MERGE BASE

USER MANUAL

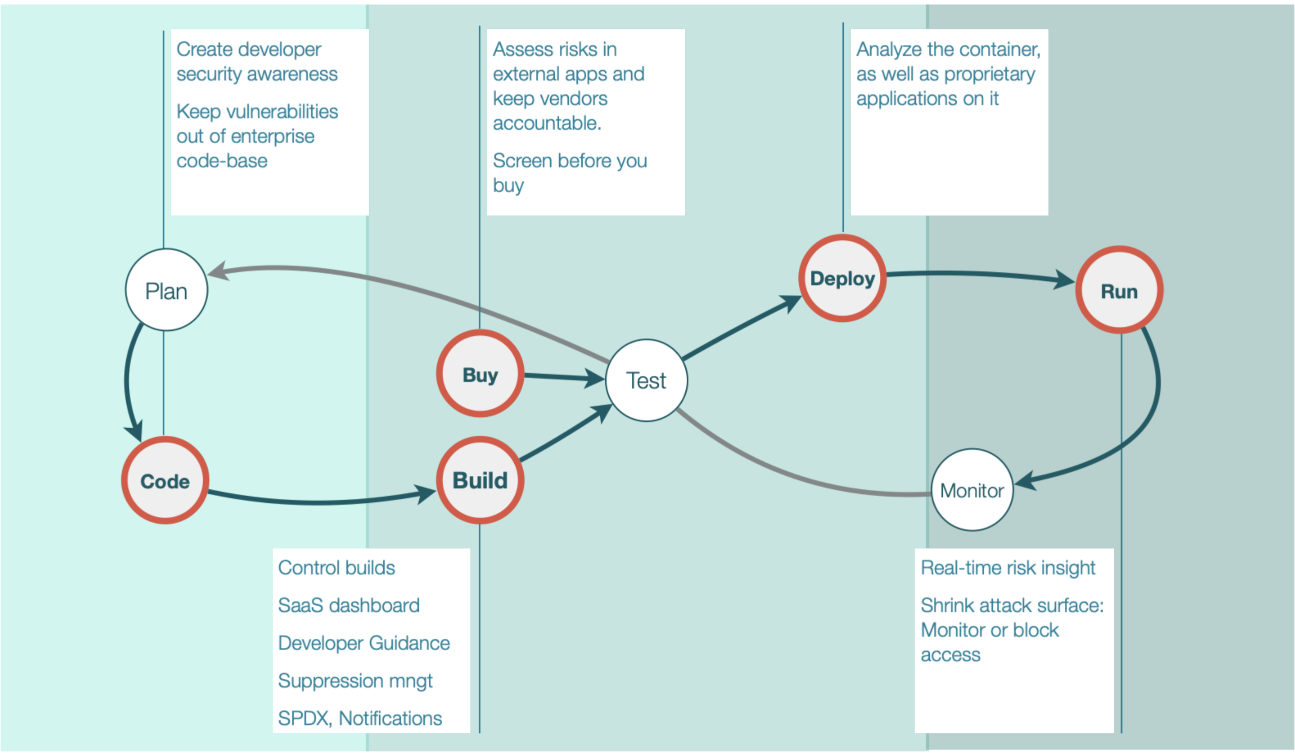
Table of Contents

Introduction

Eliminating vulnerabilities can overload the security team, decrease developer productivity, and increase the security budget. MergeBase frees developers through advanced developer guidance and auto-patching.

We dramatically reduce the load on the security team by instant visibility into true production risks, streamlined issue management, the industry's lowest false positive rate and automated risk mitigation solutions.

SCA analyzes applications for known vulnerabilities in their components and libraries. Such vulnerabilities are the primary cause of data data breaches. According to IBM, 57% of organizations do not know which vulnerabilities pose the highest risk. Gartner research concludes that 90% of all enterprises use open source software.


Create developer security awareness
Keep vulnerabilities out of enterprise code-base
Assess risks in external apps and keep vendors accountable.
Screen before you buy
Analyze the container, as well as proprietary applications on it
Plan
Deploy
Run
Buy
Test
Code
Build
Monitor
Control builds
SaaS dashboard
Developer Guidance
Suppression mngt
SPDX, Notifications
Real-time risk insight
Shrink attack surface: Monitor or block access

Awareness: MergeBase alerts developers to known vulnerabilities early in the development process, enabling overall cost savings and quick resolution.

Enterprise Controls: MergeBase prevents vulnerabilities from entering the enterprise code base.

Identification: MergeBase accurately identifies and reports vulnerabilities during the build and deployment process with very low false positive rates.

Control: MergeBase stops builds or deployments (containers) that contain vulnerabilities outside enterprise policy levels.

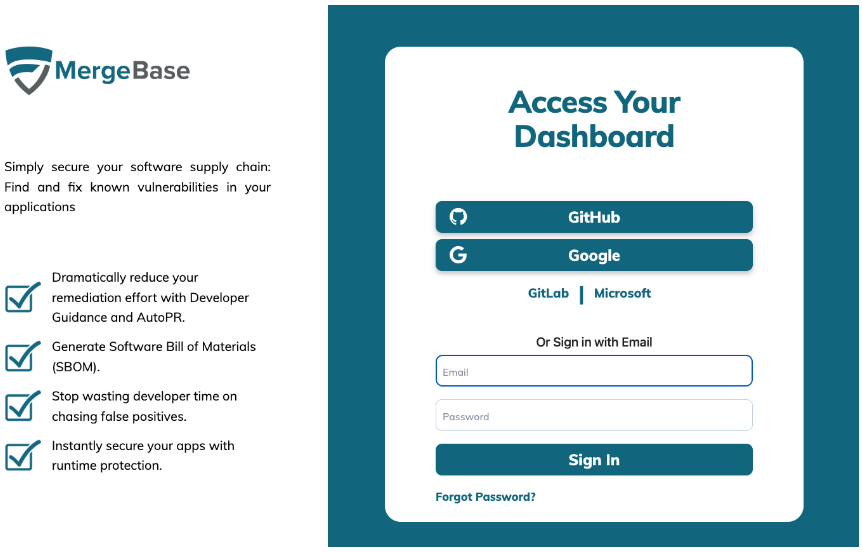
Visibility: MergeBase tracks application instances in all data centres, including cloud. It provides a complete and real-time overview of of both risk and unused, but exploitable, code.

Protection: MergeBase can instantly reduce vulnerabilities at run-time.

The Dashboard - Logging In

Free Trial Users

se the login method you selected when you registered, which is either a social login provider such as GitHub or Google, or email registration. Your dashboard server URL is generated by the registration process and will be emailed to you so you can add it to your bookmarks. Alternatively, you can login from mergebase.com and you will be redirected to your free trial dashboard once you are authenticated.



MergeBase

Simply secure your software supply chain: Find and fix known vulnerabilities in your applications

- Dramatically reduce your remediation effort with Developer Guidance and AutoPR.
- Generate Software Bill of Materials (SBOM).
- Stop wasting developer time on chasing false positives.
- Instantly secure your apps with runtime protection.

Access Your Dashboard

Login Page Screen

MergeBase Clients

MergeBase clients are provided with custom URLs. When a user accesses the URL, they will be redirected to the login service, which can be managed by MergeBase or, for customer managed single sign-on, integrated with a corporate directory server/identity provider (IdP) that supports SAML 2.0.

Customer Managed Single Sign-on

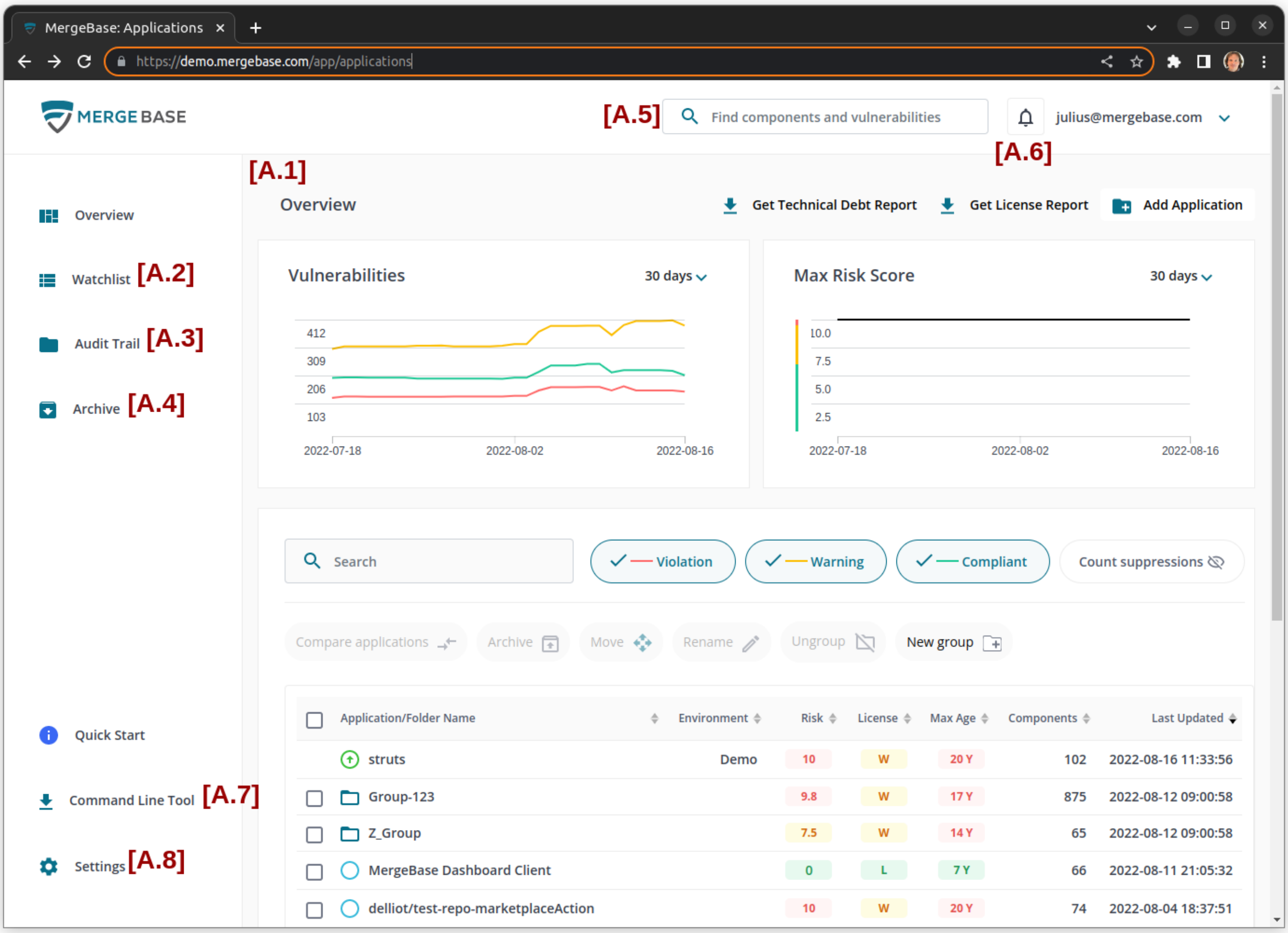
MergeBase integrates with any IdP that supports SAML 2.0 including Okta, OneLogin, and Google Workspace. Once MergeBase is integrated with the IdP, authorized users can open the MergeBase dashboard directly from their IdP dashboard or using the custom MergeBase URL.

MergeBase Managed

MergeBase can provision users on behalf of the customer. The customer provides a list of authorized users and MergeBase will create user accounts, sending an email to each user with a link to access the dashboard and a temporary password.

After entering the temporary password, each user will enter a permanent password and then be redirected to the customer's MergeBase URL to see their MergeBase dashboard.

Home Page and Navigation

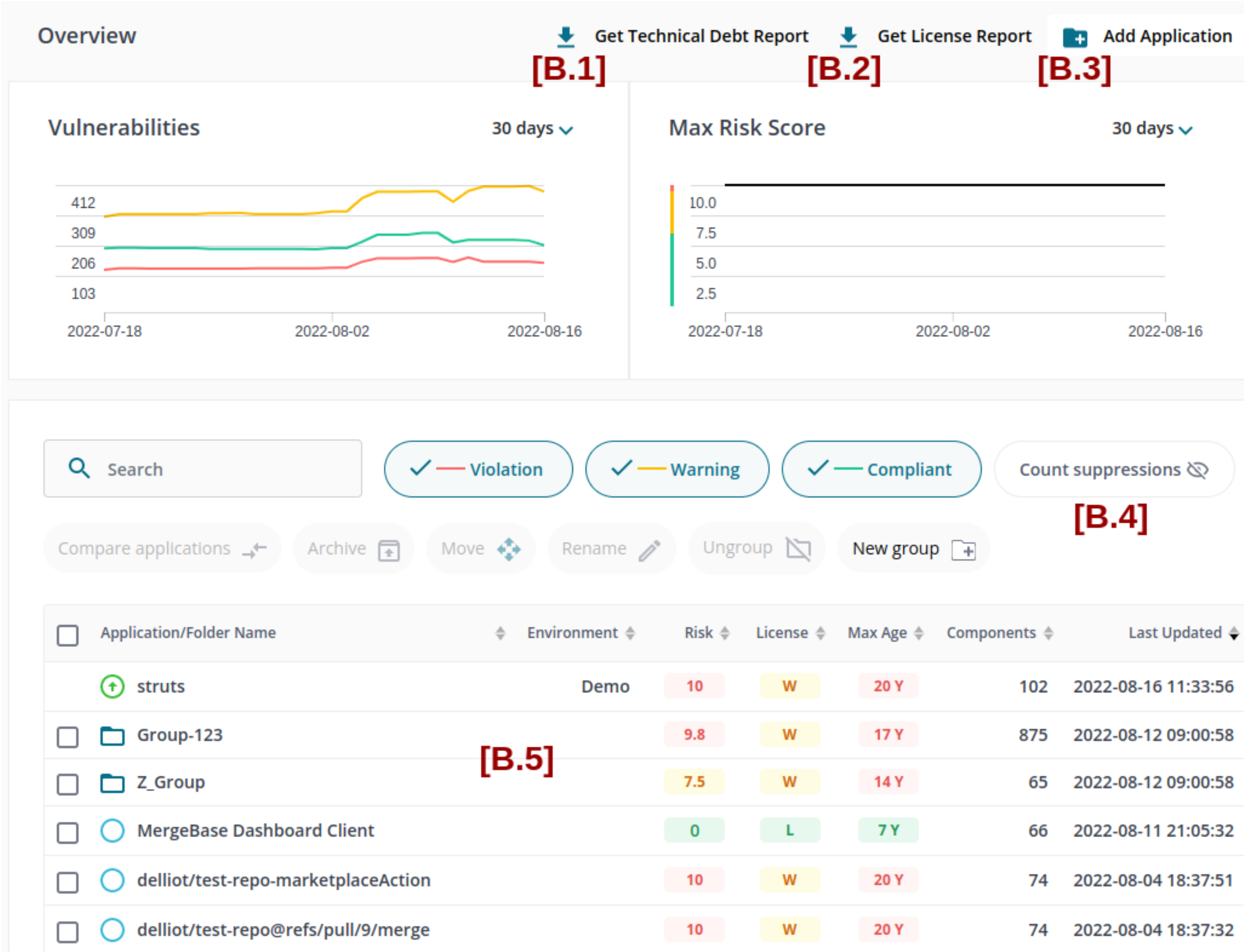


□	Application/Folder Name	Environment	Risk	License	Max Age	Components	Last Updated
	struts	Demo	10	W	20 Y	102	2022-08-16 11:33:56
□	Group-123		9.8	W	17 Y	875	2022-08-12 09:00:58
□	Z_Group		7.5	W	14 Y	65	2022-08-12 09:00:58
□	MergeBase Dashboard Client		0	L	7 Y	66	2022-08-11 21:05:32
□	delliot/test-repo-marketplaceAction		10	W	20 Y	74	2022-08-04 18:37:51

The MergeBase dashboard home page provides an overview of all the applications scanned and shows how risks are evolving over time. It consists of the main functional area (A.1), and navigation elements or utilities in the header and left sidebar (A.2 to A.8).

Applications Overview

The Overview shows a set of graphs summarizing risk and vulnerability trends above a table of scanned and inoculated applications. A scanned application is produced by scanning an application binary or source code. An inoculated application is produced by applying MergeBase run-time protection to an application binary. For complete details on the difference between scanned and inoculated applications and how to create them, see the Command Line Tool section.



□	Application/Folder Name	Environment	Risk	License	Max Age	Components	Last Updated
	struts	Demo	10	W	20 Y	102	2022-08-16 11:33:56
□	Group-123		9.8	W	17 Y	875	2022-08-12 09:00:58
□	Z_Group		7.5	W	14 Y	65	2022-08-12 09:00:58
□	MergeBase Dashboard Client		0	L	7 Y	66	2022-08-11 21:05:32
□	delliot/test-repo-marketplaceAction		10	W	20 Y	74	2022-08-04 18:37:51
□	delliot/test-repo@refs/pull/9/merge		10	W	20 Y	74	2022-08-04 18:37:32

Obsolescence Report [B.1]

The Get Obsolescence Report button allows the user to download a CSV formatted report of obsolete components. There are two types of obsolete components: ones that have an age exceeding the *update* threshold and have a more recent release; and ones that have an age exceeding the *abandon* threshold and do not have a more recent release. The default values for these thresholds are ‘12 months’ for *update* and ‘48 months’ for *abandon*. The values can be configured in the Risk Policies section of the Setting page.

Notes:

- The *update* threshold only triggers if the current component is older than the threshold *and* there is a newer release available. E.g., “Upgrade because this version is 13 months old (1 month past the 12 month threshold) and a newer version is available.”

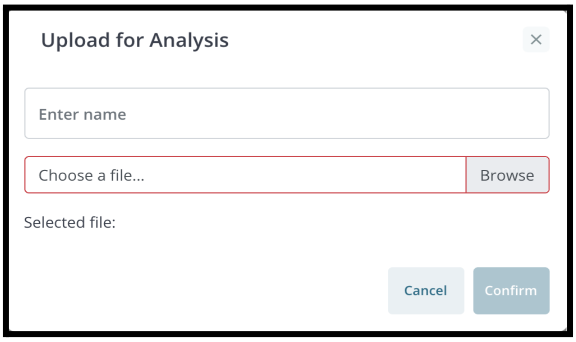
- The *abandon* threshold ignores the current component version, and only triggers if the *newest known version* is older than the *abandon* threshold. E.g., “Abandon because there have been no releases in the last 48 months.”

License Report [B.2]

The Get License Report button allows the user to download a report of the licenses used by all open-source components in all applications. The report is in CSV format suitable for importing into a spreadsheet, for example.

Add Application [B.3]

The Upload for Analysis function allows the user to upload a zip file or jar file for scanning. The file cannot exceed 200 MB. The upload feature is limited to analyzing compiled Java code. To scan source code in multiple languages, you must use the Command Line Tool.



Upload for Analysis

Enter name

Choose a file...

Selected file:

Application Filters [B.4]

The application filters allow the user to filter the application list and graph data. There are three filters:

- Application name - filter by application name (case insensitive). Filters the list dynamically (as the user types). However, the graph data will not be updated until the user presses the enter key or clicks outside the filter.
- Environment - filter by environment label. This is a drop down list.
- Include mitigated and accepted risks - if checked, the graph data will include all vulnerabilities regardless of their risk mitigation status. Otherwise, the graphs are adjusted to take into account mitigations and risk suppression. See Application Detail - Inoculated Applications below.

Application Table [B.5]

The following information is displayed for each application (scan or inoculation):

- **State**: indicates whether the application is purely a scan (blue), a live inoculated application (green), or a “stale” inoculated application (yellow). An application is considered stale if it has not contacted the dashboard for one hour.
- **Application name**: the name provided when the application was scanned or inoculated
- **Risk:** a global risk score (based on the known vulnerabilities) and a license policy violation indicator (yellow = one or more policy warnings, red = one or more policy violations)
- **Environment**: for scanned applications, this is always “SCAN”. For inoculated applications, this is the value of the environment variable assigned in the running application (see Running Inoculated Applications below)
- **Components**: a count of the components used by the application
- **Vulnerabilities**: a count of the vulnerabilities detected
- **Last updated**: for scans, the time of the last scan; for inoculated applications, the time of the last communication between the dashboard and the inoculated application.

