INTERNET2

TechEXtra

KICKOFF

OCT 6-7
2020

RPKI ROV Service in the Cloud

Presenters:
Ryan Harden - Sr Cyberinfrastructure Security Architect
Steve Wallace - Cyberinfrastructure Security Architect

Agenda

- Big Picture, take away for this presentation
- Quick Background of RPKI/ROV
- Overview of Internet2's internal ROV Deployment
- Introduction of Internet2 Community ROV Service
- Internet2 RPKI Stats and Metrics
- Q&A / Discussion

Big Picture

- Internet2 is adding ROV (turning into production)
- We want to share our approach and lessons learned to make it easier for others to adopt ROV
- We intend to duplicate our approach in the cloud to host a backup service for those that implement a local ROV capability

What is RPKI and ROV?

- RPKI (Resource Public Key Infrastructure)
  - The mechanism for asserting a route/prefix may be originated from a specific (or list) of ASNs via a ROA (Route Origin Authorization)
  - ROAs are published to an authority like at RIR (ARIN for most of us)
- ROV (Route Origin Validation)
  - The process of validating a prefix received via BGP matches its published ROA
- ROV Server
  - A process that contains all ROAs across all RPKI authorities and keeps them updated
  - Required for configuring a router to do ROV.

Requirement to protect your networks from being hijacked...

- Before an ROA can be published for an network, the network must be covered by a recent ARIN L/RSA
- Over 50% of the Internet2 membership have legacy network resources that aren’t covered by an L/RSA agreement.

Helpful Links

- ARIN RPKI/ROA Guide
  - https://www.arin.net/resources/manage/rpki/roa_request/
- Wikipedia
  - https://en.wikipedia.org/wiki/Resource_Public_Key_Infrastructure
- Cloudflare RPKI Update
  - https://blog.cloudflare.com/rpki-2020-fall-update/
- RIPE RPKI Validator
  - https://rpki-validator.ripe.net/roas

Internet2 Internal ROV Deployment

CloudFlare’s OctoRPKI and GoRTR

- https://github.com/cloudflare/cfrpki
- Deployed late 2018
- Chosen due to limited options at the time
- Deployed in Internet2 Datacenter

Internet2 Internal ROV Deployment

NLnetLabs Routinator

- https://github.com/NLnetLabs/routinator
- Deployed May 2020
- Chosen due to ease of deployment (mostly)
- Running in AWS East Region

Cloud Deployment

Why the Cloud?

- Ease of deployment
- Reproducibility
- Richly connected

Cloud (Docker + AWS Elastic Container Service + EC2)

- Docker + AWS Elastic Container Service + EC2
- Docker
  - NLnetLabs provides a nice container
    - Almost Zero Effort...Almost…
    - Must build locally to accept ARIN's Legal Agreement (groan)

- AWS Elastic Container Service
  - Automated container service
    - Set up the service
    - Set up EC2 instances
    - Set Service Parameters
    - Deploy
  - Easy access inside AWS EC2

