Bulletproof Trust User's Guide

Dark Sky Technology

Bulletproof Trust

Contents

Proprietary and Confidential Statement

Document Version

This document is version 25.3.2.

Revision History

Summary

Date	User Guide Version	Bulletproof Trust Release	Comments
Apr 16, 2025	25.3.2	Merope+	Minor bug patch release
Apr 07, 2025	25.3.1	Merope	Major feature upgrade release
Mar 05, 2025	25.2.2	Leo+	Minor feature upgrade release
Feb 28, 2025	25.2.1	Leo	Major feature upgrade release
Jan 23, 2025	25.1.2	Indus+	Minor bugfix release
Jan 17, 2025	25.1.1	Indus	Minor feature upgrade release
Dec 18, 2024	24.12.2	Hercules+	Minor bugfix release
Dec 13, 2024	24.12.1	Hercules	Major feature upgrade release

Date	User Guide Version	Bulletproof Trust Release	Comments
Dec 05, 2024	24.11.2	Gemini+	Bugfix release
Nov 25, 2024	24.11.1	Gemini	Minor feature upgrade release
Nov 07, 2024	24.10.2	Fang+	Minor feature upgrade release
Oct 31, 2024	24.10.1	Fang	Minor feature upgrade release
Oct 11, 2024	24.9.2	Electra+	Minor feature upgrade release
Sept 10, 2024	24.9.1	Electra	Minor feature upgrade release
Aug 16, 2024	24.4.4	Diadem	Major feature upgrade release
May 31, 2024	24.4.3	Calypso	Minor point release
May 09, 2024	24.4.2	Bellatrix	Minor point release
Apr 30, 2024	24.04.0001	Auriga	Updated for Auriga release with new feature details and change notes
Oct 27, 2023	23.10.0027	Andromeda	Additional Alert Model configuration details and example model code
Sept 13, 2023	23.09.0013	Andromeda	Added API tools section
Jul 25, 2023	23.07.0025	Andromeda	Initial version

See release notes for details.

Overview

Bulletproof Trust is an advanced software assurance and intelligence platform designed to predict and prevent cyber vulnerabilities in open-source. The data is made available both as a hosted platform available from Dark Sky Technology and also an on-premises deployment for disconnected or restricted networks.

Concepts and Core Ideas

Bulletproof Trust provides software assurance data about `Packages` and `Contributors`. A `Package` is any piece of software (or hardware, in the case of HDLs), firmware, or data that can be referenced uniquely by a Package URL (PURL). Bulletproof Trust uses the PURL as its standard reference format. `Contributors` refer to any unique entity which contributes to or exerts influence over a `Package`. These are typically individuals working in open-source technology, however it can also be groups sharing accounts, automated agents (bots), etc. A logical grouping of `Packages` is called a `Stream`. `Streams` can be created with an existing software bill of materials (SBOM) or by individually specifying `Packages`. One or more `Streams` make up a `Project`. An `Alert Model` exists for both `Packages` and `Contributors` and is customized to your deployment. These `Alert Models` are used to create informational, caution, and warning alerts against `Packages` and `Contributors`. These alerts also impact the associated risk score for `Packages` and `Contributors`. These alerts can be customized for and are rolled up in to `Streams` and `Projects` allowing you to visualize, understand, and make quick decisions related to the riskiness of the `Packages` used in your software supply chain. Additionally, Bulletproof Trust includes an SBOM Vault feature which is a bit-for-bit encrypted storage system designed specifically for SBOMs. It allows users to create multiple encrypted vaults, which they can choose to share for collaboration or keep entirely private. Within each vault, every SBOM is revision tracked with details like the uploader and revision note, ensuring complete historical accountability. Additionally, SBOMs can be linked to an analysis `Stream` for in-depth risk analysis, and the vaults themselves can be encrypted and exported when needed, providing a robust solution for secure and flexible management of your SBOMs.

Getting Started

Bulletproof Trust provides both a web-based dashboard and an API for accessing software assurance data. The dashboard provides a way for you to upload SBOMs, create and manage projects, create and manage streams (a new feature in Bellatrix, discussed below), and visualize alerts and information about packages and other data presented via the API, and creates at-a-glance views for the packages

and projects which are being tracked. The API can be accessed directly using a JSON web token (JWT) generated via the dashboard for integration with automated CI/CD or risk assessment systems.

Getting an Account

Bulletproof Trust Managed

To get an account for the Bulletproof Trust Managed platform hosted by Dark Sky Technology, please contact your Dark Sky point of contact.

Bulletproof Trust On-premises

Bulletproof Trust deployments are managed via the `bptctl` tool. This tool is available inside the Bulletproof Trust deployment environment. Please contact your system administrators to have them create an account on your Bulletproof Trust deployment.

Using the Dashboard

The Dark Sky Technology managed dashboard can be accessed at https://bpt.darksky.technology. For on-premises installations, please contact your system administrators to get the correct URL to the dashboard.

When logging into the dashboard, you are presented with two major sections of a single view – the sidebar on the left and the current view on the right. The sidebar is always present and contains a 'My Workspace' button as well as a list of projects. Clicking the 'My Workspace' button will return you to the My Workspace view no matter where you are in the interface. The current view on the right is initially loaded with statistics and visualizations that pertain to all open-source software analyzed by Bulletproof Trust. You will later find project-specific, stream-specific, and package-specific visualizations depending on where you have navigated in the interface.

My Workspace

When logging into the dashboard, you are presented with your workspace view, labeled "My Workspace".This is initially blank, but will fill in with insights about your Projects as you add data. Additionally, you will find a Projects tile and SBOM Vaults tile below. These are both initially blank.

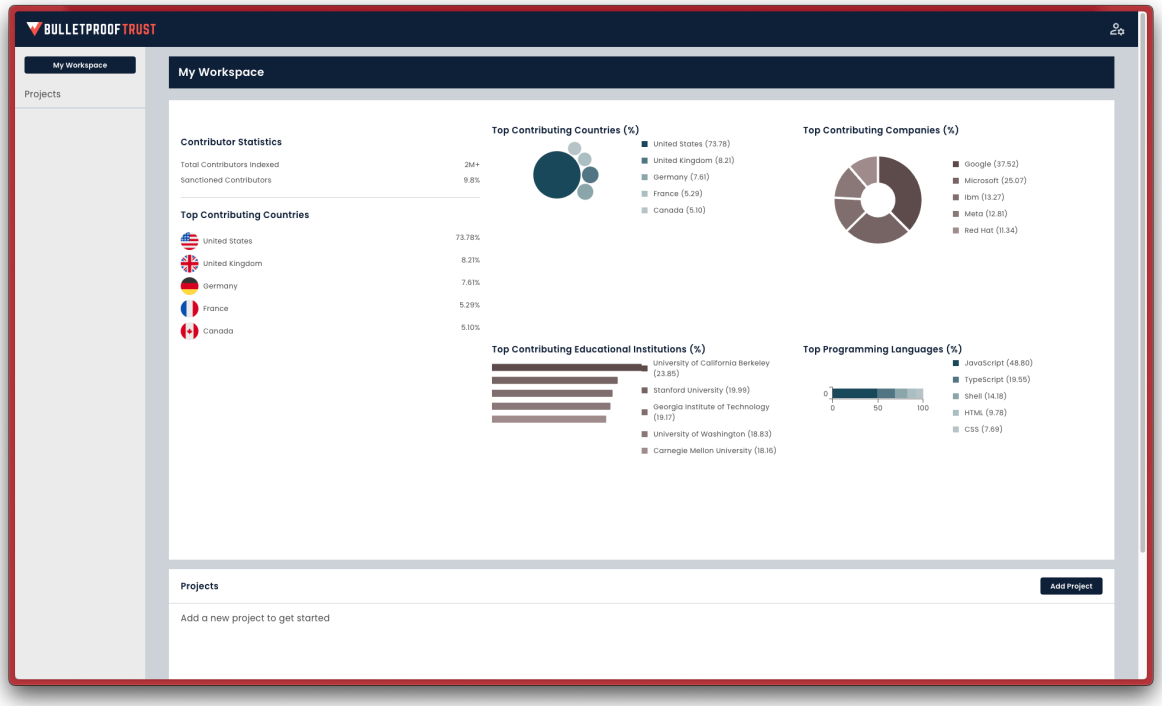


Figure 1: My Workspace view.

After creating a new or loading an existing project, you will see a high level status of each of the streams and packages inside those projects. The color of each project in the summary view and sidebar indicate a warning (red), caution (yellow), informational (grey), or good (green) status. This color scheme is used throughout the web interface to represent the status of streams, packages, contributor risk, etc.

Projects

A project is used to manage streams, which in turn manage packages and SBOMs. Every project contains one or more streams and shows up in your Workspace view. Projects can be created from scratch, loaded using an existing SBOM, or loaded from an existing project that either you own or is shared with you by another Bulletproof Trust user.

Creating a New Project

To add a new project to your workspace, click on 'Add Project' in the top right corner of the Projects tile in your Workspace view.

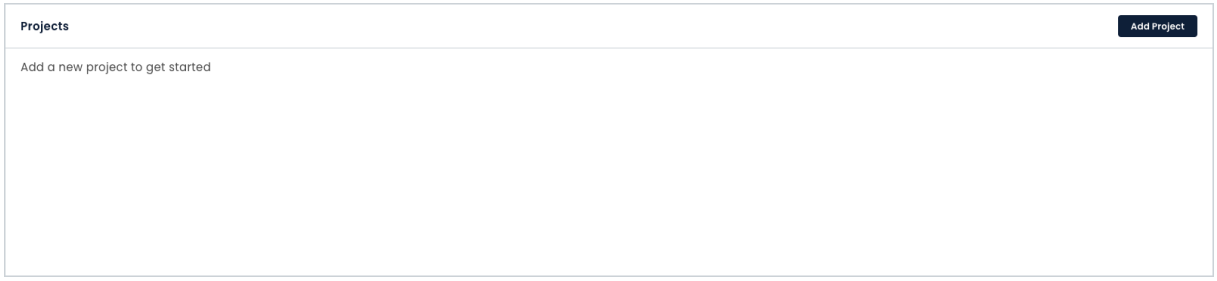


Figure 2: Adding a new project.

You will be presented with a search bar at the top of your view, allowing you to search for an existing project or create a new project.

Figure 3: Search bar for adding an existing or creating a new project.

Under the search bar, click 'Create new project'. This will display an 'Add New Project' dialogue, allowing you to create a new project from scratch or create a new project using an existing SBOM.

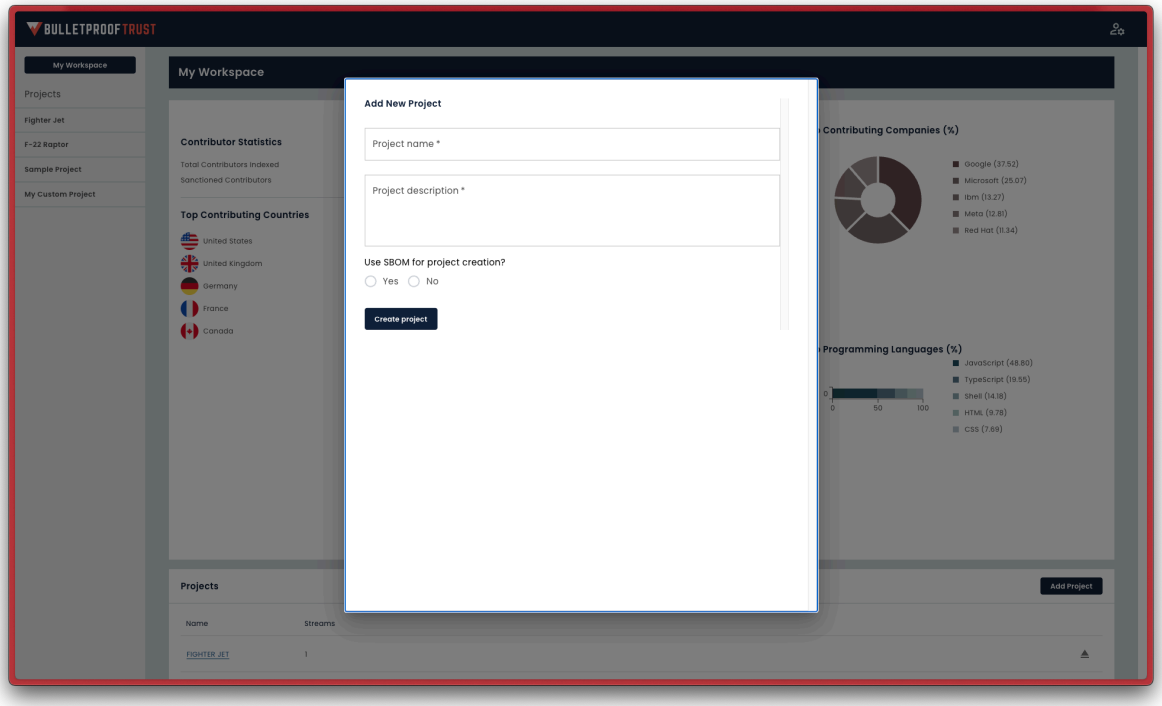


Figure 4: Add New Project dialogue for creating a new project.

Creating a New Project from an Existing SBOM

The most simple way to create a new project is to use an existing SBOM. Simply add a project name and give your project a description (you can use simple Markdown for the Project description text box), click 'Yes' under the "Use SBOM for project creation?" text, then click 'Upload SBOM'. You will be presented with your standard file browser. Navigate to and select a properly formatted SBOM file on your computer. CycloneDX JSON and CycloneDX XML are the supported SBOM types. Once the file is selected, you will see that filename next to the 'Upload SBOM' button with a small red x. To remove that SBOM and select a different one, click the red x and go through these steps again. Otherwise, click 'Create project'.

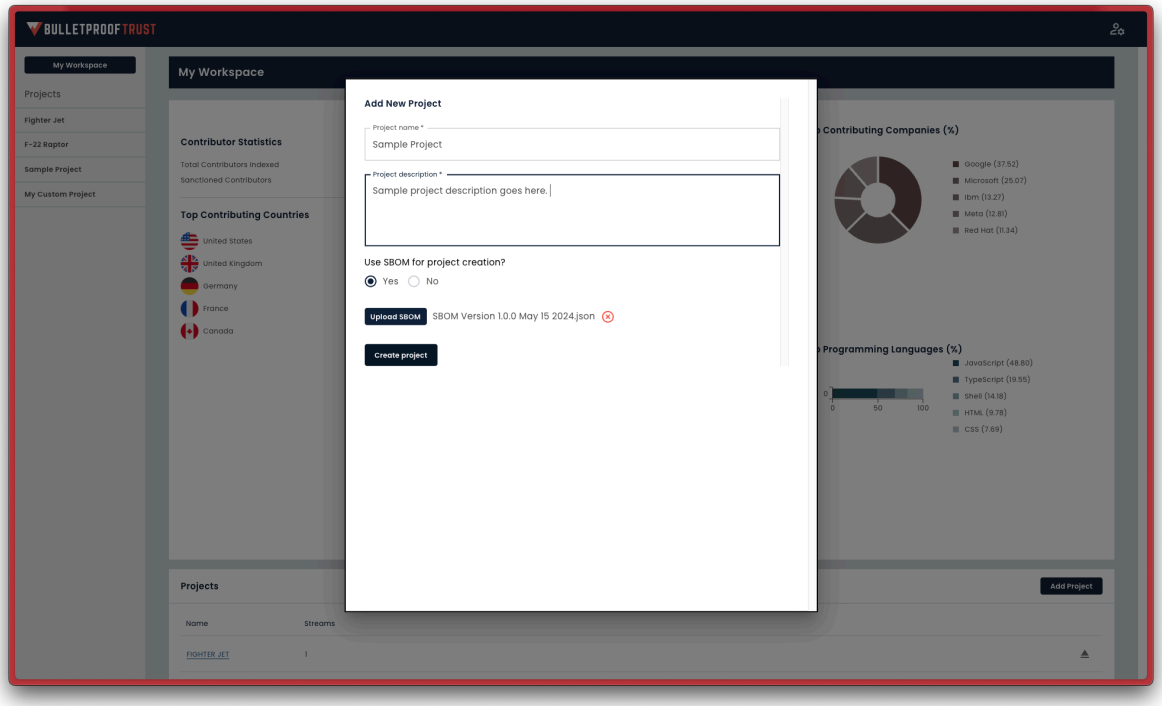


Figure 5: Add New Project dialogue for creating a new project from an SBOM.

Successful creation of a project using an existing SBOM will create a 'default' stream, which contains all of the packages in the SBOM. This stream can be viewed in the sidebar when the project is selected, and will show up in the Streams tile towards the bottom of the projects view.

Creating a New Project from Scratch

If you do not have an SBOM, enter your project name and project description, then choose 'No' under the "Use SBOM for project creation?" text. You will be presented with a few additional fields. First, enter your Project Package Url (PURL). This must be a properly formatted PURL (see this GitHub URL for the PURL specification). Next choose your Repository Type from the drop-down menu. Note that a repository is required for Bulletproof Trust to perform it's collection and analysis. You can select between Git, Subversion (SVN), Mercurial, CVS, and Bazaar. Finally, enter your Repository Url and click 'Create Project'.

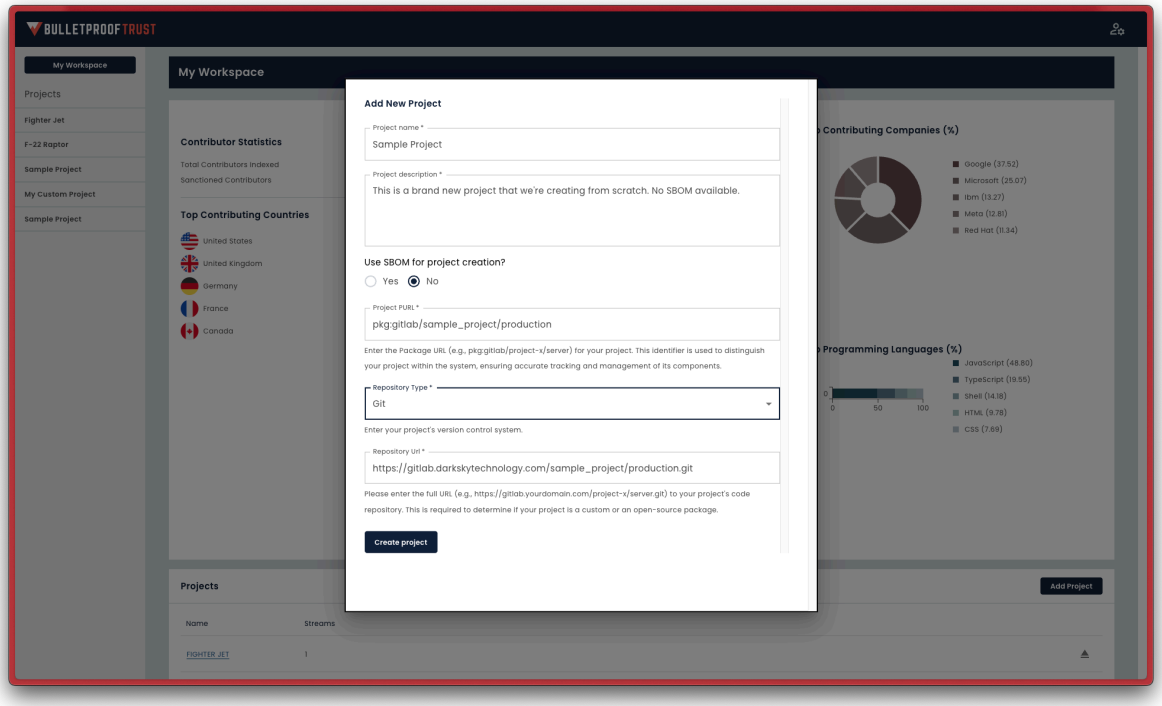


Figure 6: Add New Project dialogue for creating a new project from scratch.

Successful creation of a project from scratch will create a 'default' stream with no packages. The stream can be viewed in the sidebar when the project is selected, and will show up in the Streams tile towards the bottom of the projects view.

Regardless of whether you created a new project using an existing SBOM or from scratch, if the project was created successfully, you will see a green bar along the top of your browser window with the text "Project created!". If there were issues with the format of the SBOM or other issues creating the project, you will see a red bar at the top of your browser window with a text description of the error. Contact your Dark Sky Technology technical support with a screenshot of the error if you have questions or cannot resolve the issue. You can close the green or red project creation status bar by clicking the 'x' on the far right of the bar.

Adding an Existing Project

Just like creating a new project, you can add an existing project in your workspace by clicking on 'Add Project' in the top right corner of the Projects tile in your Workspace view.

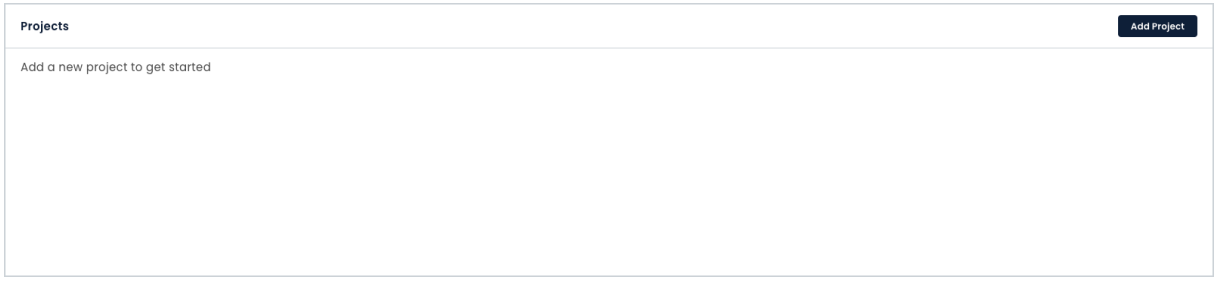


Figure 7: Adding a new project.

You will be presented with a search bar at the top of your view, allowing you to search for an existing project or create a new project.

Figure 8: Search bar for adding an existing or creating a new project.