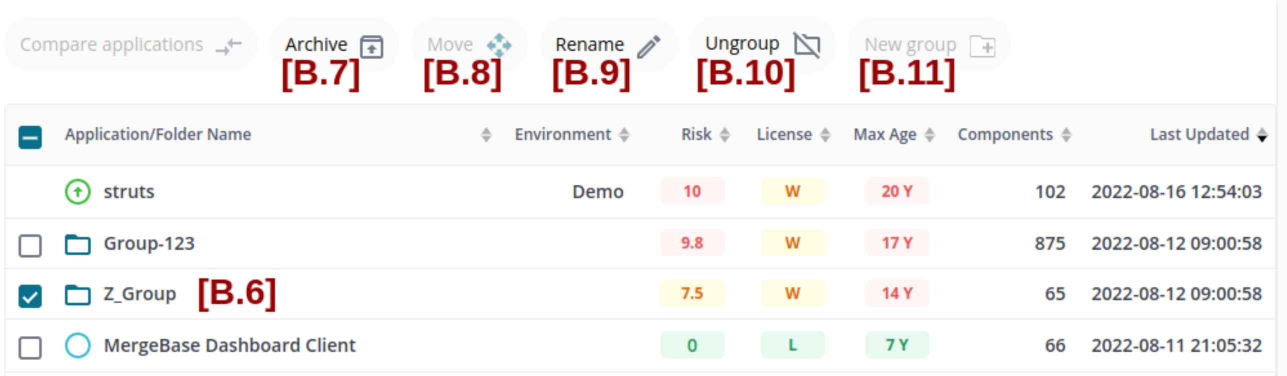
The table can be sorted by clicking on the column headers. Users can also filter the list by typing part of an application name (case insensitive) or selecting an environment label from the Environments dropdown menu.

Users click on a table row to view the details of an individual application.

Scanned applications and “stale” inoculated applications can be archived using the popup menu at the right end of each table row. Note that old scans will be automatically archived when new scans are uploaded. See the section on Automatic Archiving below.

Note that the combination of name and environment is a unique identifier for an application.

Application and Group Controls



	Application/Folder Name	Environment	Risk	License	Max Age	Components	Last Updated
	struts	Demo	10	W	20 Y	102	2022-08-16 12:54:03
□	Group-123		9.8	W	17 Y	875	2022-08-12 09:00:58
☑	Z_Group		7.5	W	14 Y	65	2022-08-12 09:00:58
□	MergeBase Dashboard Client		0	L	7 Y	66	2022-08-11 21:05:32

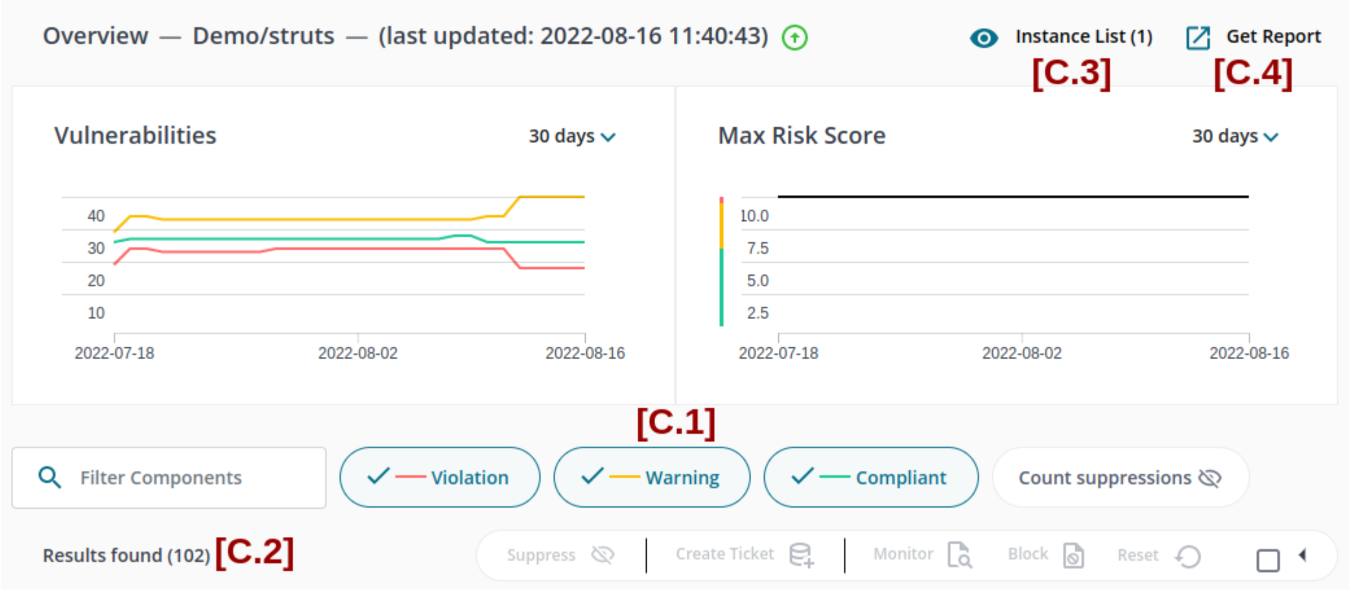
The Application and Group controls allow the user to perform the following actions:

- archive one or more applications and groups
- move one or more applications to a group
- rename an application or a group
- ungroup a group – moves all the applications in the group to the top level and removes the group
- create a new group

Application Detail View

The application detail replicates the summary graphs of the Overview but applied to the selected application only. The table below the graphs contains the list of components found in the application.

eFilters and Reports



This view provides controls [C.1] to sort and filter the table [C.2] similar to the Applications Overview. The Filter Components input allows the user to filter the component list in a similar fashion to the Applications filter in the Overview. In addition, there are three checkboxes:

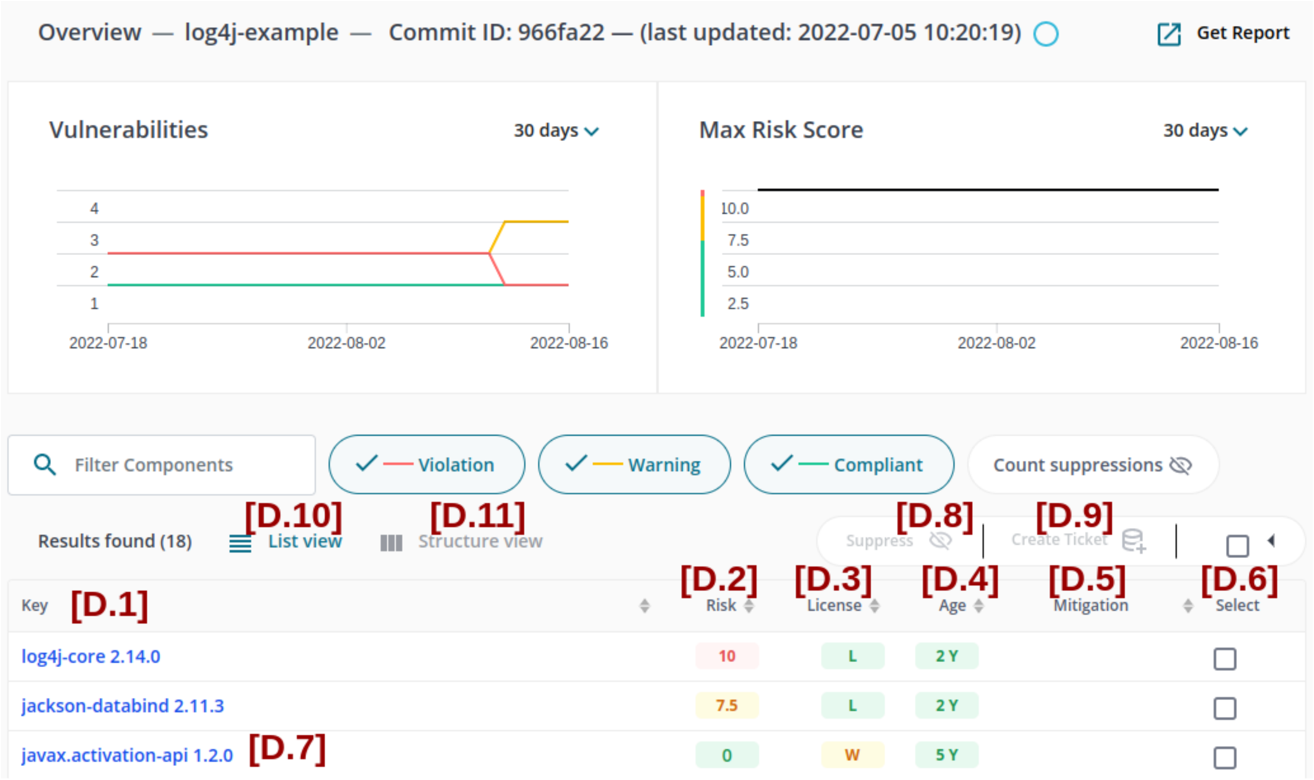
- Show vulnerable only - show only components that have vulnerabilities
- Show license warnings only - show only components that are not permitted by your organization's license policy (see License Policies in the Settings section below).
- Include mitigated and accepted risks - identical to the filter of the same name in the Applications Overview.

Finally, there are two reports: the list of active instances [C.3] and various reports [C.4]. The active instance report is only available for inoculated applications.

Component List and Mitigation Controls

The details of the component list and mitigation controls depend on whether the application is a read-only scan or an inoculated application reporting to the dashboard in real-time.

Application Detail - Scanned Applications - List View



Key	Risk	License	Age	Mitigation	Select
log4j-core 2.14.0	10	L	2 Y		□
jackson-databind 2.11.3	7.5	L	2 Y		□
javax.activation-api 1.2.0	0	W	5 Y		□

For scanned applications the dependencies can be viewed in either **[D.10] list view** or **[D.11] structured (tree) view**. When using the list view the following information is displayed for each component:

- **[D.1] - Key**: the common name and version used for the component
- **[D.2] - Risk**: the risk level associated with the component
- **[D.3] - License**: license policy compliance status associated with the component
- **[D.4] - Age**: the age of the component in years based on the original release date
- **[D.5] - Mitigation**: if Jira integration is enabled, this column will show if a ticket has been created for this component
- **[D.6] - Select**: allows the user to select multiple components to associate with a Jira ticket

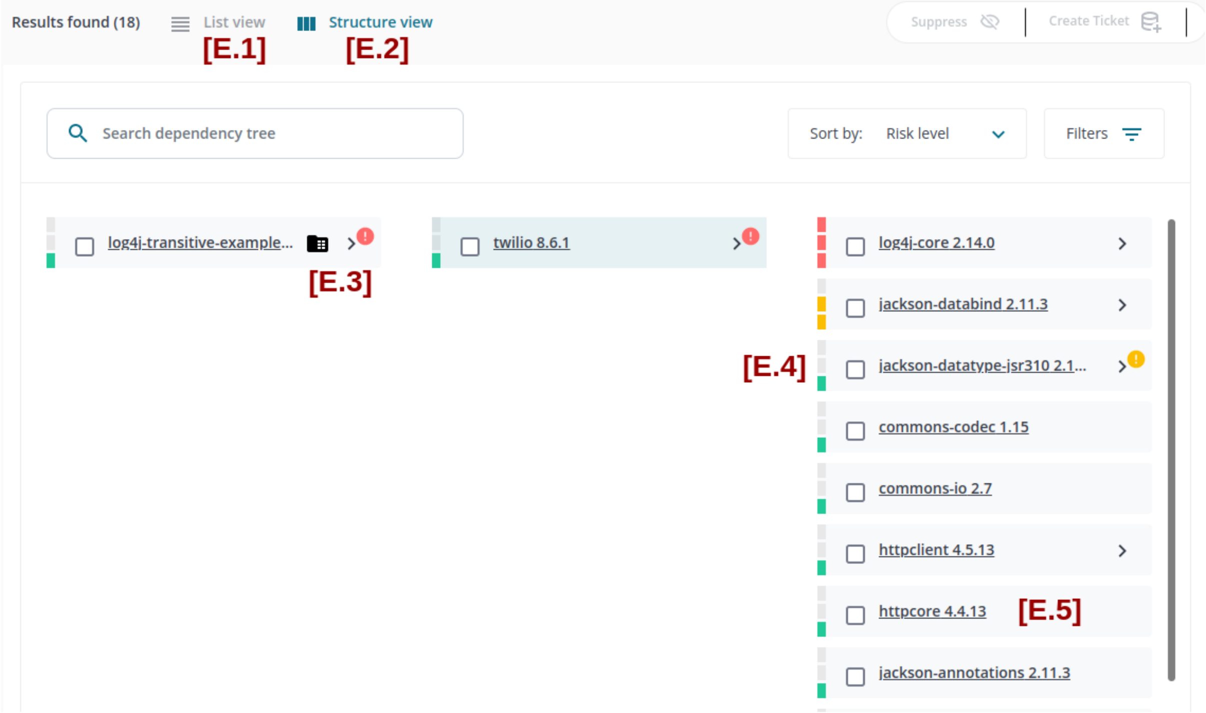
Note regarding "**[D.7] - Special components**" - Unknown components and *composite* components (components the are comprised of multiple other components) are indicated with a double-asterisk ("**") within their Key. Hovering the mouse over the double-asterisk provides additional information about these special components.

Suppressing Vulnerabilities; Creating Jira Tickets

For a given component, users can temporarily suppress associated risk and license policy violations. They can also create a Jira ticket to assign and track patching work to a development team.

- **[D.8] - Suppress**: clicking on “Suppress” tells MergeBase to temporarily ignore (stop reporting) the current known vulnerabilities for the selected components. A reason and expiry time must be provided on the dialog box that pops up before the suppression is activated.
- **[D.9] - Create Ticket**: clicking on “Create Ticket” causes MergeBase to connect to your company’s Jira instance to create patching tickets for the selected components (and their corresponding vulnerabilities).

Application Detail - Scanned Applications - Structured View (Tree View)



By default (after clicking on an application scan) the “list view” is presented, but users can toggle between the [E.1] list view and [E.2] structure view. Here we show a screenshot of the structure view.

The structure view uses a “Miller List” UI structure to help users understand the hierarchical dependency structure of their application. This structure view can be especially helpful when trying to determine what needs to change to execute a patch. E.g., Can we upgrade a vulnerable dependency directly? Or, since it comes in via transitive relationships, should we upgrade the parent dependency?

[E.3] - Sub-Dependency Expansion: click on the “>” icon to view a dependency’s sub-dependencies in the next column of the miller list. A small “!” icon (in a red circle) is displayed if vulnerabilities are present in sub-dependencies (or sub-sub dependencies, etc).

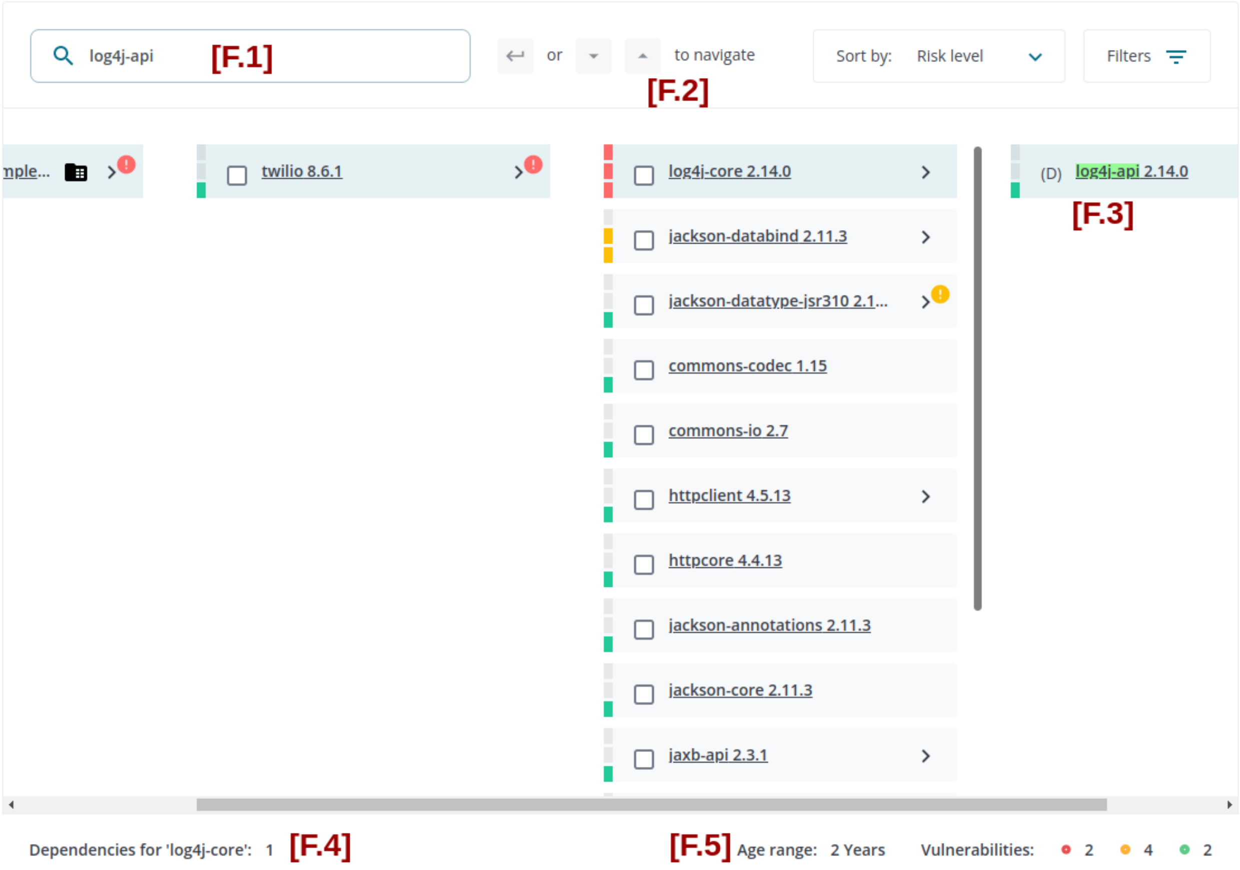
[E.4] - Vulnerable Sub-Dependencies: entries with a green indicator but a (!) yellow or red far-right indicator signify that while the dependency itself is not vulnerable, it’s sub-dependencies are.

[E.5] - Terminal Dependencies: entries lacking the “>” icon have no sub-dependencies.

Each item in the miller list is also decorated with a risk indicator (1 green square = OKAY, 2 orange squares = WARNING, 3 red squares = VIOLATION). The risk thresholds are based on vulnerability CVSS scores, and the OKAY / WARNING / VIOLATION thresholds can be adjusted globally in MergeBase’s settings. Note: the risk indicator only pertains to the dependency itself - vulnerable sub-dependencies are not incorporated into the risk indicator (vulnerable sub-dependencies are denoted via the “!” red circle instead).

