon Labs
- Publication point (PP) needs to be globally reachable
- Info about running own RPKI CA
  - https://www.slideshare.net/apnic/should-i-run-my-own-rpki-certificate-authority

Decision – Hosted vs. Delegated

- Delegated
  - Extra servers and software to run
  - Availability profile a bit unknown
- Hosted
  - Less of these risks….
- Went with Hosted at this point
  - Share fate with thousands of others
  - Consider revisiting at a later date
  - Hybrid model (CA internal, PP hosted elsewhere) has some appeal

Publishing – Creating ROAs

- Larger risk
    - Can create connectivity issues if something goes unreachable
    - Can take time to back out or correct
        - ROA distribution is on order of minutes to hours
    - Make sure to do it carefully
- Our complexity
    - Something over 100 address blocks
        - Almost all ARIN, a few from other RIRs
    - Distributed unevenly across more than 20 different ASes
        - Backbone, Regional, Data Center, Enterprise

Publishing – Creating ROAs Process

- Issuing ROA for largest blocks makes ROAs underneath “invalid”
  - Unless there’s a matching ROA for the more-specific already
- Gradually roll out
  - Sign few non-intrusive prefixes
  - Start from “bottom” (more-specific prefixes)
  - Once all filled in, issue ROAs for top-level blocks
- Integrate with IP management software in a later phase
- Ended up publishing several thousand ROAs (mostly IPv6)
  - Fewer blocks but so much more to break apart

Thanks!

Tony Tauber
<firstname>_<lastname>@comcast.com

(Not needing more spam from robots who should solve this robot Wordle instead.)