RIPE NCC Response to the Code Audit Report from Radically Open Security - December 2021

TABLE OF CONTENTS

1. EXECUTIVE SUMMARY

1.1 Introduction

Security continues to be high on our list of priorities for RPKI. Every year, we ask an external party to conduct a security audit of our RPKI systems. This is the second time that we are publishing the security report, in an effort to increase transparency and trust in the RPKI system. In this report, we list the findings of the external party that carried out this code audit and how we mitigated these issues. This code audit was requested in preparation for open sourcing the RPKI Core code. Other elements of our RPKI code, such as RPKI Commons and RPKI Publication were open source already. You can find more information on our GitHub page.

Between 26 June and 16 July 2021, Radically Open Security B.V. (ROS) carried out a penetration test and code review for the RIPE NCC. These audits were intended to assess the security level of the various components of RPKI. We are committed to having annual security assessments.

1.2 Scope

The scope of the penetration test was limited to the following target(s):
-RPKI Core (ripe-rpki-ripe-ncc)
-RPKI Trust Anchor (ripe-rpki-ta-0)
-RPKI Commons library (ripe-rpki-commons)

1.3 Project Objectives

The project objectives are described in the report from ROS.

1.4 Timeline

The penetration test took place between 26 June and 16 July 2021.

1.5 Results in a Nutshell

ROS identified one high, three moderate, and one low-severity issue during this code audit. These details are described in detail in section 3 of the ROS report.

1.6 Summary of Findings

The summary of findings is described in the report from ROS.

1.7 Summary of Recommendations

The summary of recommendations is described in the report from ROS.

2. METHODOLOGY

This section in the ROS report describes the methodology and risk classifications used for the code audit.

3. FINDINGS

3.1 RIPE-004 - Unauthenticated Access to administrative functionalities

Vulnerability type: Authentication bypass
Threat Level: High

ROS: It was found that the RPKI Core web application executes administrative actions before checking the user's session.

The RPKI core consists of multiple components. Among those components is an Apache Wicket web application that performs cookie-based session authentication to verify if the user is authenticated or not. It was determined that the authentication check is performed after the execution of critical operations. More specifically, the check is performed in the function onConfigure of the base package MinimalRPKIBasePage. This function is called on all components before any component is rendered. However, before rendering, the administrative actions have already been performed.

RIPE NCC: This was patched quickly after it was identified. Authorisation is now consistently checked before executing any actions.

3.2 RIPE-003 - XML Processing Might Lead to DoS

Vulnerability type: Denial of Service
Threat level: Moderate

ROS: The manner in which the RPKI core processes XML messages can result in denial of service.

The ripe-prki-commons library provides a DocumentBuilder configuration that is secure against XXE vulnerabilities in the first place. The attributes ACCESS-EXTERNAL-DTD and ACCESS-EXTERNAL-SCHEMA do not allow any protocol, which prevents Server-side reuest forgery (SSRF) vulnerabilities or the reading of local files via XXE. Although FEATURE-SECURE-PROCESSING limits the entity expansion to 64000 by default and the accumulated size of entities to 50.000.000 bytes, an attacker can still abuse the limited expansion for a DoS attack. If an attacker sends multiple requests in parallel containing an entity expansion payload, the server will throw a java.lang.OutOfMemoryError:GC overhead limit exceeded exception.

RIPE NCC: We no longer evaluate internal entities in XML messages.

3.3 RIPE-005 - Missing Request Size Limits can Cause DoS

Vulnerability type: Denial of Service
Threat level: Moderate

ROS: It was found that the RPKI core does not specify request size limits that prevent the exhaustion of server memory.

We identified multiple endpoints that process requests without implementing size checks, exposing the RPKI core to the risk of a denial of service attack.

RIPE NCC: We limited the up/down request size to 1 MB.

3.4 RIPE-002 - Reflected XSS in Error Page

Vulnerability type: Input sanitization
Threat level: Moderate

ROS: The RPKI core exposes extensive error messages that allows exploiting a reflected XSS vulnerability.

The RPKI core server returns extensive error messages containing stacktraces if the application is in Apache Wicket development mode. Upon further investigation, it was found that the application is always in development mode as this setting is hardcoded into the source code. During the audit, no code was found that changed this configuration.

RIPE NCC: Outside of development systems, the Apache Wicket development mode is now disabled. Also, we only show stacktraces in the UI when running in development mode.

3.5 RIPE-007 - Multiple Outdated Dependencies

Vulnerability type: Outdated software
Threat level: Low

ROS: The RPKI core relies on multiple outdated client as well as server-side dependencies with known vulnerabilities

Client-side:
Multiple deprecated client-side dependencies have been identified. The following list summarizes a set of libraries that have not been updated for a long time and contain unpatched vulnerabilities. Therefore, the web UI is exposed to the risk of various client-side vulnerabilities:
public/static/api-docs/lib/jquery-1.8.0.min.js
public/static/api-docs/lib/handlebars-1.0.0.js
public/static/api-docs/lib/underscore-min.js
public/static/api-docs/lib/swagger-custom.js
portal-theme/js/vendor/jquery-ui-1.8.16.custom.min.js
portal-theme/js/vendor/jquery-1.6.4.min.js

Server-side:
On the server-side, two libraries related to the Apache Wicket framework were found. These libraries have not been updated since 2014 and are considered deprecated.

RIPE NCC: We replaced the vulnerable springfox-swagger library to mitigate the XSS issue. Wicket is still in use for the admin UI and we are working on a full replacement for the admin UI.

3.6 RIPE-001 - Potential XXE Through Third Party or MITM

Vulnerability type: Security configuration
Threat level: Low

ROS: It was found that the RPKI server loads XML content from a third party via an insecure channel and parses the content without security configuration.

During the audit of the RPKI core component, an insecure code pattern was identified. The implemented ResourceLookupService requests three different XML documents from the external host www.iana.org via unencrypted channels and parses the contents in an insecure manner. In particular, this implementation is exposed to two vulnerabilities: Requesting data using unencrypted channels introduces the risk of a Man-in-the-Middle attack. In this scenario, an attacker would intercept and modify the response of the requested resource. Therefore the integrity of the requested data cannot be guaranteed. Parsing XML documents without enabling security features can cause XXE attacks allowing an attacker to read local files or issue requests to hosts in the internal network.

RIPE NCC: We no longer evaluate internal or external entities in XML messages. We now load IANA, and other external data over HTTPS.

4. NON-FINDINGS

In this section in the ROS report they describe what things they tried but turned out to be dead ends.

4.1 NF-008 - Permissive Deserialization Whitelist

ROS: The deserialization functionality with the XStream library of the RPKI trust anchor has been subject of the intense review. A default whitelist of types that are explicitly allowed for deserialization has been installed [..] the list in place was found to be permissive.

RIPE NCC: We now explicitly allow only the types used in the XML for offline CA communication.

4.2 NF-006 - Inadequate Handling of Content-Type Header

ROS: The exposed up/down endpoint which implements the server-side of the provisioning protocol does not properly handle the request and response Content-Type header, which might lead to exploitable vulnerabilities.

RIPE NCC: We now always set the up/down response Content-Type header to be consistent with the RFC.