RPKI MIRO: Monitoring and Inspection of RPKI Objects

Andreas Reuter
Freie Universität Berlin
andreas.reuter@fu-berlin.de

Matthias Wählisch
Freie Universität Berlin
m.waehlisch@fu-berlin.de

Thomas C. Schmidt
HAW Hamburg
t.schmidt@haw-hamburg.de

ABSTRACT

The Resource Public Key Infrastructure (RPKI) stores attestation objects for Internet resources. In this demo, we present RPKI MIRO, an open source software framework to monitor and inspect these RPKI objects. RPKI MIRO provides resource owners, RPKI operators, researchers, and lecturers with intuitive access to the content of the deployed RPKI repositories. It helps to optimize the repository structure and to identify failures.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces

Keywords

PKI monitoring, RPKI measurement, secure inter-domain routing

1. INTRODUCTION

The RPKI [1] is a public key infrastructure for Internet resources. Basically, it consists of three parts: (i) extended X.509 certificates that attest the ownership of IP prefixes and autonomous system (AS) numbers, i.e., the Internet resources, (ii) RPKI signed objects, which do not follow the X.509 data structure but are necessary to implement secure Internet routing (e.g., route origin objects), and (iii) a distributed repository system, which stores and provides all of the signed objects to Internet operators and other users. So-called relying party software fetches RPKI objects via the *rsync* protocol from the Certificate Authorities (CA) and cryptographically verify the data for further processing.

On top of the RPKI, approaches have been developed to improve routing security. A recently deployed example is prefix origin validation, which authorizes an AS to announce an IP prefix. Corresponding route origin authorization objects (short ROAs) have already been created by several operators and service providers such as AT&T, Deutsche Telekom, and Mozilla. As for any PKI, the success of RPKI applications depend on stable and coherent repository data. Surprisingly, almost no high-level monitoring tools are available to inspect (R)PKI content. Available tools mostly focus on gathering statistical measures and meta data about RPKI repositories (e.g., [2]), or only allow limited inspection of RPKI repository contents [3, 4].

SIGCOMM '15 August 17-21, 2015, London, United Kingdom

ACM ISBN 978-1-4503-3542-3/15/08.

DOI: http://dx.doi.org/10.1145/2785956.2790026

Why do we need RPKI MIRO?

The core tools for RPKI were primarily built for experts and provide only limited convenience functions. Furthermore, monitoring, in particular distributed measurements, is not easily supported. Several user groups would benefit from an extendable and easy-to-use monitoring framework.

CA Operators There are two options to deploy the RPKI: The *hosted mode*, (i.e., each Regional Internet Registry runs the CA and repositories for the resource owners), and the *delegated mode*, (i.e., each participant becomes their own RPKI and maintains the repository). Currently, the delegated mode is rarely supported by the RIRs. This will change and larger ISPs will switch to this model in the future. Our experience already showed that RPKI MIRO helps to improve the repository structure of RIRs, and we expect additional demand for RPKI MIRO with increasing deployment of the delegated model.

Resource Owners Transparency is crucial for the acceptance of the RPKI. Public RPKI data does not need to be correct [5]. RPKI MIRO aims for easy access on RPKI certificates, revocation lists, ROAs etc. to finally give Internet operators more confidence in their data. RPKI MIRO is modular to add additional views or to integrate components that alarm individual resource owners in case of failures.

Researchers Even though the basic functionality of RPKI has been standardized within the IETF, there are still open research questions, in particular with respect to the ongoing deployment. For example, the mechanism to retrieve RPKI objects from the distributed repositories is still under discussion [6, 7]. The modular architecture of RPKI MIRO allows it (a) to plug different fetching strategies in, (b) to perform distributed measurements, (c) to coherently analyse the results.

Lecturers Public Key Infrastructures and thus also the RPKI are complex systems. The interplay of the different (R)PKI objects is usually hard to grasp for students. Real data may help when teaching. However, current tools to inspect these objects are based on command line interfaces. In particular, they do not visualize the link between different object types. RPKI MIRO provides an intuitive graphical browser to visualize object content as well as the relation between the objects, which will help students to better understand the RPKI concepts. It is worth noting that the visualization is tailored to RPKI but can be easily updated for other X.509 extensions.

In the remainder of this abstract, we briefly explain the basic architecture of RPKI MIRO, report about experiences, and discuss future work.

Repository	W/o Prefetching # rsync Calls	W/o Prefetching Time	W/ Prefetching # rsync Calls	W/ Prefetching Time
AfriNIC	35 calls	96 s	3 calls	11 s
APNIC	534 calls	1745 s	2 calls	35 s
APNIC AfriNIC	5 calls	16 s		
APNIC ARIN	25 calls	81 s		
APNIC IANA	492 calls	1610 s		
APNIC LACNIC	2 calls	6 s		
APNIC RIPE	10 calls	32 s		
ARIN	1 call	7 s	1 call	7 s
LACNIC	1 call	9 s	1 call	9 s
RIPE	1 call	23 s	1 call	23 s

Table 1: Performance comparison of collecting RPKI data without and with prefetching

2. ARCHITECTURE OF RPKI MIRO

The source code of RPKI MIRO is publicly available (https://github.com/rpki-miro) and licensed under MIT. RPKI MIRO consists of the following components:

Backend—Validator The validator downloads, parses, and validates RPKI objects. Based on this processing chain, the validator component provides statistics which are relevant for evaluating the functions of a relying party. The data of validated repositories can be exported in multiple formats for further processing.