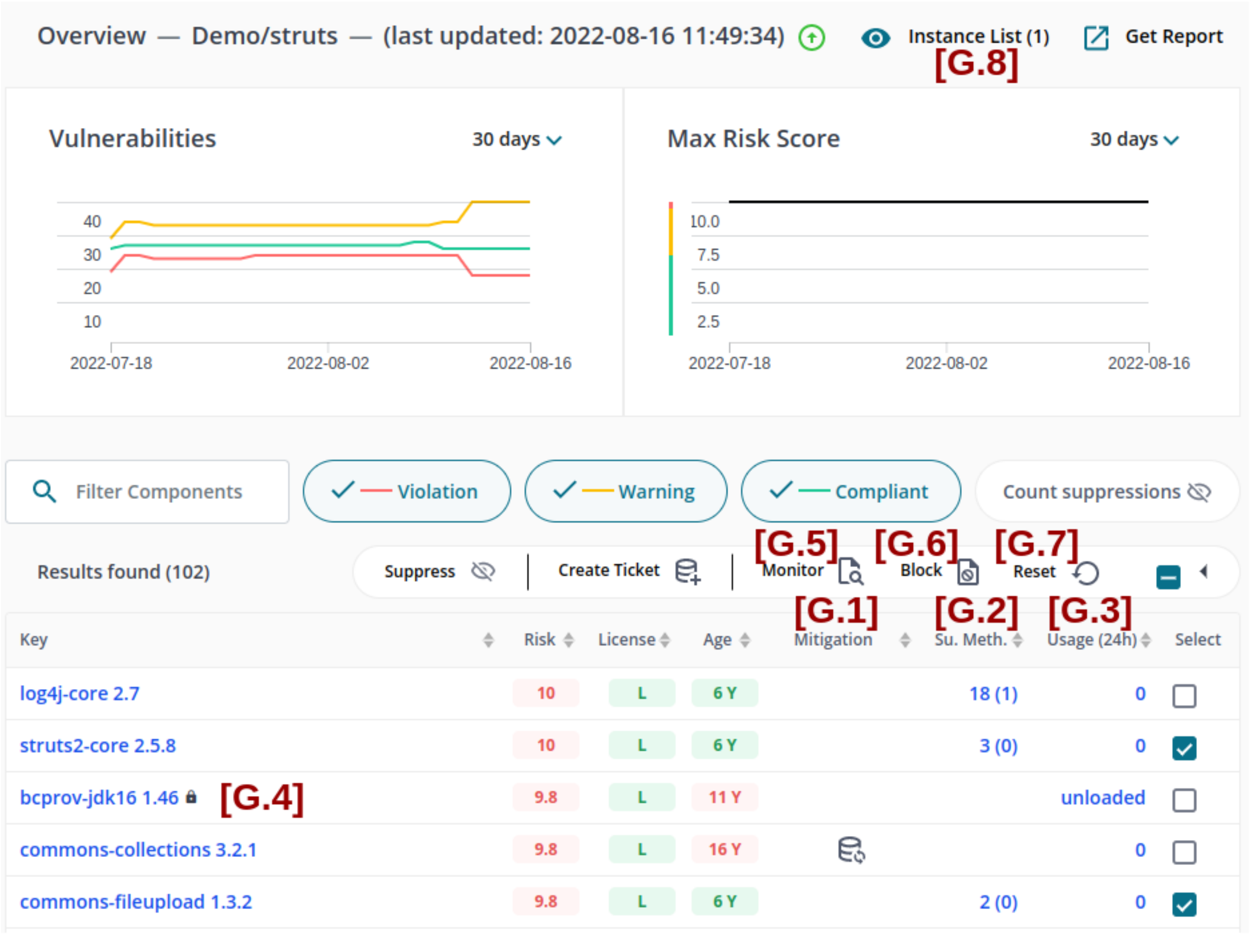
Items marked with a “(D)” are duplicate entries, and that same dependency occurs in other places throughout the dependency tree.

Application Detail - Scanned Applications - Structured View (Tree View) - Filtering



Users can quickly navigate large structure view using the **[F.1] Filter Control.** Type the text you wish to use to filter components and then use your keyboard's [ENTER], [DOWN], and [UP] arrows to **[F.2] navigate** through **[F.3] matches**. The navigation order is based on BFS (Breadth First Search). This means matching dependencies at depth 1 are matched first, and then dependencies at depth 2, and so on.

Application Detail - Inoculated Applications



Overview — Demo/struts — (last updated: 2022-08-16 11:49:34)

Results found (102)

Key	Risk	License	Age	Mitigation	Su. Meth.	Usage (24h)	Select
log4j-core 2.7	10	L	6 Y		18 (1)	0	☐
struts2-core 2.5.8	10	L	6 Y		3 (0)	0	☑
bcprov-jdk16 1.46 🔒	9.8	L	11 Y			unloaded	☐
commons-collections 3.2.1	9.8	L	16 Y			0	☐
commons-fileupload 1.3.2	9.8	L	6 Y		2 (0)	0	☑

For inoculated applications, some additional information is provided:

- **[G.1] - Mitigation:** in addition to Jira, this column will display a mitigation status symbol if the component is monitored or blocked.
- **[G.2] - Susp. Meth**: the number of suspicious methods/functions (e.g., “29 (0)”). The number to the left (not in parentheses) represents the total number of suspicious methods identified. The number to the right (in parentheses) represents the number with mitigations applied (either monitoring or blocking).
- **[G.3] - Usage (24h):** the number of times the component was used in the past 24 hours (or “unloaded” if the component is not currently loaded within its running application)

Regarding **“[G.4] - Signed Components,”** both scanned and inoculated applications might include a small lock icon within the artifact-id. This indicates that the artifact is signed and cannot be inoculated by MergeBase unless a re-signing certificate is obtained. Blocking and monitoring mitigations are not available for locked (signed) components.

Setting a Mitigation

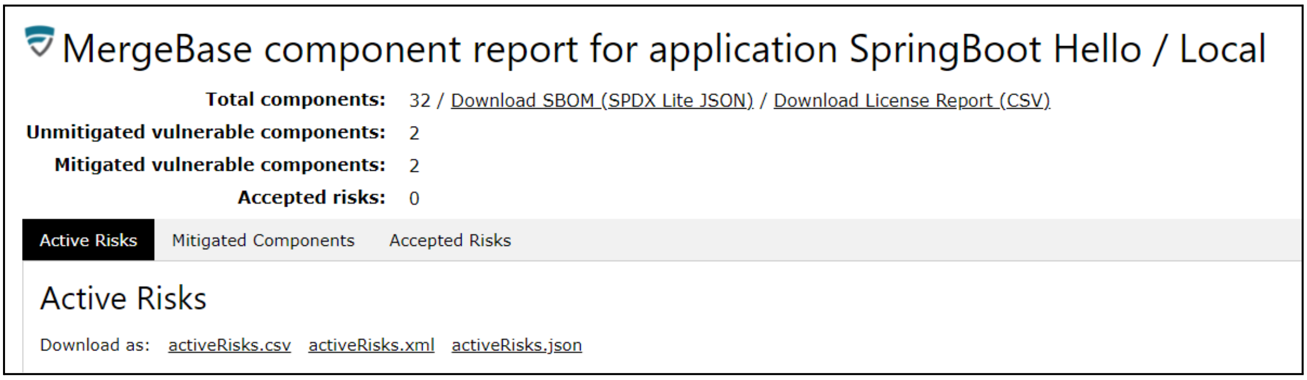
If the application being viewed is inoculated, the user can set mitigations on individual components. The mitigations are:

- **[G.5] - Monitor**: record detailed usage information for the component
- **[G.6] - Block**: prevent the component from being used (application users may see an error)
- **[G.7] - Reset:** unset all previous “monitor” and “block” mitigations previously set on the selected components.

Note that setting a mitigation may take some time to confirm while the dashboard waits for the inoculated application to acknowledge the change.

You can click on the **[G.8] - Instance List** to review all application instances (including IP addresses, mac addresses, etc) that would be affected by the mitigations.

View Report



MergeBase component report for application SpringBoot Hello / Local

Total components: 32 / Download SBOM (SPDX Lite JSON) / Download License Report (CSV)
Unmitigated vulnerable components: 2
Mitigated vulnerable components: 2
Accepted risks: 0

Active Risks

Download as: activeRisks.csv activeRisks.xml activeRisks.json

The user can also click on "View Report" to see a summary report of vulnerabilities. The report will open in a separate browser tab.

Formats

The full report can be generated in SBOM (SPDX Lite JSON) or CSV format. You can also view an HTML report for each of the sub-reports on the report page.

There is also an option to generate an SBOM with a VEX report. This VEX report allows you to generate analysis fields in your SBOM. These fields can be filled out in your component detail view as detailed in the Component Detail View section.

Sub-reports

There are currently three sub-reports which can be generated:

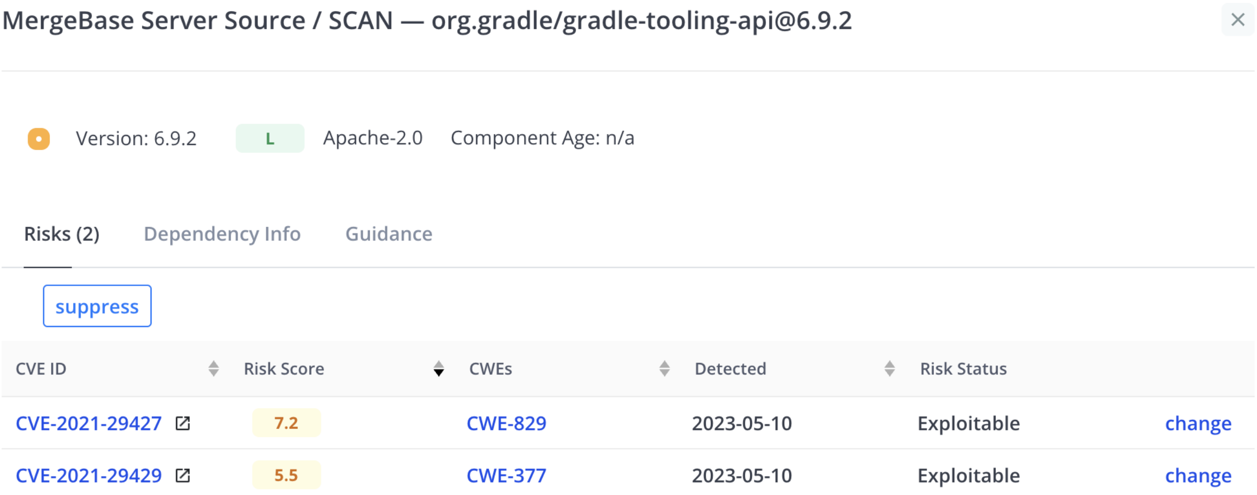
- **Active Risks:** The Active Risk report lists all current vulnerabilities in the application which have not have not been accepted as an acceptable risk or had a mitigation applied to them.
- **Mitigated Components**: This sub-report shows all components which have had a "Block" or "Monitor" mitigation set by the user.
- **Accepted Risks**: This sub-report lists all components which are currently marked as an acceptable level of risk by the user.

All sub-reports are available as CSV, XML, or JSON.

Component Detail View

For scanned or inoculated applications, the user can click on the component name to display a window with component detailed information, including the open-source license identifier (e.g. Apache-2.0) and the original release date.

Component Detail - Source Scans



MergeBase Server Source / SCAN — org.gradle/gradle-tooling-api@6.9.2

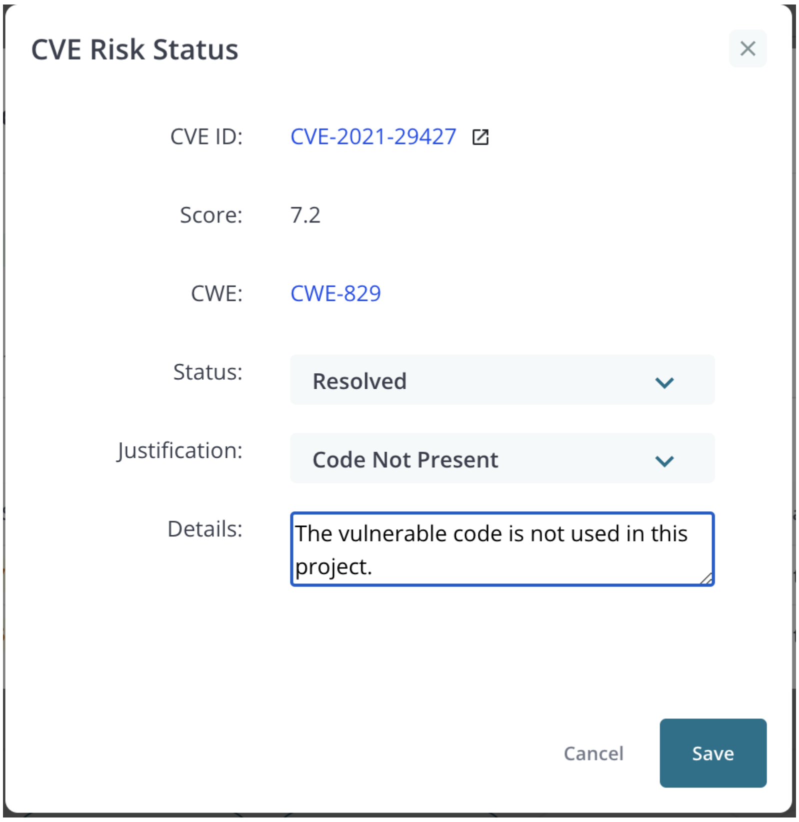
Version: 6.9.2 L Apache-2.0 Component Age: n/a

CVE ID	Risk Score	CWEs	Detected	Risk Status	
CVE-2021-29427	7.2	CWE-829	2023-05-10	Exploitable	change
CVE-2021-29429	5.5	CWE-377	2023-05-10	Exploitable	change

Component Detail - Source Scan

For source scans, the Component Detail view displays the version information including the component age and release date. There are four tabs with more data: Risks, Suspicious Methods, Dependency Info, and Guidance.

The Risk Status field is configurable by clicking the change link.



CVE Risk Status

CVE ID: CVE-2021-29427

Score: 7.2

CWE: CWE-829

Status: Resolved

Justification: Code Not Present

Details: The vulnerable code is not used in this project.

Select the status of your vulnerability from the Status dropdown, the reason for the status selected from the Justification dropdown, and then add a short description of the reason for your selections. These fields will populate the VEX analysis fields that are included in the SBOM + VEX report.

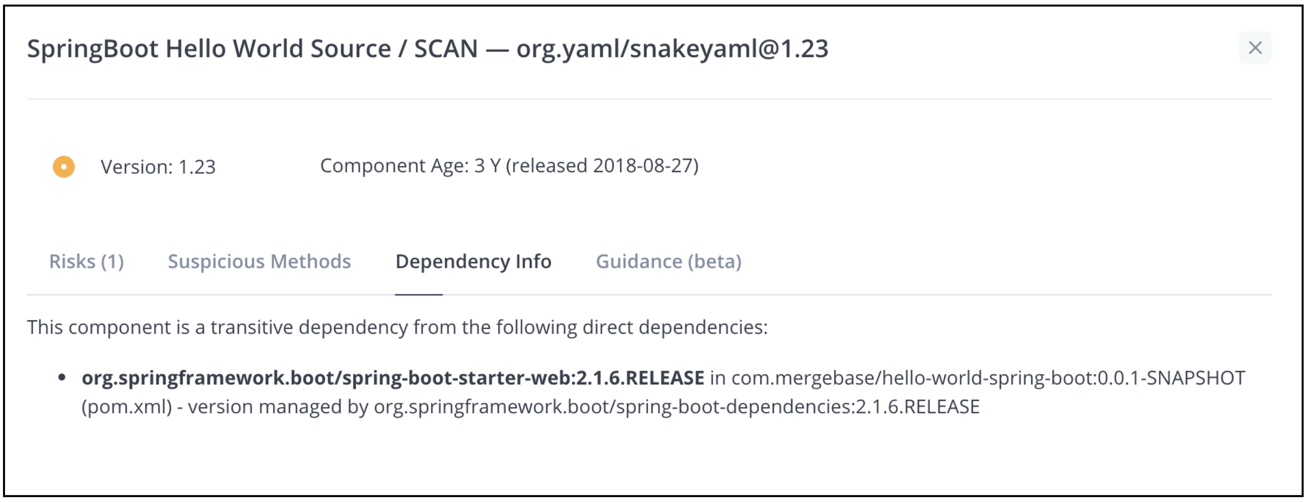
Risks Tab

For vulnerabilities sourced from the NVD database, the risks view provides a link to the complete vulnerability description at nvd.nist.gov (opens in a new window). This view also provides controls to accept risks for specific vulnerabilities. If a risk is accepted, the user must provide a reason. Once a risk has been accepted, the vulnerability counts displayed in other views will be adjusted accordingly.

The user can also suppress all the risks for the component using the “suppress” button, which works the same way as the “suppress” button in the Application Detail view. If the component is already suppressed, the “suppress” button is replaced with “remove” and “edit” buttons.

Dependency Info

This tab shows whether the component is included in the application as a direct dependency or if it is included because it is a transitive dependency. For Java applications using Maven, the information includes the pom.xml file that contains the direct dependency and, if the component version is managed, the location of the dependency management.


SpringBoot Hello World Source / SCAN — org.yaml/snakeyaml@1.23

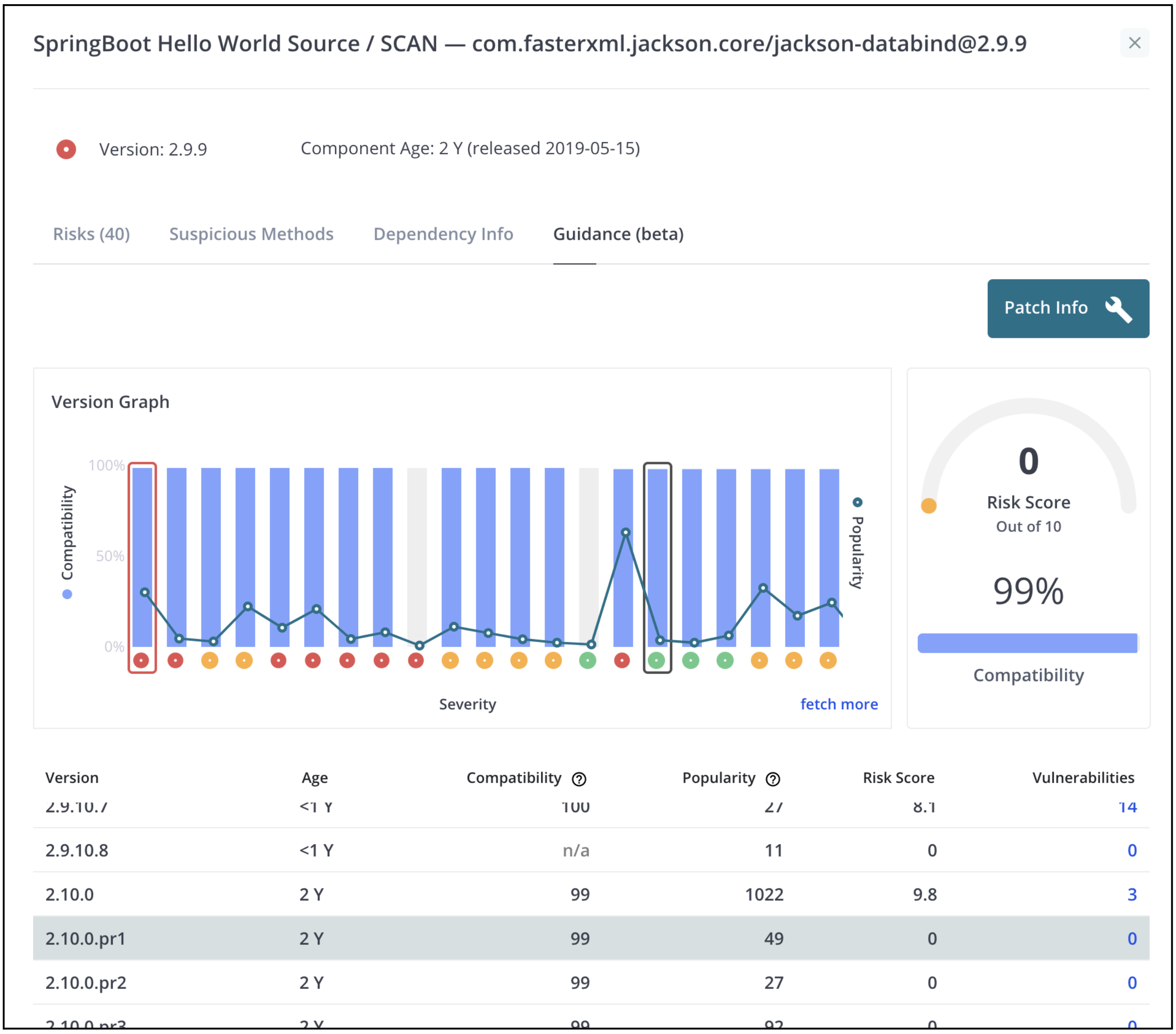
Version: 1.23

Component Age: 3 Y (released 2018-08-27)

This component is a transitive dependency from the following direct dependencies:

- **org.springframework.boot/spring-boot-starter-web:2.1.6.RELEASE** in com.mergebase/hello-world-spring-boot:0.0.1-SNAPSHOT (pom.xml) - version managed by org.springframework.boot/spring-boot-dependencies:2.1.6.RELEASE

