Google

Experience and observations in deploying Cloud scale routing security

NANOG 93

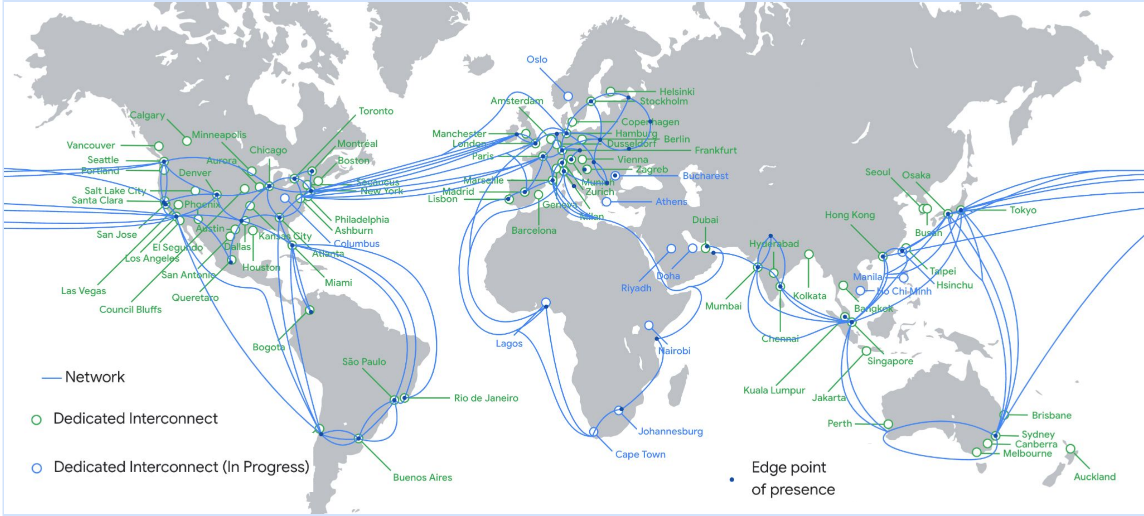
Anees Shaikh

Google Global Backbone

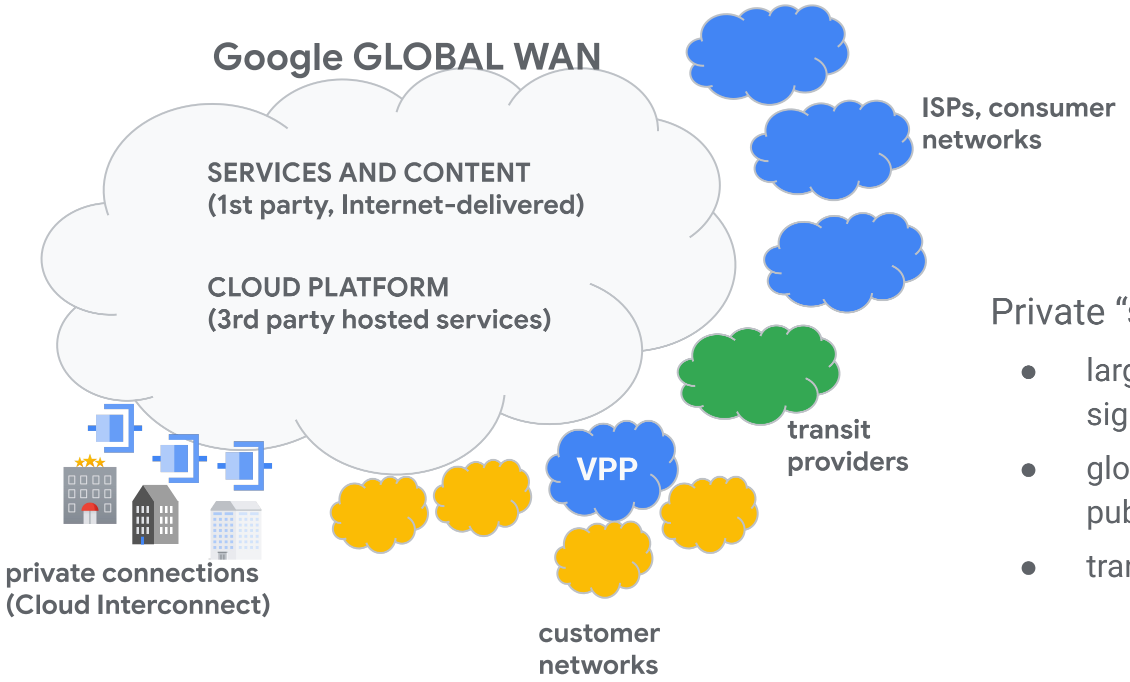
- **41** regions
- **124** zones
- **187** network edge locations
- **32** subsea cables
- **2 million miles** lit fiber
- **200+** countries & territories

