An update to the RPKI validation algorithm
draft-spaghetti-sidrops-rpki-validation-update-00

Job Snijders
job@fastly.com
Fastly

Ben Maddison
benm@workonline.africa
Workonline

Quick recap, what problem are we solving?

When an intermediate CA shrinks, any subordinate CA also needs to shrink ASAP. Following the validation algorithm described in RFC 6487, * ALL *** objects subordinate to an over-claiming CA become invalid.**

!!! Translation: IP transfers lead to RPKI object outages !!!

Real life report on *the problem:*

https://www.ripe.net/ripe/mail/archives/routing-wg/2021-January/004220.html

Don’t be distracted, in the same thread 2 *different* problems are discussed:

1: Routinator’s poor manifest handling (now fixed!)

2: Two perfectly usable ROAs were considered invalid (THIS IS WHAT WE FOCUS ON!)

How did we end up here?

Both the RFC 6487 and RFC 8360 algorithms are "*valid algorithms*" in the sense that they describe a procedure which results in some certificates being accepted and some rejected.

There is a degree of subjectivity as to which algorithm is *better*

The authors favor less operational brittleness, as long as it does not come at the expense of security. The 8360 algo is superior.

Validity is in the eye of the beholder.

The RFC 8360 algorithm does not introduce weakness

Citing from “RFC 8360 Section 7. Security Considerations”

The authors believe that the revised validation algorithm introduces no new security vulnerabilities into the RPKI, because it cannot lead to any ROA and/or router certificates to be accepted if they contain resources that are not held by the issuer.

In this context - RIRs don’t need to dictate how RPs validate

A “visible” change (such as setting a different Policy OID) is not likely to happen, out of fear of Relying Parties losing access to the RPKI information.

The Relying Parties are the ones executing the validation algorithm, only RPs are in a position to choose to use an improved algorithm.

Requiring CAs to set a new policy OID in order for RPs to begin using the improved algorithm is an unnecessary step: there is nothing to signal here.

The Profile Agility procedure as described in RFC 6487 changes a ‘backwards compatible’ change into a ‘breaking’ change, for no good reason.

The plan:

Update RFC 6487 to document how Relying Parties can apply the new algorithm to existing objects

Remove the RFC 6487 section that led to this impasse

Deprecate RFC 8360

RIRs (or intermediate CAs) are not required to take any action

Implementation status - this is entirely doable

FORT:

https://github.com/job/FORT-validator/commit/ff5f4b9313d5c553fa13bae427acb69665977727

Routinator:

https://github.com/job/rpki-rs/commit/d9fa8c72cf83ed6f25e4420eaaa9054078f15bc3

OpenBSD rpki-client:

https://marc.info/?l=openbsd-tech&m=161011710120123&w=2

What about existing work?

Yeah… as RPKI community we've wasted an incredible amount of time on this problem.

The good news: the X.509 "Policy" Extension is not 'burned', it can be used in the future if ever a need arises to use it in the future.

Only the RFC 8360 OIDs are 'burned' (but it is not used in practice anyway, and comes from an unlimited code point space)

Most RPs can just delete a bunch of code (simplifying their software)

Next steps?

The authors would like to request the chairs to start a call for Working Group Adoption.