Software Risk Manager Documentation (v2025.9.4)

Contents

Software Risk Manager User Guide

Getting Started with Software Risk Manager

Software Risk Manager (SRM) is a complete application security posture management (ASPM) solution. SRM enables you to set up policy-driven workflows to orchestrate AST tools like Coverity and Black Duck, prioritize issues, and monitor compliance across your software assets.

Software Risk Manager allows you to do the following with your AppSec data:

- Correlate results
- Prioritize vulnerabilities
- Track remediation
- Centralize risk visibility

SRM also provides issue tracking functionality as well as policy management solutions.

About This Guide

This guide provides descriptions of SRM functionality and instructions to maximize SRM deployment. Additional information can be found in the following guides:

- *Software Risk Manager Installation Guide.* This guide provides instructions for installing and configuring Software Risk Manager on Windows and Linux platforms.
- *Software Risk Manager Plugins Guide.* This guide provides information and instructions for integrating Software Risk Manager with a variety of development tools and environments.
- *Software Risk Manager API Guide.* This guide documents the various REST resources provided by Software Risk Manager, which allows external applications and scripts to interface with SRM core functionality.
- *Black Duck Bridge CLI.* This guide explains how to run scans and receive results from the command line.

Conventions Used in this Guide

The following conventions are used in this guide:

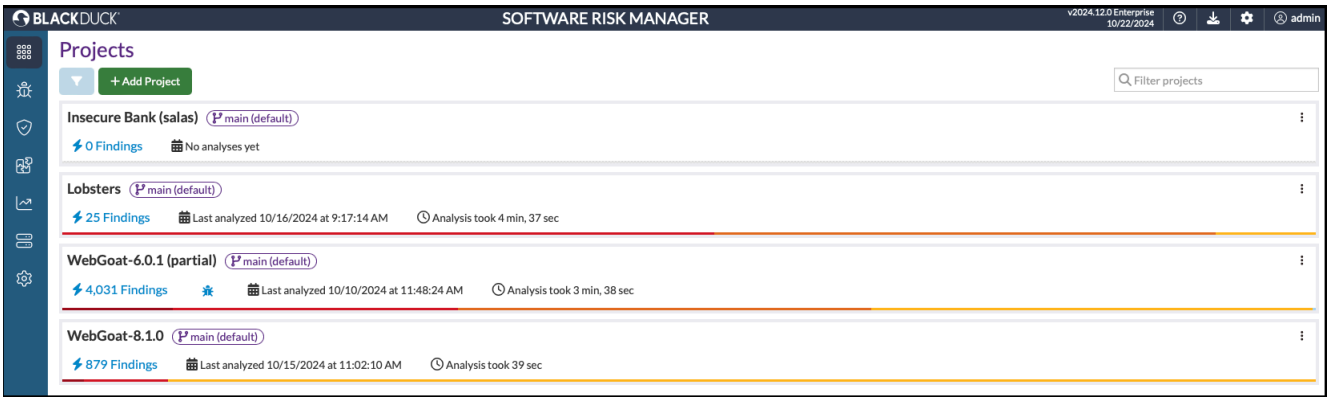
- **Page names.** The Software Risk Manager UI consists of a series of *pages.* The name of the page appears in the top-left corner of the screen. In this guide, the page name begins with a capital letter. For example, the Settings page.
- **Button names.** Tasks are performed by clicking buttons. The button name begins with a capital letter. For example, Click Save.
- **Icons.** Icons appear throughout the Software Risk Manager UI. Icons can provide a visual indication of a state or status, such as a policy violation. Icons can also serve as links to other pages. In this guide, the icons are indicated by the name of the icon, beginning with a capital letter. For example, Click the Settings icon.
- **Menu items.** Several pages in Software Risk Manager include sub-pages, which are listed as menu items along the top or left of the screen. A menu item begins with a capital letter. For example, Select License from the top menu.