100%

Carbon neutral since 2007



External connectivity from Google Cloud WAN



Private "stub" network with unique characteristics

- large peering surface – independent reachability to significant portion of the Internet
- global presence – peering in private facilities and public IXPs
- transit connections primarily for resilience

Protection for multiple types of routes

Protect traffic against multiple kinds of routing disruptions

- *routes to CSP services and content* (reachability from users)
- *routes to reach users of CSP services and content* (return path)
- *routes to third-party services / content* (needed by Cloud customers)
- *third-party owned routes announced from the CSP* (e.g., BYOIP)

Not all routes are equal

- traffic volume associated with routes
- type of traffic on routes
- routes associated with critical services or high-value customers

Key principles for reliable service deployment and operations

regionalization and replication

sharding strategies to avoid global blast radius of failures

progressive rollouts

gradual rollouts of code and configuration, including canaries, health checking, alerting, …

defense-in-depth

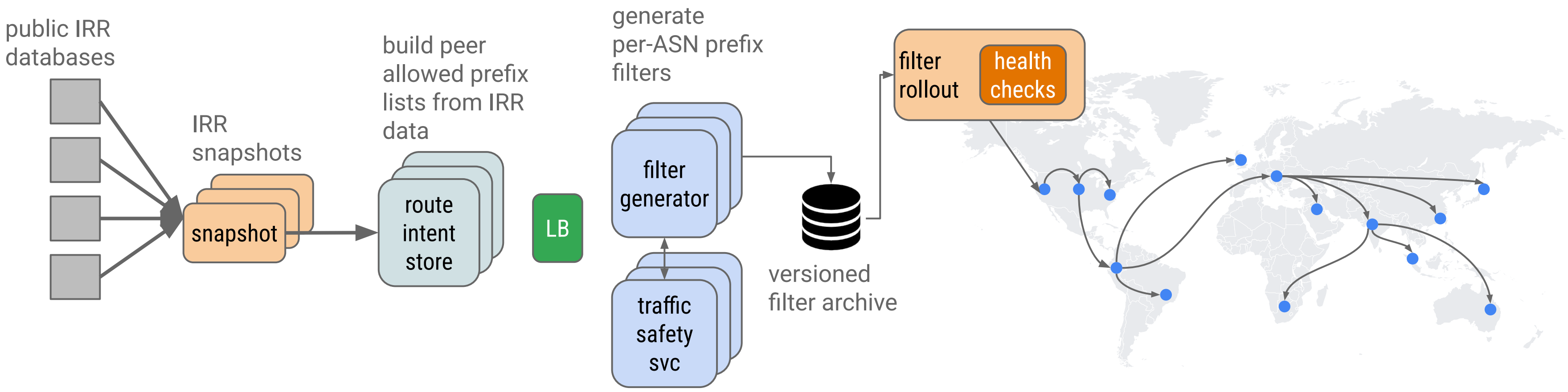
multiple layers of safety checks, including input validation, impact analysis, real-time checks, …

Routing security is implemented via distributed software services – leverage the best practices used for all Google software services