If you already own a project or if someone else has shared a project with you that is not in your workspace, you can search for it in this search bar. A drop-down list of projects that match your search query will display. Click the project you would like to add to your workspace to have it immediately added to the sidebar and your projects tile in your workspace view below.

Projects View

After creating a new or adding an existing project, you can navigate to that specific projects view by clicking the name of the project in the sidebar or using the project link in the Projects tile on your My Workspace view. This will present you with your projects view page and will highlight that project name in the sidebar while also showing any streams (and their status) contained in the project under the project name highlighted in the sidebar.

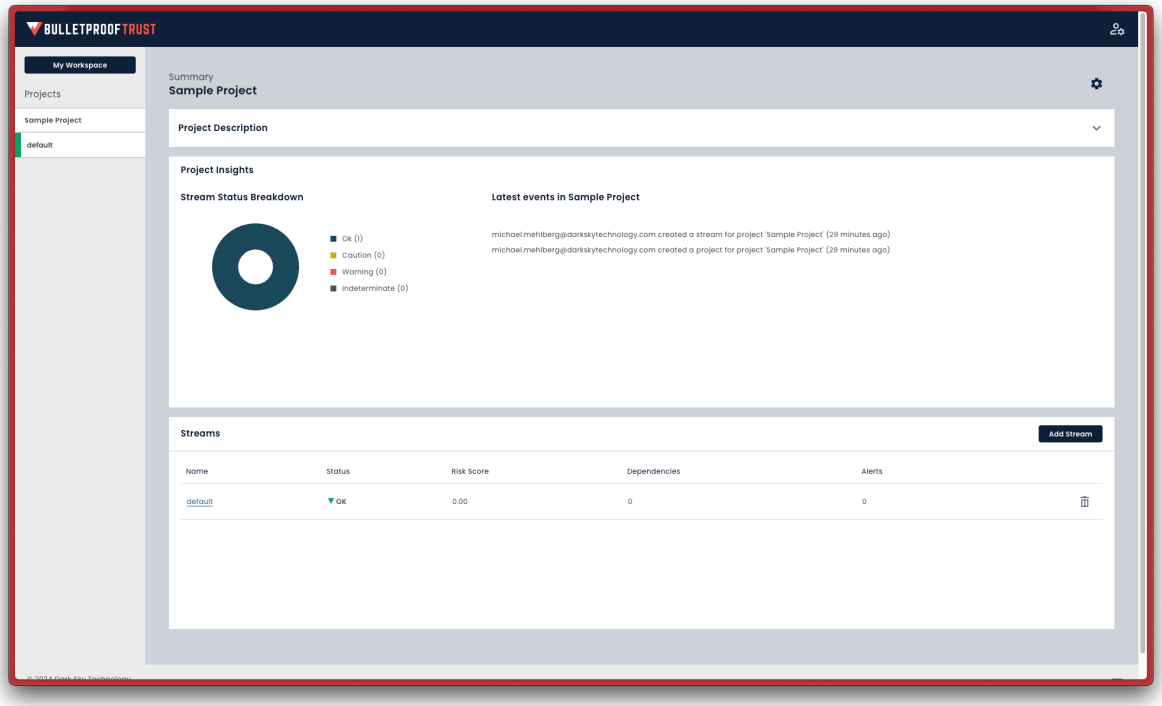


Figure 9: Projects view.

The project view is broken up into multiple tiles. The Project Description tile contains the description entered while adding a new project. The project description is initially hidden, but can be viewed by clicking the small grey down arrow at the far right of the Project Description tile. The Project Insights tile contains a visualization showing the status of all streams contained in the project as well as the latest events in the project event log. Finally, the Streams tile at the bottom of the project view contains a list of all project streams. Unless you've imported a shared project with different streams or added or renamed the default stream in your project, there will be on default stream labeled 'default' in your project.

Project Settings

Every project contains project settings, which can be accessed by clicking on the blue gear icon in the upper right hand corner of the project view page. Project settings allow you to change the project name, modify the project description, edit user permissions, view the project event log, and delete the project. To return to the project view page, either click the back button in your browser, navigate to your project using the sidebar, or click the project name at the top of the project settings page.

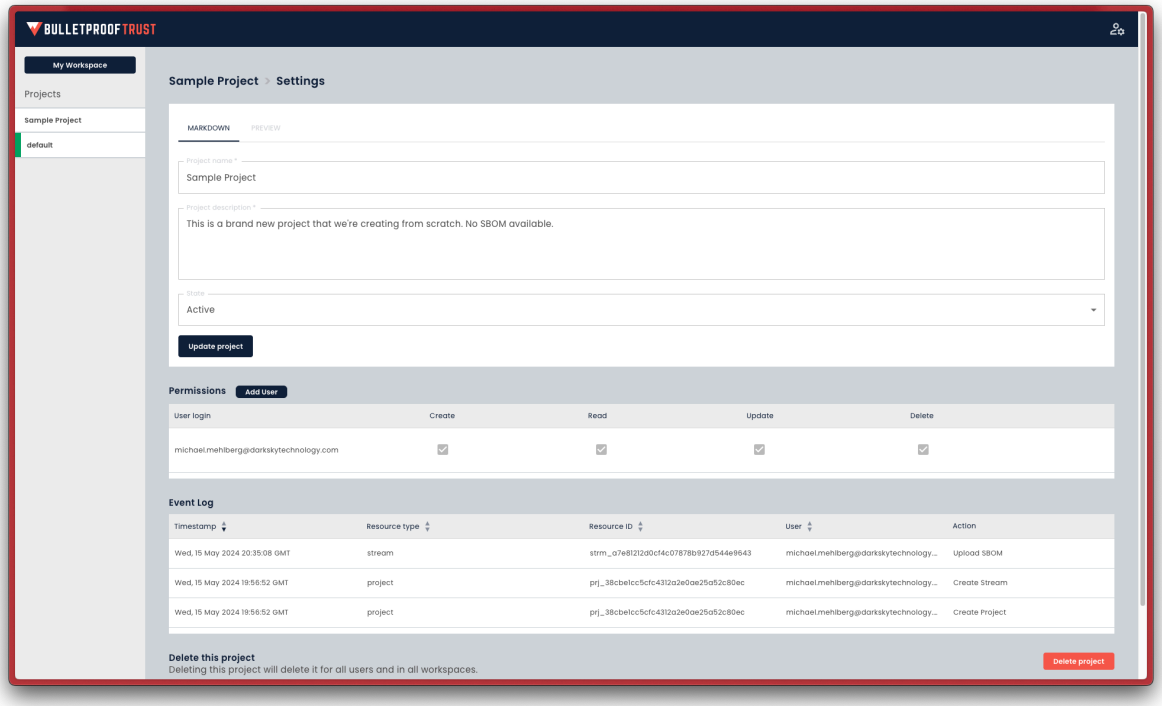


Figure 10: Projects settings.

Project Name, Description, and State

The project name field is a simple text field that contains the current name of the project. The project description field is a markdown text field that contains the current markdown description of the project. Both of these fields can be updated and the rendered markdown can be viewed by clicking the 'Preview' tab at top of this tile. Finally, the state of the project can be changed from Active to Archived if you would like to save the project but remove it from others views. Any updates made to the project name, project description, or project status can be saved by clicking the 'Update project' button at the bottom of this tile.

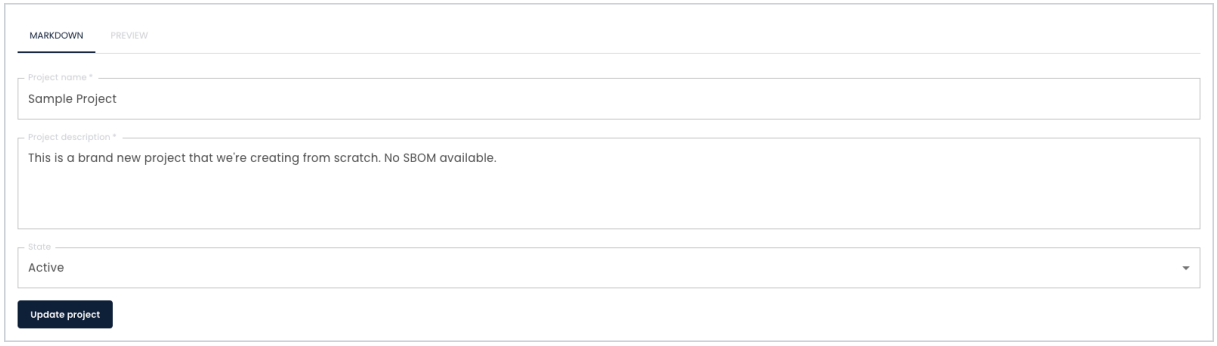


Figure 11: Updating the project name, description, and state settings.

Project Permissions

The Permissions tile contains a list of all users in the project who have create, read, update, or delete privileges for that project. The users login username will be displayed (usually an email address) along with their specific privileges for the project, which can be granted or removed by the project owner. Granting create privileges will allow that user to create new streams within the project. Granting read privileges will allow that user to view that project. Granting update privileges will allow that user to update project settings. Granting delete privileges will allow that user to delete the project for everyone (which is distinctly different than their privilege to eject that project from their workspace, which removes the project from their view while keeping the project available for other users).

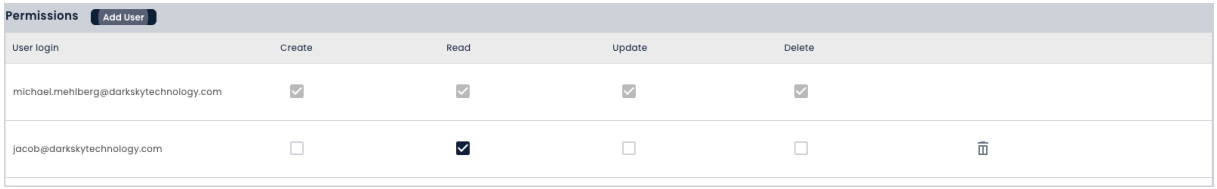


Figure 12: Project permissions

To add a new user to the project, click the 'Add User' button above the Permissions tile. This will drop down a 'Search for a user…' dialogue at the top of the window, allowing you to search for other users who have permission to add that project by username/email address. Select the appropriate user and click 'Add' to add them to the project. By default, the new user will be granted read privileges.

Figure 13: Projects settings.

To delete a user from the project, click the garbage can icon to the far right of the row for that user.

Project Event Log

The project event log is a list of dated and time-stamped events that have happened in the project itself, any of the streams contained in the project, or any of the packages contained in any of the project streams. For example, if an SBOM is uploaded to the project or a new stream is created, these events will be captured in the project event log.

Figure 14: Projects event log.

Events can be sorted by time of event, resource type (stream, project, package, etc.), resource ID (a unique identifier for that resource event), or username. Events in the log cannot be deleted, making for an immutable list of events carried out by users granted privileges to the project.

Deleting a Project

To permanently delete the project, click the red 'Delete Project' in the bottom right hand corner of the project settings page. As the text states, deleting a project will delete it for all users in all workspaces.

Figure 15: Deleting a project.

Clicking this button will bring up a confirmation dialogue, warning you that you are about to delete this shared project for everyone, immediately removing it from your and their workspace views. click the red 'Yes, delete project for everyone' button if you wish to do so. Otherwise, click 'Cancel' to go back to the project settings page.

Streams

Every project contains one or more streams, which are used to manage SBOMs and packages. A stream can be thought of as a branch in a project. For example, you may have one stream for development,

another for testing, another for a specific deployed version of your software, etc. Streams can be loaded from an existing SBOM, or created from scratch.

Adding a New Stream

To add a new stream from an existing SBOM, navigate to the project in which you wish to create your stream using the sidebar or your My Workspace view. From the project view, click 'Add Stream' in the Streams tile at the bottom of your project view.

Figure 16: Add a new stream to your project.

This will bring up an "Add New Stream" dialogue, where you will be able to add a new stream from an existing SBOM, or add a new stream from scratch.

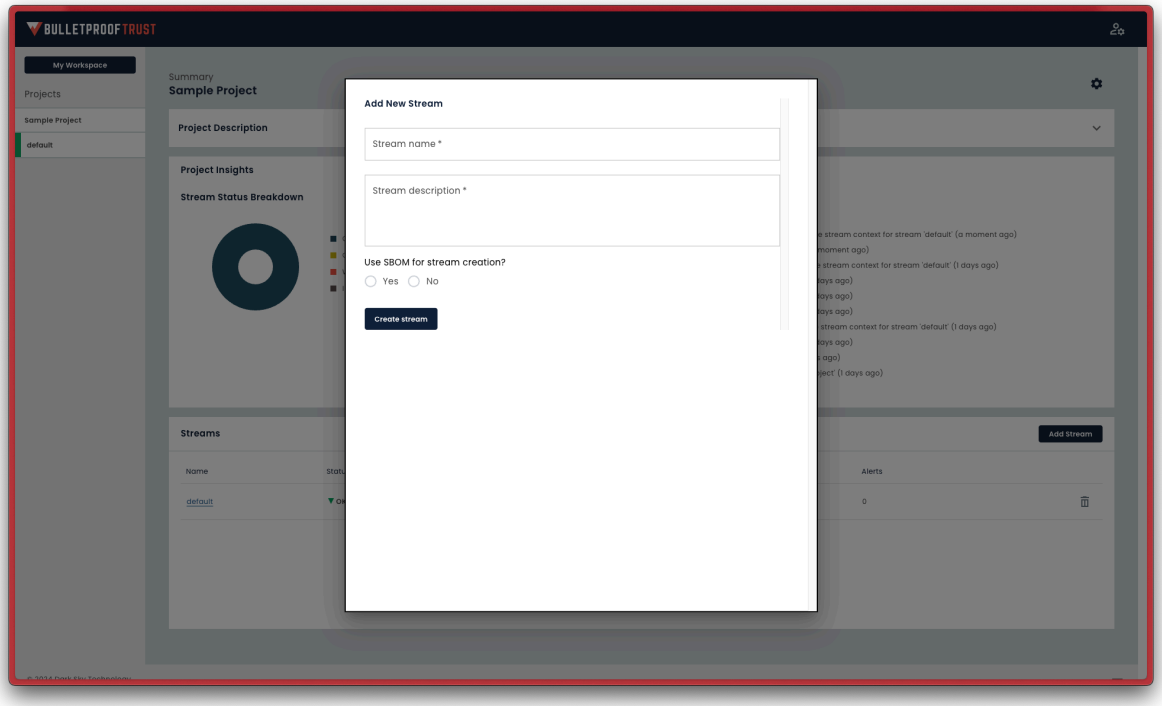


Figure 17: Add a new stream dialogue.

Adding a New Stream from an Existing SBOM

If you have an existing SBOM, you can add a new stream using that SBOM in the "Add New Stream" dialogue. Simply add your stream name, enter your stream description (you can use simple Markdown for the Stream description text box), click 'Yes' under the "Use SBOM for stream creation?" text, then click 'Upload SBOM'. You will be presented with your standard file browser. Navigate to and select a properly formatted SBOM file on your computer. CycloneDX JSON and CycloneDX XML are the supported SBOM types. Once the file is selected, you will see that filename next to the 'Upload SBOM' button with a small red x. To remove that SBOM and select a different one, click the red x and go through these steps again. Otherwise, click 'Create stream'.

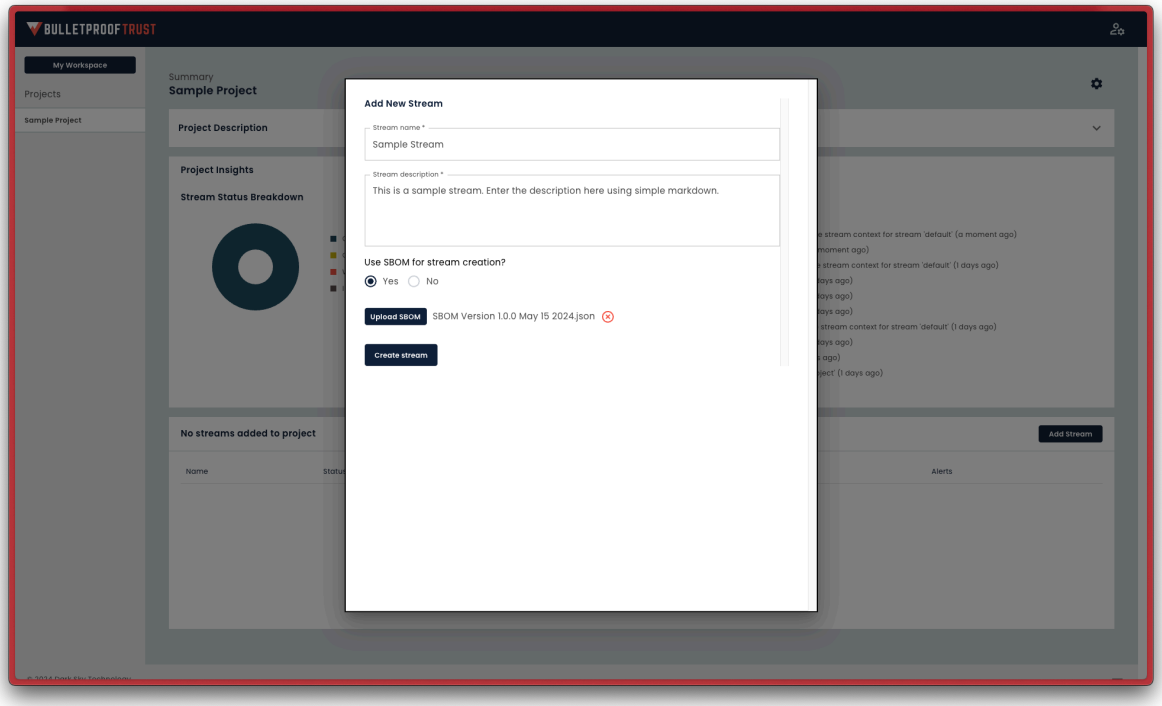


Figure 18: Add New Stream dialogue for creating a new stream from an SBOM.

Successful creation of a stream using an existing SBOM will create a new stream by the stream name provided, which contains all of the packages in the SBOM. This stream can be viewed in the sidebar when the project is selected, and will show up in the Streams tile towards the bottom of the projects view.

Adding a New Stream from Scratch

If you do not have an SBOM, enter your stream name and stream description, then choose 'No' under the "Use SBOM for stream creation?" text. You will be presented with a few additional fields. First, enter your Stream identifier. This must be a properly formatted PURL (see this GitHub URL for the PURL specification). It will contain default text based on the project name. You can accept this text unmodified, or change it to your liking. Next enter your repository tag in the "Repository Tag" field. This tag is used in conjunction with the URL for the project (listed above) to map this stream to a tag, branch, or reference in your version control system.

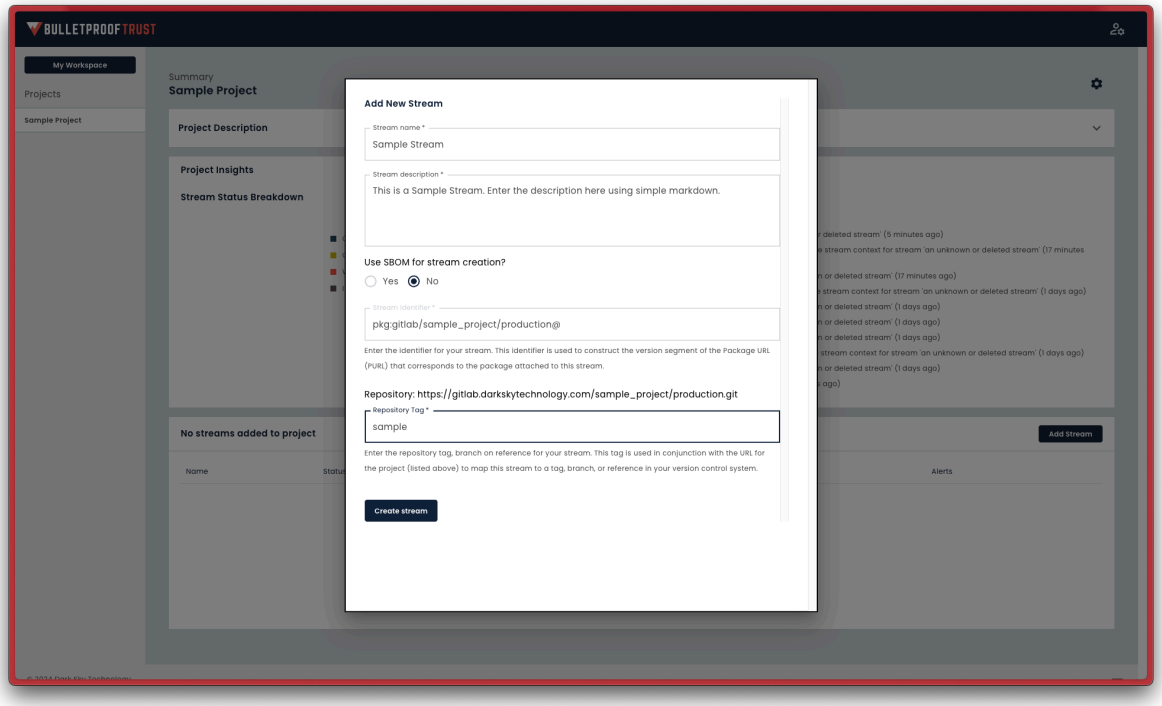


Figure 19: Add New Project dialogue for creating a new project from scratch.

Successful creation of a project from scratch will create a new stream using the provided stream name with no packages. The stream can be viewed in the sidebar when the project is selected, and will show up in the Streams tile towards the bottom of the projects view.

Regardless of whether you created a new stream using an existing SBOM or from scratch, if the stream was created successfully, you will see a green bar along the top of your browser window with the text "Stream created!". If there were issues with the format of the SBOM or other issues creating the stream, you will see a red bar at the top of your browser window with a text description of the error. Contact your Dark Sky Technology technical support with a screenshot of the error if you have questions or cannot resolve the issue. You can close the green or red stream creation status bar by clicking the 'x' on the far right of the bar.