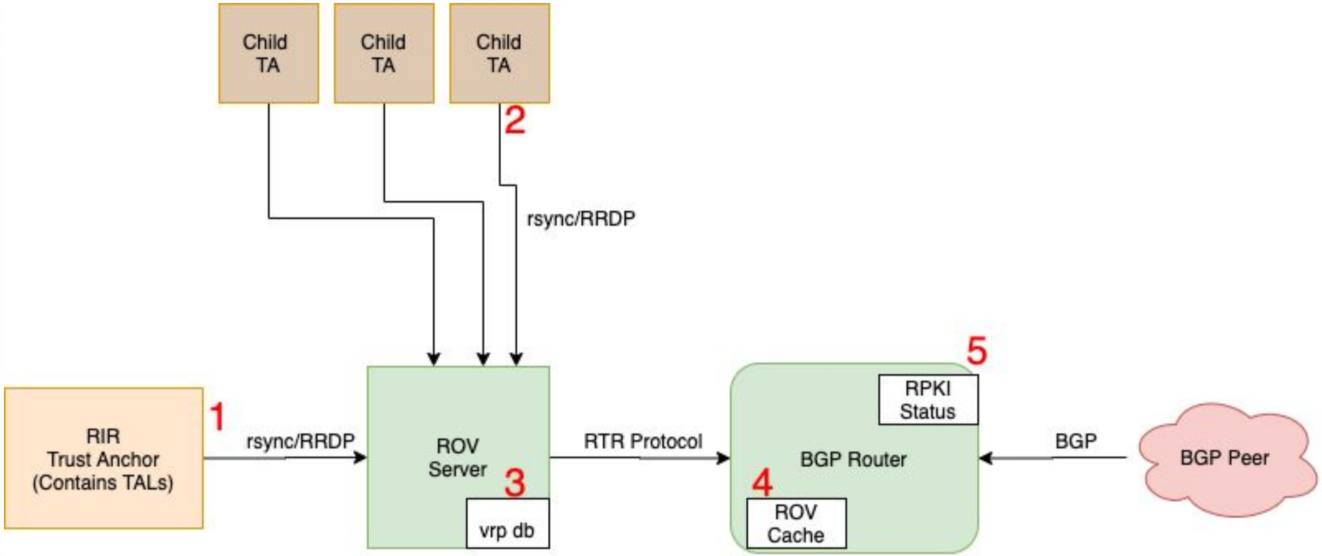
- AWS EC2
  - t3.medium (could probably be smaller)
  - No Config Necessary (due to ECS)

Cloud Deployment Details

Items or Improvements Under Consideration

- Multi-Region?
- ECS -> Fargate?
- Retire “local” service?

RPKI ROV Process Flow



1: ROV Server downloads Root TALs and ROAs from RIR TA

2: ROV Server downloads Children TA ROAs and pointers to their children, etc...

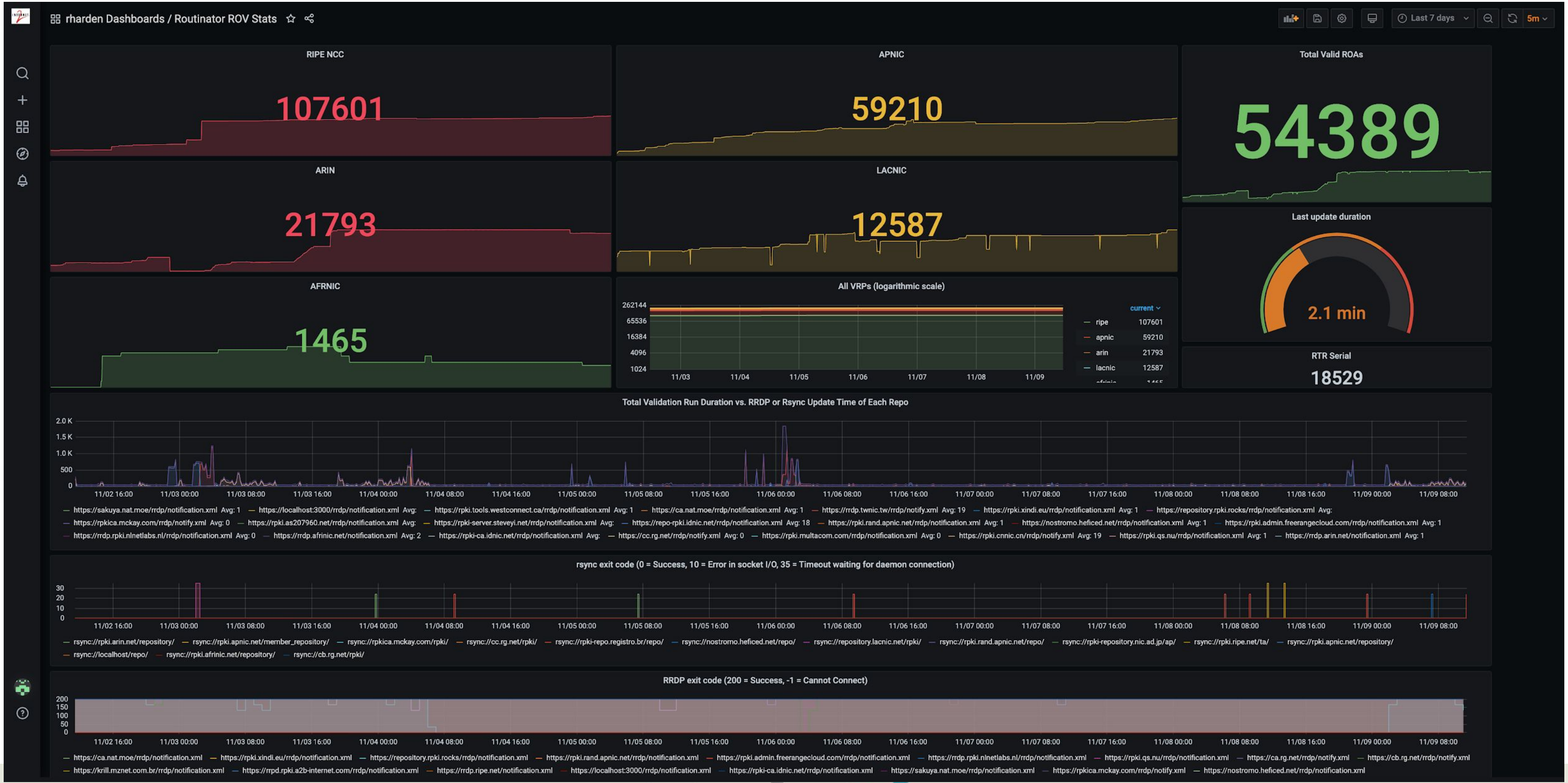
3: ROV Server verifies ROAs and creates VRPs (Validated ROA Payload)