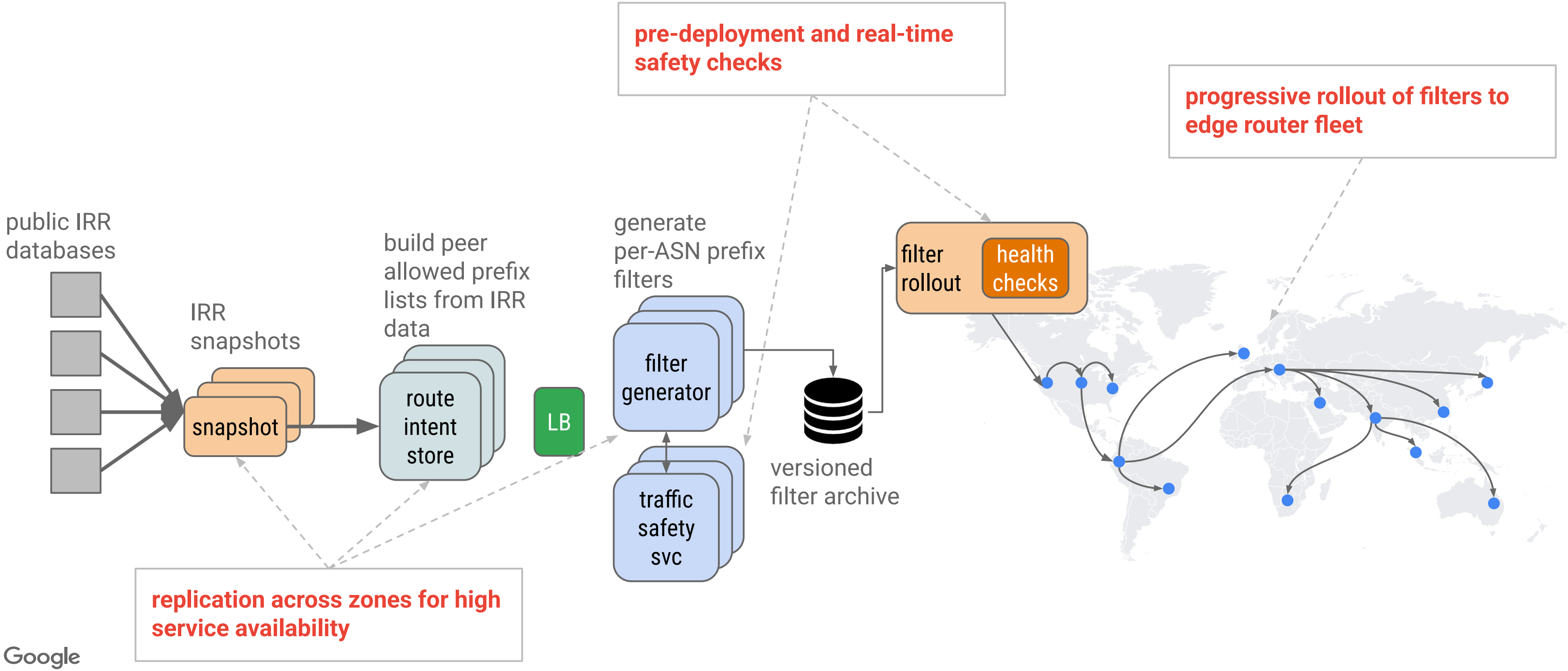
IRR filtering pipeline

Per-peer IRR filter lists generated and rolled out to the network fleet periodically