Streams View

After creating a new stream, you can navigate to that specific streams view by navigating to the project that contains the stream (either through the sidebar or your My Workspace view), then click the stream name under the project name in the sidebar or in the Streams tile at the bottom of the project view. This will present you with your streams view page and will highlight that stream name in the sidebar.

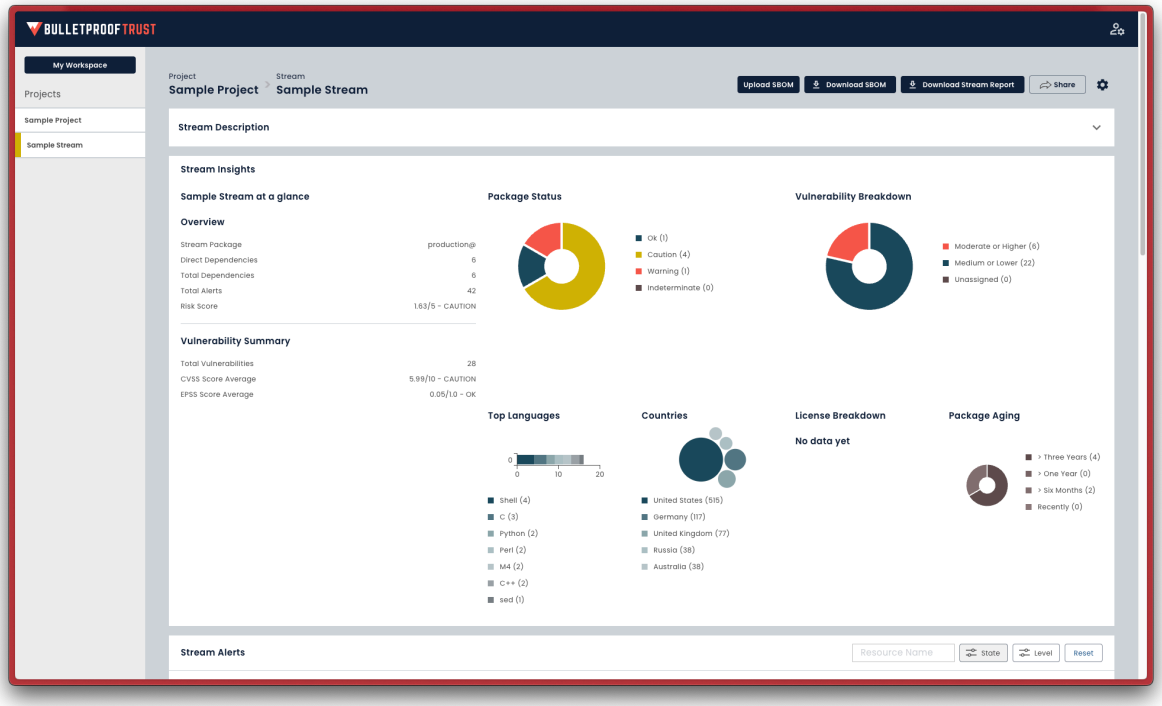


Figure 20: Streams view.

Like the project view, the streams view is broken up into multiple tiles. The Stream Description tile contains the description entered while adding a new stream. The stream description is initially hidden, but can be viewed by clicking the small grey down arrow at the far right of the Stream Description tile. The Stream Insights tile contains overview information about that stream, a vulnerability summary of the stream, a pie chart showing the breakdown by package status, a pie chart with a vulnerability severity breakdown, a chart showing the top programming languages used by all packages in the stream, a chart showing the top countries represented by all the contributors across all the packages in the stream, a breakdown of licenses used across all packages in the stream, and the age of all the packages in the stream.

Stream Alerts

Below the stream description, streams overview, and visualizations is the Stream Alerts tile. This tile contains a list of all package alerts in the stream, based on your specific alert profile.

Stream alerts can be filtered by their name by using the "Resource Name" text entry box near the top right hand corner of the tile. Stream alerts can also be filtered by state (active or dismissed) or level (critical, high, medium, low, or informational). If you wish to reset all filters back to their default status,

simply click the 'Reset' button in the upper right of the tile.

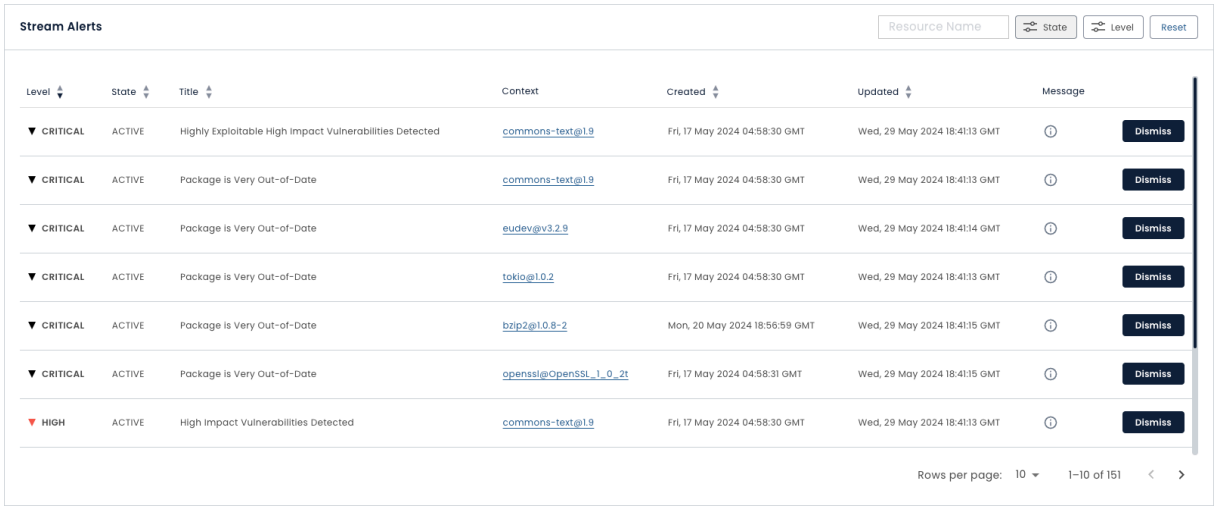


Figure 21: Stream Alerts tile in the streams view.

Stream alerts can be sorted by state (active or dismissed), level (critical, high, medium, low, or informational), alert title, context (the package in which the alert fired), date created, and date updated. There is also an informational icon in the message column with more details about that specific alert. Finally, you can dismiss the alert by clicking the 'Dismiss' button on the far right row of the alert you wish to dismiss.

The maximum number of alerts displayed in the tile is based on the "Rows per page" number in the bottom right of the tile (the default is 10). If you wish to display more than 10 alerts at a time, you can click the number of alerts displayed and select the number you wish to have displayed from the dropdown. If there are more alerts than can be displayed at once, you can page through the alerts by clicking the right or left arrows in the lower right corner of the tile.

Stream Vulnerabilities

Stream vulnerabilities can be viewed in the Vulnerabilities tile. This tile contains a list of all vulnerabilities from all packages in the stream.

Figure 22: Vulnerabilities tile in the streams view.

Vulnerabilities can be filtered by their name using the "Package Name" text entry form near the upper right hand corner of the tile. Vulnerabilities can also be filtered by their severity (critial, high, medium, low, or informational). If you wish to reset all filters back to their default status, simply click the 'Reset' button in the upper right of the tile.

Vulnerabilities can be sorted by severity (critical, high, medium, low, or informational), CVSS Score, EPSS Score, package name, identifier (e.g., the CVE number and link to the original vulnerability disclosure), and the date. Additional details can be found in the Details column by hovering over the little circle "i" in the row for that vulnerability.

The maximum number of vulnerabilities displayed in the tile is based on the "Rows per page" number in the bottom right of the tile (the default is 10). If you wish to display more than 10 vulnerabilities at a time, you can click the number of vulnerabilities displayed and select the number you wish to have displayed from the dropdown. If there are more vulnerabilities than can be displayed at once, you can page through the vulnerabilities by clicking the right or left arrows in the lower right corner of the tile.

Stream Packages

Packages contained in the stream can be viewed from the Packages tile at the bottom of the streams view. This tile contains a list of all packages in that stream. If you've added a new stream from an existing SBOM, the packages in that SBOM will show up in this tile. If you have created a new stream from scratch, this tile will be blank.

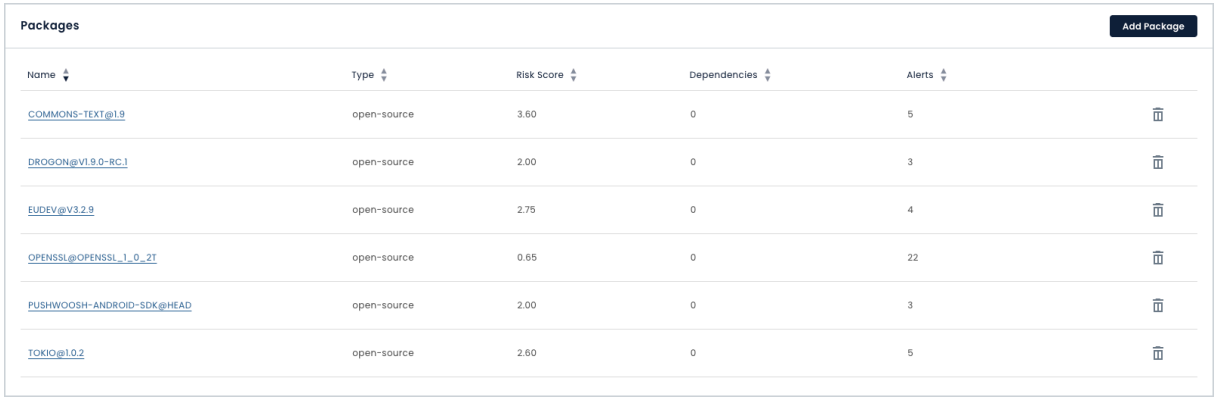


Figure 23: Packages tile in the streams view.

Stream packages can be sorted by name, type, risk score, number of dependencies, or number of alerts. Details on how to add and delete packages can be found in the Packages section below.

Stream Reports

A stream report is a PDF representation of what is displayed in the streams view. Stream reports are useful to share with others who may not have access to Bulletproof Trust or your specific project. They are also useful for archival purposes. Stream reports contain overall information about the stream, packages, alerts discovered in all packages in the stream, and vulnerabilities discovered across all packages in the stream. A stream report can be downloaded by clicking the "Download Stream Report" button near the top right of the stream view. Once clicked, you will see a notification "Generating report - this could take a few moments…". Do not navigate away from this stream view while the report is generating. Once finished, you will be presented with a link to download the PDF report to your local computer.

Stream Embeds

Stream embeds are small snippets of code that represent the overall trust status of the stream. They can be useful in development environments or custom dashboards where you need a quick, at-a-glace view of an individual stream. To copy the code necessary to embed this stream overview into a custom dashboard, simply click the 'Share' button in the upper right hand corner of the stream view. You will be presented with a dialogue, giving you choice between markdown or HTML code snippet. Copy the code snippet by selecting the code in the appropriate text box or by clicking 'Click to Copy'. This will place that code snippet in your clipboard for you to paste elsewhere.

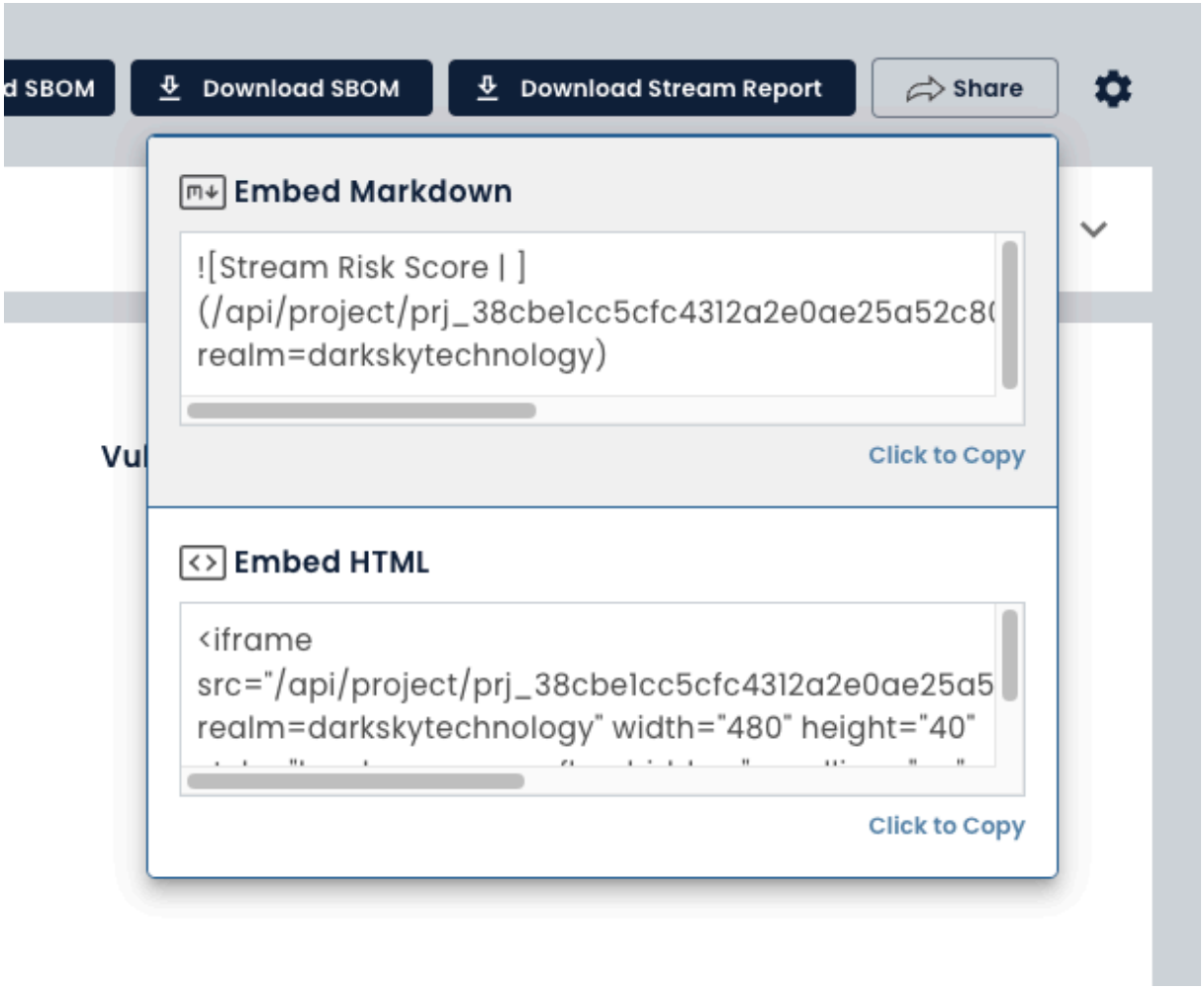


Figure 24: Embed streams dialogue.

Stream Settings

Every stream contains stream settings, which can be accessed by clicking on the blue gear icon in the upper right hand corner of the streams view page. Stream settings allow you to change the stream name, modify the stream description, or view the stream event log. To return to the stream view page, either click the back button in your browser, navigate to your stream using the sidebar, or click the stream name at the top of the stream settings page.

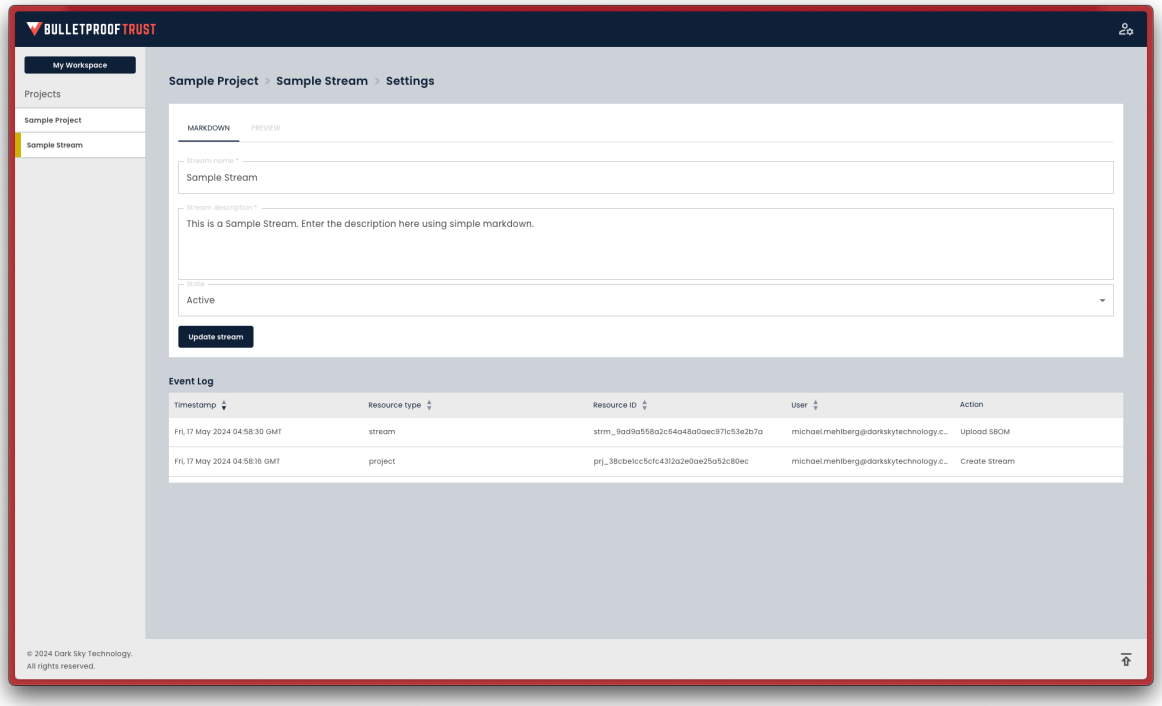


Figure 25: Stream settings.

Stream Name, Description, and State

The stream name field is a simple text field that contains the current name of the stream. The stream description field is a markdown text field that contains the current markdown description of the stream. Both of these fields can be updated and the rendered markdown can be viewed by clicking the 'Preview' tab at top of this tile. Finally, the state of the stream can be changed from Active to Archived if you would like to save the stream but remove it from others views. Any updates made to the stream name, stream description, or stream status can be saved by clicking the 'Update stream' button at the bottom of this tile.

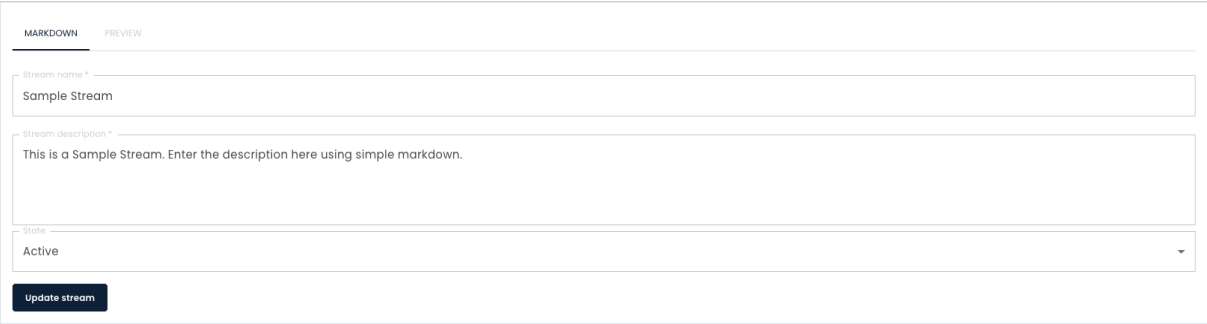


Figure 26: Updating the stream name, description, and state settings.

Stream Event Log

The stream event log is a list of dated and time-stamped events that have happened in the stream itself or any of the packages contained in the stream. For example, if an SBOM is uploaded to the stream, these events will be captured in the project event log.

Figure 27: Projects event log.

Events can be sorted by time of event, resource type (stream, project, package, etc.), resource ID (a unique identifier for that resource event), or username. Events in the log cannot be deleted, making for an immutable list of events carried out by users granted privileges to the stream.

Packages

Packages are open-source software libraries or third party libraries contained within a stream in one or more of your Bulletproof Trust projects. Every package is represented by a unique package URL (see this GitHub URL for the PURL specification, and is the basis for all package health, contributor risk, and vulnerability analysis performed in your projects and streams. While a package in a repository like GitHub will contain code and branches and issues, a package in Bulletproof Trust contains references to that repo along with information about the licenses used, issues, pull requests, popularity, contributors, sentiment, vulnerabilities, code quality, languages, etc. in that repo. This information, once analyzed by Bulletproof Trust, drives alerts based on your alert model, which inform you about the riskiness and trustworthiness of each packages, stream, and project.

Adding a New Package

You can add a new package to your stream one at a time, create a new package to add to your stream, or upload an existing SBOM with one or more packages to your stream. To start, navigate to the stream in which you wish to add your package.

Adding a New Package Manually

To add a new package one at a time, click 'Add Package' in the Packages tile at the bottom of your streams view.

