RPKI availability concerns

Executive Summary

The Resource Public Key Infrastructure (RPKI, RFC6480) is a PKI that supports routing security. Internet resource holders are issued X.509 certificates by the registry responsible for those resources, and the resource holder can, in turn, issue RPKI signed objects, such as Route Origin Authorizations (ROAs), that can be used by network operators in order to make routing-related decisions.

The RIRs operate Trust Anchors (TAs) and associated repositories that are fundamental to the operation of the RPKI. Account holders, such as the National Internet Registries (NIRs), have raised concerns about the reliability and availability of those systems, and the effects of problems with those systems. In general, APNIC considers the risks in this area to be minimal: APNIC's TA and associated repositories are highly resilient to technical issues, and even in the event of a complete failure, network operator configuration should be such that routing is unaffected. APNIC does not recommend that any changes be made in this space. More generally, downstream registries should consider whether delegated Certification Authority (CA) operation (as opposed to APNIC-hosted CA operation) mitigates any of their concerns in this area, and network operators should consider whether their reliance on the availability of RPKI is aligned with the obligations owed to them by the various parties that operate RPKI's repositories and related systems.

1 Introduction

The RPKI is a PKI that supports routing security. Internet resource registries (RIRs, NIRs, and Local Internet Registries [LIRs]) manage infrastructure associated with this PKI, such as TAs, CAs, repositories, and web portals and APIs for making updates to the system. Network operators use protocols such as rsync and RRDP to retrieve RPKI information, which can then be used to support routing security. Currently, the principal object in RPKI for routing security is the ROA, which is issued by an IP address holder, and permits announcements of the relevant IP address space by the holder of the nominated AS number. While ROAs are the main object in use today, other object types are in active development, such as the Autonomous System Provider Authorization (ASPAs).

To use RPKI, network operators must deploy Relying Party (RP) software, such as Routinator or rpki-client, which retrieves RPKI information by way of a set of TAs. Most RP software defaults to using the TAs issued by the RIRs. As a result, the systems operated by the RIRs are central to the operation of RPKI. This criticality has prompted concerns about the risks of unavailability of these systems, and the possible need for fallbacks in the event of problems. This document includes an elaboration of these concerns, as well as evaluation of potential suggestions on the steps registries and operators may take to reduce risk in this space.

2 Concerns

The main issue can be summarized as: What happens to network operators if APNIC's RPKI repository is no longer accessible?

- RP instances that have cached state, that is, were running regularly before the repository becoming inaccessible, will continue to make use of that cached state until it expires. Objects managed by APNIC directly have validity periods of at least five days, in the normal course of things.
- RP instances that do not have cached state, that is, are being newly set up after the repository has become inaccessible, will have no RPKI state. As a result, all BGP routes will be treated as having an RPKI status of 'unknown' by a router relying on such an instance. Operators are recommended to accept such routes (RFC 6483), so the practical outcome is that connectivity is unaffected, unless an operator is doing something unusual.
- For the RP instances that have cached state, if inaccessibility persists for long enough that the cached objects expire, then the result is per the previous dot point.

In other words, for operators with conventional configurations, the effect of long-term repository inaccessibility is that BGP routes that would previously have been marked as 'invalid', due to there being a ROA that excluded their use, would be accepted. Inaccessibility does not lead to all BGP routes being rejected, or similar.

Additionally, the chance of APNIC's repository becoming inaccessible due to technical issues is very low. Cloudflare acts as a reverse proxy for the RPKI Repository Delta Protocol (RRDP) service, caching the objects that are returned by APNIC's system and handling the bulk of the traffic. Aside from that, Cloudflare currently has 62 points of presence in the APNIC service region, so there is significant redundancy underpinning the operation of the service. The rsync repository service, by contrast, is hosted in Brisbane directly, but most validators in deployment today use RRDP by default, with fallback to rsync only when RRDP is unavailable.

3 Potential actions

Notwithstanding the above, there has been interest in various actions that are perceived as addressing some of the reliability concerns that have been raised. Some of those actions are documented below. APNIC does not recommend that any of these actions be carried out.