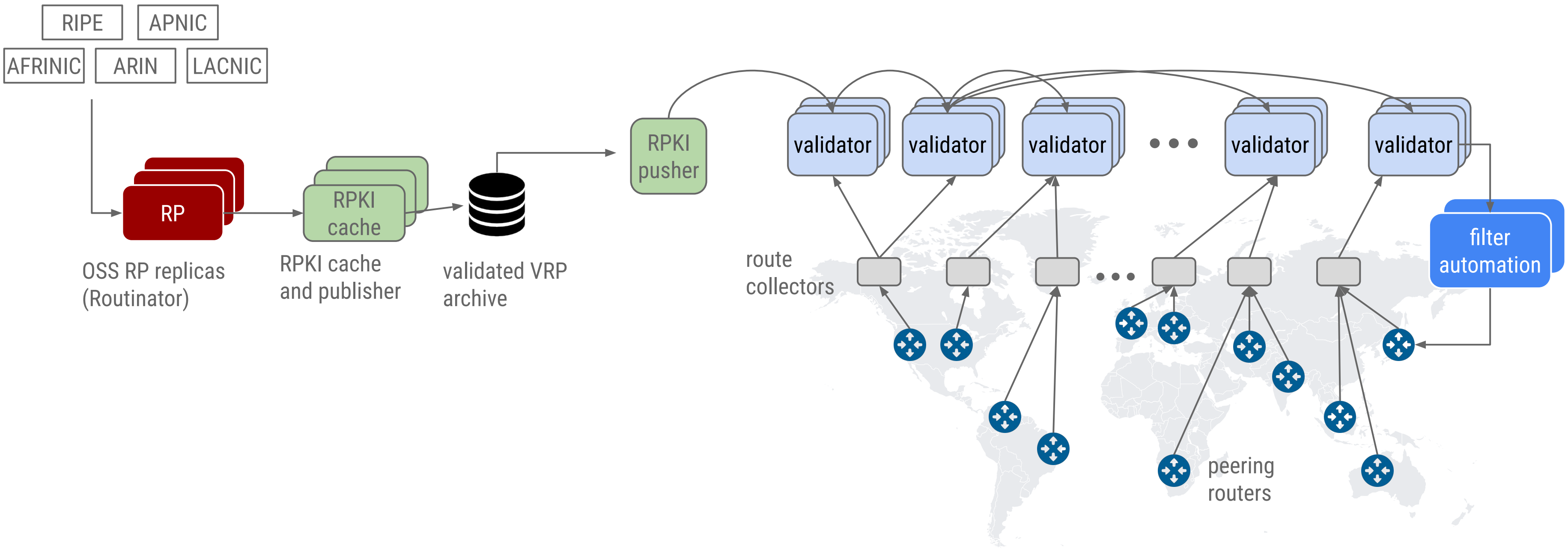
IRR filtering proceeded through stages: MARK → DEPREF → REJECT to allow data cleanup, infrastructure hardening, and safety check tuning