4: BGP Router downloads VRPs from ROV server and inserts it into a local cache.

5: BGP Prefixes are compared with ROV Cache and assigned RPKI Status

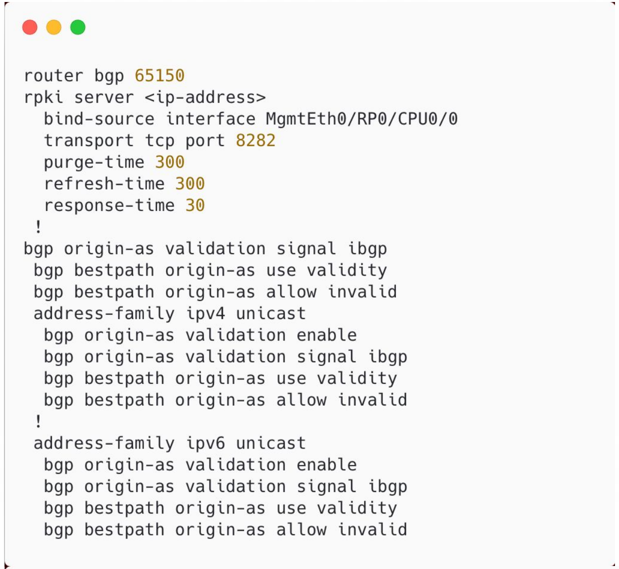
BGP actions based on RPKI Status are determined by your routing policy.

Grafana Dashboard (Routinator)