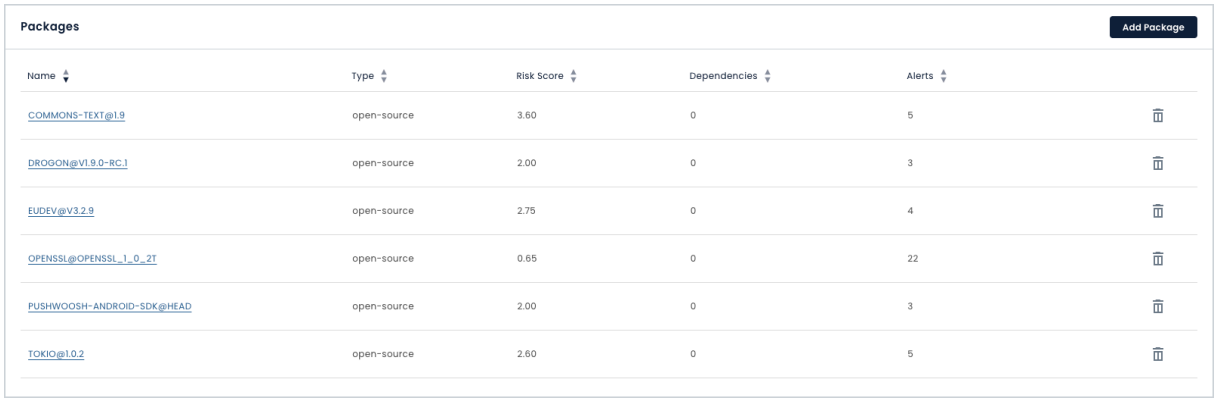


Figure 28: Add new package manually by clicking the 'Add Package' button

You will be presented with a search bar at the top of your view, allowing you to search for a package that Bulletproof Trust has previously analyzed.

Figure 29: Search bar for adding a package that Bulletproof Trust has analyzed.

In the "Search for a package…" text field, enter the name of a package Bulletproof Trust has previously analyzed. The search query can be either freeform text, or a label-based search corresponding to the components in the desired PURL. For example, if you wanted to find a library named `parse-json`, you could simply type this into the search bar and the top 100 results will be populated as you type. Alternatively, if you wanted to find a library named `accepts`, from the `npm` ecosystem with `1.x` version value you could type `name:accepts type:npm version:1.` into the search bar. Searching is done as you type and will show up as a drop-down list of matches. Click the package you wish to add to your stream, then click the 'Add' button.

Note: Adding a package manually can only be done with packages that have already been analyzed by Bulletproof Trust. To add a package that has not yet been analyzed by Bulletproof Trust, refer to the Bulletproof Trust API documentation below to anonymously request that a package be analyzed. That package can be added to your stream after the analysis is complete.

Creating a New Package

If you have proprietary software or 3rd party libraries that are not identifiable in a version control systems, you may want to track those components in your SBOM. To do that, you will create a new package. Just like adding a new package manually, click 'Add Package' in the Package tile at the bottom of your streams view.

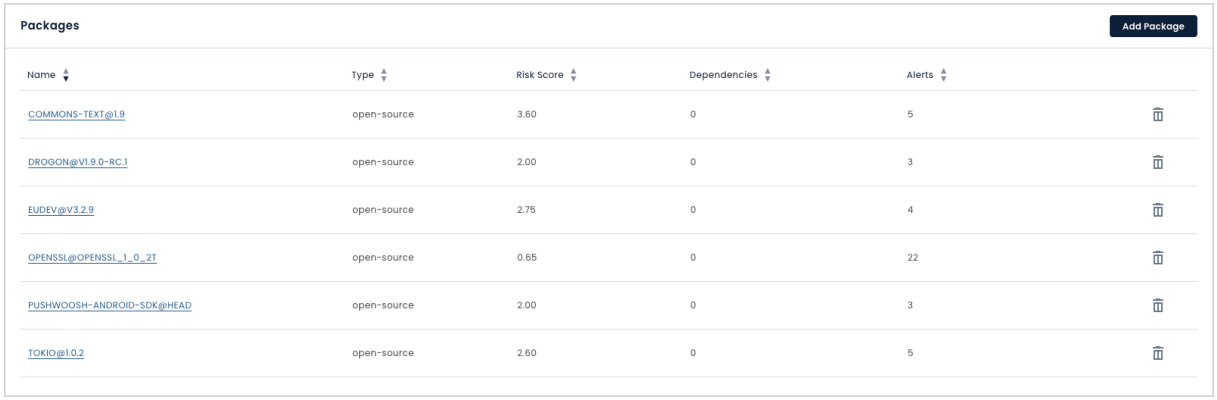


Figure 30: Create a new package to add to your stream by clicking the 'Add Package' button

You will be presented with a search bar at the top of your view. Below the "Search for a package…" text field, click the 'Create new package' link. This will bring up a dialogue where you can enter information about this custom package.

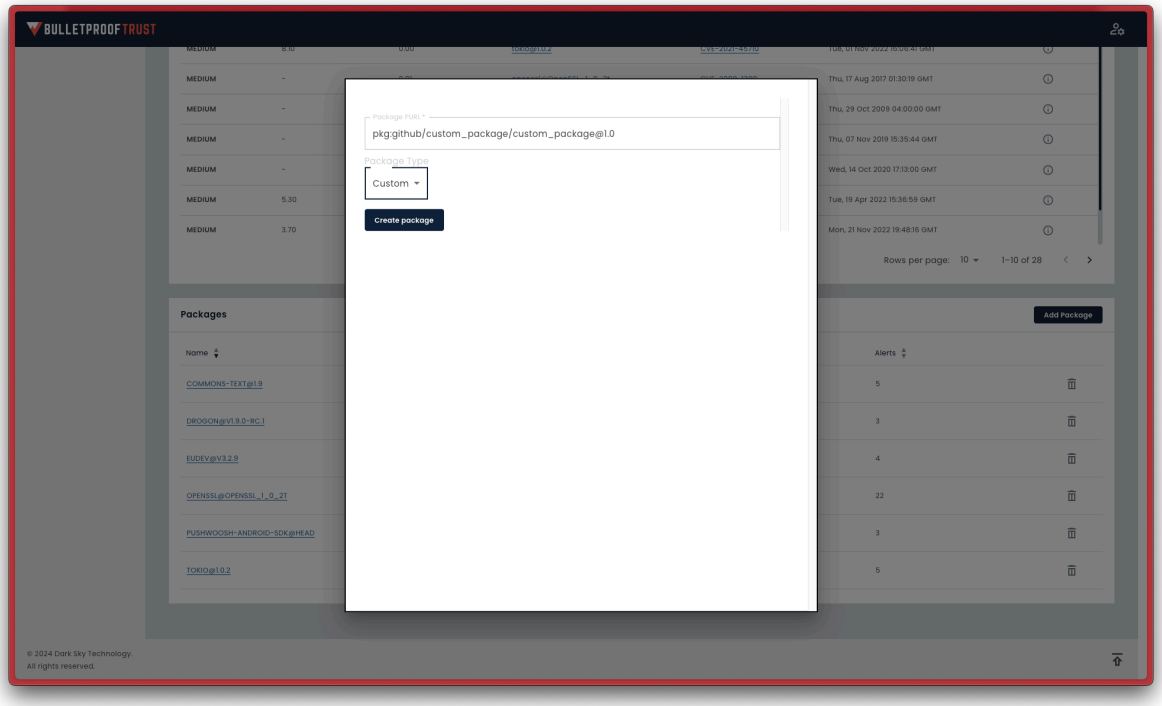
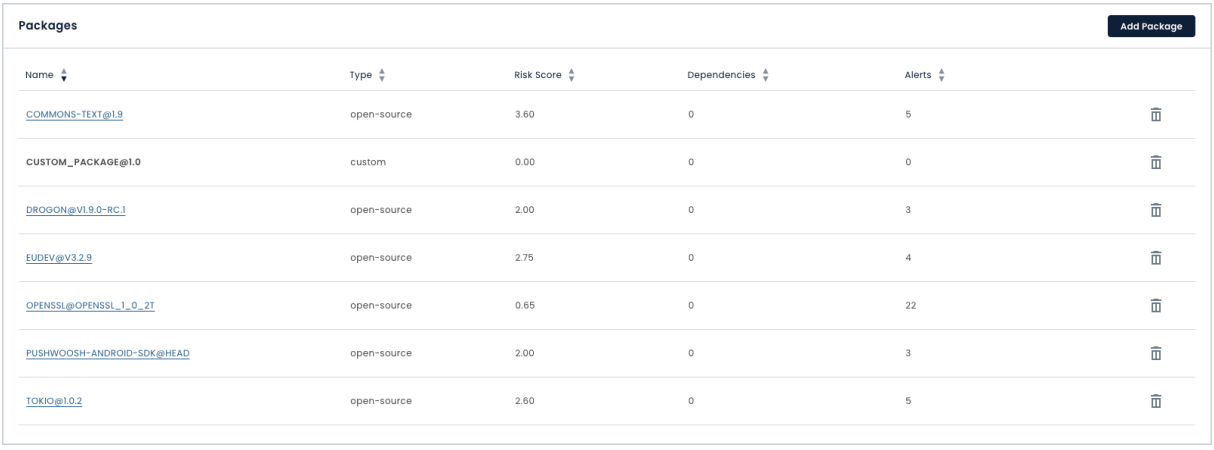


Figure 31: Creating a new package to add to your stream.

Enter a valid package URL (see this GitHub URL for the PURL specification, select the package type (either open-source or custom), then click 'Create package' This will add the new package to your Package tile in your stream view.



Name	Type	Risk Score	Dependencies	Alerts	
COMMONS-TEXT@1.9	open-source	3.60	0	5	
CUSTOM_PACKAGE@1.0	custom	0.00	0	0	
DROGON@V1.9.0-RC.1	open-source	2.00	0	3	
EUDEV@V3.2.9	open-source	2.75	0	4	
OPENSSL@OPENSSL_1_0_2T	open-source	0.65	0	22	
PUSHWOOSH-ANDROID-SDK@HEAD	open-source	2.00	0	3	
TOKIO@1.0.2	open-source	2.60	0	5	

Figure 32: New package added to the Packages tile in your stream view

Notice that the new package will not be hyperlinked, as no analysis can be performed on a package

that is not contained in a source code repository. That said, this new package will be maintained in your SBOM along with all other packages in the stream.

Adding Packages from an Existing SBOM

To add one or more packages to your stream from an existing SBOM, click the 'Upload SBOM' button at the top of your stream view. CycloneDX JSON and CycloneDX XML are the supported SBOM types.

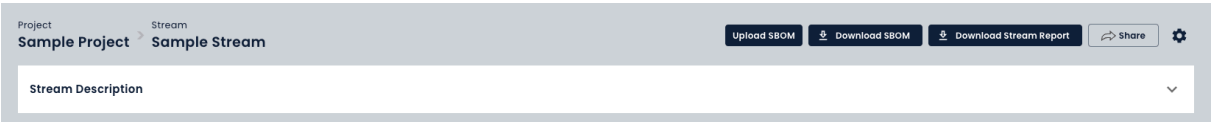


Figure 33: Upload SBOM button for loading in one or more packages from an existing SBOM.

This will bring up your operating systems standard file browswer. Browse to the location of your existing SBOM and upload it to Bulletproof Trust. While the upload is taking place, you will see the following dialogue displaying the SBOM upload and analysis progress, how many packages have been found, and a list of any issues that came up during the import process.

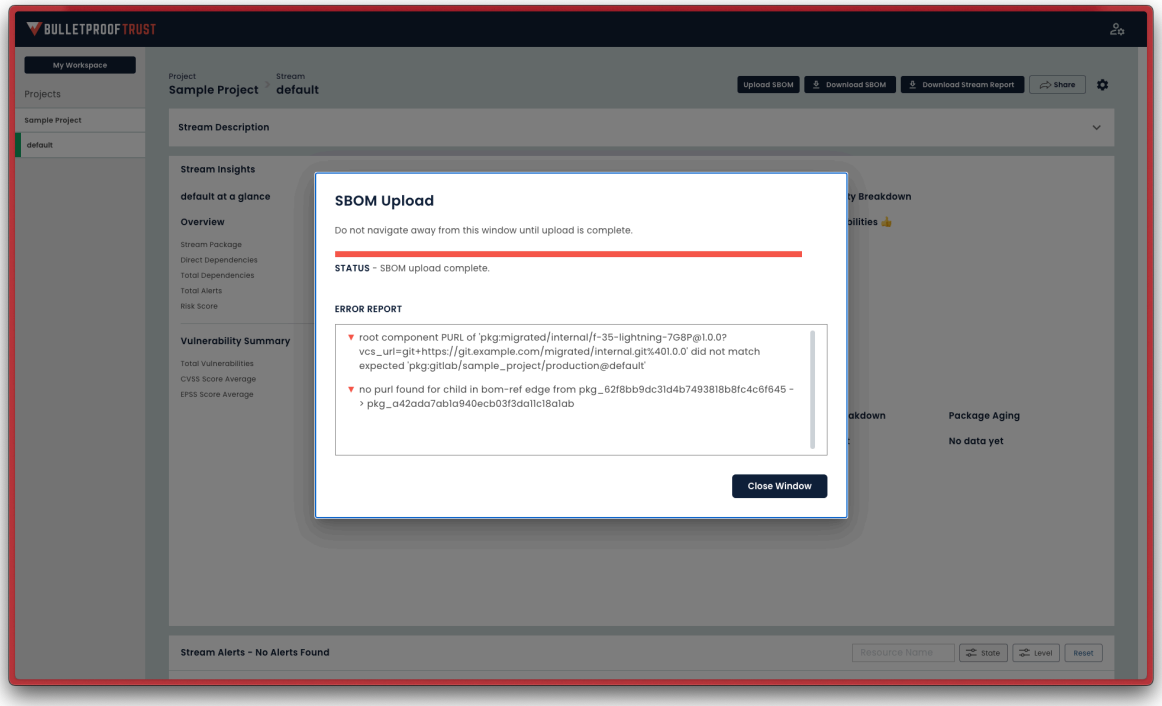


Figure 34: Importing an existing SBOM into a stream.

Assuming the upload completed successfully, click 'Close window' to view the packages uploaded from the SBOM in your stream in the Packages tile at the bottom of your stream view.

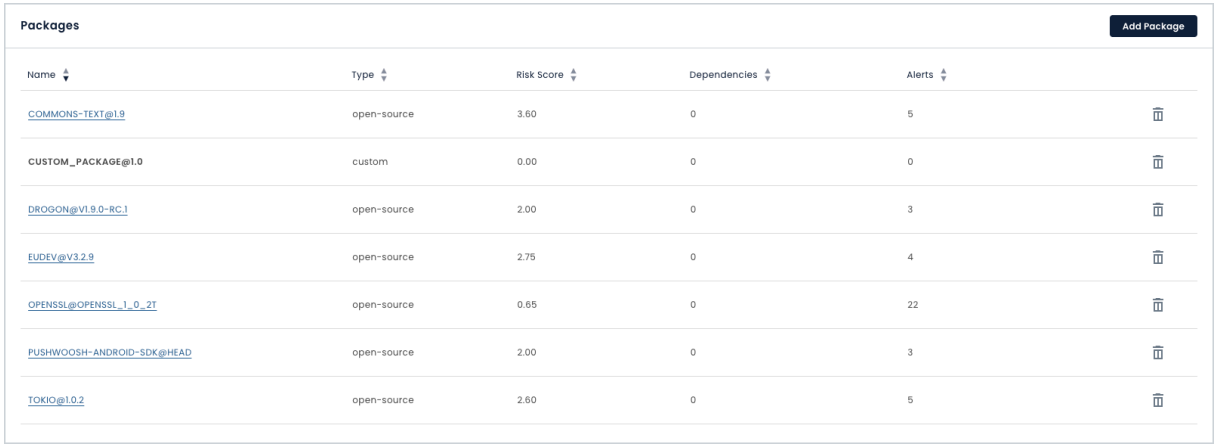


Figure 35: Packages from an existing SBOM added to the Packages tile in your stream view

Note: Uploading an existing SBOM to your stream will delete any existing packages you have in that stream, overwrite any packages you already have in that stream. The resulting stream package list will contain only those packages in the SBOM you imported.

Package View

The package view shows critical information about the package's risk and trustworthiness in a stream. Like the project and stream views, the package view is broken up into multiple tiles.

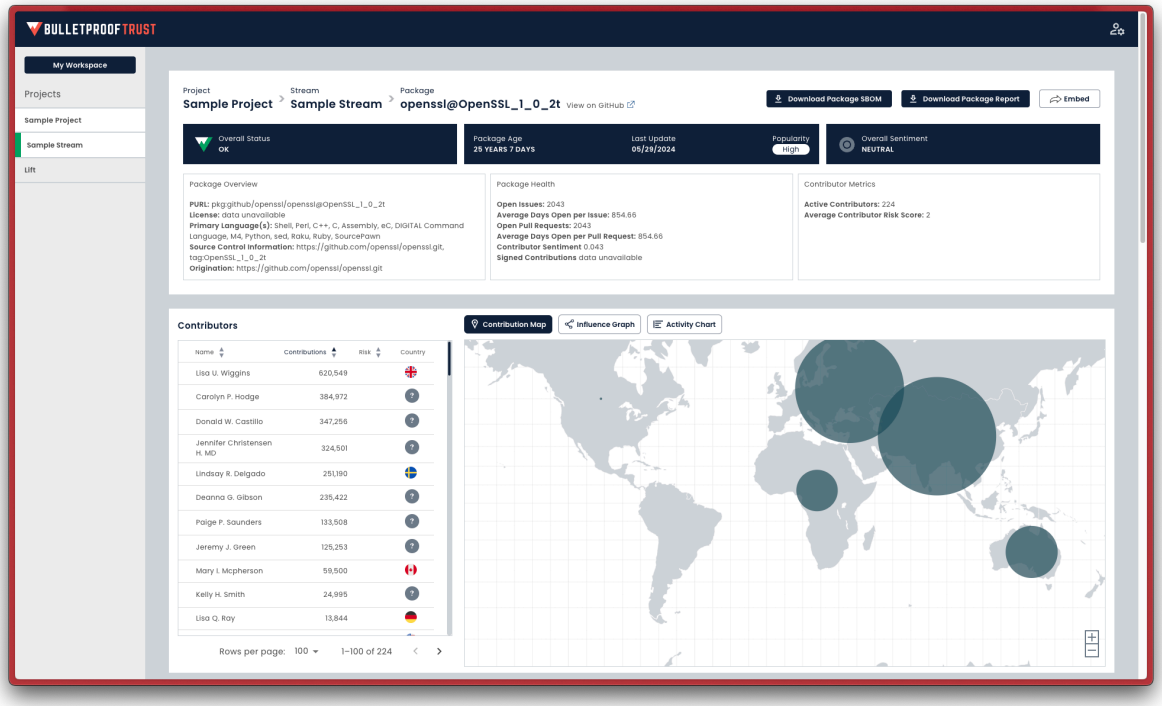


Figure 36: The package view, showing critical information about a package's risk and trustworthiness.

Package Overview

The package overview tile at the top contains navigation links to other views in the system associated with the package, various controls for extracting information from the package, and high level overview information about the package.

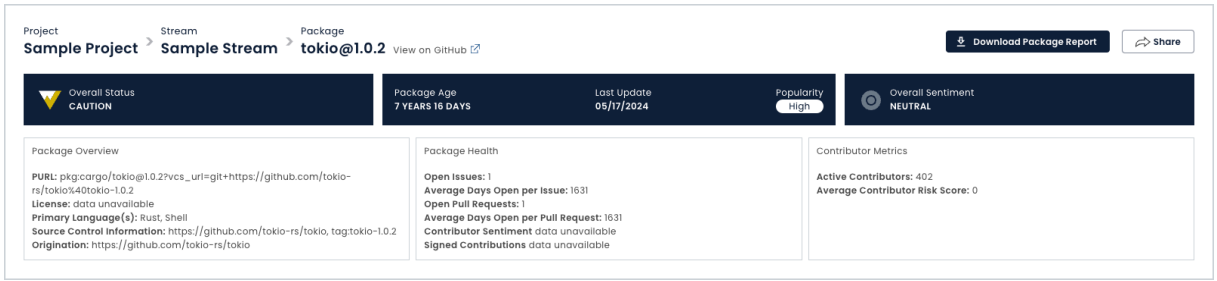


Figure 37: Package overview tile in the package view.

The breadcrumbs link in the top left of the package overview tile contains navigation links to the stream that the package is a part of and the project that that contains that stream. Clicking either

one of these will take you back to the stream or project that the package is a part of respectively.

Three blue information boxes can be seen across the middle of the package overview tile containing the overall status of the package (warning, caution, or ok), the age of the package, the date the package was last updated, the popularity of the package, and the overall sentiment of the package (a measurement of the sentiment found in the contributors code commit comments, which can indicate the possible presence of bugs or other issues in the code if poor, or lack thereof if positive).

Finally, three additional information boxes across the bottom of the package overview tile contain information about the package itself (such as the package URL, license information, programming languages used in the package, source control information, and origination of the package), health of the package (such as number of open issues, the number of days an issue remains open on average, the number of open pull requests, the average number of days a pull request is open, the contributor sentiment score, and the percent of signed contributions in the package), and contributor metrics (such as number of active contributors and the average contributor risk score). These indicators can give you an at-a-glance assessment of the health and status of the package, outside the normal package alerts and vulnerabilities tiles described below.

Package Contributors

Below the package overview tile, the Contributors tile shows a listing of contributors and, by default, a map showing the location of all contributions to the package.

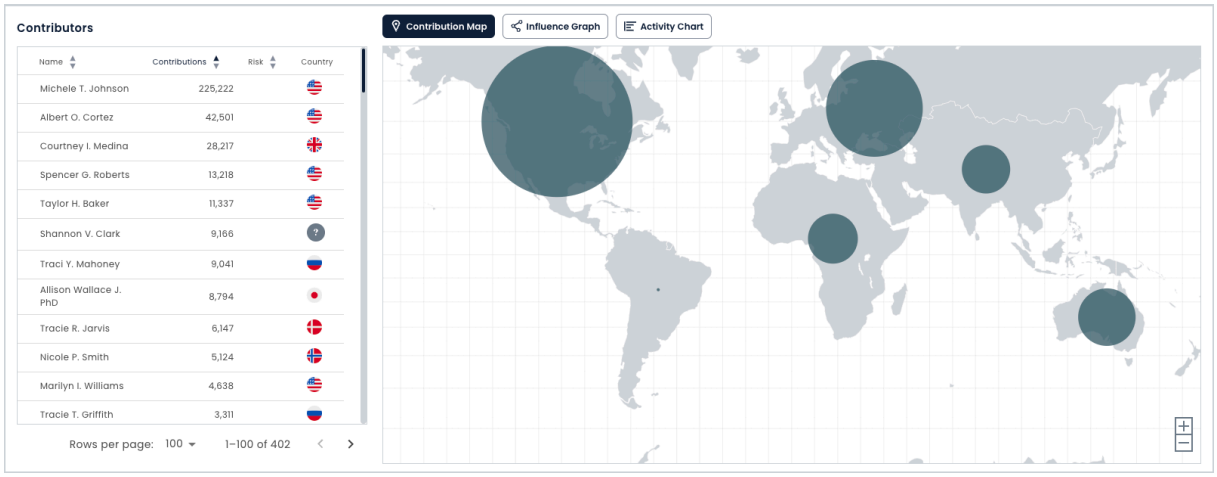


Figure 38: Contributors tile in the package view showing a list of all contributors and the locations from which they contribute.