3.1 Repository data mirroring

The current APNIC repository data could be mirrored to a repository server physically located in a given economy and controlled by APNIC directly, in order to provide further redundancy in the event of problems with the Cloudflare-based repository system. Relevant RPs could be directed to this service by way of BGP (anycast) or DNS, for example.

Advantages

- If the APNIC TA repository via Brisbane/Cloudflare is unavailable, then RP instances without cached state will still be able to initialize themselves based on this mirrored repository.

Disadvantages

- There would be some overhead on APNIC's part regarding setting up the new repository, mirroring data to it, and maintaining it over time.
- Member CAs have a validity period of seven days, which is the shortest validity period for objects hosted in the APNIC TA repository. These CAs are reissued every two days, which means they are valid for at least five days at any given time, all things being equal. This means that the mirror will only work as intended for about the same period of time (since objects will become stale once they reach the end of their validity period).

Assessment

- The benefits here are very marginal, as they are limited to uninitialized RPs, and even in the long-term unavailability case, the mirror is only useful for a short period of time (approximately five days).

3.2 Public rpki-rtr cache deployment

An NIR or similar entity could host a public rpki-rtr (RFC 8210) server for APNIC's data. Since rpki-rtr protocol data does not expire, operators could use this service to access APNIC's RPKI data indefinitely.

Advantages

- If the APNIC TA repository via Brisbane/Cloudflare is unavailable, then RP instances without cached state will still be able to rely on the data published by the rpki-rtr server.

Disadvantages

- RFC 8210 strongly discourages this mode of operation (reliance on an rpki-rtr server managed by a third party), since it involves delegating complete trust to the rpki-rtr server operator, without any way of verifying the results. This is a significant compromise in the default security posture of the system.
- Operators will still need to run their own RP and rpki-rtr instance for non-APNIC data, further complicating their deployments.

Assessment

- This option is similar to the first option, save that the rpki-rtr data does not expire. That ostensible benefit is more than outweighed by the risks associated with relying on an rpki-rtr server operated by a third party.

3.3 Update RP software to use stale state

Along the lines of the previous two actions, it may be possible to suggest or develop RP software updates that permit the indefinite use of stale state.

Advantages

- This would allow clients with cached state to continue relying on that state indefinitely, if APNIC's TA repository becomes unavailable. This behaviour would also extend to other repositories, rather than just APNIC's.

Disadvantages

- Since RPKI fails open anyway (see second bullet under Concerns above), the benefits of continued reliance on stale state are minimal.
- Relying on stale state runs counter to RPKI's intent that clients rely on objects only for so long as they remain valid, such that it is unlikely that these updates would be accepted or implemented by any of the major RP software vendors.

Assessment

- Per the disadvantages, it may be difficult to get this sort of update included. Even if that happens, it's unlikely that this option would be enabled by default, and getting RPs to enable it would also be difficult. Those difficulties mean that the chance of this being useful is low.

3.4 Separate TA as mirror of APNIC data

An NIR or similar entity could set up a new, standalone RPKI TA, to mirror APNIC's data. The TA maintainer would set up some sort of scheduled process that caused the new TA to publish ROAs such that the outcomes in routing were the same regardless of whether APNIC's TA or the new TA were used.

Advantages

- If the APNIC TA is unavailable but the new TA is available, then RP instances without cached state will still be able to initialize themselves based on the new TA.
- The new TA can republish its state indefinitely, such that its state prior to APNIC TA unavailability continues to have effect in routing.

Disadvantages (practical)

- Clients will need to be manually configured with the new Trust Anchor Locator (TAL) for this TA, since it is unlikely that existing TAL distributors (RP software implementations and some operating systems) will themselves make the TAL available.

- It may be difficult to ensure that operators are using both the APNIC TA and the new TA, since the results will be the same regardless of whether an operator is using the APNIC TA, the new TA, or both. This, in turn, makes it more difficult to realize the benefits of the additional TA.
- Such a system will have ongoing development costs. For example, once ASPAs are deployed, the system would need to be updated to copy ASPAs across as well.
- All RPKI object types to date are idempotent, and all new ones should be as well, but if a non-idempotent object type is created, duplicating those objects will not work as a resilience/backup measure.
- The disadvantages from Update RP software to use stale state apply here as well.