Guidance



SpringBoot Hello World Source / SCAN — com.fasterxml.jackson.core/jackson-databind@2.9.9

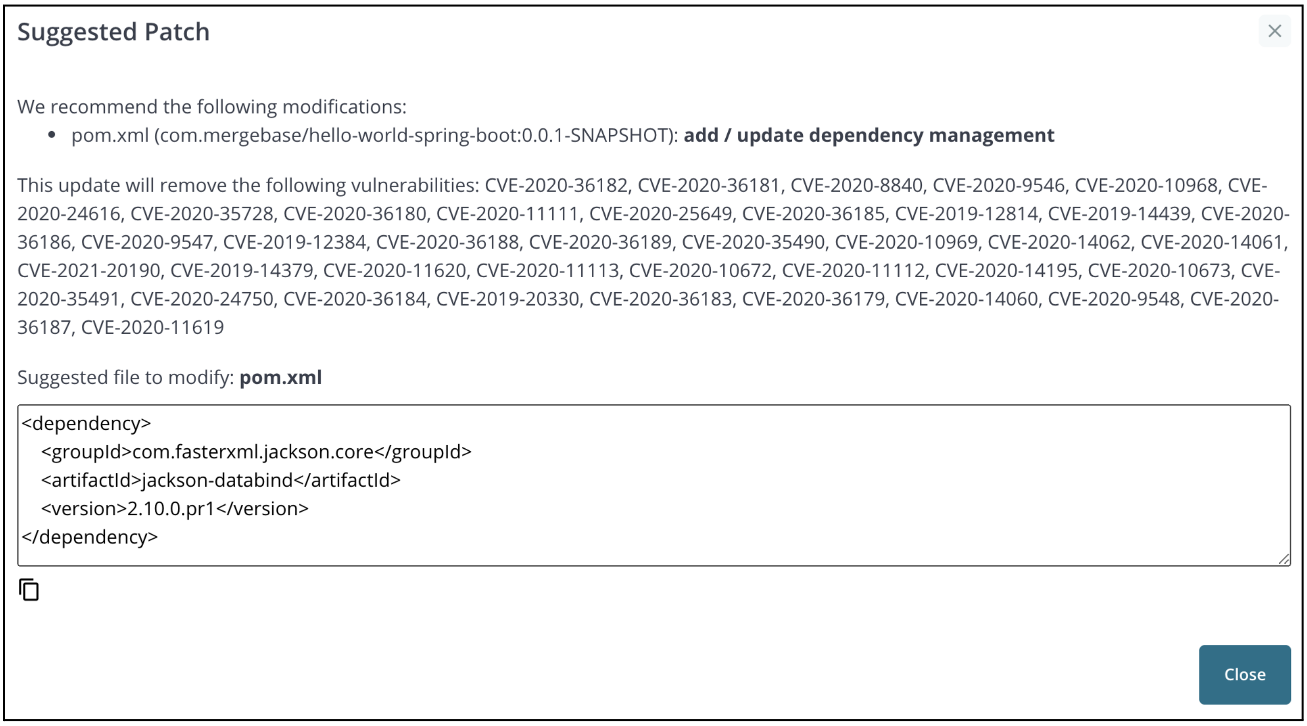
Version: 2.9.9 Component Age: 2 Y (released 2019-05-15)

Version	Age	Compatibility ⓘ	Popularity ⓘ	Risk Score	Vulnerabilities
2.9.10.7	<1 Y	100	27	8.1	14
2.9.10.8	<1 Y	n/a	11	0	0
2.10.0	2 Y	99	1022	9.8	3
2.10.0.pr1	2 Y	99	49	0	0
2.10.0.pr2	2 Y	99	27	0	0
2.10.0.pr3	2 Y	99	92	0	0

This tab provides information on potential upgrades. The user is presented with a chart showing versions that follow the current version. Below the chart is a table with information about each version. The user can click on a bar to scroll the table to the selected version. The popularity score is based on the number of open source projects that currently use that version of the component. The compatibility score is based on the difference in the public API of the current version versus the selected version. The user can click on the vulnerability count to expand the row to show all the vulnerabilities associated with that version (if any).

Patch Info

In addition to scrolling the table, clicking on a bar will enable the “Patch Info” button. Clicking on the button will display a window with information to upgrade the component.



Suggested Patch

We recommend the following modifications:

- pom.xml (com.mergebase/hello-world-spring-boot:0.0.1-SNAPSHOT): **add / update dependency management**

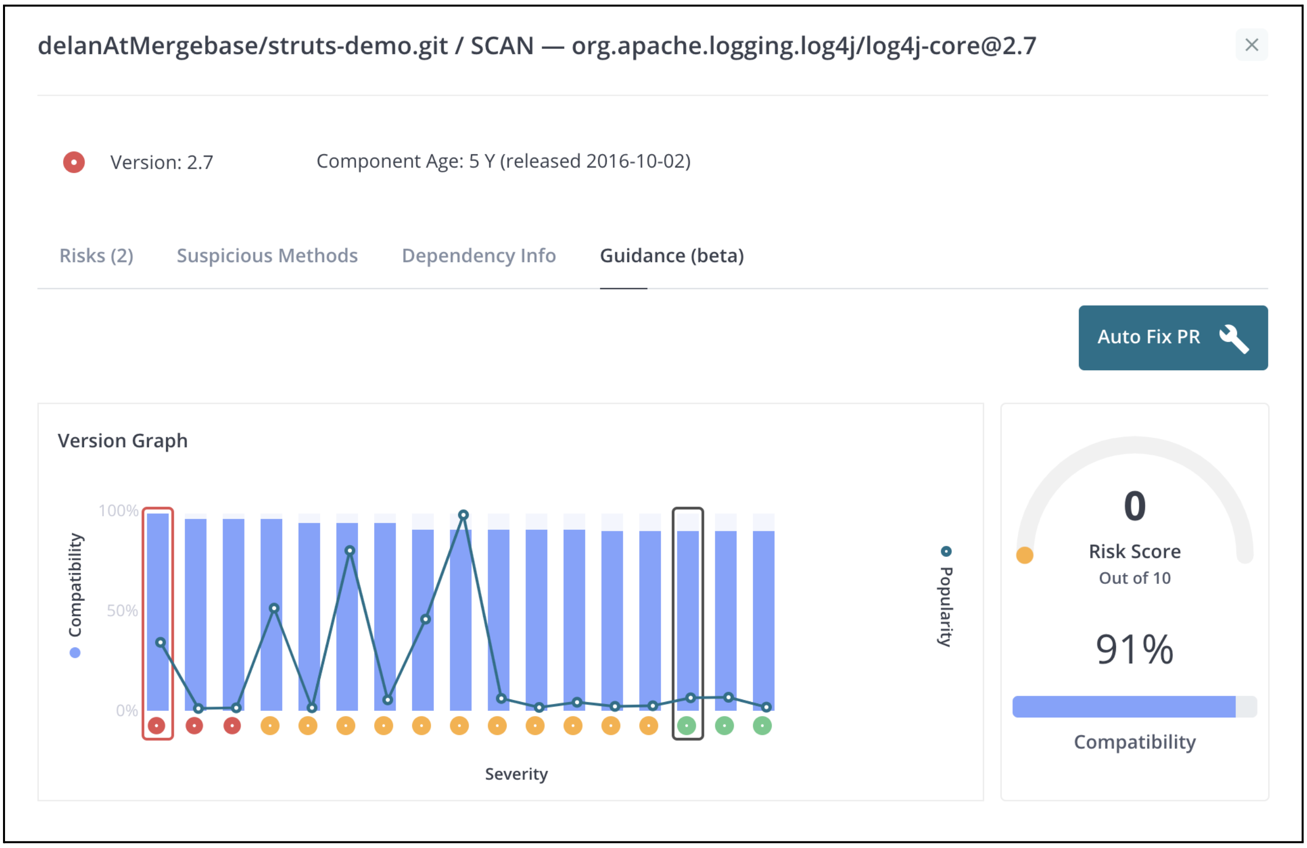
This update will remove the following vulnerabilities: CVE-2020-36182, CVE-2020-36181, CVE-2020-8840, CVE-2020-9546, CVE-2020-10968, CVE-2020-24616, CVE-2020-35728, CVE-2020-36180, CVE-2020-11111, CVE-2020-25649, CVE-2020-36185, CVE-2019-12814, CVE-2019-14439, CVE-2020-36186, CVE-2020-9547, CVE-2019-12384, CVE-2020-36188, CVE-2020-36189, CVE-2020-35490, CVE-2020-10969, CVE-2020-14062, CVE-2020-14061, CVE-2021-20190, CVE-2019-14379, CVE-2020-11620, CVE-2020-11113, CVE-2020-10672, CVE-2020-11112, CVE-2020-14195, CVE-2020-10673, CVE-2020-35491, CVE-2020-24750, CVE-2020-36184, CVE-2019-20330, CVE-2020-36183, CVE-2020-36179, CVE-2020-14060, CVE-2020-9548, CVE-2020-36187, CVE-2020-11619

Suggested file to modify: **pom.xml**

```
<dependency>
   <groupId>com.fasterxml.jackson.core</groupId>
   <artifactId>jackson-databind</artifactId>
   <version>2.10.0.pr1</version>
</dependency>
```

Auto Fix Pull Requests (Maven only)

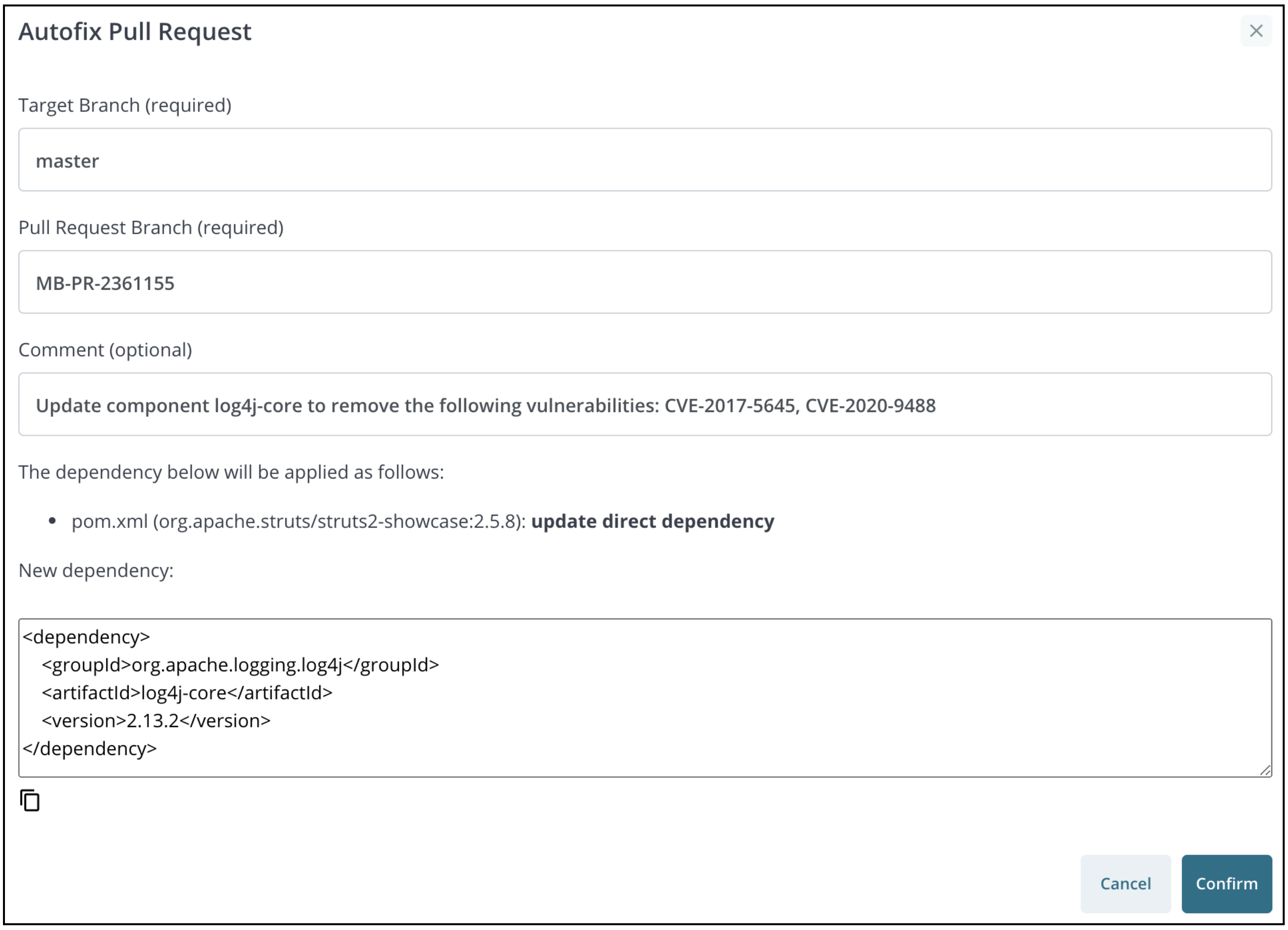
If GitHub integration is enabled and the source scan is from a Maven project in GitHub, the “Patch Info” button is replaced with an “Auto Fix PR” button.



delanAtMergebase/struts-demo.git / SCAN — org.apache.logging.log4j/log4j-core@2.7

Version: 2.7 Component Age: 5 Y (released 2016-10-02)

Clicking on the button opens a dialog that will allow the user to create a pull request to upgrade the component.



Autofix Pull Request

Target Branch (required)

master

Pull Request Branch (required)

MB-PR-2361155

Comment (optional)

Update component log4j-core to remove the following vulnerabilities: CVE-2017-5645, CVE-2020-9488

The dependency below will be applied as follows:

- pom.xml (org.apache.struts/struts2-showcase:2.5.8): **update direct dependency**

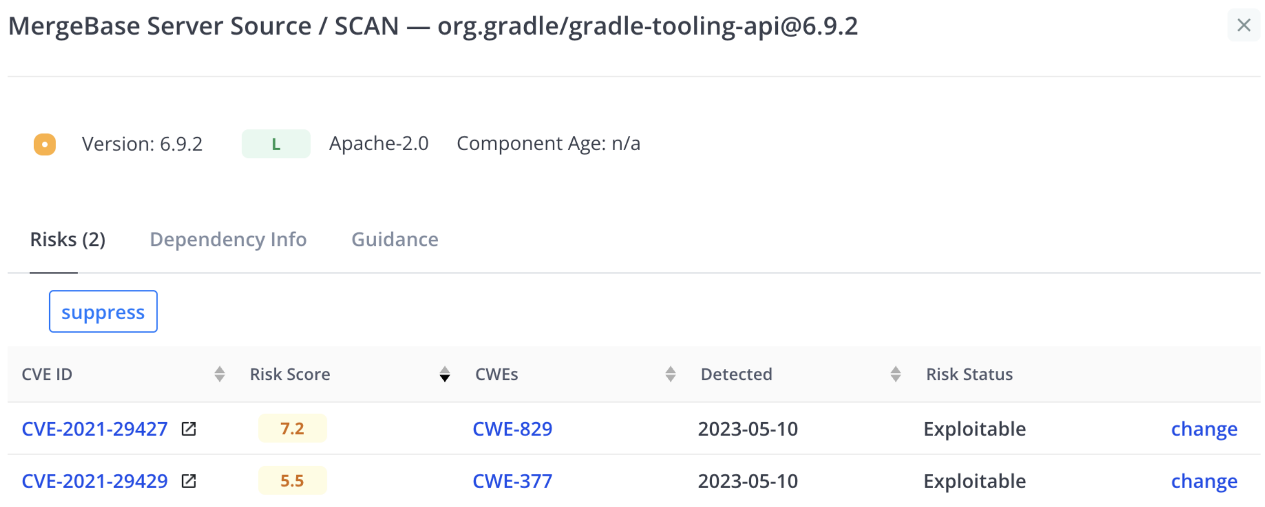
New dependency:

```
<dependency>
   <groupId>org.apache.logging.log4j</groupId>
   <artifactId>log4j-core</artifactId>
   <version>2.13.2</version>
</dependency>
```

Auto Fix Pull Request Dialog

Component Detail - Binary Scans and Inoculated Applications

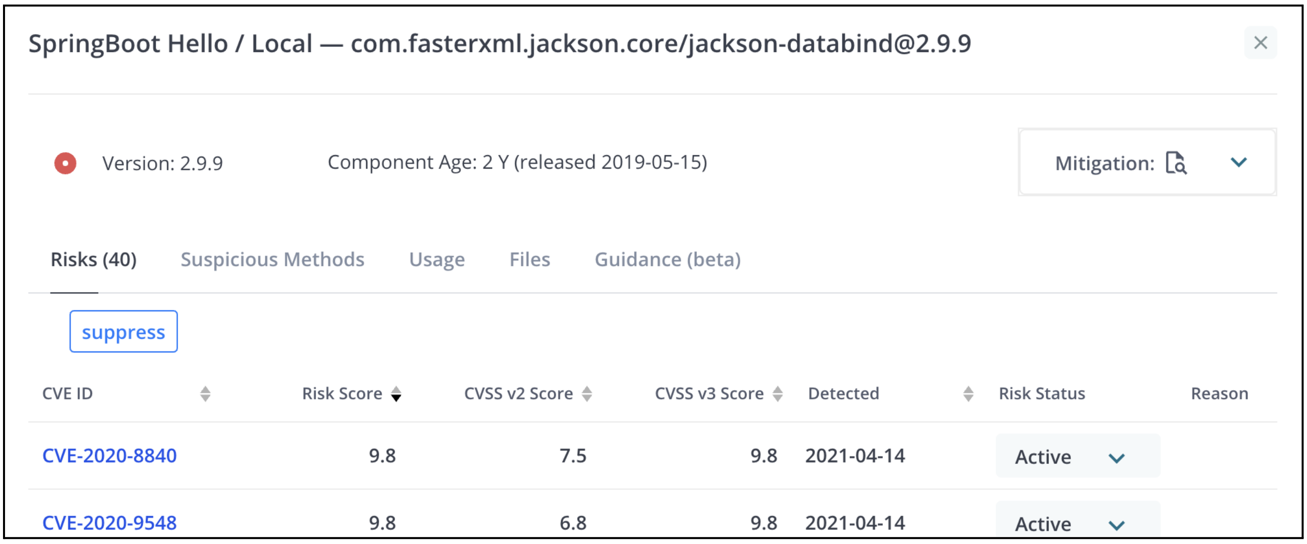
For binary scans as well as inoculated applications, different information is presented in the Component Detail view.



MergeBase Server Source / SCAN — org.gradle/gradle-tooling-api@6.9.2

● Version: 6.9.2 L Apache-2.0 Component Age: n/a

CVE ID	Risk Score	CWEs	Detected	Risk Status	
CVE-2021-29427	7.2	CWE-829	2023-05-10	Exploitable	change
CVE-2021-29429	5.5	CWE-377	2023-05-10	Exploitable	change



SpringBoot Hello / Local — com.fasterxml.jackson.core/jackson-databind@2.9.9

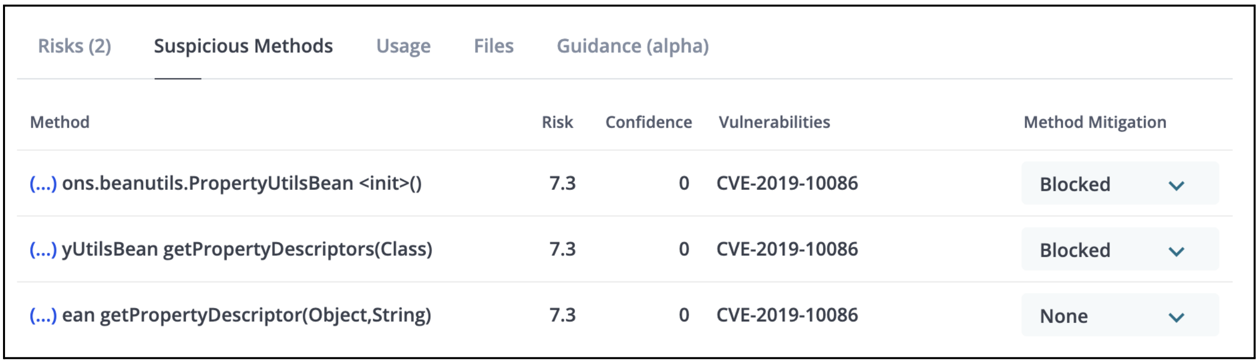
● Version: 2.9.9 Component Age: 2 Y (released 2019-05-15)

CVE ID	Risk Score	CVSS v2 Score	CVSS v3 Score	Detected	Risk Status	Reason
CVE-2020-8840	9.8	7.5	9.8	2021-04-14	Active	
CVE-2020-9548	9.8	6.8	9.8	2021-04-14	Active	

As with source scans, the component detail includes the release date and the "suppress" button. There is also a drop down menu to set a mitigation on the component (e.g. block or monitor).