- **Dropdown configuration options.** When working with certain elements, such as a project or finding, a configuration icon appears to the right of the page. The icon appears as three horizontal dots. Clicking this icon displays a dropdown list of options. Options appear in this guide by name, starting with a capital letter. For example, Click the project's dropdown configuration icon and select New Analysis.
- **Code strings and filenames.** Code strings and filenames are shown in a mono-spaced font. For example, Enter the following command: `run srm.install`

  Note: A command that is designated as "code" needs to be entered exactly as shown.

Software Risk Manager Navigation

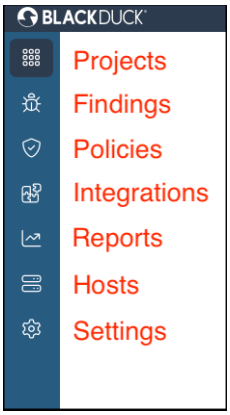
The Software Risk Manager UI consists of a series of pages, sub-pages, menus, buttons, and so on. Each page includes common elements for navigation, as can be seen the sample image below and the descriptions that follow.



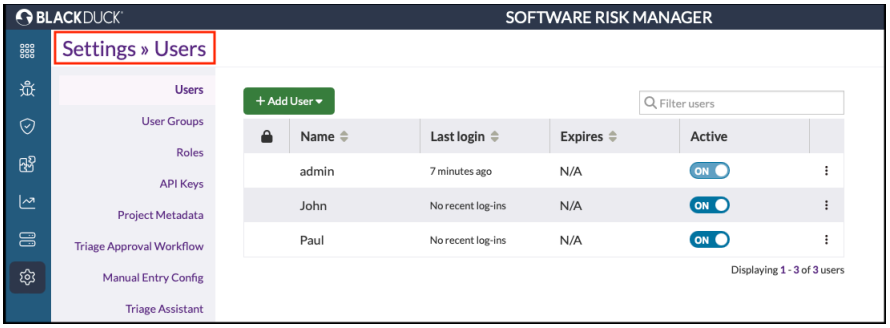
The Software Risk Manager UI is structured around seven main pages:

- **Projects.** This page shows all the currently defined projects in Software Risk Manager.
- **Findings.** This page displays all the findings from a selected analysis.
- **Policies.** This page lists all the currently defined policies, policy violations, and other policy-related data
- **Integrations.** This page provides links to all the tool integrations supported by Software Risk Manager.
- **Reports.** This page provides a list of existing reports and allows you to create reports to automatically be run on a custom schedule based on saved filters from the Findings page.
- **Hosts.** This page lists the currently defined hosts and host-related data.
- **Settings.** This page provides links to sub-pages where you can configure SRM. The Settings page and sub-pages allow you configure users, define user groups, view server logs, and so on.

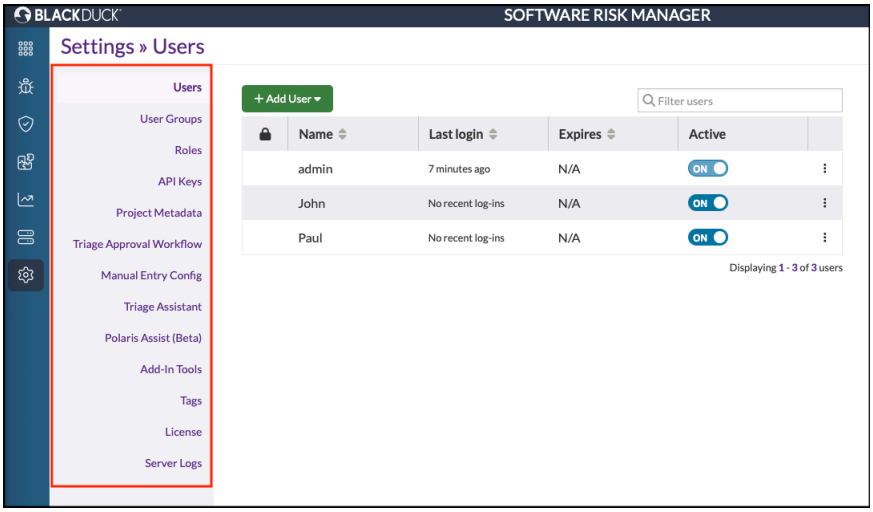
These pages are accessed by clicking their respective icons (links) in the SRM navigation bar, as shown in the figure below.