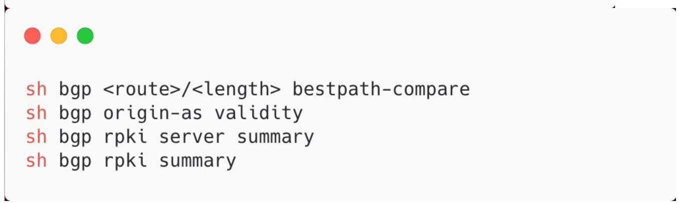

rharden Dashboards / Routinator ROV Stats
Last 7 days
5m
RIPE NCC
107601
APNIC
59210
Total Valid ROAs
54389
ARIN
21793
LACNIC
12587
Last update duration
2.1 min
AFRNIC
1465
All VRPs (logarithmic scale)
262144
65536
16384
4096
1024
11/03
11/04
11/05
11/06
11/07
11/08
11/09
current
ripe 107601
apnic 59210
arin 21793
lacnic 12587
RTR Serial
18529
Total Validation Run Duration vs. RRDP or Rsync Update Time of Each Repo
2.0 K
1.5 K
1.0 K
500
0
11/02 16:00
11/03 00:00
11/03 08:00
11/03 16:00
11/04 00:00
11/04 08:00
11/04 16:00
11/05 00:00
11/05 08:00
11/05 16:00
11/06 00:00
11/06 08:00
11/06 16:00
11/07 00:00
11/07 08:00
11/07 16:00
11/08 00:00
11/08 08:00
11/08 16:00
11/09 00:00
11/09 08:00
https://sakuya.nat.moe/rrdp/notification.xml Avg: 1
https://localhost:3000/rrdp/notification.xml Avg:
https://rpki.tools.westconnect.ca/rrdp/notification.xml Avg: 1
https://ca.nat.moe/rrdp/notification.xml Avg: 1
https://rrdp.twnic.tw/rrdp/notify.xml Avg: 19
https://rpki.xindi.eu/rrdp/notification.xml Avg: 1
https://repository.rpki.rocks/rrdp/notification.xml Avg:
https://rpkica.mckay.com/rrdp/notify.xml Avg: 0
https://rpki.as207960.net/rrdp/notification.xml Avg:
https://rpki-server.steveyi.net/rrdp/notification.xml Avg:
https://repo-rpki.idnic.net/rrdp/notification.xml Avg: 18
https://rpki.rand.apnic.net/rrdp/notification.xml Avg: 1
https://nostromo.heficed.net/rrdp/notification.xml Avg: 1
https://rpki.admin.freerangecloud.com/rrdp/notification.xml Avg: 1
https://rrdp.rpki.nlnetlabs.nl/rrdp/notification.xml Avg: 0
https://rrdp.afrinic.net/notification.xml Avg: 2
https://rpki-ca.idnic.net/rrdp/notification.xml Avg:
https://cc.rg.net/rrdp/notify.xml Avg: 0
https://rpki.multacom.com/rrdp/notification.xml Avg: 0
https://rpki.cnnic.cn/rrdp/notify.xml Avg: 19
https://rpki.qs.nu/rrdp/notification.xml Avg: 1
https://rrdp.arin.net/notification.xml Avg: 1
rsync exit code (0 = Success, 10 = Error in socket I/O, 35 = Timeout waiting for daemon connection)
30
20
10
0
rsync://rpki.arin.net/repository/
rsync://rpki.apnic.net/member_repository/
rsync://rpkica.mckay.com/rpki/
rsync://cc.rg.net/rpki/
rsync://rpki-repo.registro.br/repo/
rsync://nostromo.heficed.net/repo/
rsync://repository.lacnic.net/rpki/
rsync://rpki.rand.apnic.net/repo/
rsync://rpki-repository.nic.ad.jp/ap/
rsync://rpki.ripe.net/ta/
rsync://rpki.apnic.net/repository/
rsync://localhost/repo/
rsync://rpki.afrinic.net/repository/
rsync://cb.rg.net/rpki/
RRDP exit code (200 = Success, -1 = Cannot Connect)
200
150
100
50
0
https://ca.nat.moe/rrdp/notification.xml
https://rpki.xindi.eu/rrdp/notification.xml
https://repository.rpki.rocks/rrdp/notification.xml
https://rpki.rand.apnic.net/rrdp/notification.xml
https://rpki.admin.freerangecloud.com/rrdp/notification.xml
https://rrdp.rpki.nlnetlabs.nl/rrdp/notification.xml
https://rpki.qs.nu/rrdp/notification.xml
https://ca.rg.net/rrdp/notify.xml
https://cb.rg.net/rrdp/notify.xml
https://krill.mznet.com.br/rrdp/notification.xml
https://rrpd.rpki.a2b-internet.com/rrdp/notification.xml
https://rrdp.ripe.net/notification.xml
https://localhost:3000/rrdp/notification.xml
https://rpki-ca.idnic.net/rrdp/notification.xml
https://sakuya.nat.moe/rrdp/notification.xml
https://rpkica.mckay.com/rrdp/notify.xml
https://nostromo.heficed.net/rrdp/notification.xml

