Red Hat Trusted Profile Analyzer 1

Reference Guide

Additional reference information for Red Hat Trusted Profile Analyzer

Last Updated: 2024-07-03

Red Hat Trusted Profile Analyzer 1 Reference Guide

Additional reference information for Red Hat Trusted Profile Analyzer

Legal Notice

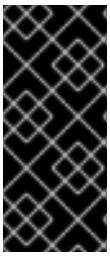
Abstract

This Reference Guide gives users additional information about Red Hat's Trusted Profile Analyzer service. Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message

Table of Contents

PREFACE

Welcome to the Red Hat Trusted Profile Analyzer Reference Guide!



IMPORTANT

Red Hat Trusted Profile Analyzer is a Technical Preview release. A Service Preview release has features that are in early development.

To give feedback or inform our engineering team of any technical issues with Trusted Profile Analyzer, email us at rhtc-support@redhat.com.

CHAPTER 1. FREQUENTLY ASKED QUESTIONS

Do you have questions about Trusted Profile Analyzer? Here is a collection of common questions and their answers to help you understand more about Red Hat's Trusted Profile Analyzer service.

Q: What is Red Hat's Trusted Profile Analyzer service?

A: Red Hat's Trusted Profile Analyzer service is a proactive service that helps you evaluate the security and vulnerability risks of using Open Source Software (OSS) packages and dependencies in your application stack.

Q: How can I use Red Hat's Trusted Profile Analyzer service?

A: There are two ways you can use Red Hat's Trusted Profile Analyzer service. First, by using the Dependency Analytics extension for integrated development environment (IDE) platforms, such as Microsoft's Visual Studio Code, or Jet Brains' IntelliJ IDEA. Using Dependency Analytics gives you in-line guidance on vulnerabilities as you write your application. Second, by searching for Software Bill of Materials (SBOM) and Vulnerability Exploitability eXchange (VEX) information for Red Hat products on Red Hat's Hybrid Cloud Console.

Q: What kind of content will be available with the Trusted Profile Analyzer service?

A: You have access to application libraries for Java, NodeJS, Python, Go, and Red Hat Enterprise Linux packages. Vulnerability information about open source packages comes directly from internal Red Hat resources, Red Hat's partner ecosystem, such as Snyk, and open source community data sources.

Q: What content will be available with the Trusted Profile Analyzer Service Preview release?

A: The following content will be available for Service Preview:

Quarkus Java Framework for Java Archive (JAR) files with associated SBOM files.

Red Hat Enterprise Linux Universal Base Image (UBI) version 8 and 9 with associated SBOM files.

Vulnerability information about open source Java packages.

Q: How does a Trusted Profile Analyzer SBOM help me?

A: A Trusted Profile Analyzer Software Bill of Materials (SBOM) can help you by understanding the software components within an application stack, and any related vulnerabilities those software components can have. An SBOM can improve visibility and transparency of open source code within the software supply chain by component's provenance, license information, and attestation of how it was built.

Q: Who is using Red Hat's Trusted Profile Analyzer service?

A: The primary audience for Red Hat's Trusted Profile Analyzer service is Quarkus Java developers, and cloud-native container image builders are using the Red Hat Enterprise Linux UBI.

Q: To use Red Hat's Trusted Profile Analyzer service, do I need to learn anything new, or change my development workflows and processes?

A: No.

Q: I am not a Quarkus Java developer, can I still gain any value from Red Hat's Trusted Profile Analyzer service?

A: Yes. The Trusted Profile Analyzer service still provides security risk information about open source packages that are not currently included in the Trusted Profile Analyzer repository.

CHAPTER 2. GLOSSARY

Common terms and definitions for Red Hat's Trusted Profile Analyzer service.

Exhort

The backend endpoint of Trusted Profile Analyzer where all the API requests get sent, to retrieve the necessary data to analyze, including package dependencies and vulnerabilities. The Red Hat Dependency Analytics (RHDA) integrated development environment (IDE) plug-in uses this endpoint to generate vulnerability reports within the IDE framework.

Software Bill of Materials

Also known by the acronym, SBOM. A manifest of dependent software packages needed for a particular application.

Single Pane of Glass

Also known by the acronym, SPOG. The RESTful application programming interface (API) for the Trusted Profile Analyzer web dashboard, and notifications.

Vulnerability Exploitability eXchange

Also known by the acronym, VEX. A security advisory issued by a software provider for specific vulnerabilities within a product.

Common Vulnerability Exposures

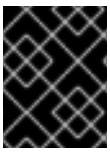
Also known by the acronym, CVE. A CVE indicates a product's exposure to attacks and malicious activities by giving it a score 1-10, where 1 is the lowest exposure level and 10 is the highest exposure level.

Common Vulnerability Score System

Also known by the acronym CVSS. The CVSS calculates CVE scores according to specific formulas when trying to calculate CVEs in a broad range of products and networks.

CHAPTER 3. CREATING A SOFTWARE BILL OF MATERIALS MANIFEST FILE

Red Hat Trusted Profile Analyzer can analyze both CycloneDX and Software Package Data Exchange (SPDX) SBOM formats by using the JSON file format. Many open source tools are available to you for creating Software Bill of Materials (SBOM) manifest files from container images, or for your application. For this procedure we are going to use the Syft tool.



IMPORTANT

Currently, Trusted Profile Analyzer only supports CycloneDX version 1.3, and SPDX version 2.2.

Prerequisites

- Install Syft for your workstation platform:
  - Red Hat Ecosystem Catalog
  - GitHub

Procedure

1. To create an SBOM by using a container image.
   CycloneDX format:

   Syntax

   ```
   syft IMAGE_PATH -o cyclonedx-json
   ```

   Example

   ```
   $ syft registry:example.io/hello-world:latest -o cyclonedx-json
   ```

   SPDX format:

   Syntax

   ```
   syft IMAGE_PATH -o spdx-json
   ```

   Example

   ```
   $ syft registry:example.io/hello-world:latest -o spdx-json
   ```

   NOTE

   Syft supports many types of container image sources. See the official supported source list on Syft's GitHub site.

2. To create an SBOM by scanning the local file system.
   CycloneDX format:

Syntax

```
syft dir: DIRECTORY_PATH -o cyclonedx-json
syft file: FILE_PATH -o cyclonedx-json
```

Example

```
$ syft dir:. -o cyclonedx-json
$ syft file:/example-binary -o cyclonedx-json
```

SPDX format:

Syntax

```
syft dir: DIRECTORY_PATH -o spdx-json
syft file: FILE_PATH -o spdx-json
```

Example

```
$ syft dir:. -o spdx-json
$ syft file:/example-binary -o spdx-json
```

Additional resources

- Scanning an SBOM manifest file by using the Red Hat Trusted Profile Analyzer managed service.
- National Telecommunications and Information Administration's (NTIA) How-to Guide on SBOM generation.