The default map view displays the relative size of the bubbles over the various regions and countries indicate the distribution of contributions across the globe. Zooming in will provide more detailed data

as the view moves from a region-centric view to a country-centric view. Hovering over a given region or country will provide the contribution count.

To see how contributors influence each other on this project, click the 'Influence Graph' button at the top of the Contributors tile. This will show an edge-node connected graph of all contributors and their relationship to one another.

Figure 39: Contributors tile showing the a contributor influence graph.

To see the activity of all contributors click the 'Activity Chart' button at the top of the Contributors tile. Contributors will be sorted by most to least active. The bar graph represents the number of contributions each contributor has made overall, broken into contributions that add code vs. contributions that remove code. In general, most contributors add slightly more code than they remove. If you see a contributor who is adding far more code than they are removing or vice versa, this may indicate a trust problem.

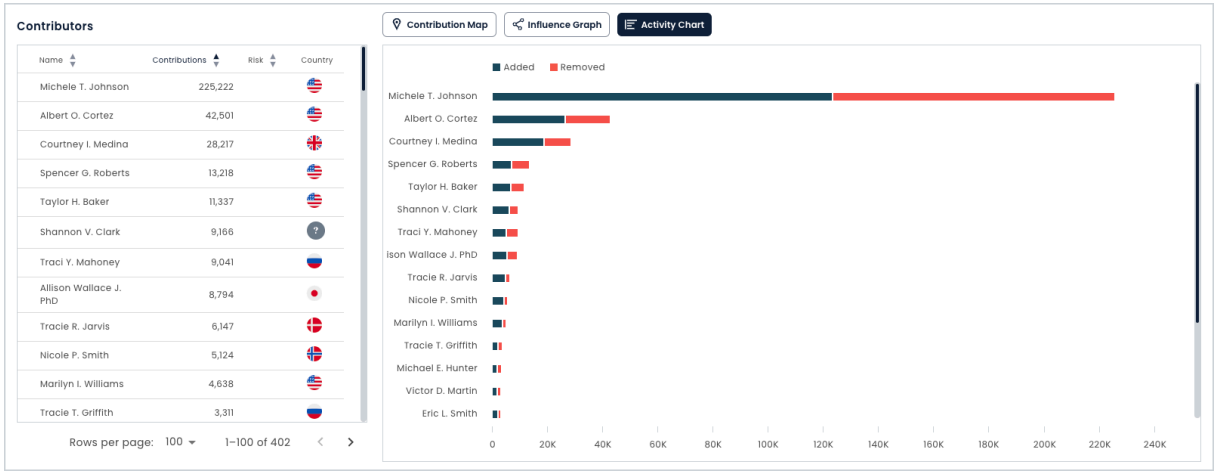


Figure 40: Contributors tile showing the a activity of each contributor.

Contributors in the contributor list can be sorted by name, number of contributions, risk score, and country. To see additional details about an individual contributor, simply click the name of the contributor of interest. This will expand the Contributors tile to show additional information about that individual contributor.

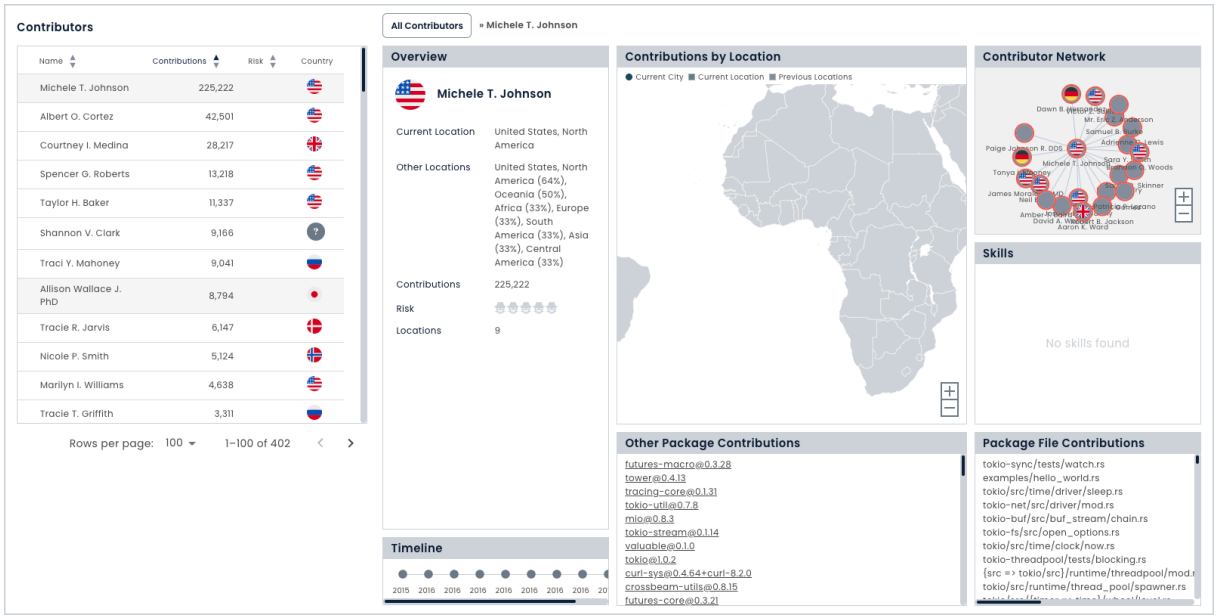


Figure 41: Contributors tile with a single contributor expanded to show details.

In this contributor details view you can see any alerts that impact the Contributor risk score, more details about the contributor data points including a list of files they have modified in this package, a list of other packages Bulletproof Trust has indexed which contain contributions from this contributor,

their contribution location history, a set of skills if known, and a timeline of data points associated with this contributor. To go back to the main Contributors tile view, simply click 'All Contributors at the top of the Contributors tile.

The maximum number of contributors displayed in the tile is based on the "Rows per page" number in the bottom right of the contributors list in the tile (the default is 100). If you wish to display more than 100 contributors at a time, you can click the number of contributors displayed and select the number you wish to have displayed from the dropdown. If there are more contributors than can be displayed at once, you can page through the contributors by clicking the right or left arrows in the lower right corner of the contributors list.

Package Alerts

Below the Contributors tile is the Package Alerts tile. This tile contains a list of all alerts for that package, based on your specific alert profile.

Package alerts can be filtered by their name by using the "Resource Name" text entry box near the top right hand corner of the tile. Package alerts can also be filtered by state (active or dismissed) or level (critical, high, medium, low, or informational). If you wish to reset all filters back to their default status, simply click the 'Reset' button in the upper right of the tile.

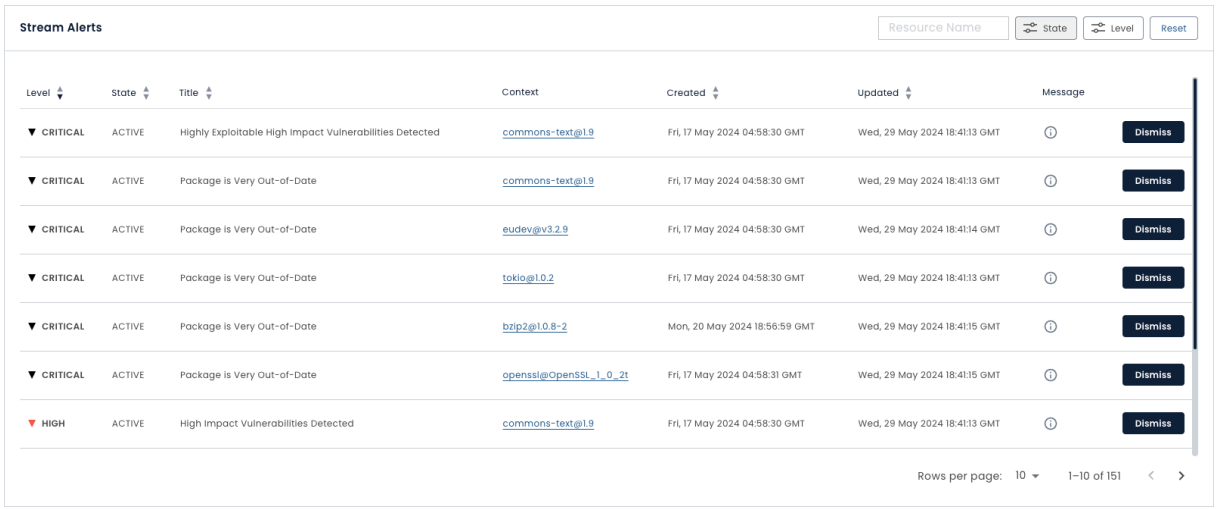


Figure 42: Package Alerts tile in the streams view.

Package alerts can be sorted by state (active or dismissed), level (critical, high, medium, low, or informational), alert title, context (the package in which the alert fired), date created, and date updated. There is also an informational icon in the message column with more details about that specific alert.

Finally, you can dismiss the alert by clicking the 'Dismiss' button on the far right row of the alert you wish to dismiss.

The maximum number of alerts displayed in the tile is based on the "Rows per page" number in the bottom right of the tile (the default is 10). If you wish to display more than 10 alerts at a time, you can click the number of alerts displayed and select the number you wish to have displayed from the dropdown. If there are more alerts than can be displayed at once, you can page through the alerts by clicking the right or left arrows in the lower right corner of the tile.

Package Vulnerabilities

Package vulnerabilities can be viewed in the Vulnerabilities tile. This tile shows known vulnerabilities for the package of interest and any of its direct dependencies. The can include National Vulnerability Database (NVD) CVEs, OSV alerts, or Github Security Advisories.

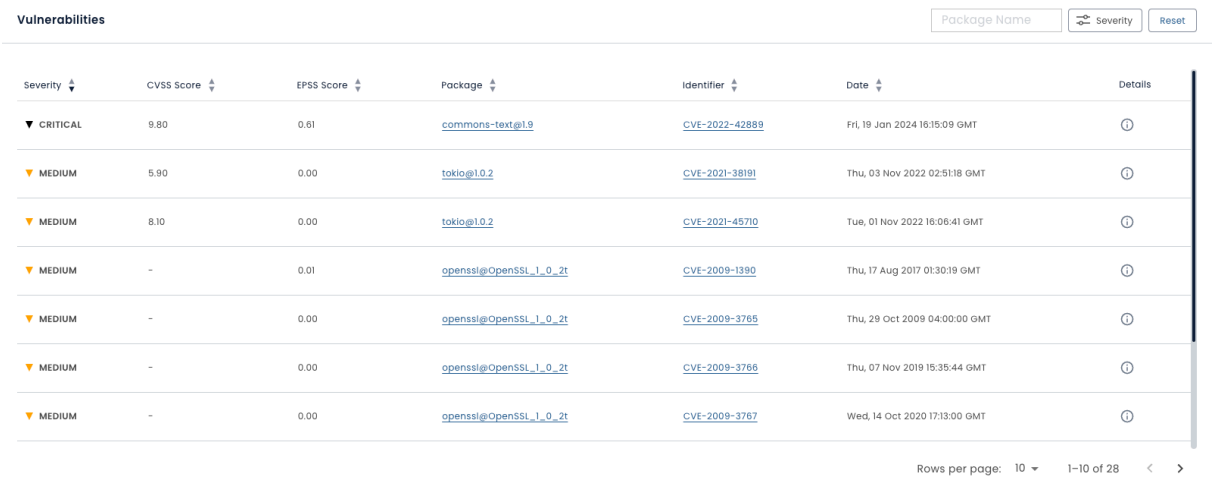


Figure 43: Vulnerabilities tile in the package view.

Vulnerabilities can be filtered by their name using the "Package Name" text entry form near the upper right hand corner of the tile. Vulnerabilities can also be filtered by their severity (critial, high, medium, low, or informational). If you wish to reset all filters back to their default status, simply click the 'Reset' button in the upper right of the tile.

Vulnerabilities can be sorted by severity (critical, high, medium, low, or informational), CVSS Score, EPSS Score, package name, identifier (e.g., the CVE number and link to the original vulnerability disclosure), and the date. Additional details can be found in the Details column by hovering over the little circle "i" in the row for that vulnerability.

The maximum number of vulnerabilities displayed in the tile is based on the "Rows per page" number in the bottom right of the tile (the default is 10). If you wish to display more than 10 vulnerabilities at a time, you can click the number of vulnerabilities displayed and select the number you wish to have displayed from the dropdown. If there are more vulnerabilities than can be displayed at once, you can page through the vulnerabilities by clicking the right or left arrows in the lower right corner of the tile.

Package Health Findings

Package health findings can be viewed in the Package Health Findings tile. This tile contains a list of all package health findings from the package being viewed.

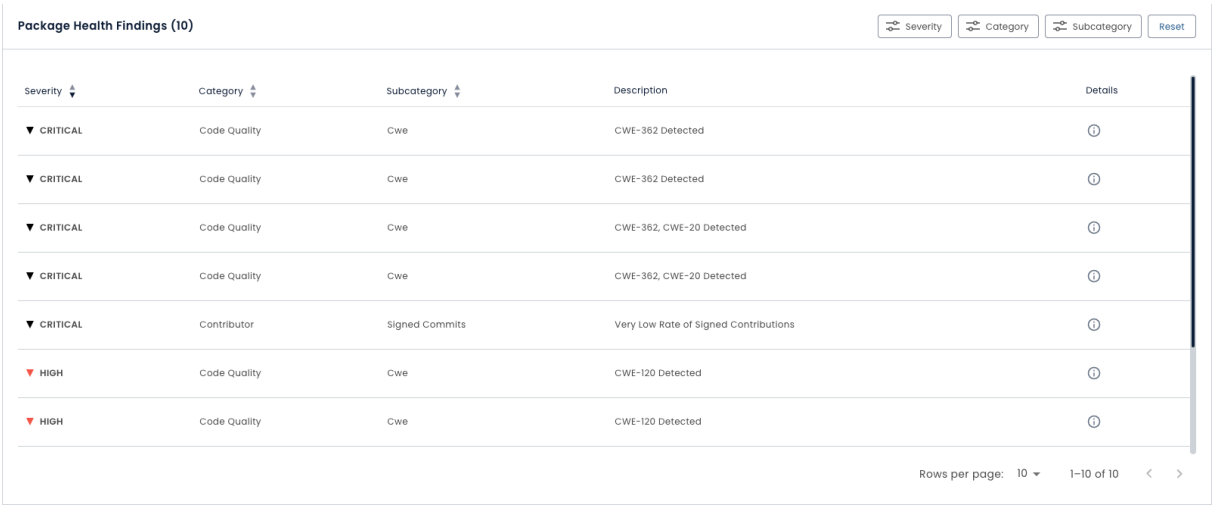


Figure 44: Package Health Findings tile in the package view.

A package health finding is an issue that could potentially increase the riskiness or lower the trustworthiness of the package in one or more categories. Package health findings categories include discovered malware (different from a vulnerability, which is a potential exposure to exploit vs. an actual exploit found), legal issues (such as license mismatches in the code), maintainability issues (such as deprecated dependencies), contributor issues (these include issues that affect all contributors, such as overall low code signing rates), code quality issues (various CWE's that are detected), and integrity issues).

Package health findings can be sorted by severity (critical, high, medium, low, or informational), category (malware, legal, maintainability, contributor, code quality, or integrity), and subcategory (signature detection, explicit license, implicit license, no license, copyright detection, deprecation, seniment, signed commits, cwe, generic). Additional details can be found in the Details column by hover-

ing over the little circle "i" in the row for that package health finding.

The maximum number of package health findings displayed in the tile is based on the "Rows per page" number in the bottom right of the tile (the default is 10). If you wish to display more than 10 package health findings at a time, you can click the number of package health findings displayed and select the number you wish to have displayed from the dropdown. If there are more package health findings than can be displayed at once, you can page through the vulnerabilities by clicking the right or left arrows in the lower right corner of the tile.

Package Reports

A package report is a PDF representation of what is displayed in the package view. Package reports are useful to share with others who may not have access to Bulletproof Trust or your specific project. They are also useful for archival purposes. Package reports contain overall information about the package, alerts discovered in the package, contributor risks discovered in the package, vulnerabilities discovered in the package, and package health findings. A package report can be downloaded by clicking the 'Download Package Report' button near the top right of the package view. Once clicked, you will see a notification "Generating report - this could take a few moments…". Do not navigate away from this package view while the report is generating. Once finished, you will be presented with a link to download the PDF report to your local computer.

Package Embeds

Package embeds are small snippets of code that represent the overall trust status of the package. They can be useful in development environments or custom dashboards where you need a quick, at-a-glace view of an individual package. To copy the code necessary to embed this package overview into a custom dashboard, simply click the 'Share' button in the upper right hand corner of the package view. You will be presented with a dialogue, giving you choice between markdown or HTML code snippet. Copy the code snippet by selecting the code in the appropriate text box or by clicking 'Click to Copy'. This will place that code snippet in your clipboard for you to paste elsewhere.

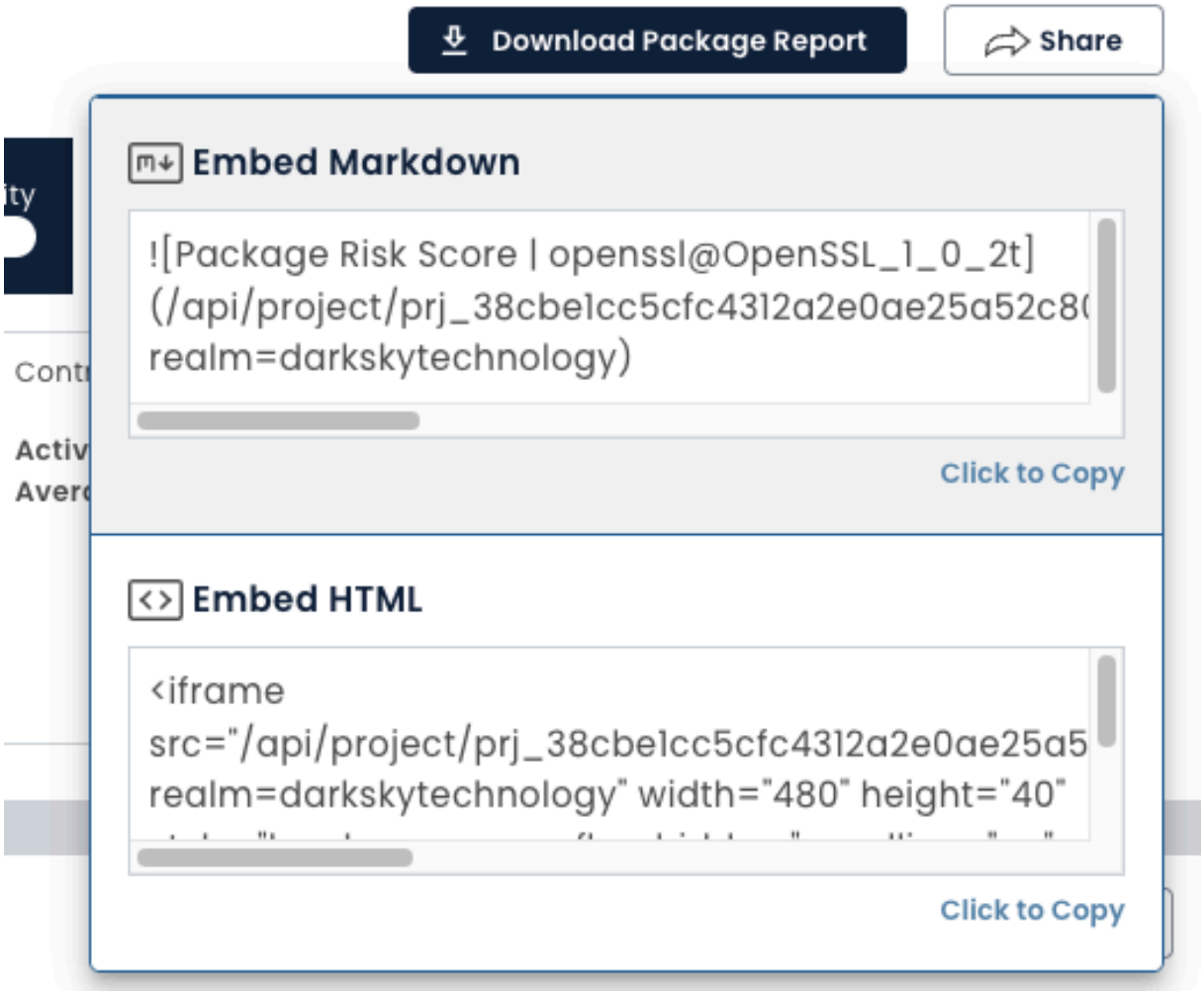


Figure 45: Embed package dialogue.

Package SBOM

If a package SBOM is available, you will see a button in the upper right hand corner of the package view labeled 'Download Package SBOM'. Click this button to generate the SBOM for that specific package. Once clicked, you will see a notification "Generating SBOM - this could take a few moments…". Do not navigate away from this package view while the package SBOM is generating. Once finished, you will be presented with a link to download the package SBOM to your local computer.