Sample Router Config

Cisco

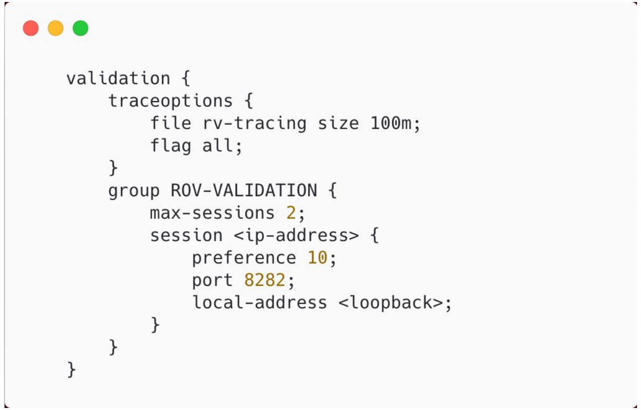


```
router bgp 65150
rpki server <ip-address>
  bind-source interface MgmtEth0/RP0/CPU0/0
  transport tcp port 8282
  purge-time 300
  refresh-time 300
  response-time 30
 !
bgp origin-as validation signal ibgp
 bgp bestpath origin-as use validity
 bgp bestpath origin-as allow invalid
 address-family ipv4 unicast
  bgp origin-as validation enable
  bgp origin-as validation signal ibgp
  bgp bestpath origin-as use validity
  bgp bestpath origin-as allow invalid
 !
 address-family ipv6 unicast
  bgp origin-as validation enable
  bgp origin-as validation signal ibgp
  bgp bestpath origin-as use validity
  bgp bestpath origin-as allow invalid
```

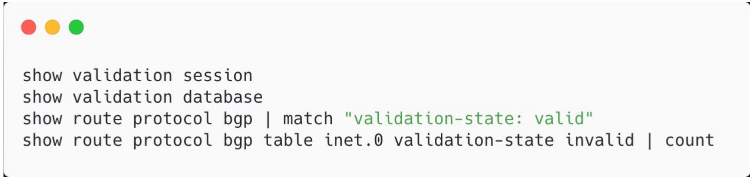


```
sh bgp <route>/<length> bestpath-compare
sh bgp origin-as validity
sh bgp rpki server summary
sh bgp rpki summary
```

Juniper

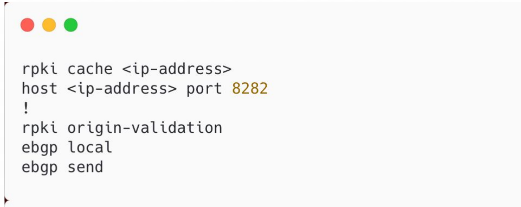


```
    validation {
        traceoptions {
            file rv-tracing size 100m;
            flag all;
        }
        group ROV-VALIDATION {
            max-sessions 2;
            session <ip-address> {
                preference 10;
                port 8282;
                local-address <loopback>;
            }
        }
    }
```

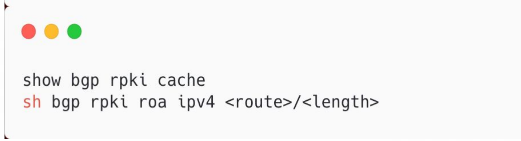


```
show validation session
show validation database
show route protocol bgp | match "validation-state: valid"
show route protocol bgp table inet.0 validation-state invalid | count
```

Arista



```
rpki cache <ip-address>
host <ip-address> port 8282
!
rpki origin-validation
ebgp local
ebgp send
```



```
show bgp rpki cache
sh bgp rpki roa ipv4 <route>/<length>
```

ROV In the Cloud Blog Post

Detailed Deployment Guide for Routinator in AWS

Stay Tuned!