The Risks and Guidance tabs are the same as for source scans. However, the Guidance tab will not display an "Auto Fix PR" button even if GitHub integration is enabled. This is because scans from jar files do not contain the git repository information required to create the patches and pull requests. There is also no information about dependencies.

Suspicious Methods



Method	Risk	Confidence	Vulnerabilities	Method Mitigation
(...) ons.beanutils.PropertyUtilsBean <init>()	7.3	0	CVE-2019-10086	Blocked
(...) yUtilsBean getPropertyDescriptors(Class)	7.3	0	CVE-2019-10086	Blocked
(...) ean getPropertyDescriptor(Object,String)	7.3	0	CVE-2019-10086	None

Suspicious methods are methods identified by MergeBase as being implicated in an exploit for the identified vulnerabilities. The MergeBase user can choose to block or monitor the individual methods displayed in this list.

Usage

If monitoring has been enabled at the component or suspicious method level, the information collected will be displayed here.

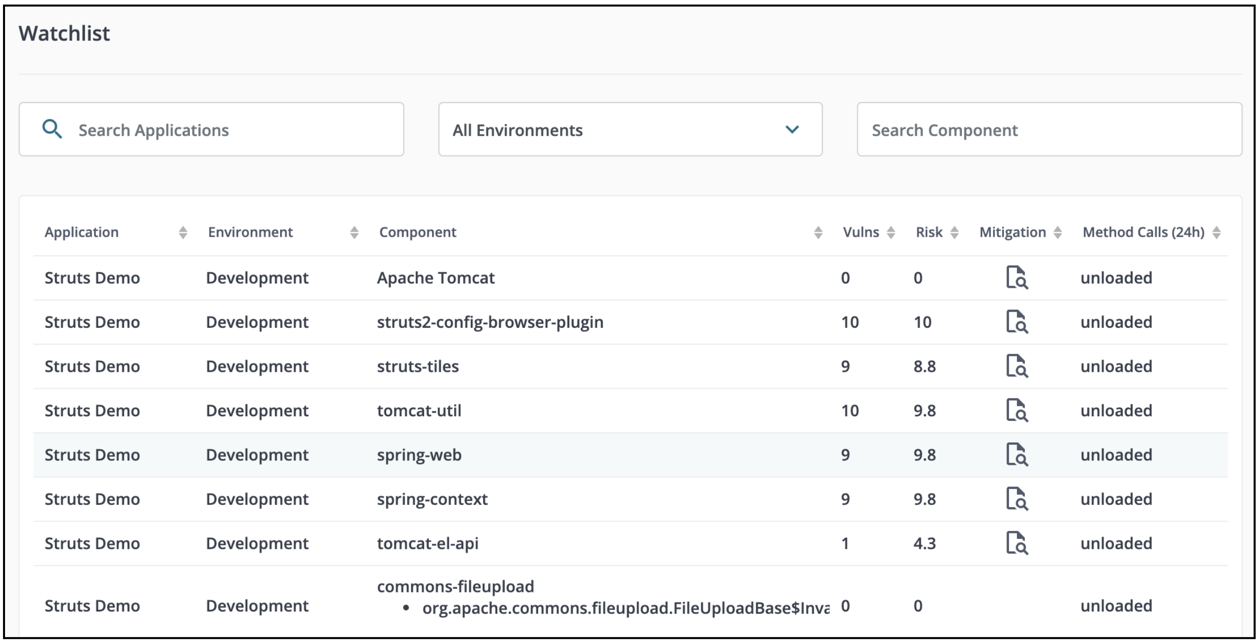
Files

This tab will display the file path(s) associated with the component when it was scanned.

Watchlist

The Watchlist provides a view of all currently monitored, blocked or suppressed components and methods.

The MergeBase user can click on a row to navigate to the Application Detail view showing the affected component.

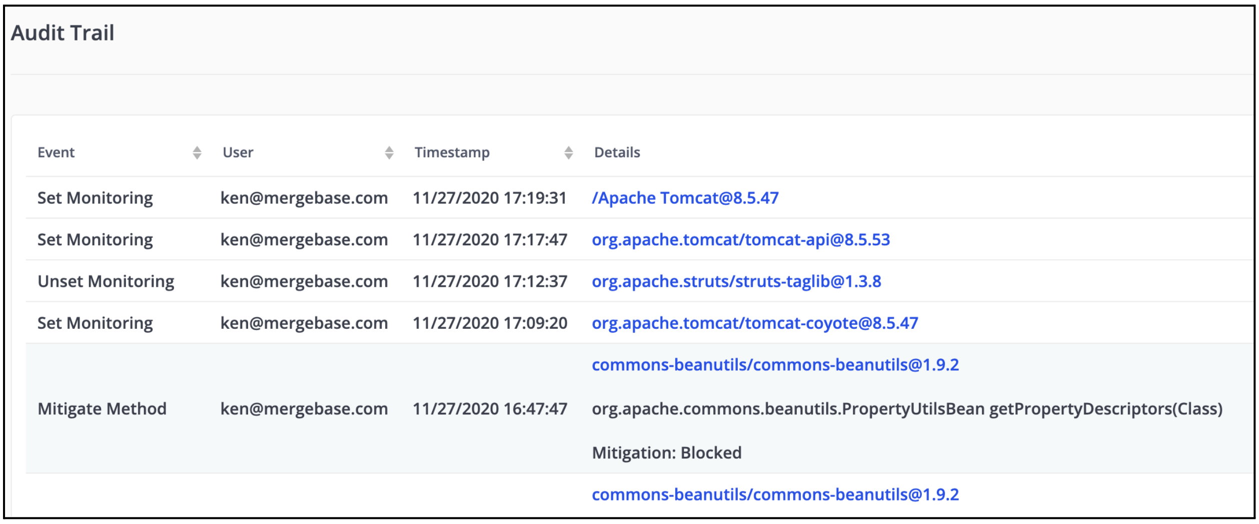


Watchlist

Application	Environment	Component	Vulns	Risk	Mitigation	Method Calls (24h)
Struts Demo	Development	Apache Tomcat	0	0	🔍	unloaded
Struts Demo	Development	struts2-config-browser-plugin	10	10	🔍	unloaded
Struts Demo	Development	struts-tiles	9	8.8	🔍	unloaded
Struts Demo	Development	tomcat-util	10	9.8	🔍	unloaded
Struts Demo	Development	spring-web	9	9.8	🔍	unloaded
Struts Demo	Development	spring-context	9	9.8	🔍	unloaded
Struts Demo	Development	tomcat-el-api	1	4.3	🔍	unloaded
Struts Demo	Development	commons-fileupload • org.apache.commons.fileupload.FileUploadBase$Inva	0	0		unloaded

Audit Trail

The Audit Trail provides a history of actions taken by MergeBase users. This includes creating scans, setting mitigations, and accepting risks.



Audit Trail

Event	User	Timestamp	Details
Set Monitoring	ken@mergebase.com	11/27/2020 17:19:31	/Apache Tomcat@8.5.47
Set Monitoring	ken@mergebase.com	11/27/2020 17:17:47	org.apache.tomcat/tomcat-api@8.5.53
Unset Monitoring	ken@mergebase.com	11/27/2020 17:12:37	org.apache.struts/struts-taglib@1.3.8
Set Monitoring	ken@mergebase.com	11/27/2020 17:09:20	org.apache.tomcat/tomcat-coyote@8.5.47
Mitigate Method	ken@mergebase.com	11/27/2020 16:47:47	commons-beanutils/commons-beanutils@1.9.2 org.apache.commons.beanutils.PropertyUtilsBean getPropertyDescriptors(Class) Mitigation: Blocked
			commons-beanutils/commons-beanutils@1.9.2

Users can click on the links to navigate to the component in question.

Archive

The Archive section displays old scans and obsolete inoculated applications. If a scan has been archived by mistake, the scan can be restored using the popup menu at the right end of every table row.

This view has similar filtering controls as the Overview and users can click on a row to open the Application Detail View for the archived application.

Automatic Archiving

When a user scans an application that they have previously scanned, the previous scan is automatically archived to help keep the dashboard data clean and useful.

When a user runs the Command Line Tool to produce a scan or inoculated application, it generates an "application profile" that is stored on the customer's MergeBase server. The profile is uniquely identified by its name and an environment label. For scans, the environment label is always "SCAN". For inoculated applications, the environment label is determined by the application's run time environment.

When the MergeBase server receives an application profile and there already exists a profile with the same name and environment, the server automatically archives the existing profile. Otherwise, the MergeBase server simply creates a new profile.

In the case of an inoculated application, the profile is created when the Command Line Tool is run to inject MergeBase's runtime protection code into the application binary. This profile has the environment set to "INOC" and is immediately archived. When the inoculated application is started, it will report to the customer's MergeBase server, which will then create a new profile with the name and the environment configured in the application's runtime environment.

If a new version of the application is inoculated and deployed, the new version will replace the old version.

Other Features

Scans API (REST Endpoint)

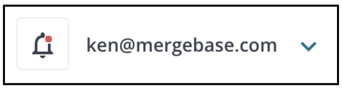
MergeBase provides a stable "SCANS" API Rest Endpoint to help customers import MergeBase scan data into other systems and reports.

Notifications

The customer's MergeBase server will generate the following notifications:

- A new vulnerability was discovered against an existing application's component and the risk score meets or exceeds the warning threshold
- An existing vulnerability was re-scored and either the old or the new score meets or exceeds the warning threshold
- A blocked component or method was accessed
- A monitored component or method was accessed

Recent notifications can be accessed by clicking on the bell icon in the upper right corner of the window (next to the user menu). A red dot indicates the presence of new notifications since the user's previous login.



Global Search

The global search function allows users to search for components and vulnerabilities. Users can discover which components are impacted by a particular vulnerability and which applications contain a particular component. For example, a search for "CVE-2019" will find all vulnerabilities from 2019 that pertain to all scanned and inoculated applications in your dashboard. A search for "struts" will find all components that contain "struts" in their name.

Download Command Line Tool

The Command Line Tool button allows the user to download the tool, which is used to scan and inoculate applications. Usage of the tool is explained in the Command Line Tool (CLT) section below.

Settings (Configuration)

The Settings menu item ⚙️ in the bottom left hand side of the dashboard, provides access to MergeBase configuration settings for the customer's organization. It consists of three tabs: Status, Integrations, and License Policies.

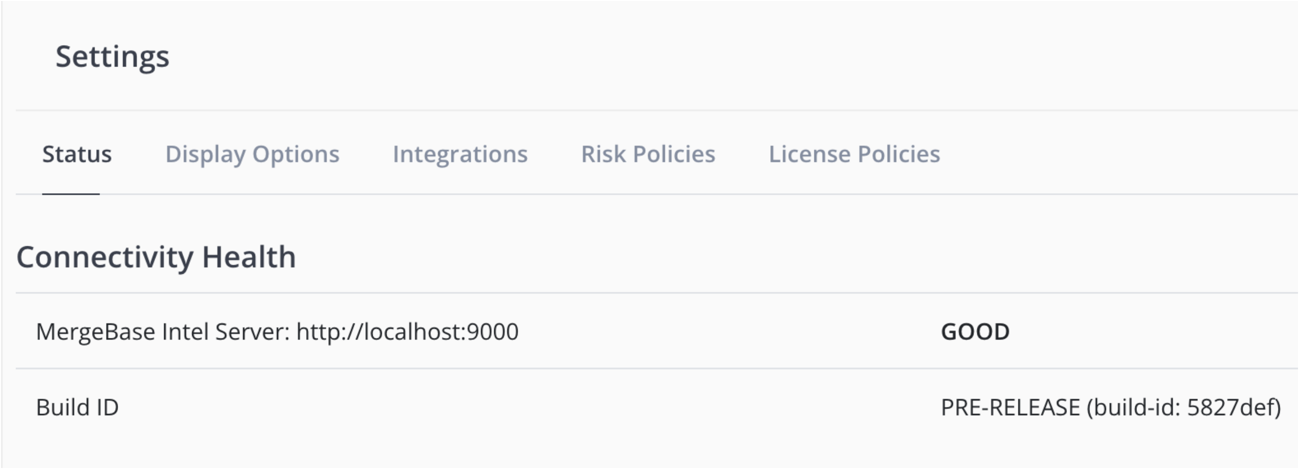
Status Tab

The Status Tab provides some basic network connectivity health checks for your system. If you have enabled Slack or other integrations, you can generate a test notification.

Build ID

This displays the Build ID (version number) of the MergeBase Dashboard. Please include your current "Build ID" when communicating with the MergeBase support team about any problems you encounter.

Note: The CLT (Command Line Tool) also displays its Build ID on every invocation. We recommend using a version of the CLT that matches the same version as the MergeBase Dashboard (where the Build ID's are equivalent).



Connectivity Health

MergeBase Intel Server: http://localhost:9000	GOOD
Build ID	PRE-RELEASE (build-id: 5827def)

Integrations Tab

The Integrations Tab provides the forms to manage integration with external services. The currently supported services are:

- Issue tracker (Jira or Microsoft Boards)
    - Jira (on-premise or cloud)
        - Provide the project key you want to show in Jira (the prefix of the issue)
            - for example: ISSUE
        - Provide the Jira specific issue type
            - for example: Task, Epic, Story, Bug
        - Provide the URL for your Jira instance (the root).
            - for example: https://company.atlassian.net
        - The login ID you would like to use
            - for example: person@example.com
        - Your Atlassian api token key
            - https://support.atlassian.com/atlassian-account/docs/manage-api-tokens-for-your-atlassian-account/
    - Microsoft Boards
        - Provide the URL for your Boards project. The URL has the form:

https://dev.azure.com/{organization}/{project}

- Your personal access token (PAT) generated for your Azure account. The permission required for this token is “Read & write” on Work Items.

- Slack
- Syslog / IBM QRadar
- MergeBase REST API (Nucleus compatible)
- Splunk

Jira Integration

MergeBase supports integration with Jira Cloud and Jira Server (on-premise). For Jira Cloud, you must create an API token for a particular Jira user. On the MergeBase side, you select “API Token” as the Authorization Type and enter the client ID and token in the corresponding fields.

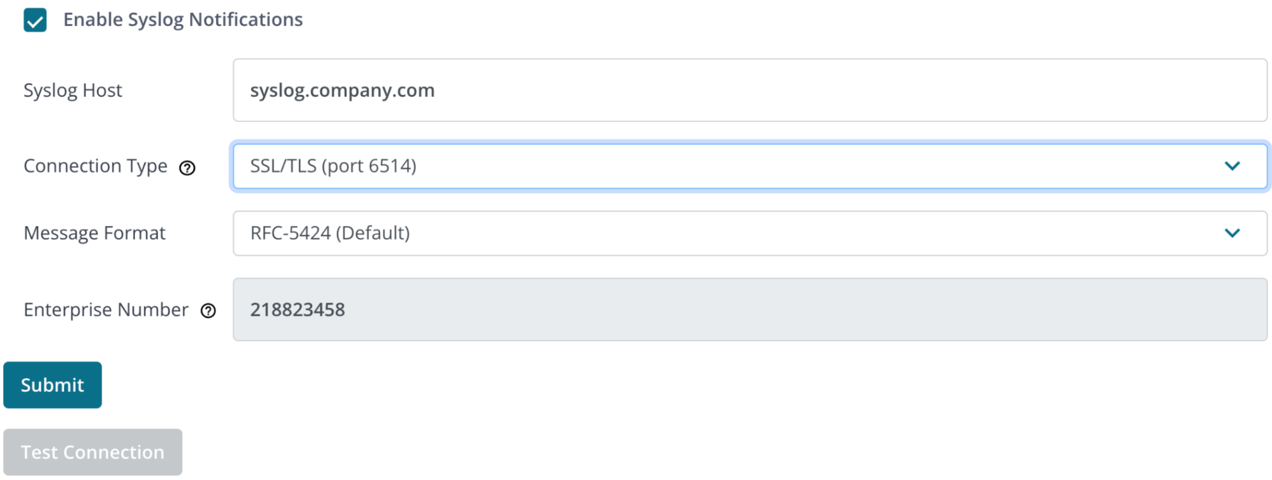
For Jira Server, you create a Personal Access Token in Jira. On the MergeBase side, you select “Personal Access Token” as the Authorization Type and enter the token in the corresponding field. You must be using Jira Server 8.4 or later.

Once you have saved your configuration, you can test the connection to ensure it is working properly.

If you are using Jira Server and need to create firewall rules to allow MergeBase to connect to it, you must request a permanent IP address from MergeBase. The REST API endpoints used by MergeBase are:

- /rest/api/2/project
- /rest/api/2/project/<project-key>

Syslog / IBM QRadar Integration



☑ Enable Syslog Notifications

Syslog Host	syslog.company.com
Connection Type	SSL/TLS (port 6514)
Message Format	RFC-5424 (Default)
Enterprise Number	218823458

To configure Syslog integration, enter the Syslog server host name and select the connection type. There are three connection types, SSL, UDP, and TCP. SSL is recommended unless you are running MergeBase on a server in a secure network.

If you are using QRadar, change the message format to LEEF.

Use the Test Connection button to test your connection and send sample messages.

Risk Policies Tab

The Risk Policies Tab allows the user to configure the thresholds related to vulnerabilities and component age. There are two vulnerability thresholds: violation and warning. They are used in the risk status indicators for components with vulnerabilities. The value set for the violation threshold will be used by the Command Line Tool if the option to verify license violations is enabled (see the Command Line Tool section).

There are two component age thresholds: update and abandonment. When a component exceeds the update threshold and has a more recent release, it is considered updatable. When a component exceeds the abandonment threshold and does not have a more recent release, it is considered abandonable.

License Policies Tab

The License Policies Tab allows the user to configure which open-source licenses are permitted, warnings, or violations. The tab is pre-configured according to our interpretation of industry practices.

Command Line Tool (CLT)

The Command Line Tool (hereafter referred to as the CLT) is an executable Java .jar file that has two main uses:

- Scanning
  - source code projects that are built with any of the supported build systems / languages
  - compiled Java (JAR, WAR, or EAR files) or compiled Microsoft files (DLL or EXE files)
  - Docker Linux images that use a supported package manager
- Preparing the application for run-time protection by inoculating Java applications with MergeBase's runtime monitoring and protection instrumentation.

In both cases, the tool produces a vulnerability report and, unless offline mode is specified, the scan data is uploaded to the customer's MergeBase dashboard where the components are monitored for new vulnerabilities.

Using the CLT

Downloading

To use the CLT, download the mergebase.jar file from your MergeBase dashboard. This jar file is a wrapper that will download the actual tool. It will check for updates and automatically upgrade if there is a new version.