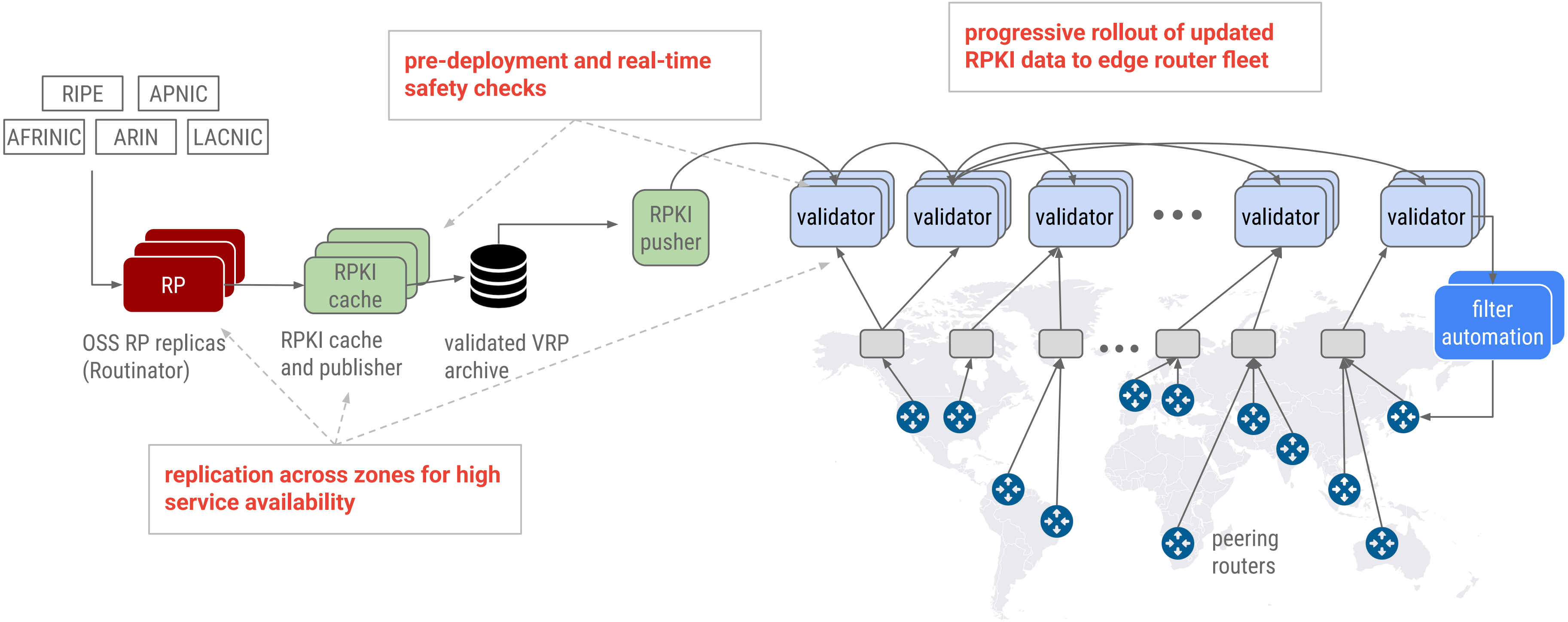
IRR filtering pipeline



RPKI origin validation pipeline



RPKI origin validation pipeline



Safety: IRR filtering

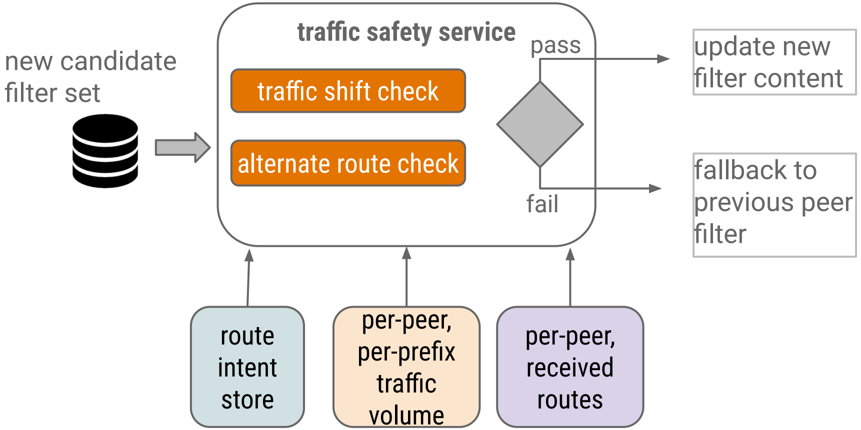
progressive introduction of filtering action: MARK → DEPREF → REJECT
(inform) → (fix) → (protect)

Pre-deployment safety checks:

- prevent peering link congestion and path impact
- prevent / detect potential blackholes

Real-time checks during rollouts:

- ensure overall device and traffic health



Safety: RPKI origin validation (WIP)

Validation of public RPKI data – avoid using data directly from external sources

- periodic “release” of public RPKI data into the filtering pipeline
- validate each release – i) significant changes in #VRPs, ii) impact on reachability

Reachability checks

- automate checks for reachability impacts due to RPKI INVALIDs
- develop policies for override decisions

Overrides and escalations

- tooling to add overrides for specific `<prefix, origin>`
- enable / disable filtering on peer ASNs, or specific sessions