SBOM Vaults

The SBOM Vault is a bit-for-bit encrypted storage system designed to securely manage your SBOMs. In the SBOM Vault, you can create multiple vaults and decide whether to share them for collaboration or keep them private. Each SBOM is revision tracked with details including the uploader and a revision note, ensuring that every change is clearly documented. Additionally, you can link SBOMs to an analysis Stream to facilitate risk analysis. Vaults can also be encrypted and exported when necessary, providing you with a robust and flexible system for managing your SBOMs.

Creating a new Vault

Creating a new SBOM vault is straightforward. To begin, click the "Add Vault" button. You will be prompted to search for an existing vault or you can click to "Create a new vault". This will bring up a window where you will fill in details such as the vault's name. Once completed, click the "Create Vault" button. A new vault will be created for you, with default permissions set so that only you have access.

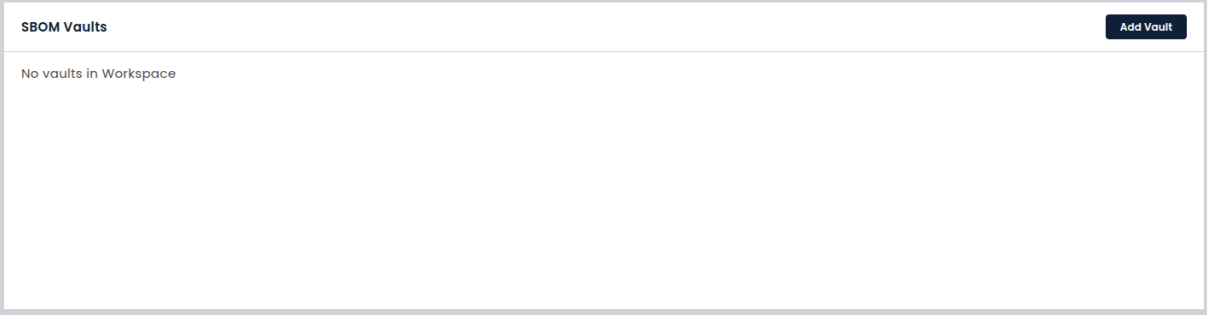


Figure 46: Add a new vault

Adding SBOMs to a Vault

To add SBOMs to a vault, first navigate to the desired vault. Click the "Add SBOM" button. You will then be prompted to fill in the SBOM name and description. After that, select the SBOM file you wish to upload and click "Add to SBOM Vault." This will create a new SBOM entry in that vault.

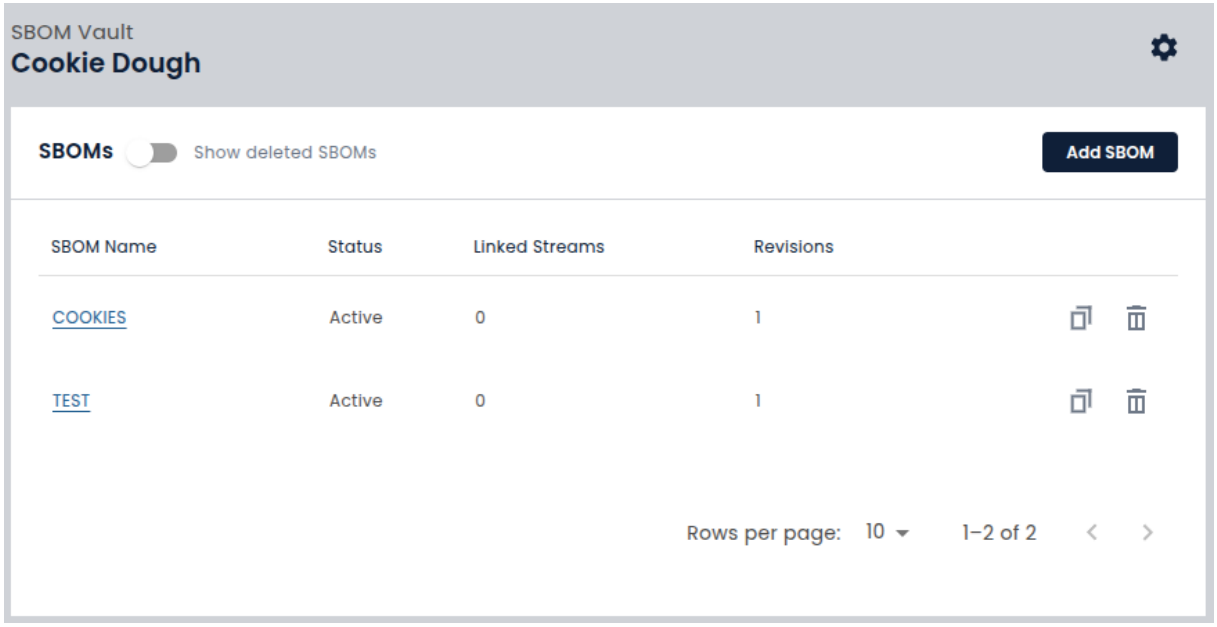


Figure 47: Add a new SBOM

Updating an SBOM in a Vault

To update an existing SBOM or add a revision, navigate to the vault and select the SBOM you wish to update. Click the "Add a Revision" button, then fill in the updated SBOM description. Next, select the new SBOM file and click "Add SBOM Revision" to complete the update.

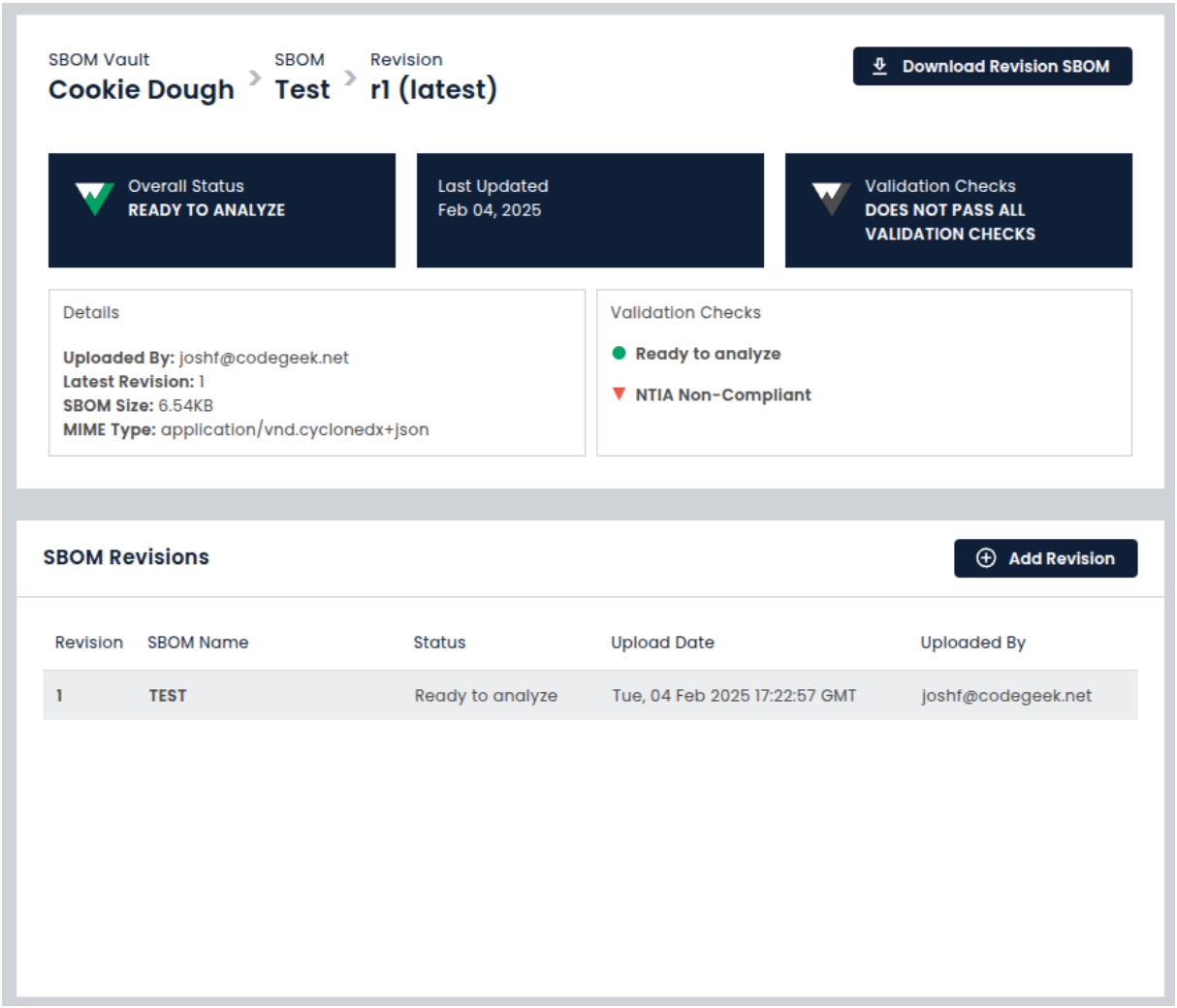


Figure 48: Add a new Revision

Sharing a Vault

To share access to a vault with other users, navigate to the vault settings page by clicking the gear icon in the top right-hand corner of the vault. Then, select "Add User" to add users to this vault. You can assign specific permissions for each user.

Exporting a Vault

To export a vault, navigate to the vault you want to export and click the gear icon in the top right-hand corner. Then, click the "Download Vault Archive" button. A box will appear asking you to enter an

archive encryption password, which will be used to encrypt the ZIP file. After supplying a password, click "Download" to export your vault as an encrypted ZIP file.

Account Management

From any page in the dashboard, open the Account Management panel by clicking on the gear icon in the top right. Here, your account email address as well as your API keys can be viewed and managed.

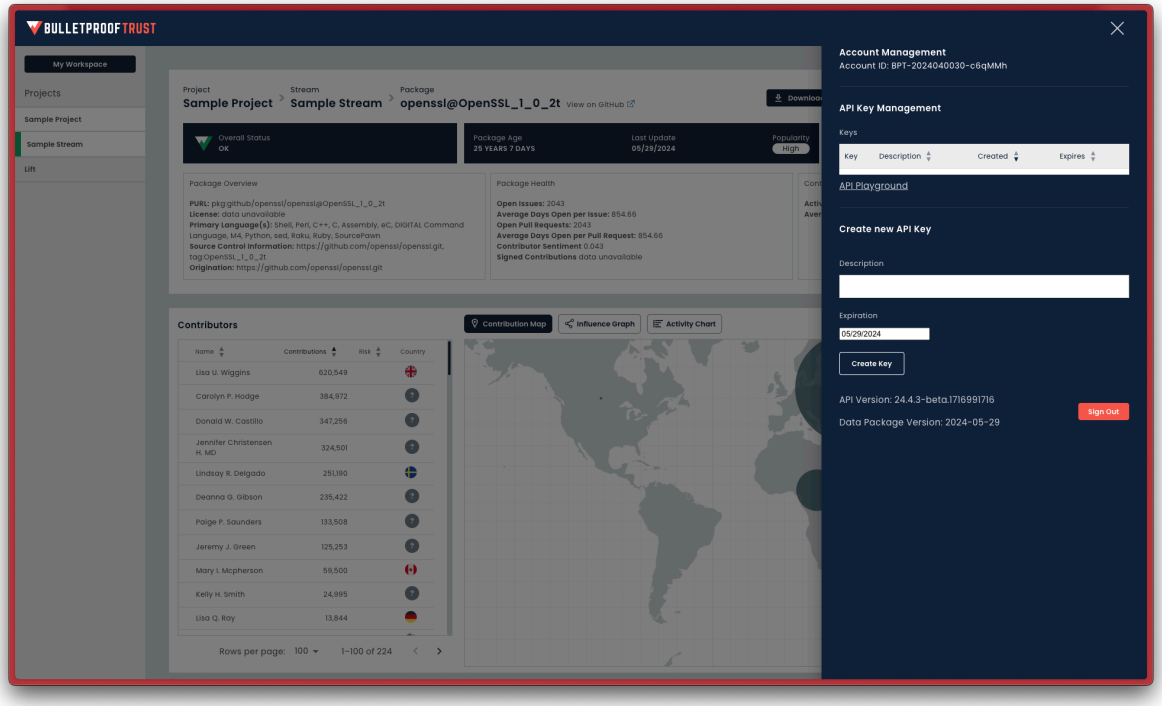


Figure 49: Account management panel.

To generate, manage, and use your API keys, please see the Using the API section below.

To sign out of Bulletproof Trust, click the red 'Sign Out' button in the lower right hand corner of the panel.

Notification Management

User notifications play a crucial role in keeping users informed about various events within the system. These notifications can cover a range of events, such as analysis streams being updated, new SBOM

revisions being uploaded, alerts changing for a subscribed stream and more. Users can be notified about events via the in-app notification inbox as shown below, by email, or both.

Users have the ability to customize their notification preferences to suit their needs. To edit these preferences, simply navigate to the User Notification Inbox by clicking the inbox button in the top right, and click on the gear icon located inside your inbox.

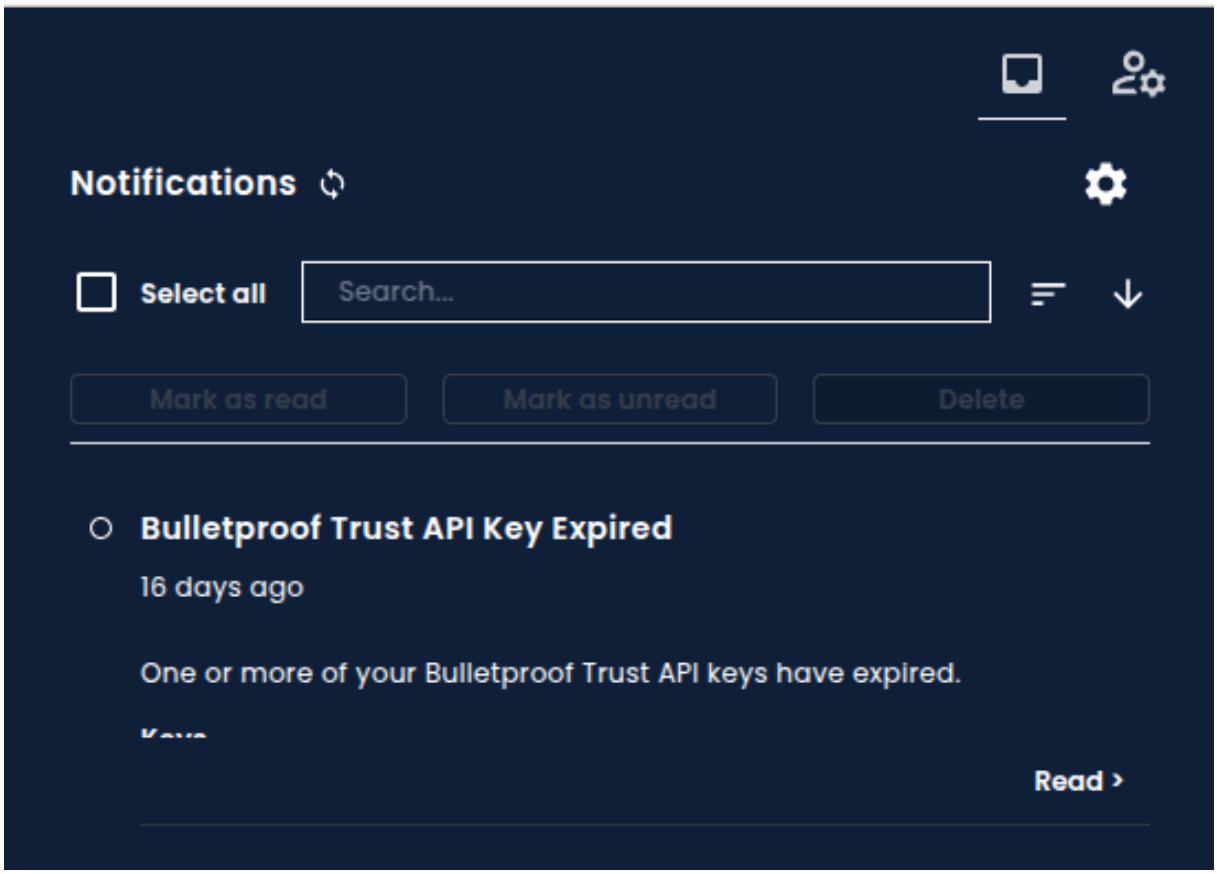


Figure 50: User notification inbox.

Events and Subscriptions

There are numerous types of events that users can stay informed about. For each event type, you have the option to subscribe to notifications delivered via the in-app Notification Inbox, email, or both. This flexibility ensures that you receive notifications in the manner that best suits your preferences.

For events related to Streams, Vaults, SBOMs, and Projects, you need to subscribe to those individual resources to receive the specific event notifications. Within the Event Notification Preferences area, you can view which subscriptions are currently active for your user account.

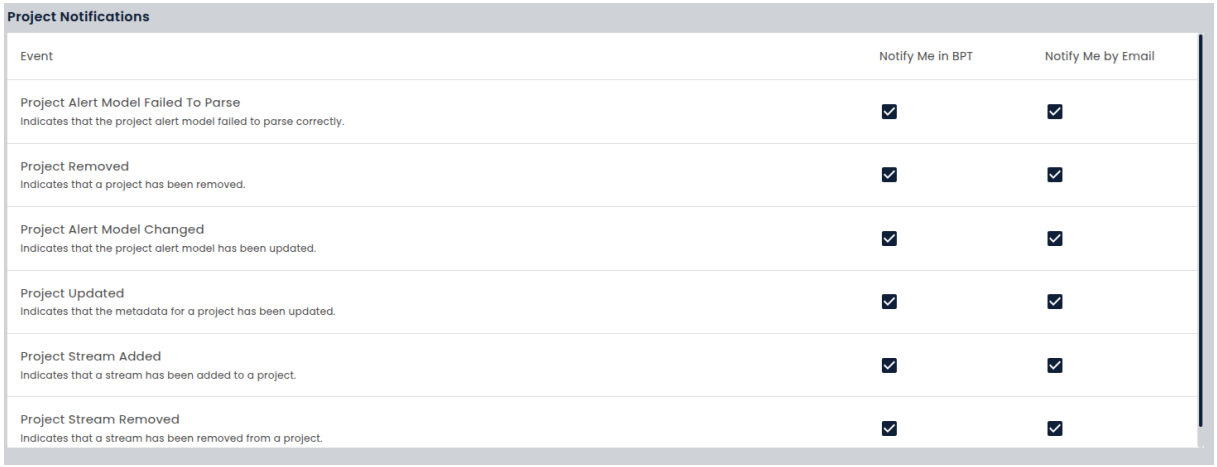


Figure 51: User notification inbox.

Resources you create will automatically be subscribed to, however to subscribe to other resources, simply navigate to the resource and click the bell icon in the top right. To unsubscribe, you can click this icon again, or navigate to your user notifications preferences page where all your active subscriptions can be managed in one place.

Using the API

The Bulletproof Trust API can be accessed at https://api.darksky.technology/bulletproof-trust/stable. For on-premises installations, please contact your system administrators to get the correct URL to the API. For both managed and on-premises systems, a Swagger API Playground exists to aid usage and understanding of the API's methods. This Playground also shows the schema used when data is returned from the API.

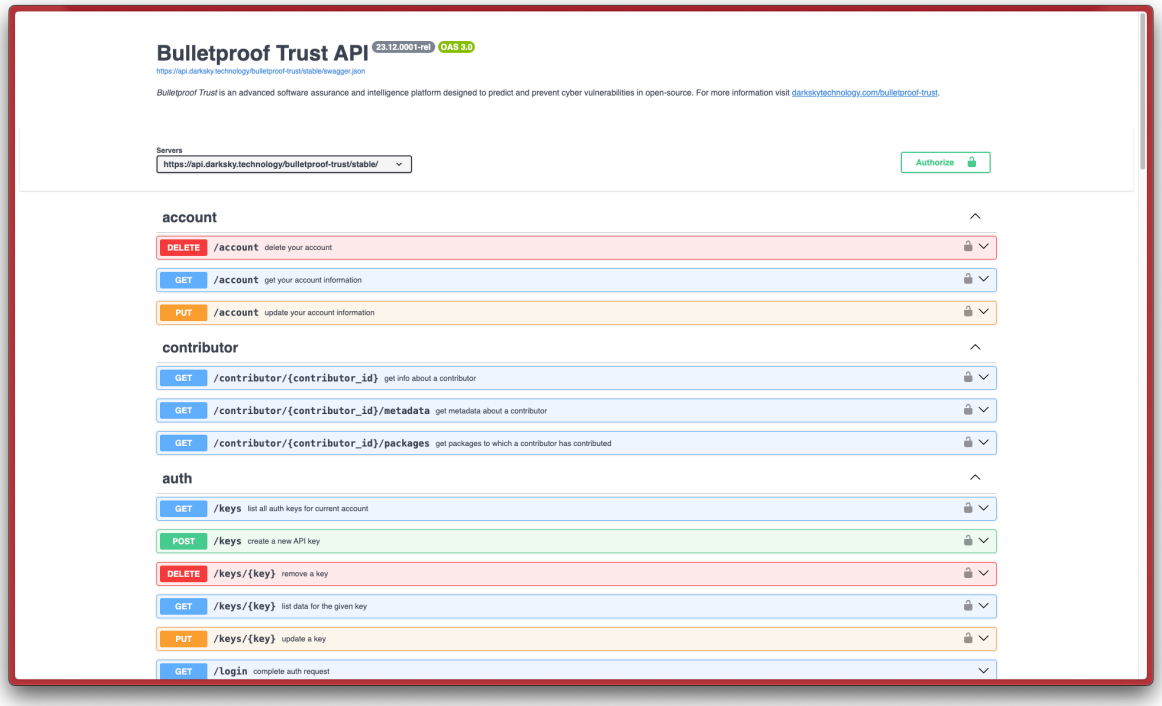


Figure 52: The Bulletproof Trust API Playground

For the managed platform, this Playground is at https://api.darksky.technology/bulletproof-trust/stable/playground. For on-premises installations, please contact your system administrators to get the correct URL to the Playground.