Download via API

You can also download the mergebase.jar via API from your dashboard server:

Usage:

```
GET : [dashboard-domain]/api/update/clt/mergebase.jar

Authorization:
       Must include "X-Authorization: [customer-token]" HTTP header with your unique customer
token.
       You can retrieve this token from the main Settings page on your dashboard.
```

Quick Help Page

Running the CLT with the "--help" option prints out the CLT's quick-help page. As of version 3.4.9, it prints the following:

```
Usage:
  java -jar mergebase.jar [arguments...] [scan targets...]

A scan target can be a directory, file, or docker image name with tag (e.g. "alpine:latest").

Exit codes:          0  - No vulnerabilities above threshold found.
                     1  - One or more vulnerabilities were found.
                     2  - Failed to connect to MergeBase Intelligence service.
                   100  - This help message was printed.
                   101  - Error with command arguments.
                   200  - Other errors.
                  1xxx  - Tool execution errors where xxx is the tool exit code.

Arguments:

  --help             Print this help.

Processing option must be one of the following modes:

  --scan (default)   Scan the dependencies of a single source project per scan path.
                     To scan recursively, use --all.

  --binary           Scan for binary files recursively (e.g. Java jars, dotNet DLLs, .whl files)

  --import           Import SBOM files in CycloneDX JSON format
  --inoculate        Inject runtime monitoring and mitigation code into Java jars

  --revert           Undo the inoculation (jars are restored to their original contents)

Required for all modes except --revert:

  --name=NAME        The name of the application being scanned or inoculated.

Optional for scan mode:

  --group=NAME       Assign the application to the named group. If the group does not exist,
                     it will be created.

  --all              Scan for projects/subprojects recursively. Applies to --scan mode only.
                     Exercise caution with this option as it may generate a very large scan.
                     In particular, it should normally not be used with multi-module Maven
projects.

  --threshold=X      Minimum CVSS score (between 0.0 and 10.0) that
                     vulnerabilities must possess to be included in
                     results. If not specified, the default is 0.0.

                     NOTE: Vulnerabilities with no CVSS entry in the
                     federal NVD database are implicitly assigned a
                     score of 10.0 by this tool (a rare situation).

  --enforceLicensePolicy Return an error exit code if the scan detects
                     a component with a license that violates a license
                     policy defined in the MergeBase dashboard (not applicable to offline scans)

  --mavenOptions=X   Specify options for use with Maven scanning (e.g. "-Dprofile-name")

  --includeOnly=X    When scanning a Gradle project, include only subprojects with ids
                     matching the prefix

  --configuration=X   When scanning a Gradle project, use 'X' as the Gradle configuration
                     The default is 'runtimeClasspath'

  --offline          Scan in offline mode (Enterprise version only). See User Guide for
limitations.

Optional for all modes, excludes --revert:

  --sbom             Write an SBOM (SPDX Lite) to stdout or file if --outputFile is used.

  --csv              Output scan results in the specified format
  --json             (e.g., csv, json, or xml)
  --xml

  --outputFile=FILE  Write vulnerability report to FILE instead of STDOUT.

  --verbose          Show all CVEs associated with each file (normally
                     the results display only the worst 2 or 3), and
                     show every file processed, including files with no
                     vulnerabilities.

  --quiet            Do not print any output during the scan.

  --exitZero         Always return exit code 0.
                     (E.g., don't kill build scripts, just report findings).

Optional for offline usage (Enterprise version only):

  --disableIgnore    Disable the CVE ignore list (".mergebase.ignore" file
                         in the current project directory, or if such does not
                         exist, a file in "~/.mergebase/ignore".

See: https://mergebase.com/
(C) Copyright 2022 Mergebase Software Inc. All Rights Reserved.

MergeBase CLT version=v3.4.9 (build-id: 4ea80c2)
```

Source Code Scan

To scan a source code project and upload the results to your dashboard, run:

```
java -jar mergebase.jar --name=”App Name” <target>
```

where “`App Name`” is the name you want to see in the dashboard and `<target>` is the root directory of a project.

Specific requirements for each project type are detailed below in the section on *Supported Languages and Build Systems*.

The tool will print out a report and exit with an error code if any vulnerabilities are found. You can modify this behaviour with the `--exitZero` or `--threshold` options.

Four report formats are supported: plain text, CSV, JSON, and XML. These reports contain only information on components with vulnerabilities.

A complete list of all components with vulnerability and licensing information is available if you use the `--sbom` option.

Binary Scan

<target> can be any of the following:

- A Java deployable artifact (e.g. .jar, .war, .ear, .aar)
- A Windows .dll or .exe file
- A Python .whl (wheel) file
- A directory containing the above files

Inoculate and Revert (Runtime Protection only)

To inoculate an application, run the following command:

```
java -jar mergebase.jar --inoculate --name=”App Name” <target>
```

where “App Name” is the name you want to see in the dashboard.

<target> can be one of the following:

- A Java deployable artifact (e.g. .jar, .war, .ear)
- A directory containing Java deployable artifacts

If you are inoculating a third party application with a large number of .jar files (e.g. several hundred) and you have limited memory available on the machine you are using to run the CLT, you may need to inoculate the .jar files in smaller batches (or even one at a time). To assemble the results together into a single application in the dashboard, use `--append` on every invocation of the CLT after the first.

To revert an inoculation, use the `--revert` option:

```
java -jar mergebase.jar --revert <target>
```

Proxy Settings

The Command Line Tool (CLT) needs to be able to open a network connection to the customer’s MergeBase dashboard server in order for the results of scans to be available. If the CLT is unable to establish a connection, it will immediately halt and display an error message.

In complex environments to open this type of network connection might require the use of proxy servers.

The Java Virtual Machine (JVM) has built-in support for HTTP and HTTPS proxies, which can be set using Java system properties. For HTTPS, the relevant system properties are:

`https.proxyHost` – the host name of the proxy server

`https.proxyPort` – the port number (default value is 443)

For example, in order to run the CLT using a proxy my.proxy.com on port 443, you will need to add the system property definition as follows:

```
java -Dhttps.proxyHost=my.proxy.com mergebase.jar …
```

Setting the proxy for an inoculated application will vary depending on the application platform and how it is deployed.

Reference: Java Networking and Proxies

Proxy User Authentication

The CLT supports proxy user authentication using the https.proxyUser and https.proxyPassword (optional) system properties provided by the Java environment.

Option 1: Password provided as parameters on the command line

This option is likely more relevant as part of automated workflows and scripts.

Usage:

```
java -Dhttps.proxyHost=my.proxy.com -Dhttps.proxyUser=userid\
    -Dhttps.proxyPassword=password -jar mergebase.jar …normal parameters…
```

Option 2: Password entered interactively by a user

This option is likely more relevant in development or testing situations.

Usage:

```
java -Dhttps.proxyHost=my.proxy.com -Dhttps.proxyUser=userid\
    -jar mergebase.jar  …normal parameters…

Enter proxy password: <type password>
```

CLT Auto-Upgrading

The mergebase.jar CLT checks the MergeBase dashboard server to see if the CLT should self-upgrade itself. This check happens with every invocation. If the CLT's version is older compared to the dashboard server's version, then the self-upgrade logic is triggered.

Behind the scenes the CLT maintains a "mergebase-actual.jar" file in a hidden ".mergebase" directory under <HOME>/.mergebase/clt/ on the machine where it runs. When "mergebase.jar" sees that an upgrade is required it downloads and replaces the "mergebase-actual.jar" with the newer version.

In some situations some users prefer to not have files downloaded into the user's HOME directory. You can override the download location by specifying a "`-DMERGEBASE_HOME=X`" value among the command-line arguments. This "-DMERGEBASE_HOME=X" value must be specified before "-jar". For example:

```
java -DMERGEBASE_HOME=/tmp/ -jar mergebase.jar ...
```

Offline Usage (for Enterprise customers)

The command line tool can run in an offline mode with some limitations. In this case it produces a vulnerability report but does not attempt to upload the scan data to the customer's MergeBase server. To run in offline mode, add the command line option "`--offline`". This feature is available to enterprise customers. Contact MergeBase for further details.

Supported Languages and Build Systems

MergeBase supports the following languages and build systems:

- Java - Maven and Gradle
- DotNet - NuGet
- Go
- JavaScript - NPM and Yarn
- PHP - Composer
- Elixir - Hex
- Python - PIP
- Ruby - GEM
- C/C++ - CMake, Make, and Conan

Maven

The CLT requires a globally installed maven executable or a mavenw/mavenw.bat script. You do not need to pre-build the project but scanning time will be faster if the Maven packages are already installed locally and do not need to be downloaded.

If you have a multi-module Maven project and wish to scan a particular profile, you can pass Maven options to select the profile using --mavenOptions. For example:

```
java -jar mergebase.jar ... --mavenOptions=”-Dprofile-name”
```

Gradle

The CLT requires a globally installed gradle executable or a gradlew/gradlew.bat script. You do not need to pre-build the project but scanning time will be faster if the Maven packages are already installed locally and do not need to be downloaded.

In the case of a multi-module Gradle project, you can use --includeOnly to filter the sub-modules by their ids. For example:

```
java -jar mergebase.jar ... --includeOnly=:storage:remote
```

will select all sub-modules with ids that start with “:storage:remote” and exclude all other sub-modules.

The CLT uses the default configuration ‘runtimeClasspath’ to determine the scope of the scan. You can override this with the --configuration option.

```
java -jar mergebase.jar ... --configuration=customConfig
```

For Gradle projects with default configurations for Android, use **releaseRuntimeClasspath** as the configuration name.

Note: if the CLT fails to scan a sub-project in a multi-project Gradle project, the tool will output an error message and continue to scan the remaining sub-projects.

DotNet

The DotNet project must be built before running the CLT.

Go

The CLT requires Go to be installed. You can verify this by running “`go mod`” in your project root directory.

NPM and Yarn

The CLT requires both a package.json file and a package-lock.json (NPM) or yarn.lock (Yarn) file.

PHP Composer

The CLT requires both a composer.json file and a composer.lock file.

Elixir

You must have the Elixir development tools installed, including the Hex package manager.

Python

The recommended method is to install the pipdeptree tool and save the output in a file called pipdeptree.txt. The alternate method is to use the ‘pip freeze’ command and save the output in a file called requirements.txt. If you use the latter method, you will not have the structure view that is provided using pipdeptree.

You can also scan for .whl files using the --binary option.

Ruby

You must have the Ruby development tools installed.

C/C++ Make, CMake, and Conan

Most C or C++ projects must be built before running the CLT, although MergeBase can scan Conan dependency files pre-build. To scan C/C++ source projects we recommend including the “--all” flag (but not the “--binary” flag). This will search the file-system under the supplied paths for dependency information by examining Makefiles, CMake config, Auto-Config (*.ac), and other similar C/C++ build-related metadata files.

To scan C/C++ binary artifacts, be sure to include the “--binary” flag. MergeBase currently supports EXE, DLL, and ELF file formats when scanning C/C++ binary artifacts (e.g., executables and shared-libraries, including *.DLL and *.so).

Supported Docker Package Managers

The CLT can scan Linux images if the target supplied on the command line is of the form **image-name:tag**. In this case, the tool will run a binary scan as described above on the image files and will also scan packages installed by any of the following package managers:

- APK - Alpine Linux
- APR - Debian
- RPM

You must run ‘docker pull’ on the image you wish to analyze before invoking the scan.

Example:

```
java -jar mergebase.jar --name=Ubuntu ubuntu:latest
```

Running Inoculated Applications

The process of inoculation injects a small amount of code into the application that allows it to report back to your dashboard and to apply mitigations. You should be able to run the application in your normal fashion with one important change: you must define an environment variable **mb_environment** with a suitable value.

```
export mb_environment=PROD
java -cp yourApp.jar com.your.app.Main …
```

Note that you can also override the application name you provided when running the CLT by defining an environment variable **mb_application**. The value of this variable will override whatever value you set with the --name option.

MergeBase Proxy Settings

MergeBase might require proxy configuration settings to function. These settings are applied when running the target application.

Inoculated applications report to the customer's MergeBase dashboard server. The inoculated applications must be able to initiate a connection with the dashboard server. If the application is unable to connect, it will continue to run but create error messages in the application logs.

When the inoculated application is launched it needs similar settings applied to the Java JVM as indicated with the Build proxy settings.

For example, in order to run the inoculated application using a proxy my.proxy.com on port 443, you will need to add the system property definition as follows:

```
java -Dhttps.proxyHost=my.proxy.com -cp yourApp.jar …
```

Setting the proxy for an inoculated application will vary depending on the application platform and how it is deployed.

Reference: Java Networking and Proxies

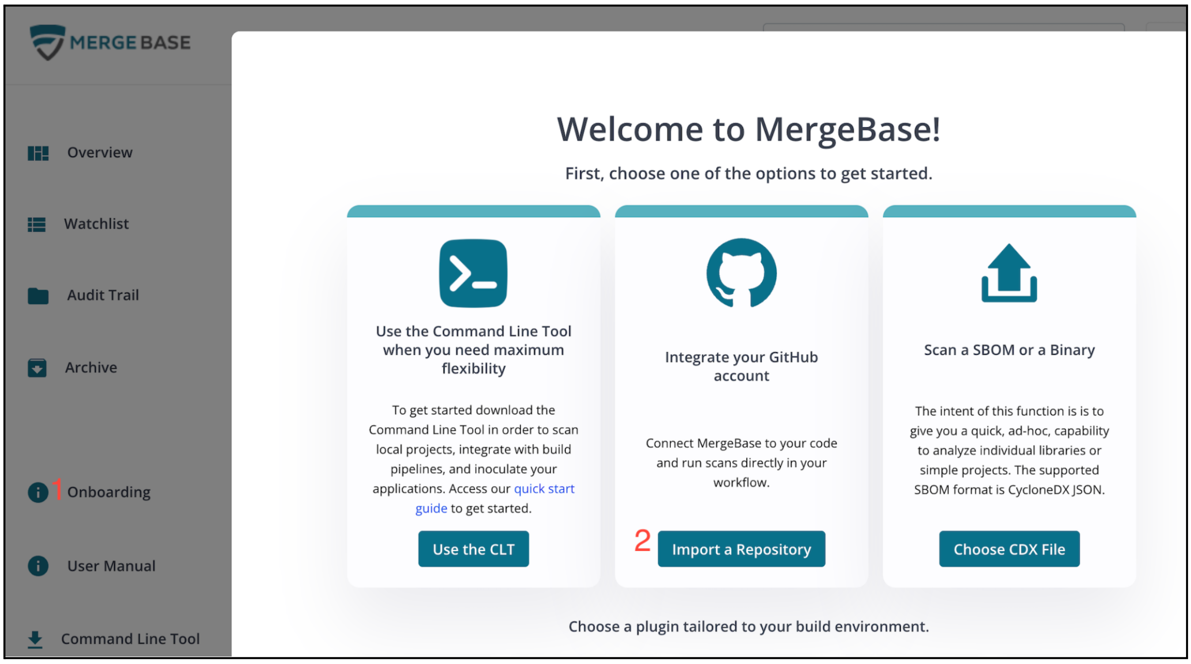
Limitations and Known Issues

- Users cannot change their password except by using the "Forgot your password?" link on the login page

GitHub Integration

MergeBase provides a Github app to seamlessly integrate code from designated GitHub repositories into MergeBase for comprehensive scanning. Additionally, it enables near real-time scanning of submitted commits. It scans build files from your Github repositories and integrates the results directly to the MergeBase dashboard for your centralized view.

Integrating Github Code Repositories with MergeBase



Welcome to MergeBase!

First, choose one of the options to get started.

Use the Command Line Tool when you need maximum flexibility

To get started download the Command Line Tool in order to scan local projects, integrate with build pipelines, and inoculate your applications. Access our quick start guide to get started.

Integrate your GitHub account

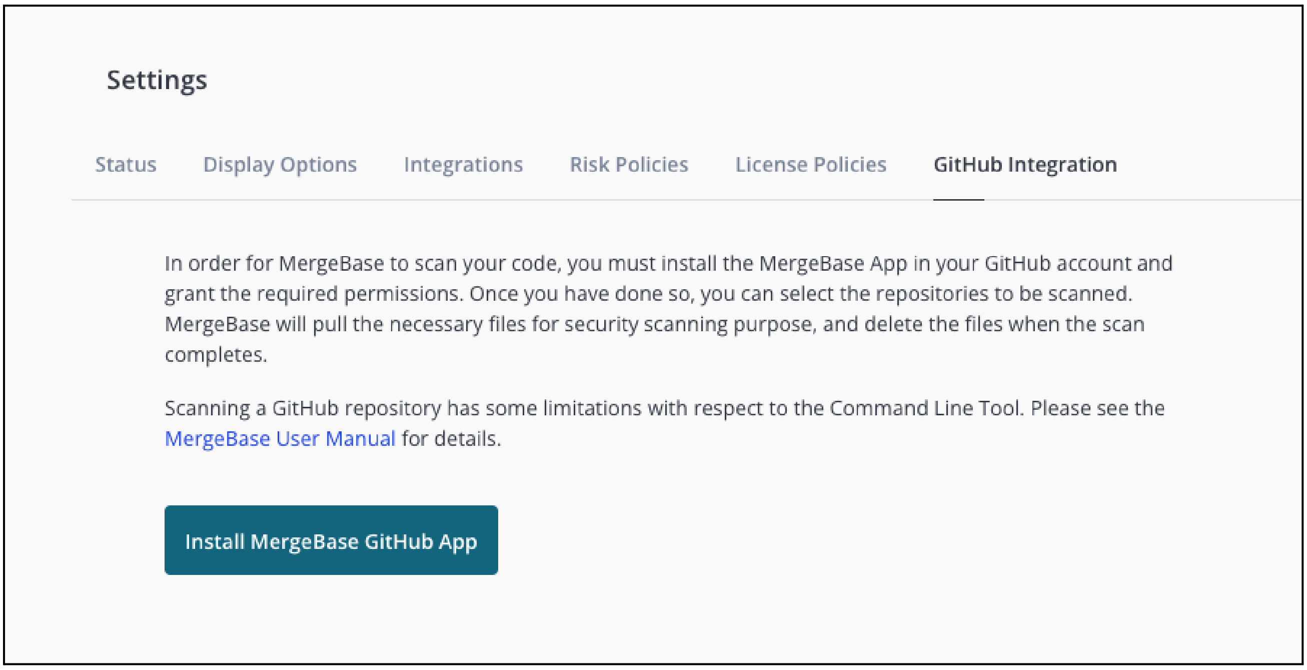
Connect MergeBase to your code and run scans directly in your workflow.

Scan a SBOM or a Binary

The intent of this function is is to give you a quick, ad-hoc, capability to analyze individual libraries or simple projects. The supported SBOM format is CycloneDX JSON.

Choose a plugin tailored to your build environment.

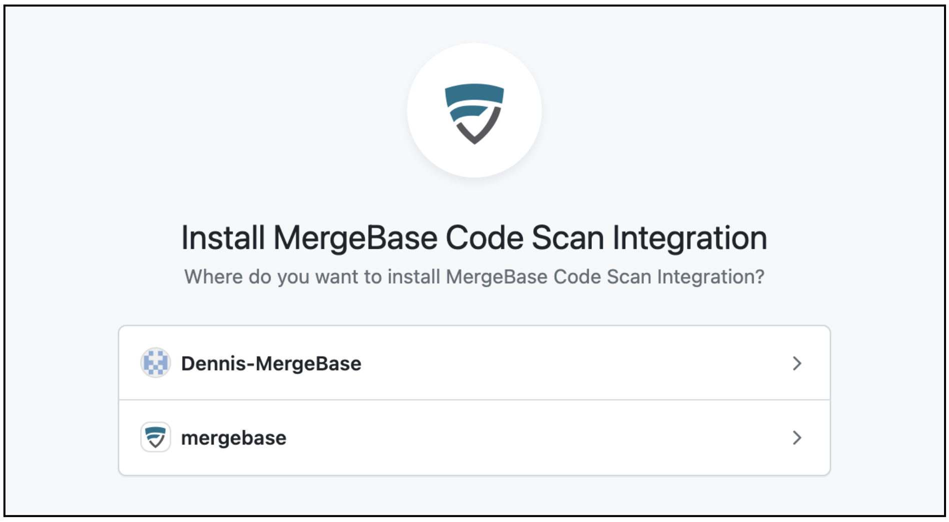
First, on the MergeBase dashboard home page illustrated above, click on the 1 "Onboarding" button on the left, followed by 2 "Import a Repository" button.


Settings

In order for MergeBase to scan your code, you must install the MergeBase App in your GitHub account and grant the required permissions. Once you have done so, you can select the repositories to be scanned. MergeBase will pull the necessary files for security scanning purpose, and delete the files when the scan completes.

Scanning a GitHub repository has some limitations with respect to the Command Line Tool. Please see the MergeBase User Manual for details.

Then click the "Install MergeBase GitHub App" button.

