Red Hat Trusted Profile Analyzer 1

Quick Start Guide

Using the Red Hat Trusted Profile Analyzer managed service on Red Hat Hybrid Cloud Console

Last Updated: 2024-07-03

Red Hat Trusted Profile Analyzer 1 Quick Start Guide

Using the Red Hat Trusted Profile Analyzer managed service on Red Hat Hybrid Cloud Console

Legal Notice

Abstract

This Quick Start Guide gives users the essential information to start using the Red Hat Trusted Profile Analyzer managed service on Red Hat Hybrid Cloud Console. 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's Trusted Profile Analyzer (RHTPA) Quick Start Guide! This is a quick guide on how to use the RHTPA managed service on Red Hat's Hybrid Cloud Console.

CHAPTER 1. SEARCHING FOR VULNERABILITY INFORMATION

You can use the Trusted Profile Analyzer service to find existing Software Bill of Materials (SBOM), Vulnerability Exploitability eXchange (VEX) documents, and common vulnerability and exposure (CVE) information for Red Hat products and packages.

IMPORTANT

For this Technical Preview release, Trusted Profile Analyzer provides only information for the following Red Hat products:

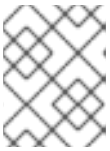
- Red Hat Enterprise Linux Universal Base Image (UBI) versions 8 and 9.
- The Java Quarkus library.

Prerequisites

- A Red Hat user account to access the Red Hat Hybrid Cloud Console .

Procedure

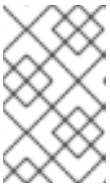
1. Open a web browser.
2. Go to the **Application and Data Services** home page on the Hybrid Cloud Console.
3. If prompted, log into the Hybrid Cloud Console with your credentials.
4. On the navigation menu, click **Trusted Profile Analyzer**.
5. On the Trusted Profile Analyzer home page, click the **Subscribe and launch** button. A new web browser window opens to the Trusted Profile Analyzer console home page.



NOTE

By subscribing, your registered email address goes onto the product mailing list, so you can receive information on new product developments.

6. On the **Home** page, in the search field, enter your search criteria and click **Search**.
7. On the search results page, you can filter the results by Red Hat products, download SBOM files, view package vulnerability information, and view any possible remediations.



NOTE

The number shown on the **Advisories** tab is how many times your search criteria made a match. On the **Products and containers** tab, the number in the **Product advisories** column shows the number of advisories for that specific product.

CHAPTER 2. SCANNING A SOFTWARE BILL OF MATERIALS FILE

You can scan your custom software bill of materials (SBOM) manifest files for analysis by Red Hat.

IMPORTANT

Red Hat does not retain a copy of your scanned SBOM files.

Prerequisites

- A Red Hat user account to access the Red Hat Hybrid Cloud Console .
- An existing CycloneDX 1.3 or Software Package Data Exchange (SPDX) 2.2 manifest files.

Procedure

1. Open a web browser.
2. Go to the **Application and Data Services** home page on the Hybrid Cloud Console.
3. If prompted, log in to the Hybrid Cloud Console with your credentials.
4. On the navigation menu, click **Trusted Profile Analyzer**.
5. On the Trusted Profile Analyzer home page, click the **Subscribe and launch** button. A new web browser window opens to the Trusted Profile Analyzer console home page.

   NOTE

   By subscribing, your registered email address goes onto the product mailing list, so you can receive information about new product developments.

6. Click **Scan SBOM** from the navigation menu.
7. You can drag-and-drop an SBOM manifest file onto the page, or click **Load an SBOM**
8. After scanning the SBOM file, you get a summary of the analysis, and specific vulnerability information for the packages included in your SBOM file.

Additional resources

- To learn how to create a software bill of materials file, see the Trusted Profile Analyzer Reference Guide for details.

CHAPTER 3. CONFIGURING VISUAL STUDIO CODE TO USE DEPENDENCY ANALYTICS

You can gain access to Red Hat's Trusted Profile Analyzer service by using the Dependency Analytics extension for Microsoft's Visual Studio Code (VS Code) editor application. With this extension you get access to the latest open source vulnerability information, and insights about your application's dependent packages. The Red Hat Dependency Analytics extension uses the following data sources for the most up-to-date vulnerability information available:

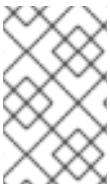
- The ONGuard service, integrates the Open Source Vulnerability (OSV) and the National Vulnerability Database (NVD) data sources. When given a set of packages to the ONGuard service, a query to OSV retrieves the associated vulnerability information, and then a query to NVD for public Common Vulnerability and Exposures (CVE) information.