API Overview

The API for Bulletproof Trust is broken out into six sections:

1. Contributor Data
2. Package Data
3. Project Management
4. Auth Management (logging in and out, key management)
5. Tools
6. Account Management

These six sections and their related methods can be viewed in the API documentation as part of the Playground. The API is secured by Bearer Authorization and requires that an API key be present for most operations. Operations which require authorization are marked by a lock (🔒) icon.

Note: The tools in the `tools` sections are only available in the Dark Sky managed deployment.

API Playground Authorization

You will need an API key to authorize a session in the API playground. For details on how to do this, see Generating API Keys. Navigate to the Playground URL and click on the green 'Authorize' button in the top right. Paste your API key into the resulting popup as shown below and click 'Authorize'.

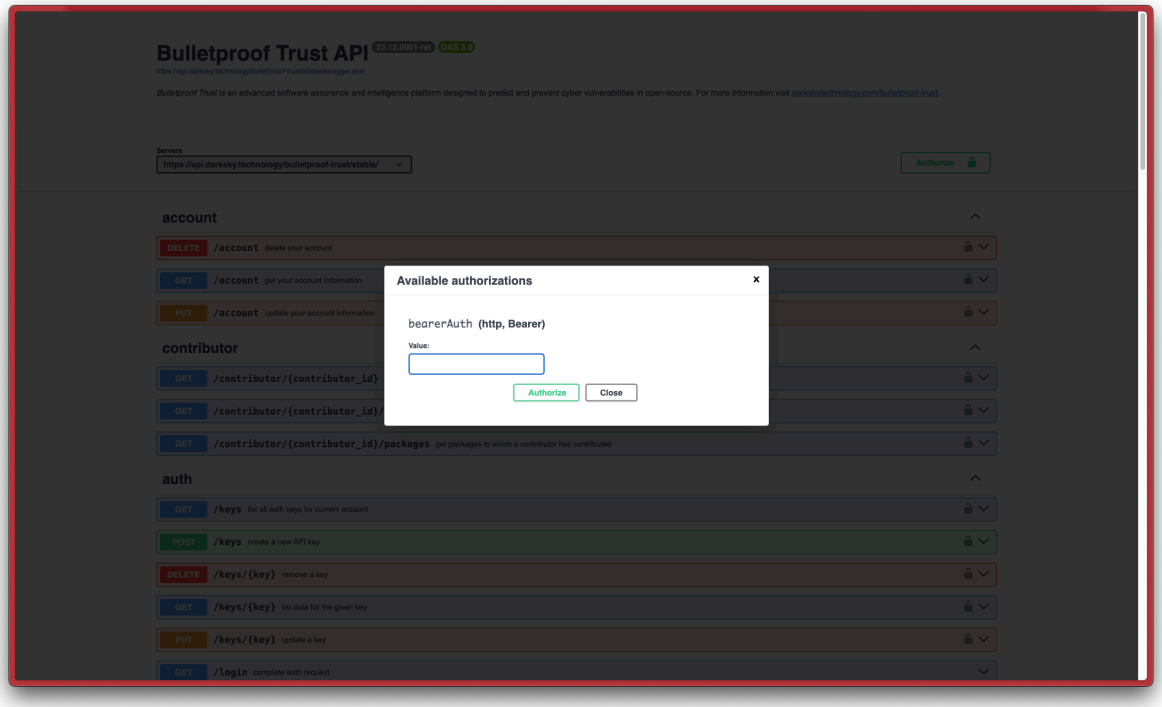


Figure 53: API playground authorization.

Generating API Keys

To use the API without interacting with the dashboard, create one or more API keys from the Account Management panel. Enter a key description and an expiration date, and your key will be generated and populated in the key table above. You can then copy, reveal, or delete this key by clicking the key of interest in the key list.

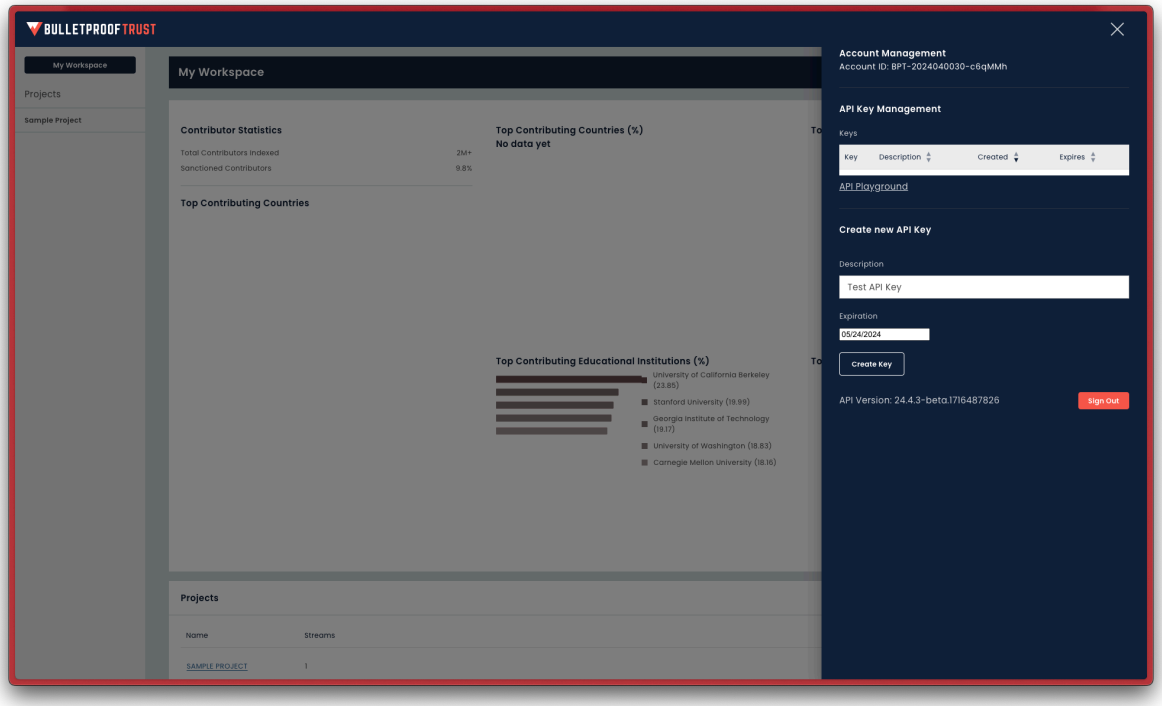


Figure 54: Generating an API key.

Once copied, visit the API Playground. For the managed platform, this Playground is at https://api.darksky.technology/bulletproof-trust/stable/playground. For on-premises installations, please contact your system administrators to get the correct URL to the Playground.

Common API Actions

Finding a Package

To find a package already in the Bulletproof Trust index, use the `GET /package` endpoint. Click to drop down the configuration for this endpoint and fill the search and limit variables. The `search` variable takes the same inputs as the search box detailed in Adding a New Package Manually. Click execute and view the results in the box below. Note that the Playground prints a `cURL` command for generating this query. You can copy this command into your terminal to replicate the results outside the Playground environment.

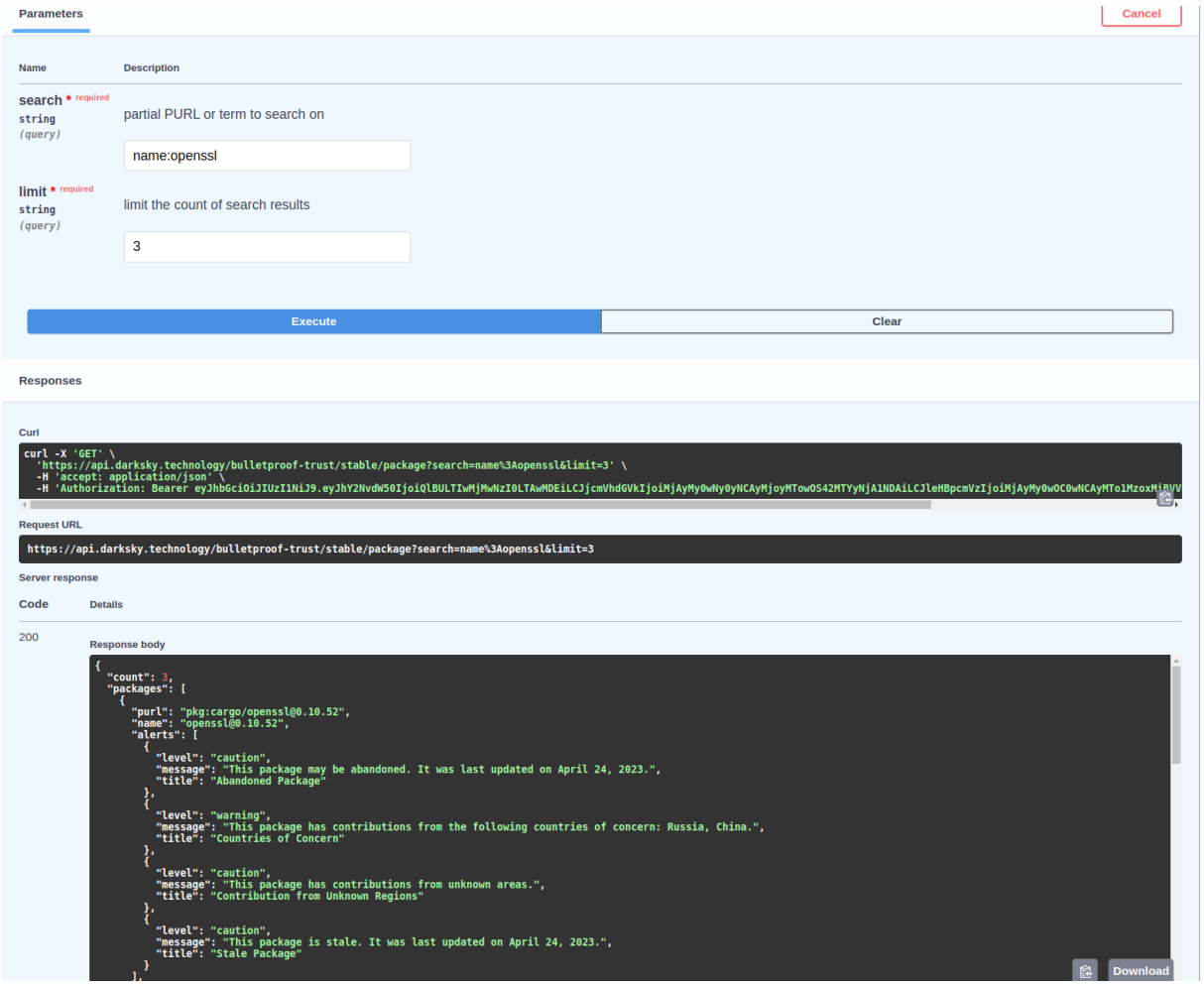


Figure 55: Finding a package using the API.

Retrieving a Package's Data

To get data for a specific package, use its Package Id.

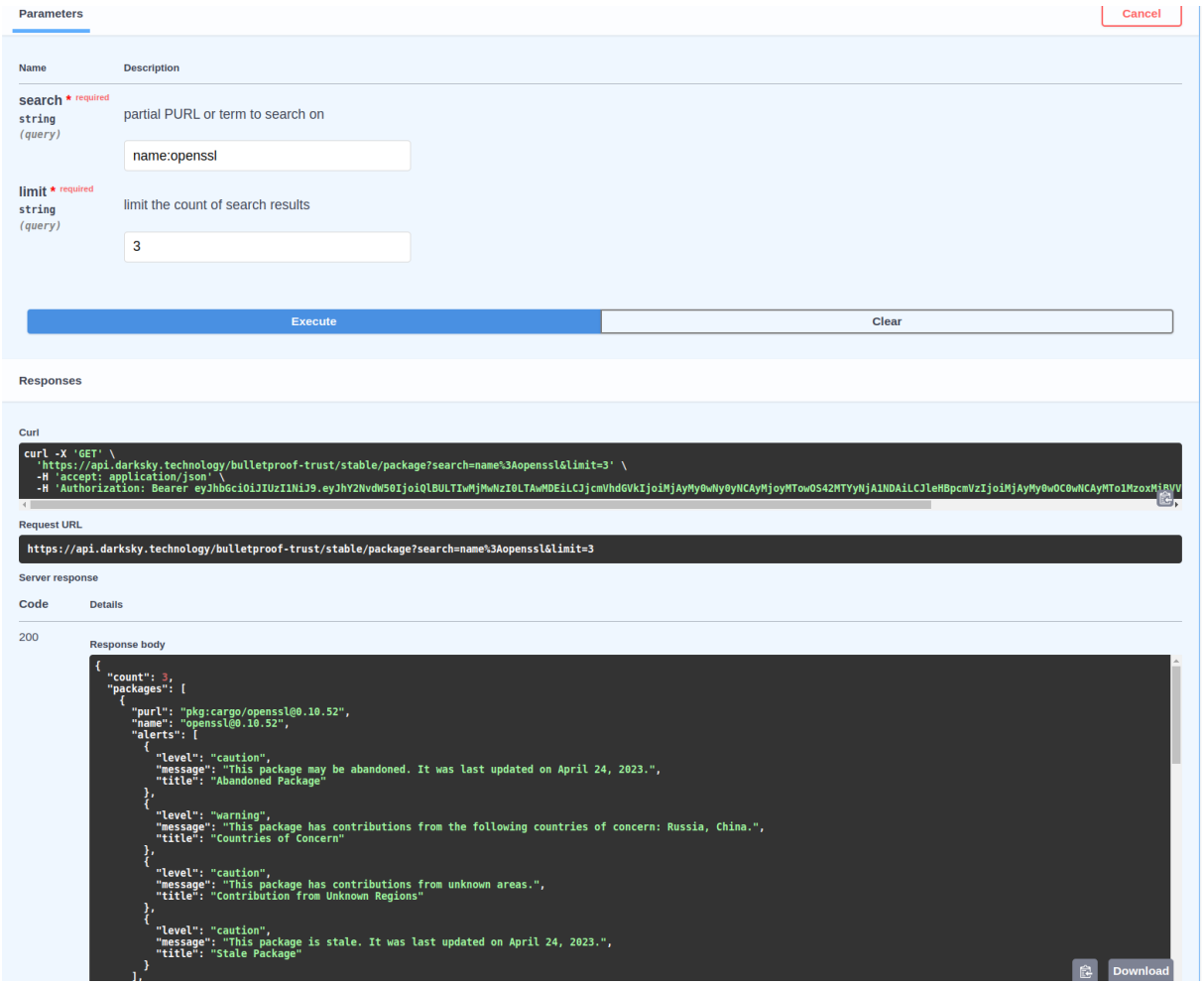


Figure 56: Retrieving a package's data.

In this example, the package `pkg:cargo/openssl@0.10.52` was shown with an ID of `fde69d964dd04aa7af336083d9d68fec`. Use the `GET /package/{package_id}` endpoint to retrieve its data.

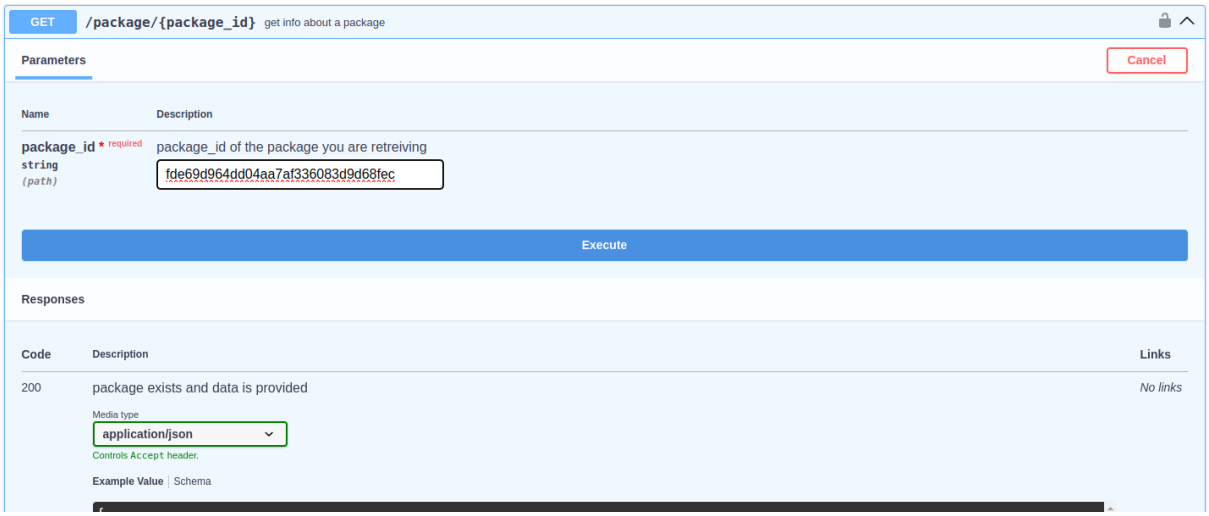


Figure 57: Retriving a package using the API.

Reviewing a Specific Contributor

If you want to drill into a specific contributor listed in the Package data, use the contributor's ID. This ID is unique across Bulletproof Trust. Copy this ID into the `GET /contributor/{contributor_id}` endpoint to get this contributor's record.

Figure 58: Retriving contributor information using the API.

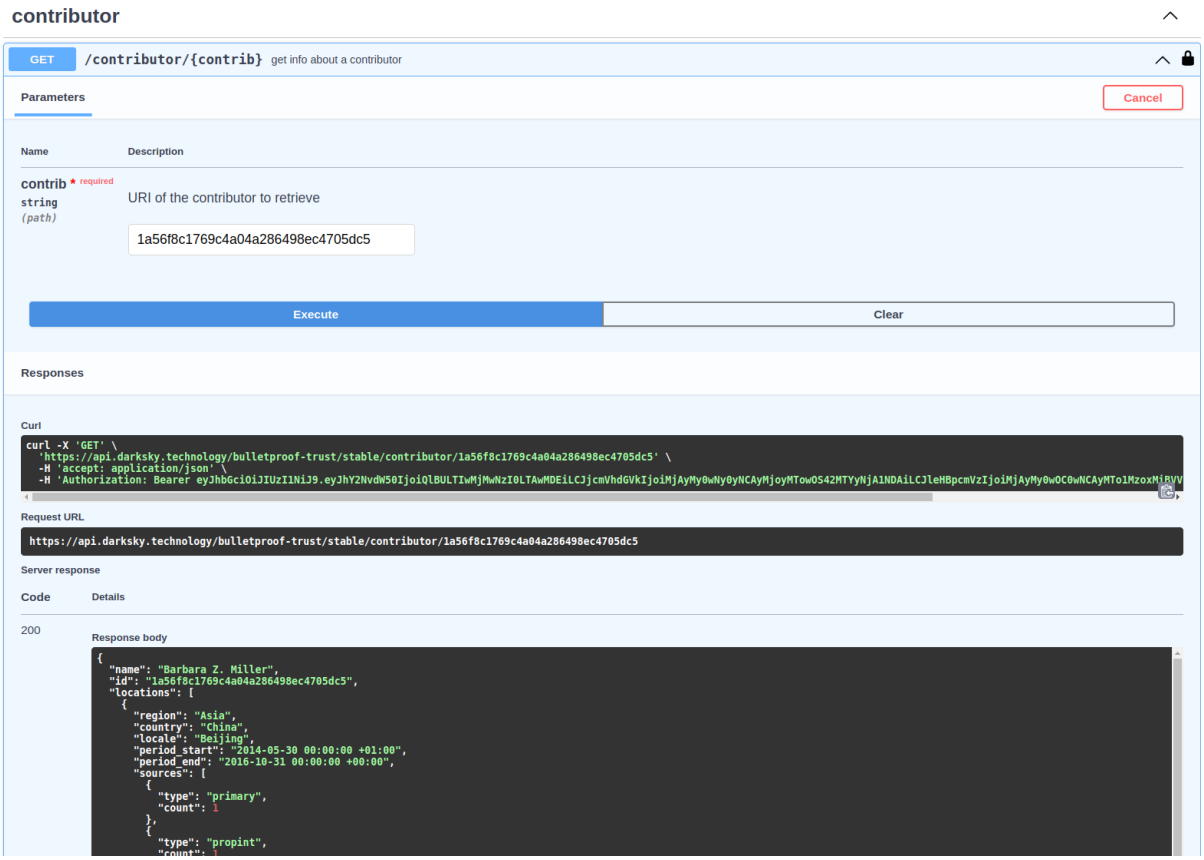


Figure 59: Viewing a contributor record using the API.

Submitting a Package

To submit a new package for indexing into Bulletproof Trust, use the `POST` **`/package`** endpoint. The PURL submitted must be a valid PURL or it will be rejected by the engine. The PURL must be populated in the request body object. Select an option for advanced analytics.