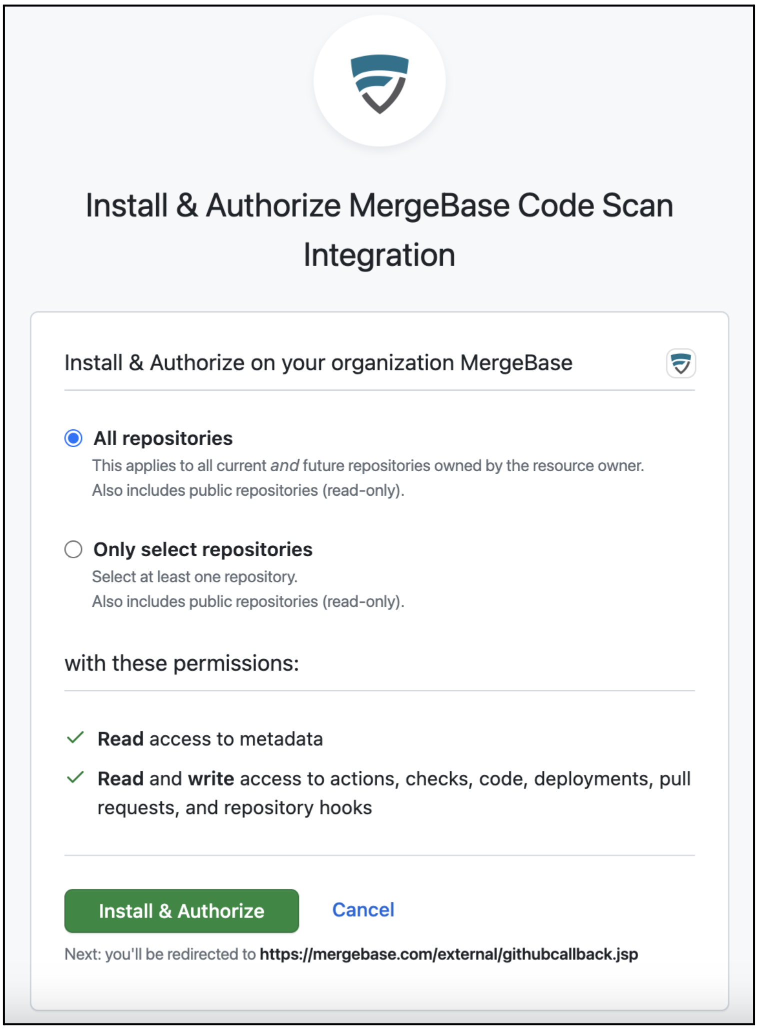


Install MergeBase Code Scan Integration

Where do you want to install MergeBase Code Scan Integration?

You will then be redirected to the GitHub website.

- If you haven't logged into GitHub, you'll encounter the login page to authenticate your credentials.
- For users with multiple GitHub accounts, an account selection page will appear, as shown above.
- Otherwise, you will proceed to the "Install & Authorize" page as depicted below.



Install & Authorize MergeBase Code Scan Integration

Install & Authorize on your organization MergeBase

● **All repositories**
This applies to all current *and* future repositories owned by the resource owner.
Also includes public repositories (read-only).

○ **Only select repositories**
Select at least one repository.
Also includes public repositories (read-only).

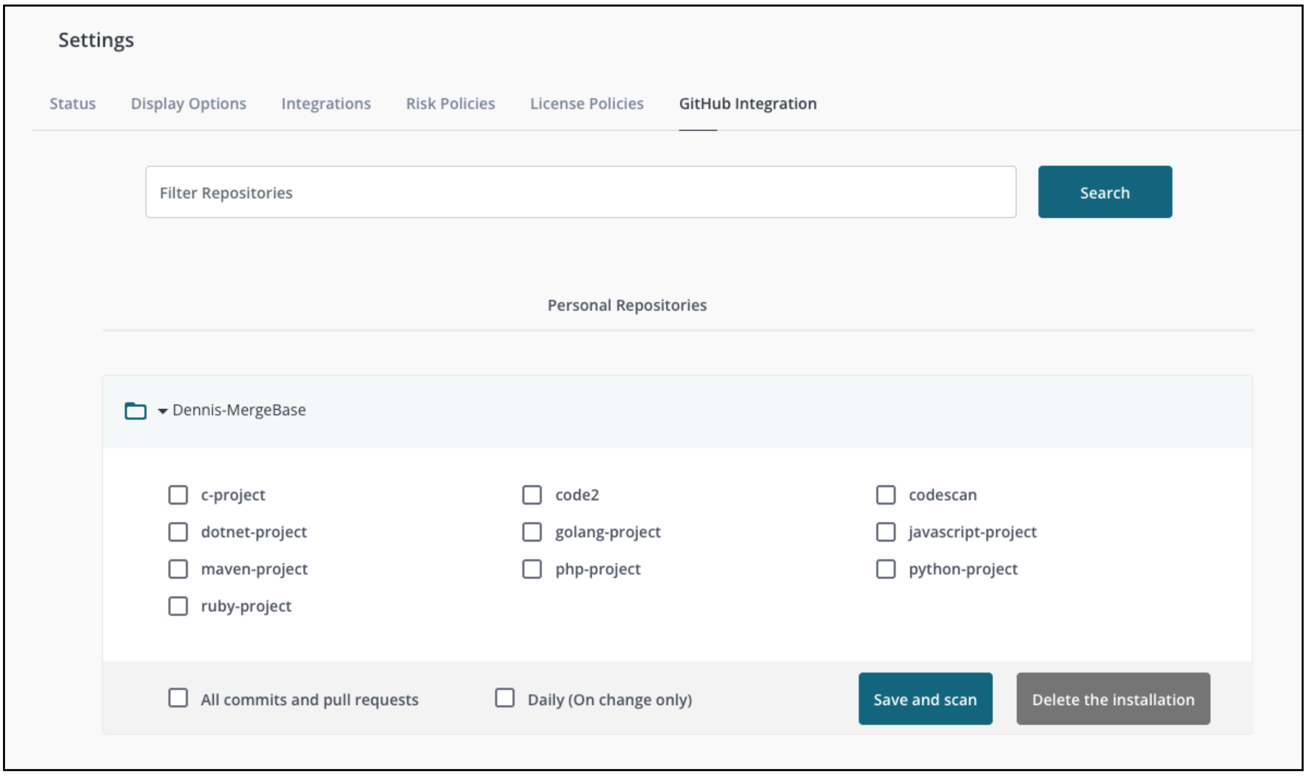
with these permissions:

✓ **Read** access to metadata

✓ **Read** and **write** access to actions, checks, code, deployments, pull requests, and repository hooks

Next: you'll be redirected to **https://mergebase.com/external/githubcallback.jsp**

Upon reviewing the list of permissions, click the “Install & Authorize” button.


Settings

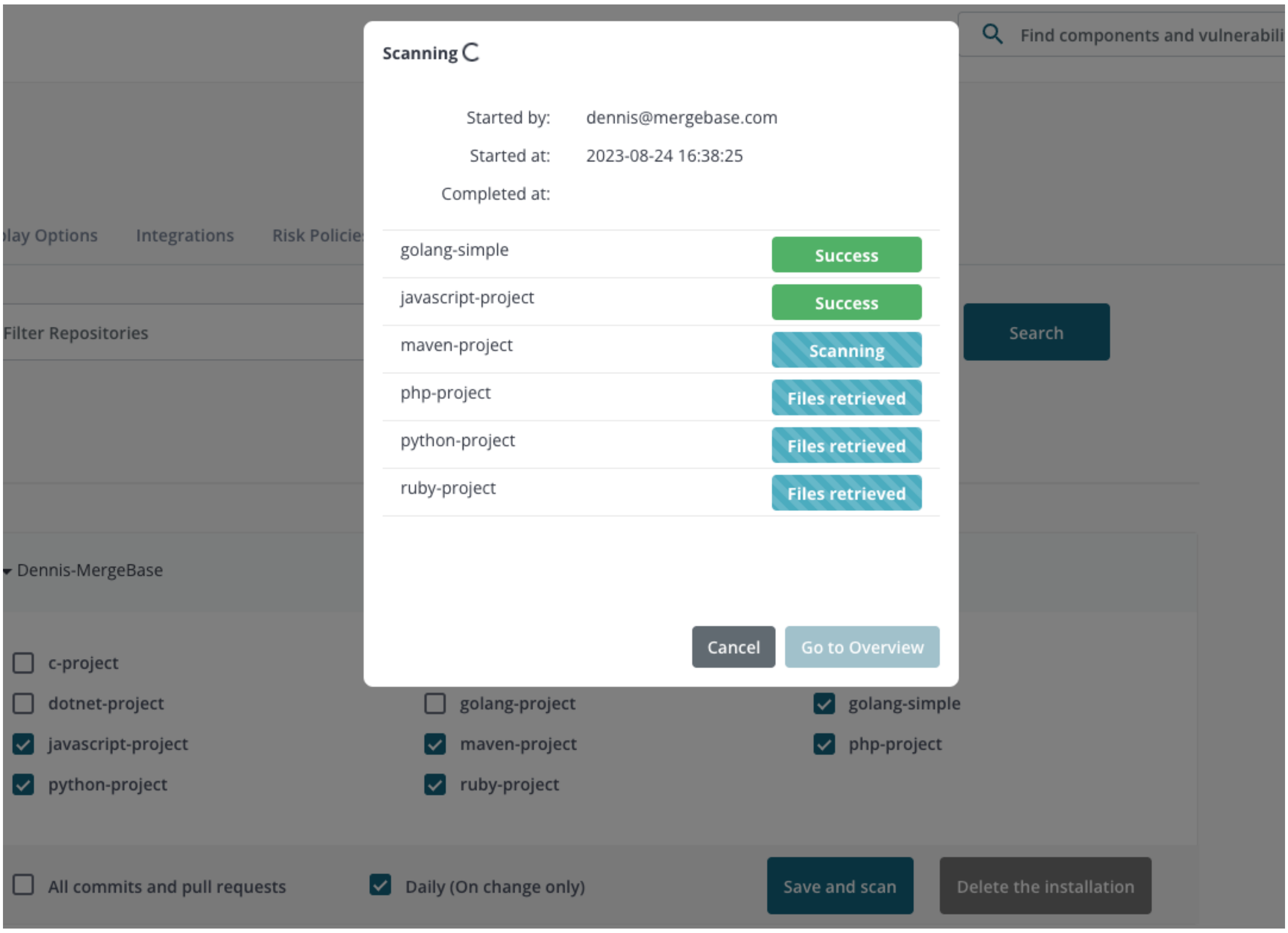
□ c-project
□ code2
□ codescan
□ dotnet-project
□ golang-project
□ javascript-project
□ maven-project
□ php-project
□ python-project
□ ruby-project

□ All commits and pull requests
□ Daily (On change only)

After being redirected back to the MergeBase, the authorized repositories will be listed on the page. You can select repositories for scanning as needed. If multiple GitHub accounts have been integrated, they will all be visible on this page.

Managing Your GitHub Integration

You can utilize the search function to find repositories. Selected repositories that do not match the search keywords will also appear in the search results.



Scanning

Started by: dennis@mergebase.com

Started at: 2023-08-24 16:38:25

Completed at:

golang-simple	Success
javascript-project	Success
maven-project	Scanning
php-project	Files retrieved
python-project	Files retrieved
ruby-project	Files retrieved

□ c-project
□ dotnet-project
☑ javascript-project
☑ python-project
□ golang-project
☑ maven-project
☑ ruby-project
☑ golang-simple
☑ php-project

□ All commits and pull requests
☑ Daily (On change only)

Upon clicking the “Save and Scan” button, The checklist’s selection status will be saved for further actions. The selected repositories will be scanned immediately and asyncronously. The popped-up modal window will display the scanning progress of each repository. When all scans are completed, the “Go to Overview” button will become enabled. Clicking it will allow you to navigate to the overview page where the scan results are displayed.

A current limitation of the integration is that it will only scan the default branch of the repository (generally, the “main” branch).

- Please note that the scanning process might take some time, so it’s advisable not to select too many repositories at one time.
- There are some specific cases related to programming languages of the repository where scanning is unavailable:

  The repository requires compilation:

  C/C++

  DotNet

  The repository requires lock files:

  NPM (JavaScript): package-lock.json

  Yarn (JavaScript): package-lock.json

  Gem (Ruby): Gemfile.lock

  Composer (PHP): composer.lock

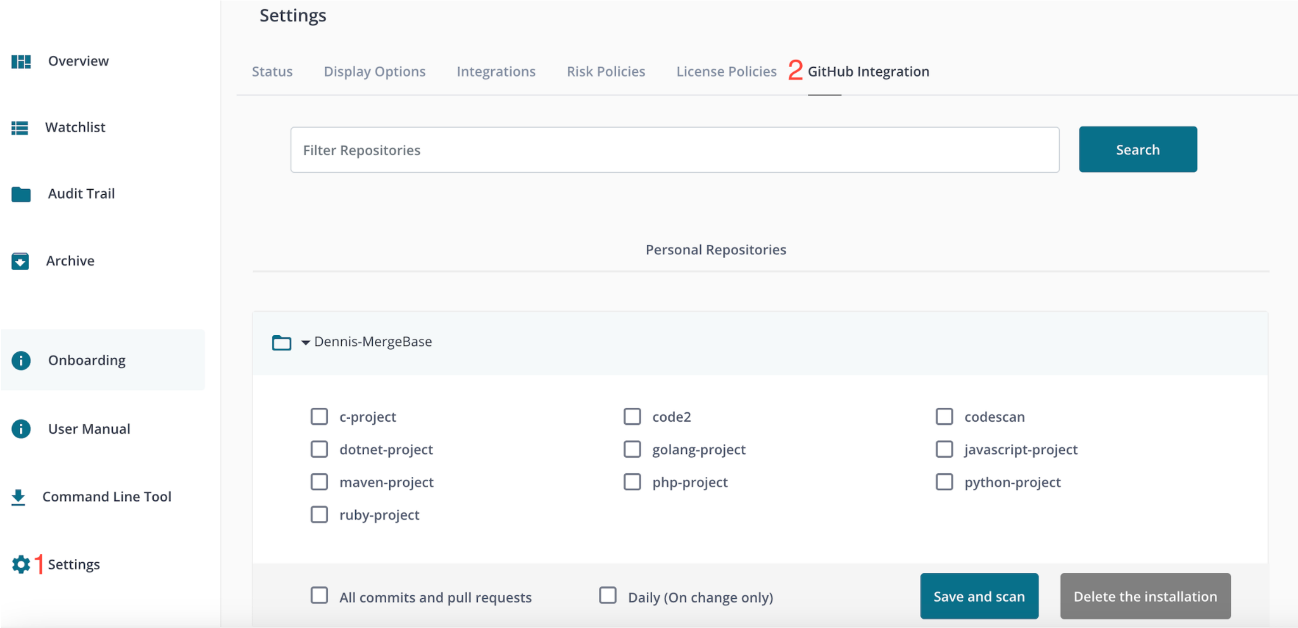
If the “All commits and pull requests” checkbox is selected, scans will be initiated for new commits and pull requests.

- Please note that it involves scanning with a 10-minute interval instead of a real time scanning.

Opting for "Daily (On change only)" will trigger daily scanning whenever there are new code changes.

To delete an installation, click the “Delete the installation” button corresponding to the specific GitHub account. Subsequently, both the installation within MergeBase and the corresponding installation on GitHub will be erased.

- Please note that, if the installation is directly removed from GitHub, the corresponding installation will also be eliminated the next time you log in to MergeBase and navigate to the settings page.



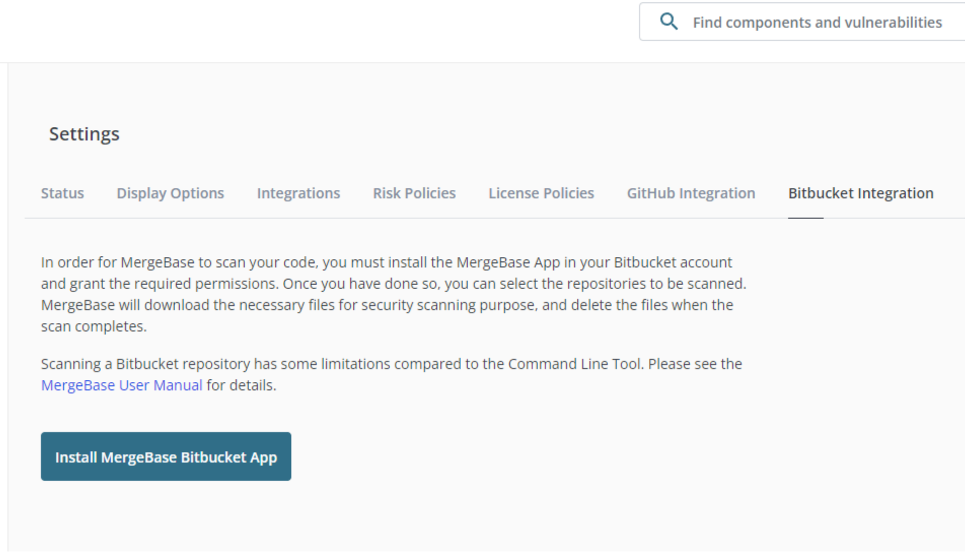
If you want to access this page directly, proceed to the 1 “Settings” page in the left hand navigation as illustrated below, and then access the 2 “GitHub Integration” tab along the top.

Bitbucket Integration

MergeBase provides a Bitbucket Connect app to seamlessly integrate code from designated Bitbucket repositories into MergeBase for comprehensive scanning. Additionally, it enables near real-time scanning of submitted commits. It scans build files from your Bitbucket repositories and integrates the results directly to the MergeBase dashboard for your centralized view.

Integrating Bitbucket Repositories with MergeBase

First, on the MergeBase dashboard home page illustrated above, go to the Settings page, and click the Bitbucket Integration tab.



In order for MergeBase to scan your code, you must install the MergeBase App in your Bitbucket account and grant the required permissions. Once you have done so, you can select the repositories to be scanned. MergeBase will download the necessary files for security scanning purpose, and delete the files when the scan completes.

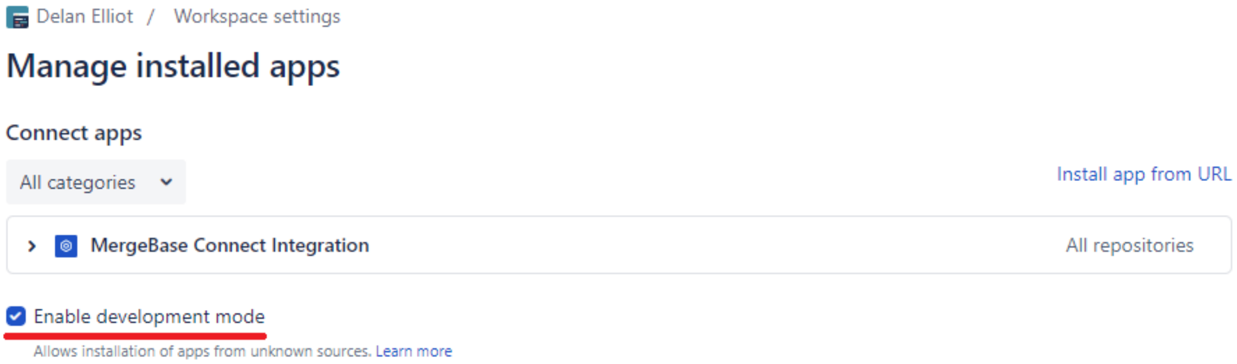
Scanning a Bitbucket repository has some limitations compared to the Command Line Tool. Please see the MergeBase User Manual for details.

Then click the “Install MergeBase Bitbucket App” button.

You will then be redirected to the Bitbucket website.

- If you haven’t logged into Bitbucket, you'll encounter the login page to authenticate your credentials.
- For users with multiple Bitbucket workspaces, the workspace selection screen will be shown.