Efficient fetching of objects is crucial for any relying party. A new rsync call will result in a new TCP connection, which leads to overhead on the server side. However, timing is important for a secure Internet backbone. To minimize the amount of separate calls, we implemented a pre-fetching strategy that derives the longest common prefix of multiple publishing points, which then is used to access the data. Table 1 shows the performance gain in terms of rsync calls and download time thanks to reduced communication overhead.

Backend—Statistics Separate modules calculate statistical measures (e.g., number of valid objects) to evaluate the repository state. Such information is helpful for two reasons: First, it allows an analysis of the coherence of the distributed repository among different vantage points. Second, it allows the comparison of different implementations of the validator. Our work revealed some bugs in existing relying party software.

Frontend—Browser The browser component visualizes the content of RPKI objects and their relationship. It is worth noting that the current tree view per trust anchor is *not* a direct representation of the underlying file system structure—this would overload the presentation. Instead of that, we decided to show only certificates and ROAs within the tree and present their meta data as well as relationships to manifest and CRLs in a separate detail view. Due to the model view controller concept of RPKI MIRO, additional views can be easily added. A public instance of the RPKI browser is available under http://rpki-browser.realmv6.org/.

Frontend—Statistics This component visualizes previously gathered statistics. It allows for an easy comparison of the repository state with other relying parties.

3. EXPERIENCES AND FUTURE WORK

RPKI MIRO is used by the research as well as the operator community. It has already helped to better understand the current state of RPKI deployment. For example, with RPKI MIRO it was easy to identify that the cross-RIR resource space has increased and includes route origin attestation objects. Cross-RIR resources are resources

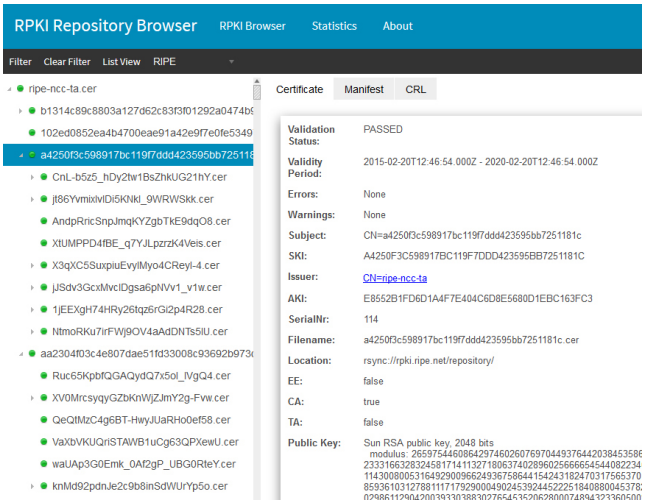

Figure 1: One component of RPKI MIRO: The repository browser showing the content of a certificate of the RIPE CA.

which have been signed by one RIR and are transferred to another RIR. This requires that the provisioning protocol [8] is functional between the Regional Internet Registries—something that lacked attention.

RPKI MIRO has also helped to better understand how the five RIRs have implemented majority-minority address space, i.e., IP address space of which a super prefix is managed by one RIR but some sub-prefixes of this prefix are managed by another RIR. This fragmentation has implications for creating resource certificates.

Part of our ongoing research work will be the analysis of the different protocol proposals to retrieve RPKI objects from the repositories. For this we will leverage the distributed measurement functionality of RPKI MIRO. We will also refine our pre-fetching strategy to automatically adapt to the repository structure.

On the feature side of RPKI MIRO, we will extend the web interface to allow for the uploading of external repository data. We will also enable users to configure monitoring of repository changes and to initiate alarms.

Acknowledgments We thank the SIDR community and the RIR community for discussions and their valuable feedback, special thanks to Rob Austein, Tim Bruijnzeels (RIPE), and Carlos M. Martinez (LACNIC). This work has been supported within the BMBF project Peeroskop.

4. REFERENCES

[1] M. Lepinski and S. Kent, "An Infrastructure to Support Secure Internet Routing," IETF, RFC 6480, February 2012.
[2] J. Kloots, "RPKI Dashboard," http://rpki.surfnet.nl/, 2015.
[3] RIPE NCC, "RPKI Validator," https://github.com/RIPE-NCC/rpki-validator, 2015.
[4] LACNIC, "Origin Validation Looking Glass," http://www.labs.lacnic.net/rpkitools/looking_glass/, 2015.
[5] E. Heilman, D. Cooper, L. Reyzin, and S. Goldberg, "From the consent of the routed: Improving the transparency of the rpki," in *Proc. of ACM SIGCOMM* 2014.
[6] T. Bruijnzeels, O. Muravskiy, B. Weber, R. Austein, and D. Mandelberg, "RPKI Repository Delta Protocol," IETF, Internet-Draft – work in progress 00, February 2015.
[7] E. Osterweil, T. Manderson, R. White, and D. McPherson, "Sizing Estimates for a Fully Deployed RPKI," Verisign Labs, TR 1120005 version 2, December 2012.
[8] G. Huston, R. Loomans, B. Ellacott, and R. Austein, "A Protocol for Provisioning Resource Certificates," RFC 6492.