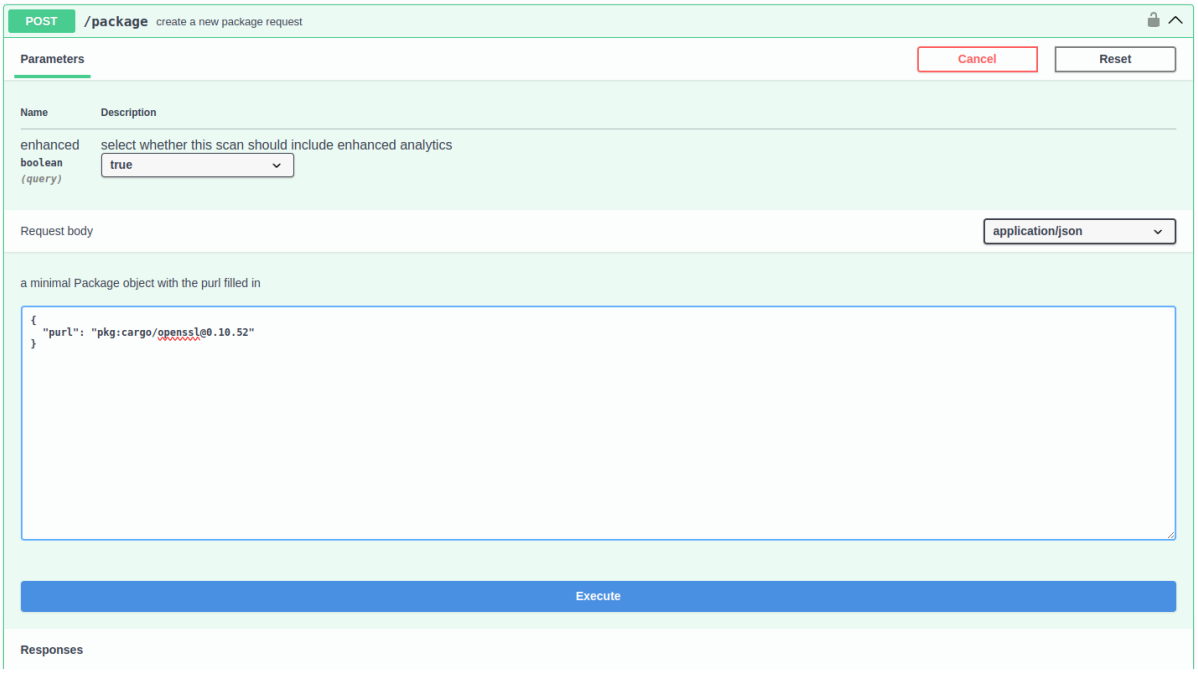


Figure 60: Submitting a new package for analysis via the API.

API Tools

Dark Sky offers a set of tools in the managed platform working with package and SBOM data for generating PURLs, CPEs, deriving source code location, etc.

Purlizer

The Purlizer is a component of the Bulletproof Trust platform used to ease the generation of PackageURLs (PURLs) for SBOM and other dependency-tracking use cases when only sparse or poorly formatted information is available about a given package. Frequently all that is known about a dependency is its name and a nominal version number. With this sparse information, various strategies are applied until a PURL is successfully generated or all strategies are exhausted. The ideal outcome is a PURL which includes, or can be used to derive, the SCM information for the given package.

Note: This is not always possible using the information given (for example when RPM packages are given, it can be difficult to automatically locate the source information).

Purlizer Usage

The Purlizer tool is asynchronous because some of the identification methods require multiple long-running steps. As such it is broken into three API endpoints.

- Request: this endpoint takes in an *identifier* string; returns a tracking ID
- Status: this endpoint takes the tracking ID and returns status of *complete*, *failed*, *in progress*
- Result: this endpoint takes the tracking ID and returns the results of the operation (if completed)
  The *identifier* string can be any string, but is preferentially a package name and version separated by the @ symbol. For example:

```
mypackage@1.2.3
```

Making a Request

To make a Purlizer request, POST an application/json body to the API endpoit with the following keys in the JSON document body:

- identifier: the @ string that the Purlizer will operate on
- strategies: the optional ordered list of selected strategies (see: Strategy Suppression or Selection

An application/json response with a single key is returned:

- id: the request tracking ID

Identifier Hints

The identifier can contain hints for the Purlizer's strategy matcher. For example package versions ending in "el8" will be first sought out in the RHEL and CentOS repositories. Currently implemented hints are:

- identifiers beginning with "python", "python27", "python3", etc. indicate a PyPI matcher should be run first
- identifiers ending in "el7", "el8", or "ubuntu" will be matched using the respective distribution repositories first
- identifiers beginning with "github", "github.com" will be matched using the GitHub matcher first
- identifiers that are PURL-like will be matched using the PURL strategy first

Strategy Suppression or Selection

Each of the named strategies can be chosen and ordered when making a request to the Purlizer to select or suppress certain strategies. When no selections are made, the default value is all strategies. To select a subset of strategies, pass the 'strategies' key in the request and set the items to any *N* of the following:

- purl
- deb
- rpm
- pypi
- npm
- cargo
- github
- conan

Note: Setting this key will force the selected strategies to be executed in the order they were provided.

Checking Request Status

To check the status of an active Purlizer request, perform a GET request to the status endpoint with the tracking ID retreived from the Request endpoint prior. An application/json response with a single key is returned:

- status: one of [in progress|failed|complete]

Or an HTTP 404 (Not Found) if this tracking ID is invalid.

Getting the Request Results

To check the result of an active Purlizer request, perform a GET to the Results endpoint with the tracking ID retreived from the Request endpoint prior. An application/json response with one key is returned:

- result: the generated PURL

Or an HTTP 404 (Not Found) if this tracking ID is invalid.

Note: Results are cached for a maximum of 180 minutes (3 hours).

CPE Generator

The CPE Generator is a component of the Bulletproof Trust platform used to attempt matching a PURL to a known CPE as defined by the National Vulnerability Database's Common Platform Enumeration Dictionary. This aids in connecting packages with known vulnerabilities across multiple platforms.

The CPE Generator tool is asynchronous because some of the match methods require multiple long-running steps. As such it is broken into three API endpoints.

- Request: this endpoint takes in a *purl* string; returns a tracking ID
- Status: this endpoint takes the tracking ID and returns status of *complete*, *failed*, *in progress*
- Result: this endpoint takes the tracking ID and returns the results of the operation (if completed)

Making a Request

To make a CPE Generator request, POST an application/json body to the API endpoit with the following keys in the JSON document body:

- purl: the PURL of the package of interest

An application/json response with a single key is returned:

- id: the request tracking ID

Checking Request Status

To check the status of an active CPE Generator request, perform a GET request to the status endpoint with the tracking ID retrieved from the Request endpoint prior. An application/json response with a single key is returned:

- status: one of [in progress|failed|complete]

Or an HTTP 404 (Not Found) if this tracking ID is invalid.

Getting the Request Results

To check the result of an active CPE Generator request, perform a GET to the Results endpoint with the tracking ID retrieved from the Request endpoint prior.

An application/json response with one key is returned:

- result: a list of potential CPEs, or an empty list if no confident match found

Or an HTTP 404 (Not Found) if this tracking ID is invalid.

Note: Results are cached for a maximum of 180 minutes (3 hours).

Configuring the Alert Model

The alert model is configured by default globally in the instance configuration file `/etc/bulletproof-trust/global_alert_model.toml` and can be customized on a per-project basis via the API.

Alert Configuration

In the sample below, we are configuring an alert. There are 6 components to configuring an alert:

- **tag** - this is the unique ID tag for the alert and denotes the start of a new configuration item; it is placed inside square brackets `[rules.this_is_an_alert_tag]` and must always be preceded by `rules.`.
- **level** - this is the level of alert to generate when triggered; must be one of [informational, low, medium, high, critical]
- **enabled** - this is a boolean true/false to allow you to create alerts and disable them without removing them from the config file
- **title** - this is the alert title when generated; it is best to keep this title short (less than 120 characters); this can be templated
- **message** - this is the alert message when generated; it can include more verbose information and uses a template (where applicable) to dynamically fill in information about the generated alert; this field can contain markdown
- **rule** - this is the trigger condition for the alert; this can take many forms and references key data that can be used in an alert
- **description** - this is used to describe the rule and serves only as a comment area for the rule writer

Sample Alert

```
### SAMPLE ALERT ###
[rules.High_Impact_Vulnerabilities_Exploitable]
description = """
This rule tests for high impact vulnerabilities.
"""

title = "Highly Exploitable High Impact Vulnerabilities Detected"
message = """
### This package contains {package.vulnerability.id} which has a high score of {package.vulnerability.base_score}/10.0 and
has an exploitabilty prediction index of {package.vulnerability.epss_score}/1.0.

#### Details
The Exploit Prediction Scoring System (EPSS) is a predictive model that estimates the likelihood of a
software vulnerability being exploited in the wild, based on characteristics of the vulnerability and
its environment. EPSS scores range from 0 to 1, where higher values signify a greater probability of
exploitation, aiding organizations in prioritizing patching and mitigation efforts more effectively
by focusing on vulnerabilities most likely to be exploited. See the [EPSS User Guide](https://www.first.org/epss/user-guide) for
details on interpreting this measure. Your score of {package.vulnerability.epss_score} indicates a moderate likelihood of
exploitation in the next 30 days.
"""
level = "critical"
rule = "{package.vulnerability.base_score} >= 7.0 and {package.vulnerability.epss_score} >= 0.5"
enabled = true
```

Alert `Rule` Types

Bulletproof Trust currently supports 5 kinds of rule types for alerts:

- Numerical comparison - testing if a condition is true based on >, <, >=, <=, ! =, or ==
- Set intersections - testing if a set of data is *inside* another defined set; e.g., a contributor's list of countries overlaps with a list of banned countries
- Date comparison - testing if dates are older than or more recent than a reference date
- Nullity comparison - testing if a value is null/none or non-existent
- Boolean comparison - testing if a value is truthy or falsey

These rule types can be chained together with `and` to create more complex rules and narrow the scope

of their applicability.

For example, from the sample above:

```
{package.vulnerability.cvss_v3_base_score} >= 7.0 and {package.
vulnerability.epss_score} >= 0.5
```

This will only fire when both conditions are met.

Numerical Comparisons

Configuring a numerical comparison test involves selecting a test *against* value and reference value. The test against value must come from the set of values listed in the "Alertable Conditions" section below. Configuring a test involves making a logical comparison between the test value (the dot-delimited value and a reference). For example:

```
{package.vulnerability.cvss_v3_base_score} >= 7.0
```

This test condition will fire when the CVSS base score for any vulnerability against the package of interest is greater than or equal to 7.0.

In the `message` for the alert configuration, you can use the test value (`{package.vulnerability.cvss_v3_base_score}`) as well as other context-appropriate variables to provide more details about the fired alert. For example:

```
message = "This package contains {package.vulnerability.id} which
has a high score of {package.vulnerability.cvss_v3_base_score}/10.0
and has an exploitabilty prediction index of {package.vulnerability.
epss_score}/1.0. "
```

When the alert is fired, the message would read something like: *This package contains CVE-2024-01234 which has a high score of 8.8/10.0 and has an exploitabilty prediction index of 0.88/1.0.*

Set Intersections

Configuring a set intersection test involves selecting a test *against* set and reference set. The test against set must come from the set of values listed in the "Alertable Conditions" section below. Configuring a test involves making a set comparison between the test value (the dot-delimited value and a reference). For example:

```
{contributor.organization} is in ["ByteDance", "Yadro"]
```

This test condition will fire when a company listed on the right hand side (ByteDance, Yadro) is found in the reference set for contributors of this package.

In the `message` for the alert configuration, you can use a 0-indexed subsitution to extract the set intersection operation value to provide more details about the fired alert. In this rule, there is only one rule component so the substition index is `{0}`. The `{0}` gets replaced by the subset which was found. For example:

```
message = "There were contributors from the following companies found
: {0}."
```

When the alert is fired, the message would read something like: *There were contributors from the following companies found: ByteDance*

The opposite can be acheived by using `is not in` in place of `is in`.

Fuzzy Set Comparison

Configuring a fuzzy set comparison test involves selecting a test *against* set and reference set. The test against set must come from the set of values listed in the "Alertable Conditions" section below. Configuring a test involves making a set comparison between the test value (the dot-delimited value and a reference). For example:

```
{package.license.discovered} is like ["GPL", "CPL"]
```

This test condition will fire when any discovered license fuzzy matches against any one of (GPL, CPL).

In the `message` for the alert configuration, you can use a 0-indexed subsitution to extract the fuzzy set comparison operation value to provide more details about the fired alert. In this rule, there is only one rule component so the substition index is `{0}`. The `{0}` gets replaced by the subset which was found. For example:

```
message = "Copy-left licenses found: {0}."
```

When the alert is fired, the message would read something like: *Copy-left licenses found: GPLv3.0, CPL1.0, LGPL*

The opposite can be acheived by using `is not like` in place of `is like`.

Date Comparison

Configuring a date comparison test involves selecting a test *against* date and reference date. The test against date must come from the set of values listed in the "Alertable Conditions" section below. Configuring a test involves making a date comparison between the test value (the dot-delimited value and a reference). For example:


```
package.metrics.key_dates.updated} is more than 6M ago
```

This test condition will fire when the package's most recent commit was more than 6 months ago. You can also use:

```
package.metrics.key_dates.created} is less than 1M ago
```

to perform the opposite comparison. Date comparison operators accepted are:

- `xD` - representing `x` days
- `xW` - representing `x` weeks
- `xM` - representing `x` months
- `xY` - representing `x` years

In the `message` for the alert configuration, you can use the test value (`package.metrics.key_dates.updated}`) to provide more details about the fired alert. For example:

```
message = "This package was last updated {package.metrics.key_dates.updated}"
```

When the alert is fired, the message would read something like: *This package was last updated on 2021-04-18 23:18:09+0500*

Nullity and Booelan Comparisons

Configuring a nullity or boolean comparison test involves selecting a test *against* value. The test against value must come from the set of values listed in the "Alertable Conditions" section below. Configuring a test involves making a logical comparison between the test value (the dot-delimited value) and a reference. For example:

```
{package.metrics.scm.is_valid} == false
```

String Comparisons

Configuring a string comparison test involves selecting a test *against* value. The test against value must come from the set of values listed in the "Alertable Conditions" section below.

This tests whether or not a selected test value contains another substring or not.

```
{package.health_finding.details} contains "crypto"
```

This alert would fire if the substing "crypto" was found in any package health finding for a given package.

The opposite can be acheived by substituting `does not contain` for `contains`.

Alertable Conditions

Single String Values

- **package**.metrics.scm.type - the SCM type; e.g., git, svn, bzr
- **package**.metrics.scm.url - the SCM URL
- **package**.metrics.scm.tag - the SCM tag, branch, or revision
- **package**.metrics.origin_url - the origin URL
- **package**.purl.type - the <type> field of the PURL
- **package**.purl.namespace - the <namespace> field of the PURL
- **package**.purl.name - the <name> field of the PURL
- **package**.purl.version - the <version> field of the PURL
- **package**.purl.subpath - the <subpath> field of the PURL
- **package**.license.declared - the declared package license identifier; e.g., GPLv3, MIT, Apache 2.0
- **package**.vulnerability.id - a specified vulnerability ID against a particular package
- **package**.vulnerability.url - a specified URL
- **package**.vulnerability.type - a specified vulnerability type against a particular package
- **package**.vulnerability.title - a specified vulnerability title against a particular package
- **package**.vulnerability.description - a specified vulnerability description against a particular package
- **package**.vulnerability.vector - a specified vulnerability CVSS attack vector against a particular package
- **package**.health_finding.category - a package health finding category
- **package**.health_finding.subcategory - a package health finding subcategory
- **package**.health_finding.severity - a package health finding severity level
- **package**.health_finding.description - a package health finding brief description
- **package**.health_finding.details - a package health finding longer details

- `contributor.stream_packages_summary` - a text string summarizing the `contributor.stream_packages` list of PURLs
- `contributor.package_behavior.package_id`- a text string of the BPT Package ID being tested in an alert model
- `contributor.package_behavior.package_name` - a text string of the name@version for the package being tested in an alert model
- `contributor.package_behavior.package_purl` - a text string of the PURL stub for the package being tested in an alert model
- **`package`**`.newer_versions.latest` - PURL of the latest version of this package
- **`package`**`.metrics.typosquatting.summary` - a comma-delimited list of packages that may be the legitimate package for a detected typosquatting case ### Sets of String Values
- `contributor.location.region` - contributor regions
- `contributor.location.country` - contributor countries
- `contributor.location.locale` - contributor locales (this is the first lower adminstrative area below country for the given country, region pair; e.g. states in USA)
- `contributor.organization` - contributor organization
- `contributor.stream_packages` - a list of PURLs this contributor has contributed to in the context of the alert
- **`package`**`.metrics.languages.names` - detected programming languages in a package
- **`package`**`.license.discovered` - discovered license identifiers in a package
- **`package`**`.newer_versions.list` - list of newer versions of this package

Arithmetic Values

- **`package`**`.metrics.issues.closed` - the number of issues closed in the last indexing cycle
- **`package`**`.metrics.issues.closed_rate` - the number of issues closed per day in the last indexing cycle
- **`package`**`.metrics.issues.closed_acceleration` - the rate of change of opened issues between subsequent indexing cycles
- **`package`**`.metrics.issues.open` - the number of issues opened in the last indexing cycle
- **`package`**`.metrics.issues.open_rate` - the number of issues opened per day in the last indexing cycle

- **package**.metrics.issues.open_acceleration - the rate of change of opened issues between subsequent indexing cycles
- **package**.metrics.issues.updated - the number of issues updated in the last indexing cycle
- **package**.metrics.issues.updated_rate - the number of issues updated per day in the last indexing cycle
- **package**.metrics.issues.updated_acceleration - the rate of change of updated issues between subsequent indexing cycles
- **package**.metrics.pull_requests.closed - the number of pull requests closed in the last indexing cycle
- **package**.metrics.pull_requests.closed_rate - the number of pull requests closed per day in the last indexing cycle
- **package**.metrics.pull_requests.closed_acceleration - the rate of change of opened pull requests between subsequent indexing cycles
- **package**.metrics.pull_requests.open - the number of pull requests opened in the last indexing cycle
- **package**.metrics.pull_requests.open_rate - the number of pull requests opened per day in the last indexing cycle
- **package**.metrics.pull_requests.open_acceleration - the rate of change of opened pull requests between subsequent indexing cycles
- **package**.metrics.pull_requests.updated - the number of pull requests updated in the last indexing cycle
- **package**.metrics.pull_requests.updated_rate - the number of pull requests updated per day in the last indexing cycle
- **package**.metrics.pull_requests.updated_acceleration - the rate of change of updated pull requests between subsequent indexing cycles
- **package**.metrics.pull_requests.merged - the number of pull requests merged in the last indexing cycle
- **package**.metrics.pull_requests.merged_rate - the number of pull requests merged per day in the last indexing cycle
- **package**.metrics.pull_requests.merged_acceleration - the rate of change of merged pull requests between subsequent indexing cycles
- **package**.metrics.languages.count - the total number of languages used in the package
- **package**.vulnerability_summary.counts.total - the total number of vulnerabilities
- **package**.vulnerability_summary.counts.low - the number of low-severity vulnerabilities

- **package**.vulnerability_summary.counts.medium - the number of medium-severity vulnerabilities
- **package**.vulnerability_summary.counts.moderate - the number of moderate-severity vulnerabilities
- **package**.vulnerability_summary.counts.high - the number of high-severity vulnerabilities
- **package**.vulnerability_summary.counts.critical - the number of critical-severity vulnerabilities
- **package**.vulnerability_summary.counts.unassigned - the number of vulnerabilities without an assigned severity
- **package**.vulnerability_summary.scores.cvss_average - the average CVSS score of vulnerabilities
- **package**.health_finding_summary.counts.total - the total number of health findings
- **package**.health_finding_summary.counts.informational - the number of informational health findings
- **package**.health_finding_summary.counts.low - the number of low-severity health findings
- **package**.health_finding_summary.counts.medium - the number of medium-severity health findings
- **package**.health_finding_summary.counts.high - the number of high-severity health findings
- **package**.health_finding_summary.counts.critical - the number of critical-severity health findings
- **package**.metrics.popularity.stars - the number of stars in the last indexing cycle
- **package**.metrics.popularity.stars_rate - the number of stars per day in the last indexing cycle
- **package**.metrics.popularity.stars_acceleration - the rate of change of stars between subsequent indexing cycles
- **package**.metrics.popularity.forks - the number of forks in the last indexing cycle
- **package**.metrics.popularity.forks_rate - the number of forks per day in the last indexing cycle
- **package**.metrics.popularity.forks_acceleration - the rate of change of forks between subsequent indexing cycles
- **package**.metrics.popularity.watchers - the number of watchers in the last indexing cycle
- **package**.metrics.popularity.watchers_rate - the number of watchers per day in the last indexing cycle

- **package**.metrics.popularity.watchers_acceleration - the rate of change of watchers between subsequent indexing cycles
- **package**.metrics.ossf_scorecard.score - the Open Source Security Foundation's aggregate project score for this package
- **package**.metrics.contributions.contributor_pool_size - the size of the contributor pool
- **package**.sbom.dependency_count - the total number of dependencies in the software bill of materials (SBOM)
- **package**.vulnerability.base_score - the base score of a vulnerability according to CVSS (latest applicable)
- **package**.vulnerability.exploit_score - the exploitability score of a vulnerability according to CVSS (latest applicable)
- **package**.vulnerability.impact_score - the impact score of a vulnerability according to CVSS (latest applicable)
- **package**.vulnerability.epss_score - the EPSS score of a vulnerability
- **package**.vulnerability.epss_percentile - the EPSS percentile of a vulnerability
- **package**.newer_versions.count - the count of newer versions of this package

Boolean Values

- **package**.metrics.is_fork - indicates if the package is a fork of another repository
- **package**.metrics.is_outdated - indicates if the package has any newer revisions
- **package**.metrics.scm.is_valid - indicates if the source code management (SCM) repository is valid
- **package**.license.no_license - indicates if the source code management (SCM) repository is valid
- contributor.package_behavior.is_maintainer - indicates the contributor is a maintainer of the package being tested
- contributor.package_behavior.is_new_maintainer - indicates the contributor is a new maintainer of the package being tested
- contributor.package_behavior.is_dependency_changer- indicates the contributor is a changer of the dependencies of the package being tested
- contributor.package_behavior.is_new_contributor- indicates the contributor is a new contributor of the package being tested
- **package**.metrics.ossf_scorecard.is_valid - indicates that the Open Source Security Foundation's aggregate project score for this package is valid or not
- **package**.metrics.typosquatting.detected - indicates that the package may be typosquatted for another package, see package.metrics.typosquatting.summary

- **package**.vulnerability.is_kev - indicates that the the current vulnerability appears on the CISA Known Exploited Vulnerability list

Date Values

- **package**.metrics.key_dates.created - the date when the package was created
- **package**.metrics.key_dates.updated - the date when the package was last updated