Disadvantages (architectural)

- Any additional general-purpose TA increases the surface area for problems in RPKI, due to a TA being able to make statements about any resource.
- Additional TAs complicate communications around deployment. Currently, RPs default to installing the five RIR TAs, but that would need to change in future to account for a new TA like this.
- Additional TAs contribute to a sense of fragmentation around RPKI that may lead users to look at alternatives for route security.

Assessment

- This option has a large number of significant disadvantages, including some that go to the general architecture of the system, such that it is not worthwhile to pursue it.

4 Suggestions for Registries

4.1 Delegated CA operation

Most APNIC account holders run APNIC-hosted CAs, where APNIC manages the CA and its associated repository on behalf of the account holder, and updates involving the CA happen by way of the MyAPNIC portal or the registry API. An account holder that is concerned about RPKI availability issues could opt to run a delegated CA under APNIC instead, using Krill, for example. Doing this would allow them greater control over their RPKI state, and they would also gain experience with operating an RPKI CA, while remaining within the current single-APNIC-TA model. APNIC offers an RPKI repository publication service, so the account holder could rely on that for distributing their repository, or they could decide to run their own repository service.

This recommendation does not address any of the immediate concerns raised in this document, in that operating a delegated CA still involves reliance on the availability of APNIC's TA and repository. However, it does in some ways lessen the reliance on APNIC's system, since the delegated CA itself has responsibility for issuing and renewing ROAs and other objects.

5 Suggestions for operators

5.1 Handling RPKI unavailability

There are anecdotal reports of operators in the APNIC service region relying on RPKI to a greater extent than that intended by the architects of the system. For example, an operator might require that their customers publish ROAs for their networks, rejecting any announcements from their customers that are not covered by such ROAs. An operator relying on RPKI to this extent should consider the appropriateness of that reliance, given that the RPKI is managed by a diverse group of parties who generally owe no obligation to the operator with respect to availability or support. See, for example, AWS secures internet routing with RPKI plus security checks.

6 Comments on unavailability of other repositories

This document is about the potential unavailability of the APNIC RPKI repositories. A related problem is the potential unavailability of all RPKI repositories, including those for the other RIRs' TAs. In general, the discussion in this document is equally relevant to that unavailability scenario, subject to the following considerations.

- RIR repository distribution and availability characteristics are generally similar to those for APNIC. However, delegated repository management is much less consistent in this respect, with many repositories being deployed as a single instance in one geographic location.
- Most RPKI content is contained within the RIR repositories themselves, and that content will be relevant to RPs in general. For delegated repositories, the relevance is more context-specific. For example, an NIR may operate a delegated repository for its own resources, but if those resources are relied on mostly or entirely by operators and users within its economy, then the effect of unavailability for other operators and users will be minimal when compared with those within the economy itself. Along similar lines, many delegated repositories are operated as test or research systems, where unavailability is either anticipated or unproblematic.
- RPKI objects managed by APNIC directly will typically have a validity period of at least five days. However, there are no protocol requirements regarding validity periods and renewal schedules, and different repositories take different approaches. The rpki.net CA software, for example, uses a default validity period of six hours, with reissuance after four hours and thirty minutes. This affects the usefulness of a solution like Repository data mirroring with respect to content produced by an rpki.net CA that is configured in this way.
- Repository data mirroring is only available with the co-operation of the repository operator, since it depends on the existing repository URL resolving to the mirror repository server.
- Public rpki-rtr cache deployment can be extended to handle data from other RPKI repositories very easily.
- Update RP software to use stale state would work for all RPKI repositories.
- A separate TA per Separate TA as mirror of APNIC data could be used to mirror data from delegated repositories, as well as from the repositories of other RIRs.
- The proposed ERIK protocol is a potential general-purpose solution along the lines of Repository data mirroring, in that it operates as a mirror for existing RPKI content. However, it requires RPs to be using software that supports it, and to be configured to download content from the ERIK mirror, rather than from the RPKI repositories themselves, so it is not a ready solution in the near term.
- Delegated CA operation is not relevant to the scenario of unavailability of other RPKI repositories.
- Handling RPKI unavailability is equally relevant to the scenario of unavailability of other RPKI repositories.