Examples of filtering-related incidents and mitigations

Incident type	Root cause(s)	Mitigation approach
peer lost connectivity after IRR filtering introduced	• brittle routing architecture in some peers • incorrect impact analysis of filtering on peer traffic	• more careful filtering changes for enterprise and other non-ISP peers • update criteria for applying filtering
peer experiences unexpected or suboptimal routing	• incorrect advertisement accepted from unfiltered peer redirects traffic	• supplement IRR filtering with RPKI • fixes in IRR AS-SET and route object collection algorithms
peer IRR / RPKI update delays	• weekly IRR filter rollouts • periodic releases of RPKI data to filtering system	• improve communication and timeline expectations on Peering Portal • provide operations teams tools to query state of deployed filters
unexpected routing due to unfiltered peer	• incorrect advertisement accepted	• deploy ad-hoc filters to immediately mitigate • supplement with RPKI filtering for large peers

RPKI ROV – observations

IPv4 RPKI Invalids observed
Top 10 originating ASNs

Google edge		Public routing tables (NIST)	
ASN	Invalids	ASN	Invalids
22773	3431	23693	5797
834	608	2516	483
17561	397	12552	179
17670	174	41704	155
1	168	31713	142
33287	144	4787	121
18101	142	5384	96
38710	132	4804	90
984	123	9009	83
8100	123	18106	76

IPv6 RPKI Invalids observed
Top 10 originating ASNs

Google edge		Public routing tables (NIST)	
ASN	Invalids	ASN	Invalids
55836	4582	2516	4159
22677	2421	22677	2412
20115	2035	20115	1122
22773	1268	47331	594
12849	541	52257	382
64079	234	399169	256
30036	185	43357	249
207808	147	30036	182
852	117	50673	136
137409	116	9931	115

snapshots from end of January 2025

RPKI ROV – observations

IPv4 RPKI Invalids observed
Top 10 originating ASNs

Google edge		Public routing tables (NIST)	
ASN	Invalids	ASN	Invalids
22773	3431	23693	5797
834	608	2516	483
17561	397	12552	179
17670	174	41704	155
1	168	31713	142
33287	144	4787	121
18101	142	5384	96
38710	132	4804	90
984	123	9009	83
8100	123	18106	76

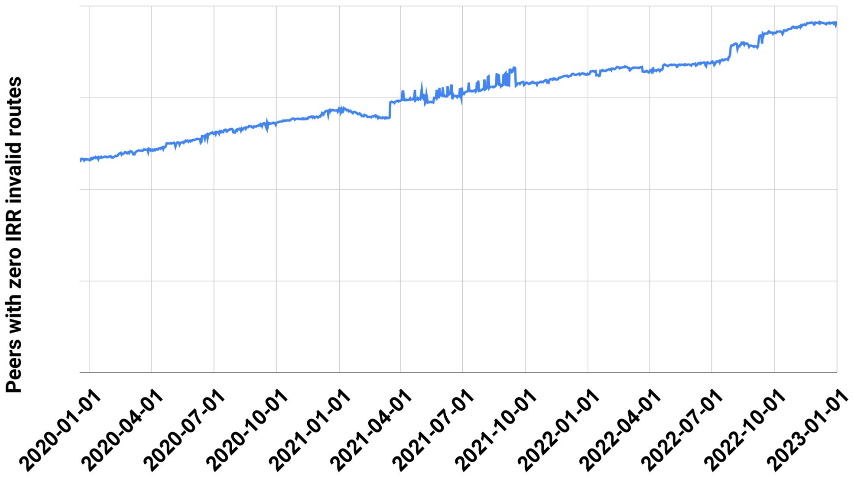
IPv6 RPKI Invalids observed
Top 10 originating ASNs

Google edge		Public routing tables (NIST)	
ASN	Invalids	ASN	Invalids
55836	4582	2516	4159
22677	2421	22677	2412
20115	2035	20115	1122
22773	1268	47331	594
12849	541	52257	382
64079	234	399169	256
30036	185	43357	249
207808	147	30036	182
852	117	50673	136
137409	116	9931	115

snapshots from end of January 2025

- public data: ~5K IPv4 / ~12K IPv6 *total* invalid routes in global routing table – we receive significantly greater number
- actively communicating with top peers sending RPKI invalids
- frequently changing distribution of origin and maxlen violations, with small proportion that violate both (14% this snapshot)