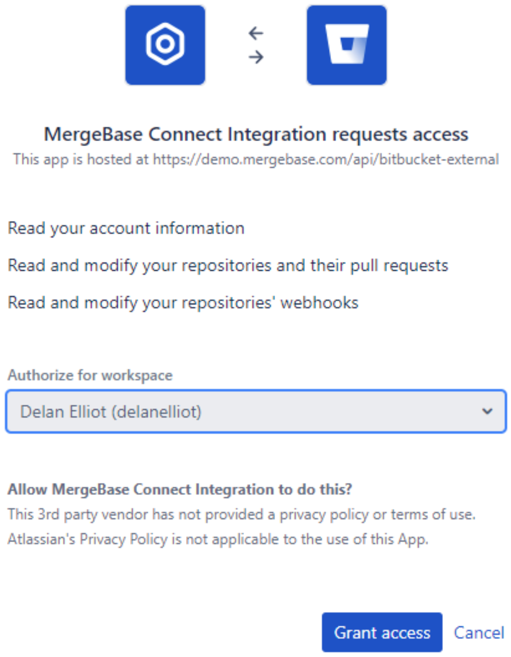
If it is not already enabled, you will need to enable development mode to install the MergeBase bitbucket app integration. Bitbucket will prompt you to enable this setting. This is because a custom app is generated for your unique dashboard.



☑ Enable development mode

Allows installation of apps from unknown sources. Learn more

Upon reviewing the list of permissions, click the "Grant Access" button.



MergeBase Connect Integration requests access

This app is hosted at https://demo.mergebase.com/api/bitbucket-external

Read your account information

Read and modify your repositories and their pull requests

Read and modify your repositories' webhooks

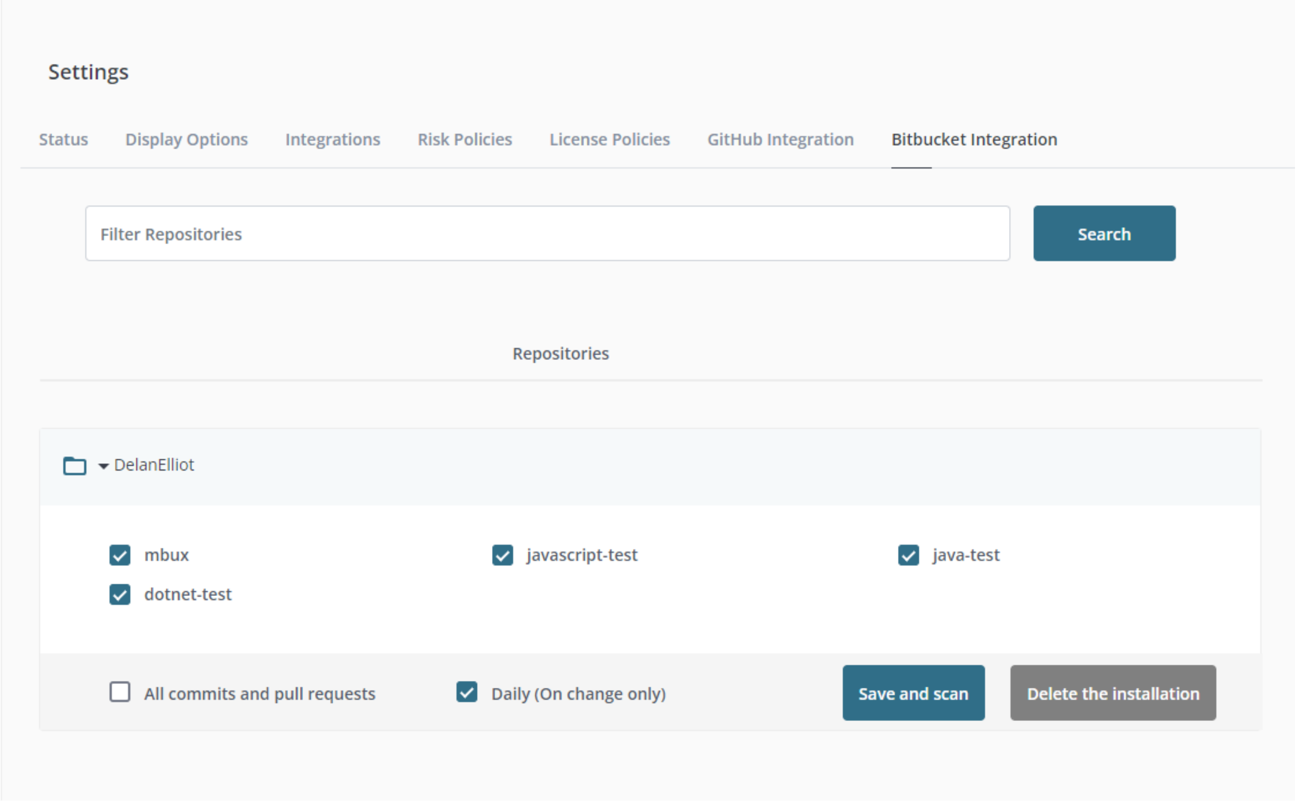
Authorize for workspace: Delan Elliot (delanelliot)

Allow MergeBase Connect Integration to do this?

This 3rd party vendor has not provided a privacy policy or terms of use.

Atlassian's Privacy Policy is not applicable to the use of this App.

After being redirected back to the MergeBase, the authorized repositories will be listed on the page. You can select repositories for scanning as needed.



Settings

Repositories

DelanElliot

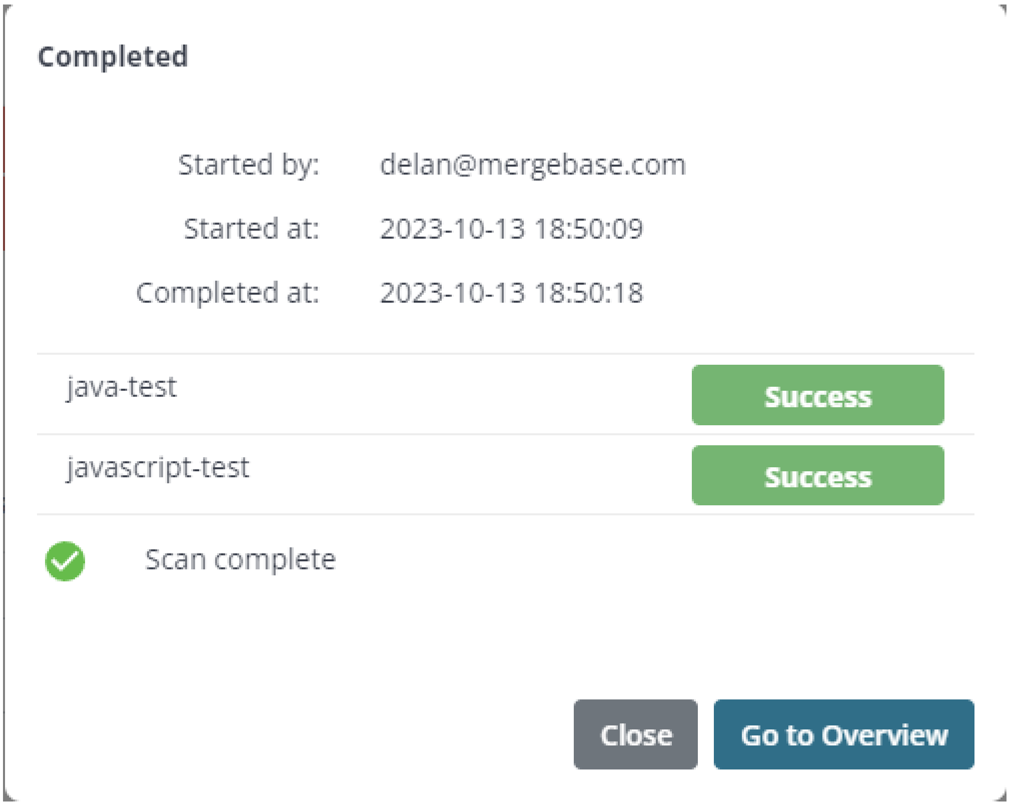
☑ mbux ☑ javascript-test ☑ java-test

☑ dotnet-test

□ All commits and pull requests ☑ Daily (On change only)

Managing Your Bitbucket Integration

You can utilize the search function to find repositories. Selected repositories that do not match the search keywords will also appear in the search results.



Completed

Started by: delan@mergebase.com

Started at: 2023-10-13 18:50:09

Completed at: 2023-10-13 18:50:18

java-test Success

javascript-test Success

✓ Scan complete

Upon clicking the “Save and Scan” button, The checklist’s selection status will be saved for further actions. The selected repositories will be scanned immediately and asynchronously. The popped-up modal window will display the scanning progress of each repository. When all scans are completed, the “Go to Overview” button will become enabled. Clicking it will allow you to navigate to the overview page where the scan results are displayed.

- Please note that the scanning process might take some time, so it’s advisable not to select too many repositories at one time.
- There are some specific cases related to programming languages of the repository where scanning is unavailable:

  The repository requires compilation:

  C/C++

  DotNet

  The repository requires lock files:

  NPM (JavaScript): package-lock.json

  Yarn (JavaScript): package-lock.json

  Gem (Ruby): Gemfile.lock

  Composer (PHP): composer.lock

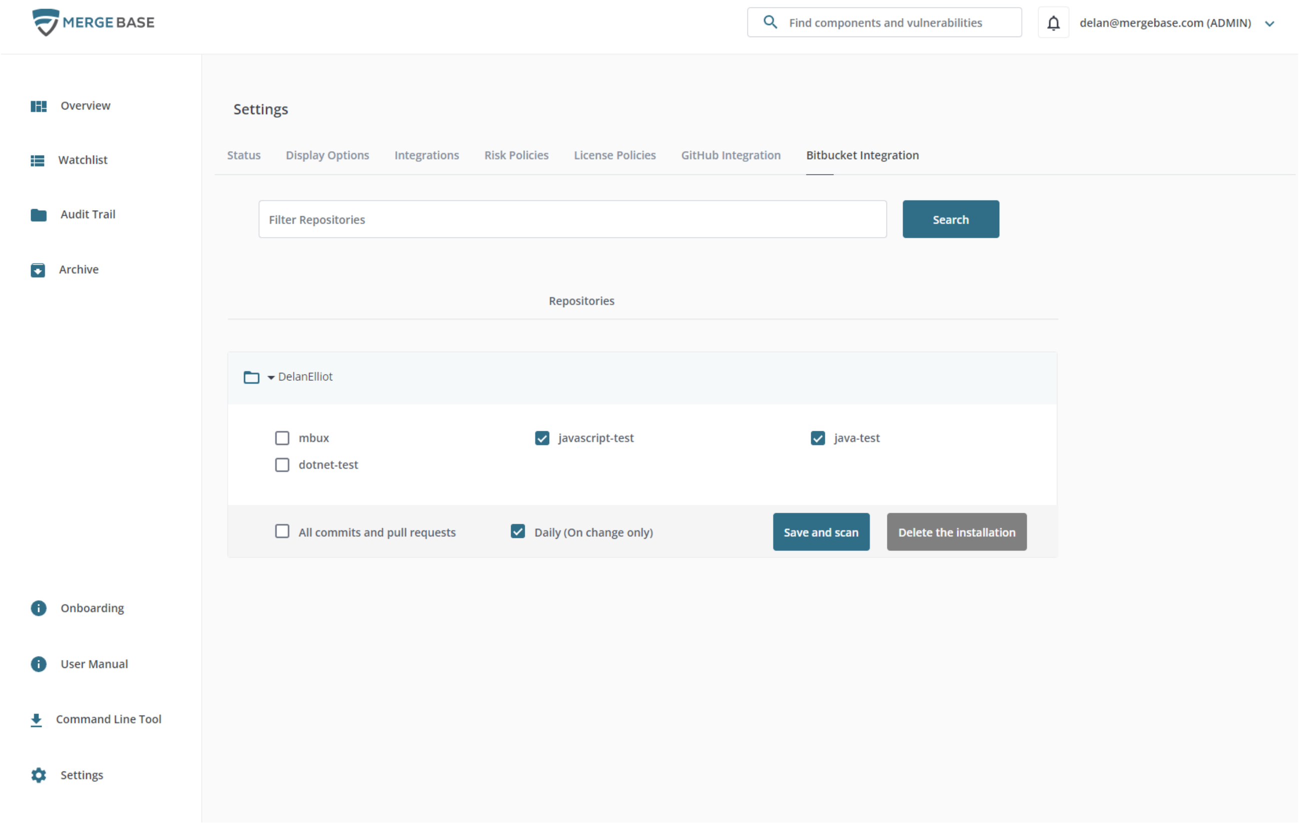
If the “All commits and pull requests” checkbox is selected, scans will be initiated for new commits and pull requests.

- Please note that it involves scanning with a 10-minute interval instead of a real time scanning.

Opting for "Daily (On change only)" will trigger daily scanning whenever there are new code changes.

To delete an installation, click the "Delete the installation" button corresponding to the specific Bitbucket account. Subsequently, both the installation within MergeBase and the corresponding installation on Bitbucket will be erased.

- Please note that, if the installation is directly removed from Bitbucket, the corresponding installation will also be eliminated the next time you log in to MergeBase and navigate to the settings page.



□ mbux ☑ javascript-test ☑ java-test
□ dotnet-test

□ All commits and pull requests ☑ Daily (On change only)

If you want to access this page directly, proceed to the   "Settings" page in the left hand navigation as illustrated below, and then access the "Bitbucket Integration" tab along the top.

Appendix A - Glossary

Application

From MergeBase’s perspective, an Application is a collection of components grouped under a single name in the MergeBase dashboard. An Application's list of components typically evolves over time as component versions are upgraded.

Binary Scan

A scan produced from jar files (Java only).

Blocking

The MergeBase dashboard allows users to block entire components and/or suspicious methods within running applications. When blocked, methods and components cannot execute any of their internal logic. This feature is currently only available when MergeBase is applied against Java and Bytecode based applications.

CLT (Command Line Tool)

The MergeBase CLT (Command Line Tool) is a Java application (“mergebase.jar”) that users can download from their MergeBase dashboard. The majority of MergeBase’s application security features require the CLT. To invoke the CLT after downloading it type “java -jar mergebase.jar” at a command prompt.

Component

From MergeBase’s perspective, a component is another word for “open source library”, “shared library”, “jar file”, “dll”, “software module”, etc. A component is essentially a piece re-useable 3rd-party code that developers can import into their systems. Components generally have a name and a version. Sometimes components have known-vulnerabilities associated with some of their versions.

CVE

CVE stands for “Common Vulnerabilities and Exposures,” but more importantly, CVEs represent individual records in NIST’s National Vulnerability Database. These records are the primary way software vendors communicate and coordinate patching of known-vulnerabilities with the public.

Inoculation

The MergeBase CLT allows users to “inoculate” Java Jar files. Inoculation is accomplished via bytecode rewriting. After inoculation the Jar files begin communicating with the MergeBase dashboard during their next execution, helping security analysts monitor and manage known-vulnerabilities affiliated with these Jar files in real time.

Inoculated Application

An “Inoculated App” is a Java application where its jar files have been inoculated by the MergeBase CLT.

License

A “License” represents an open-source license (or collection of licenses) associated with a component. For example, “Apache 2.0” is one such license.

Risk Score

The MergeBase risk-score is the largest CVSSv3 score (or CVSSv2 if CVSSv3 is not available) observed across all vulnerabilities associated with a component or application.

Source Scan

A scan from source using the project build files (e.g. Maven, Gradle, NPM, etc).

Suppression

Vulnerabilities that are not interesting (e.g., false positives or low priority issues) can be “suppressed” or “ignored” temporarily via the MergeBase dashboard.

Suspicious Method

A suspicious method represents a single function within a software library that MergeBase has associated with a known vulnerability.

Appendix B - Technical Architecture and Data Flows

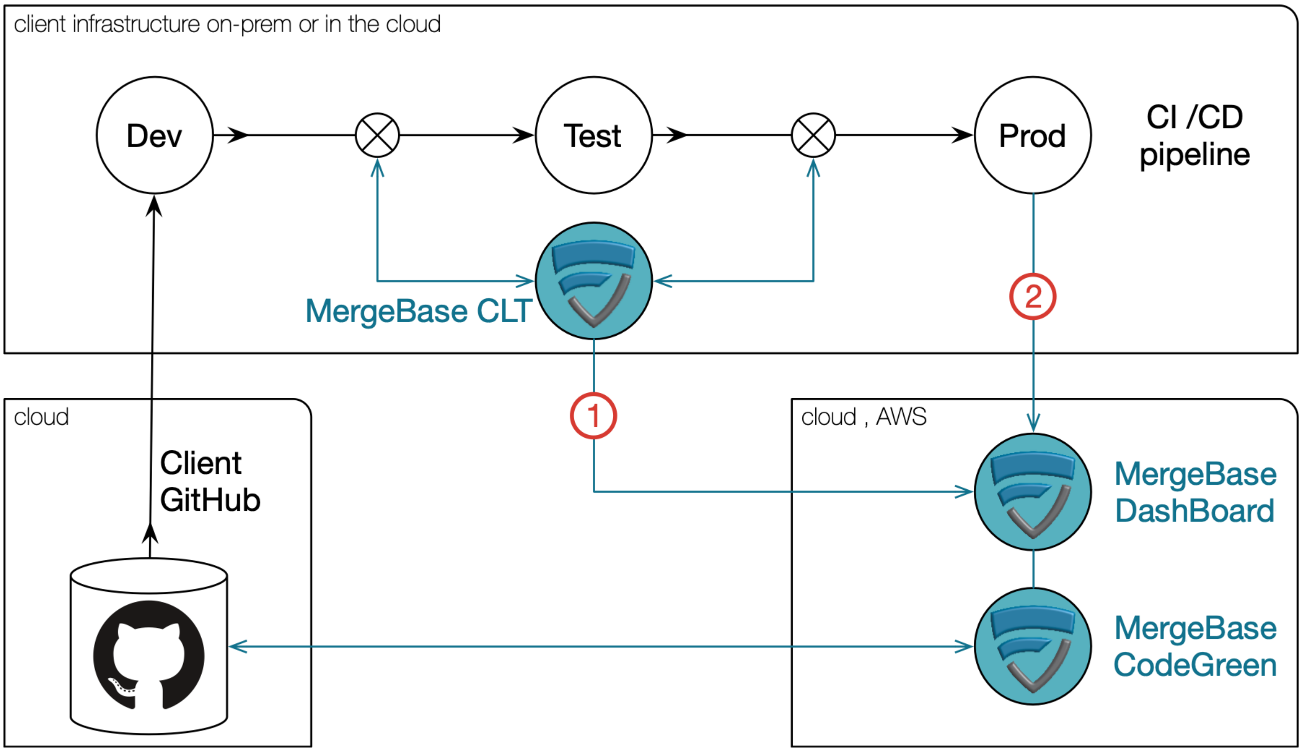
The MergBase cloud based setup is shown in this section.

MergeBase typically has three components:

Dashboard Every client gets its own dashboard with a unique URL of the form CLIENTNAME.mergebase.com.
This dashboard shows all the applications analysed and is the control center for runtime as well.

CLT The client downloads a specific version of the command line tool from its dashboard. This command line tool typically operates within the client infrastructure (on-prem, or cloud) and is for instance integrated into the buildpipeline.

CodeGreen App CodeGreen is a SaaS based app that integrates with the clients GitHub or BitBucket environment.



Connectivity

In the typical SaaS configuration the following external network connections are required:

① The CLT requires an outgoing connection to the client’s dashboard.

② The applications that are instrumented require an outgoing connection to the MergeBase dashboard in order to function properly.

Options

A number of additional configuration options are available upon request.

Offline The CLT can operate in offline mode in that case it does not require network connectivity, but its capabilities are limited and the dashboard will not be populated.

Specific AWS instance

The client can select a specific AWS instance. For instance a Canadian government entity can select to run its instance in a Montreal datacenter.

On-prem

MergeBase can also be fully deployed on prem. This applies to both the Dashboard and CodeGreen.

Appendix C - Known Issues

- The CLT cannot be used to re-sign jars on Windows
- Cannot use the ‘@’ symbol in the values for MergeBase docker image environment variables
- Uploading a Java application will fail if the application contains multiple jars that are not recognized as known open source components and they have MANIFEST.MF files with similar metadata
- Risk and Vulnerability graphs may not reflect reduction in vulnerability count if an application is scanned twice the same calendar day and:
    - the first scan produces an increase in vulnerabilities
    - the second scan removes the newly added vulnerabilities
- Inoculated applications will run on Windows 10 but may not accurately report usage or apply all mitigations.