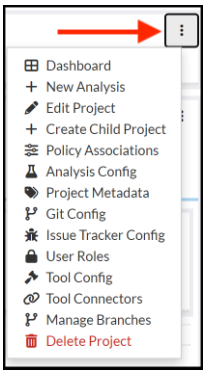
Clicking an icon from the navigation bar opens the corresponding page. The name of the page appears in the top left of the screen.



Menus, when available, appear along the left side of the page. Clicking a menu option opens the corresponding page.

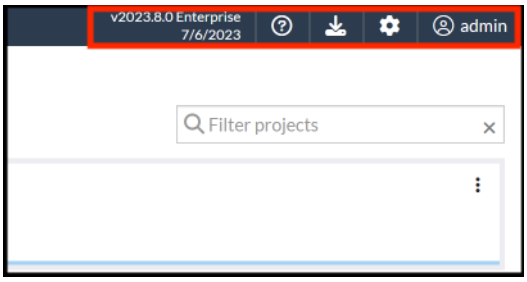


The dropdown configuration icon (three horizontal dots) is located to the right of a particular entity, such as a project or a finding, and provides a dropdown list of options.



The menu in the top right corner of the screen provides additional options and information (from left to right):

- **Software version.** This is the version number of the Software Risk Manager installation.
- **In-app documentation.** Click the question mark icon to access the Software Risk Manager documentation set. Both PDF and web versions of the guides are available.
- **Plugin downloads.** Click the plugins download icon to display a list of plugin options you can download.
- **Settings.** Click the settings icon to access the Software Risk Manager visual log.
- **[username].** Click the username icon to access the My Settings page or to log out of Software Risk Manager.



Software Risk Manager Deployment Options

Software Risk Manager has several deployment options, depending on your AppSec environment and requirements. The most common SRM deployment options are as follows:

- Data aggregation
- Tool orchestration (requires the SRM Tool Orchestration module deployed on Kubernetes)
- SRM coupled with SAST (requires the SRM Scan Farm module deployed on Kubernetes)
- SRM coupled with SCA (requires the SRM Scan Farm module deployed on Kubernetes)
- SRM coupled with SAST and SCA (requires the SRM Scan Farm module deployed on Kubernetes)

Data Aggregation from Multiple Sources

Software Risk Manager supports a large and growing list of tool connectors that can be used to collect and aggregate data from multiple sources. In addition to providing risk assessment detail for each finding, SRM provides policy violation management, dashboards, and a host of other features. For this deployment option, SRM is installed as a standalone product in a Windows or Linux environment.

Data Aggregation with Tool Orchestration

SRM with tool orchestration enables AppSec teams to build their own custom pipeline and orchestrate all scanning from a central location. If a required tool isn't currently supported, SRM allows teams to configure any commercial or open source tool to work with SRM. This option requires the SRM Tool Orchestration module and a Kubernetes deployment.

Software Risk Manager Coupled with SAST

This deployment option allows teams to run Coverity (SAST) scans seamlessly with SRM. This option requires the SRM Scan Farm module deployed on Kubernetes.

Software Risk Manager Coupled with SCA

This deployment option allows teams to run Black Duck (SCA) scans seamlessly with SRM. This option requires the SRM Scan Farm module deployed on Kubernetes.

Software Risk Manager Coupled with SAST and SCA

This deployment option allows teams to run Coverity (SAST) and Black Duck (SCA) scans seamlessly with SRM. This option requires the SRM Scan Farm module deployed on Kubernetes.

Software Risk Manager Support Information

Software Risk Manager supports the following browsers and configurations.

Supported Platforms

Software Risk Manager APIs are compatible with any operating system and hardware that can connect to the SRM server or APIs via HTTPS.