IRR filtering – observations



Steady improvements in IRR data based on invalid prefixes during DEPREF phase

Relatively small number of peer escalations due to traffic shifts

Limited filter size results in lack of filtering on peers who send a very large number of routes – address gap w/RPKI OV filtering

- some peers also have large filters due to very large AS-SETs relative to announced routes

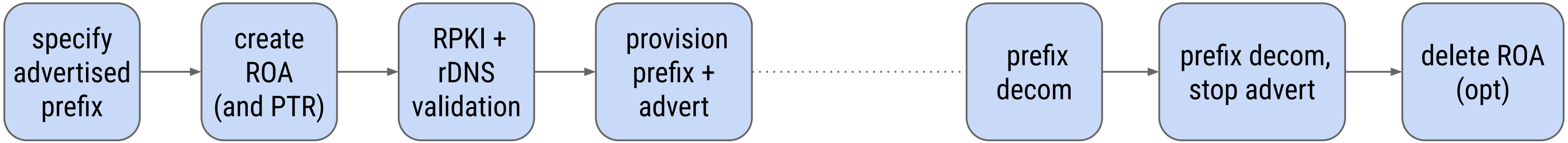
Experiments with limiting set of IRR DBs shows opportunity for restricting DB set somewhat, but will cause some impact

BYOIP and RPKI

Bring-Your-Own-IP allows customers to use their own IP ranges for cloud-hosted applications

Use cases: hosted security services, email services, VPN/remote access, legacy applications …

<u>lifecycle of BYOIP addresses (simplified)</u>



Operational challenges

- ensuring propagation / downstream acceptance of routes
  - e.g., proxy registrations in IRR to avoid filtering

- deleting ROAs ≠ service deprovisioning
- policy tradeoffs: rapid decomm vs. risk for creating outages due to accidental ROA deletion
- further complicated by very short-lived uses (e.g., using IP leasing services)

Route server decomm

- Route servers help scale BGP sessions in IXPs, but also have operational challenges
  - complexity to control traffic and manage congestion (especially inbound)
  - dependency on third party hardware, software, and security
- Main challenge: applying filtering in IXPs
  - inconsistent use of AS-SETs and inclusion of route server AS
  - large IXPs with large AS-SETs result in very large prefix lists
  - some IXPs perform route validation, but no BCP across all IXPs
- Google decided to decomm peerings with route servers
  - remove as a point of hijack vulnerability and operational toil
  - trade-offs between losing visibility and security risks
- Encouraging IXP peers to move to PNI or transit providers
  - simplify through automated peering turnups – requires correct route registrations in IRR / RPKI

Enterprise customer peering

Standard Internet Edge peering works well for network operators, not so much for enterprise cloud customers

Common issues

- no SLA on peering uptime or support
- inability to meet peering requirements – e.g., redundancy in connectivity, advertisements
- unfamiliarity with traffic failover, IP asset mgmt, routing DBs, incident management …
- lack of routing expertise
- distinguishing enterprise peers, and determining an appropriate filtering policy

Peering Provider
Verified Gold

Preferred approach: *Verified Peering Provider* (VPP)

VPP providers:

- need to meet technical requirements on peering diversity
- provide a simpler alternative to direct BGP peering for customers
- currently 18 VPP providers with more coming

Summary

Route filtering as a Cloud-scale distributed software service

- multiple layers of safety checks, especially on external data and careful deployment

Mitigation and “big red button” tools essential for SRE and operations teams

Route behavior at our edge is *very* dynamic, and significantly different from public Internet

Mix of enterprise customers and route filtering

- requires special handling when using standard peering
- move enterprises to VPP providers

THANK YOU!

aashaikh@google.com