Introduction: Internet2 Community ROV Service

- **Forward looking statements!** As you can imagine we're pretty busy implementing NGI…
- We hope to offer a best-effort validator service
- The service is intended to be backup validator server
- AWS Hosted
- Legal Mumbo-jumbo

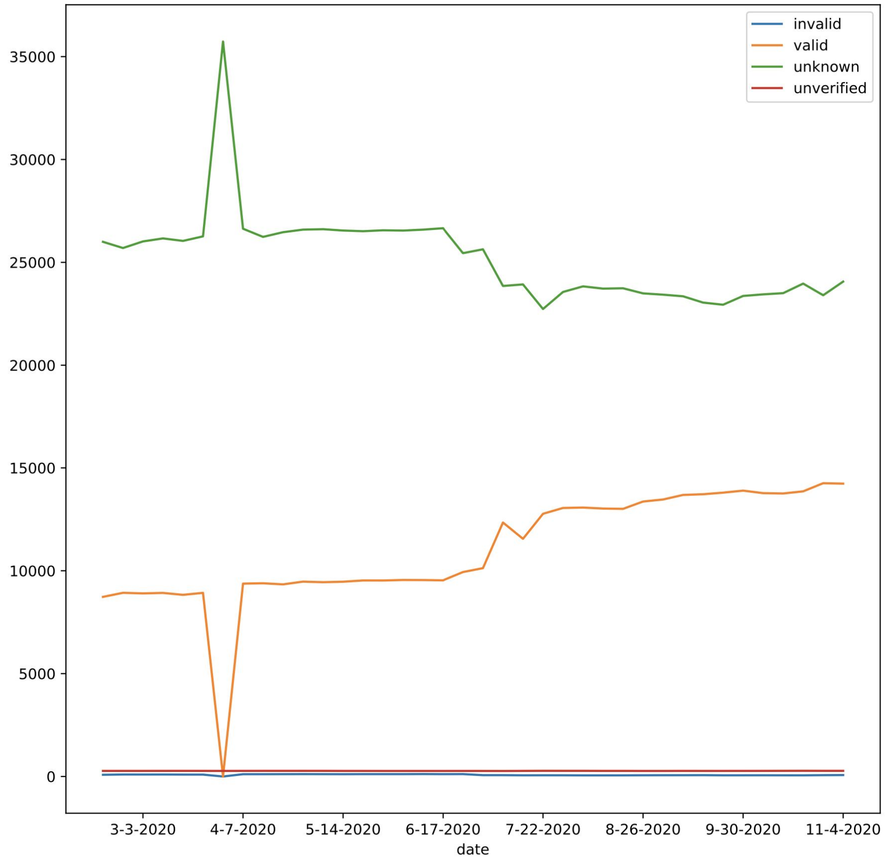
Legal Stuff (initially)

The service is intended to be offered at no additional fee, however it will require the execution of a “Network Service Schedule - RPKI Validator Service” agreement.

Internet2 is required to sign ARIN’s:

“RESOURCE CERTIFICATION **REDISTRIBUTOR** RELYING PARTY AGREEMENT” so a requirement of Internet2’s RPKI Validator Service agreement is that the organization must also sign ARIN’s RPA.

RPKI Status for the Internet2 R&E Table



Check on Internet2 R&E route status

Run the following command on the Internet2 router proxy to view Internet2’s view of RPKI route status for Internet participants’ routes in the R&E table

https://routerproxy.wash2.net.internet2.edu/routerproxy/

```
show route table inet.0 validation-state [status] community 11537:950 terse
```

where [status] is “invalid”, “unknown”, “unverified”, or “valid”.

Questions and Discussion