Browser Support

The SRM web UI can be accessed using any of the supported browsers shown in the following table:

Table 1:

Browser	Versions	Provider
Firefox	Latest	Versions supported by Mozilla
Google Chrome	Latest	Versions supported by Google
Microsoft Edge	Latest	Versions supported by Windows 10
Safari	Latest	Versions supported by Apple

Scan Farm Supported File Types and Tests

Note: Supported file types and tests apply to integrated Coverity and Black Duck only.

SAST Language Support

SRM supports the following SAST languages:

Table 2:

Language	Language Versions	Code Upload (UI)	Git Integration	CI via Black Duck Bridge CLI (CLI)
Salesforce Apex		Supported	Supported	Supported
C/C++	C++23 C++20 C++98 C++03 C++11 C++14 C++17 C89 C99 C11	Not Supported	Not Supported	Supported
C#	Up to C# 12	Supported	Supported	Supported
Dart	Version Agnostic	Supported	Supported	Supported
Go	Go 1.20–1.21	Not Supported	Not Supported	Supported
Java	Up to Java 21	Supported	Supported	Supported
JavaScript	ECMAScript 2023	Supported	Supported	Supported
Kotlin	1.8.0-1.8.22, 1.9.0	Not Supported	Not Supported	Supported
Objective-C/C++		Not Supported	Not Supported	Supported
PHP	Version Agnostic	Supported	Supported	Supported
Python	Python 3.x–3.11	Supported	Supported	Supported
Ruby	Matz's Reference Impl. (MRI) 1.9.2–3.2 and equivalents (via Breakman pro bundles into analysis kit)	Supported	Supported	Supported

Language	Language Versions	Code Upload (UI)	Git Integration	CI via Black Duck Bridge CLI (CLI)
Swift	Version Agnostic	Supported	Supported	Supported
TypeScript	TypeScript 1.0–5.2	Supported	Supported	Supported
Visual Basic	Up to Visual Basic 16	Not Supported	Not Supported	Supported

Infrastructure as Code: Static Testing

SRM supports the following Infrastructure as Code Static Testing.

Table 3:

Language	What is supported	Code Upload (UI)	Git Integration	CI via Black Duck Bridge CLI (CLI)
IaC	**Platforms:** AWS CloudFormation, Kubernetes, Terraform. **Formats:** HCL (Terraform), JSON, XML, YAML	Supported	Supported	Supported

SCA Language and Package Manager Support

SRM supports the following SCA languages and package manager support:

Table 4:

Package Manager	Language	Test Mode	Supported	Entry Point	Supported Detectors, Requirements	Accura
Apache Ivy	Various	Code upload or SCM integration	Not Supported			
		Black Duck Bridge CLI (CI/CLI)	Supported	Ivy Build Parse	Ivy Build Parse • Files: ivy.zml, build.zml	Low
BitBake	Various	Code upload or SCM integration	Not Supported			

Package Manager	Language	Test Mode	Supported	Entry Point	Supported Detectors, Requirements	Accura
		Black Duck Bridge CLI (CI/CLI)	Supported	Bitbake CLI		
Cargo	Rust	Code upload or SCM integration	Not Supported			
		Black Duck Bridge CLI (CI/CLI)	Supported	Cargo Lock	Cargo Lock • Files: Cartfile, Cartfile.resolved	High
Carthage	Various	Code upload or SCM integration	Not Supported			
		Black Duck Bridge CLI (CI/CLI)	Supported	Carthage Lock	Carthage Lock • Files: Cartfile, Cartfile.resolved	High
CocoaPods	Objective-C	Code upload or SCM integration	Not Supported			
		Black Duck Bridge CLI (CI/CLI)	Supported	Pod Lock	Pod Lock • Files: Podfile.lock	High
Conan	C/C++	Code upload or SCM integration	Not Supported			
		Black Duck Bridge CLI (CI/CLI)	Supported	Conan Lock	Conan Lock • Files: conan.lock	High
					Conan CLI • Files: conanfile.txt or conanfile.py • Executables: conan	High
				Conan CLI	Conan CLI • Files: conanfile.txt or conanfile.py • Executables: conan	High
Conda	Python	Code upload or SCM integration	Not Supported			
		Black Duck Bridge CLI (CI/CLI)	Supported	Conda CLI	Conda CLI	High