Dependency Analytics supports the following programming languages:

- Maven
- Node
- Python
- Go

IMPORTANT

Visual Studio Code by default, executes binaries directly in a terminal found in your system's **PATH** environment. You can configure Visual Studio Code to look somewhere else to run the necessary binaries. You can configure this by accessing the extension settings. Click the **Workspace** tab, search for the word *executable*, and specify the absolute path to the binary file you want to use for Maven, Node, Python, or Go.



NOTE

The Dependency Analytics extension is an online service maintained by Red Hat. Dependency Analytics only accesses your manifest files to analyze your application dependencies before displaying the results.

Prerequisites

- Install Visual Studio Code on your workstation.
- For Maven projects, analyzing a **pom.xml** file, you must have the **mvn** binary in your system's **PATH** environment.
- For Node projects, analyzing a **package.json** file, you must have the **npm** binary in your system's **PATH** environment.
- For Go projects, analyzing a **go.mod** file, you must have the **go** binary in your system's **PATH** environment.
- For Python projects, analyzing a **requirements.txt** file, you must have the **python3**/**pip3** or **python**/**pip** binaries in your system's **PATH** environment. Also, the Python application needs to be in VS Code's interpreter path.

Procedure

1. Open the Visual Studio Code application.
2. From the file menu, click **View**, and click **Extensions**.
3. Search the **Marketplace** for *Red Hat Dependency Analytics*.
4. Click the **Install** button to install the extension. Wait for the installation to finish.
5. To start scanning your application for security vulnerabilities, and view the vulnerability report, you can do one of the following:
    - Open a manifest file, hover over a dependency marked by the inline Component Analysis, indicated by the wavy-red line under a dependency name, click **Quick Fix**, and click **Detailed Vulnerability Report**.
    - Open a manifest file, and click the **pie chart** icon.
    - Right click on a manifest file in the **Explorer** view, and click **Red Hat Dependency Analytics Report...**.
    - From the vulnerability pop-up alert message, click **Open detailed vulnerability report**

Additional resources

- Red Hat Dependency Analytics Visual Studio marketplace page.
- The GitHub project.

CHAPTER 4. CONFIGURING INTELLIJ TO USE DEPENDENCY ANALYTICS

You can gain access to Red Hat's Trusted Profile Analyzer service by using the Dependency Analytics plugin for Jet Brains' IntelliJ IDEA application. With this plugin you get access to the latest open source vulnerability information, and insights about your application's dependent packages. The Red Hat Dependency Analytics plugin uses the following data sources for the most up-to-date vulnerability information available:

- The ONGuard service, integrates the Open Source Vulnerability (OSV) and the National Vulnerability Database (NVD) data sources. When given a set of packages to the ONGuard service, a query to OSV retrieves the associated vulnerability information, and then a query to NVD for public Common Vulnerability and Exposures (CVE) information.

Dependency Analytics supports the following programming languages:

- Maven
- Node
- Python
- Go

NOTE

The Dependency Analytics extension is an online service maintained by Red Hat. Dependency Analytics only accesses your manifest files to analyze your application dependencies before displaying the results.

Prerequisites

- Install IntelliJ IDEA on your workstation.
- For Maven projects, analyzing a **pom.xml** file, you must have the **mvn** binary in your system's **PATH** environment.
- For Node projects, analyzing a **package.json** file, you must have the **npm** binary in your system's **PATH** environment.
- For Go projects, analyzing a **go.mod** file, you must have the **go** binary in your system's **PATH** environment.
- For Python projects, analyzing a **requirements.txt** file, you must have the **python3**/**pip3** or **python**/**pip** binaries in your system's **PATH** environment.

Procedure

1. Open the IntelliJ application.
2. From the file menu, click **Settings** , and click **Plugins**.
3. Search the **Marketplace** for *Red Hat Dependency Analytics*.
4. Click the **INSTALL** button to install the plug-in.

5. To start scanning your application for security vulnerabilities, and view the vulnerability report, you can do one of the following:

    - Open a manifest file, hover over a dependency marked by the inline Component Analysis, indicated by the wavy-red line under a dependency, and click **Detailed Vulnerability Report**.

    - Right click on a manifest file in the **Project** window, and click **Dependency Analytics Report**.

Additional resources

- Red Hat's Dependency Analytics Jet Brains marketplace page.

- The GitHub project.