Package Manager	Language	Test Mode	Supported	Entry Point	Supported Detectors, Requirements	Accura
					• Files: environment.yml • Executables: conda	
CPAN	Perl	Code upload or SCM integration	Not Supported			
		Black Duck Bridge CLI (CI/CLI)	Supported	Cpan CLI	Cpan CLI • File: Makefile.PL • Executables: cpan	High
CRAN	R	Code upload or SCM integration	Not Supported			
		Black Duck Bridge CLI (CI/CLI)	Supported	Packrat Lock	Packrat Lock • File: packrat.lock	High
Dart	Dart	Code upload or SCM integration	Not Supported			
		Black Duck Bridge CLI (CI/CLI)	Supported	Dart CLI	Dart CLI • Files: pubspec.yaml, pubspec.lock • Executables: dart, flutter	High
					Dart PubSpec Lock • Files: pubspec.yaml,pubspec.lock	High
				Dart PubSpec Lock	Dart PubSpec Lock • Files: pubspec.yaml,pubspec.lock	High
Go Dep	Golang (Go)	Code upload or SCM integration	Not Supported			
		Black Duck Bridge CLI (CI/CLI)	Supported	GoDep Lock	GoDep Lock • File: Gopkg.lock	High
Go Gradle	Golang (Go)	Code upload or SCM integration	Not Supported			

Package Manager	Language	Test Mode	Supported	Entry Point	Supported Detectors, Requirements	Accura
		Black Duck Bridge CLI (CI/ CLI)	Supported	GoGradle Lock	GoGradle Lock • File: gogradle.lock	High
Go Modules	Golang (Go)	Code upload or SCM integration	Not Supported			
		Black Duck Bridge CLI (CI/ CLI)	Supported	GoMod CLI	GoMod CLI • Files: go.mod • Executables: go	High
Go Vendor	Golang (Go)	Code upload or SCM integration	Not Supported			
		Black Duck Bridge CLI (CI/ CLI)	Supported	Go Vendor	Go Vendor • Files: vendor/ vendor.json	High
				GoVndr CLI	GoVndr CLI • Files: vendor.conf	High
Gradle	Various	Code upload or SCM integration	Supported	Gradle Project Inspector	Gradle Project Inspector • Files: build.gradle	Low
		Black Duck Bridge CLI (CI/ CLI)	Supported	Gradle Native Inspector	Gradle Native Inspector • Files: build.gradle or build.gradle.kts • Executables: gradlew or gradle	High
					Gradle Project Inspector • Files: build.gradle	Low
Hex	Erlang	Code upload or SCM integration	Not Supported			
		Black Duck Bridge CLI (CI/ CLI)	Supported	Rebar CLI	Rebar CLI • Files: rebar.config • Executables: rebar3	High
Lerna	Node.js	Code upload or SCM integration	Not Supported			
		Black Duck Bridge CLI (CI/ CLI)	Supported	Lerna CLI	Lerna CLI • Files: lema.json, package.json	High

Package Manager	Language	Test Mode	Supported	Entry Point	Supported Detectors, Requirements	Accura

					• Executables: Lerna, and one of the following:
					  • packagelock.json
					  • npmshrinkwrap.json
					  • yarn.lock	

Maven	Various	Code upload or SCM integration	Supported	Maven Project Inspector	Maven Project Inspector • Files: pom.xml	Low
		Black Duck Bridge CLI (CI/CLI)	Supported	Maven CLI	Maven CLI • Files: pom.xml • Executables: mvnw or mvn	High
					Maven Project Inspector • Files: pom.xml	Low
				Maven Wrapper CLI	Maven Wrapper CLI • Files: pom.groovy • Executables: mvnw or mvn	High
					Maven Project Inspector • Files: pom.xml	Low
npm	Node.js	Code upload or SCM integration	Supported	NPM Package Lock	NPM Package Lock • Files: packagelock.json. For better results, include a package.json also.	High
				NPM Package Json Parse	NPM Package Json Parse • Files: package.json	Low
		Black Duck Bridge CLI (CI/CLI)	Supported	NPM Shrinkwrap	NPM Shrinkwrap • Files: npm-shrinkwrap.json. For better results, include a package.json also.	High
					NPM Package Lock • Files: packagelock.json. For better results,	High

Package Manager	Language	Test Mode	Supported	Entry Point	Supported Detectors, Requirements	Accura
					include package.json also.	
					NPM CLI • Files node_modules, package.json • Executables: npm	High
					NPM Package Json Parse • Files: packagelock.json	High
				NPM Package Lock	NPM Package Lock • Files: packagelock.json. For better results, include a package.json also.	High
					NPM CLI • Files: node_modules, package.json • Executables: npm	High
					NPM Package Json Parse • Files: package.json	Low
				NPM CLI	NPM CLI • Files: node_modules, package.json • Executables: npm	High
					NPM Package Json Parse • Files: package.json	Low
				NPM Package Json Parse	NPM Package Json Parse • Files: package.json	Low
NuGet	C#	All	Supported	NuGet Solution Native Inspector	NuGet Solution Native Inspector • Files: A solution file with a .sln extension	High
					NuGet Project Inspector	Low

Package Manager	Language	Test Mode	Supported	Entry Point	Supported Detectors, Requirements	Accura
					• Files: A project file with the .csproj or .sln extension	
				NuGet Project Native Inspector	NuGet Project Native Inspector • Files: A project file with the csproj, .fsproj, .vbproj, or .rproj extension	High .asaproj,
					NuGet Project Inspector • Files: A project file with the .csproj or .sln extension	Low
Packagist	PHP	Code upload or SCM integration	Not Supported			
		Black Duck Bridge CLI (CI/CLI)	Supported	Composer Lock	Composer Lock • Files: composer.lock, composer.json	High
PEAR	PHP	Code upload or SCM integration	Not Supported			
		Black Duck Bridge CLI (CI/CLI)	Supported	Pear CLI	Pear CLI • Files: package.xml • Executables: pear	High
pip	Python	Code upload or SCM integration	Supported	Pipfile Lock	Pipfile Lock • Files: Pipfile or Pipfile.lock	High
		Black Duck Bridge CLI (CI/CLI)	Supported	Pipenv Lock	Pipfile Lock • Files: Pipfile or Pipfile.lock • Executables: python or python3, and pipenv	High
					PIP Native Inspector • Files: setup.py, or one or more requirements.txt	High

Package Manager	Language	Test Mode	Supported	Entry Point	Supported Detectors, Requirements	Accura
					• Executables: python and pip, or python3 and pip3	
					Pipfile Lock • Files: Pipfile, Pipfile.lock	High
				Pip Native Inspector	PIP Native Inspector • Files: setup.py, or one or more requirements.txt • Executables: python and pip, or python3 and pip3	High
					Pipfile Lock • Files Pipfile, Pipfile.lock	High
				Pipfile Lock	Pipfile Lock • Files: Pipfile, Pipfile.lock	High
pnpm	Node.js	All	Supported	Pnpm Lock	Pnpm Lock • Files pnpmlock.yaml, package.json	High
Poetry	Python	All	Supported	Poetry Lock	Poetry Lock • Files Poetrylock, pyproject.toml	High
RubyGems	Ruby	Code upload or SCM integration	Not Supported			
		Black Duck Bridge CLI (CI/CLI)	Supported	Gemfile Lock	Gemfile Lock • Files: Gemfile.lock	High
					Gemspec Parse • Files: A gemspec file with the .gemspec extension	Low
				Gemspec Parse	Gemspec Parse • Files: A gemspec file with a .gemspec extension	Low

Package Manager	Language	Test Mode	Supported	Entry Point	Supported Detectors, Requirements	Accura
SBT	Scala	Code upload or SCM integration	Not Supported			
		Black Duck Bridge CLI (CI/CLI)	Supported	Sbt Native Inspector	Sbt Native Inspector • Files: build.sbt • Plugins: Dependency Graph	High
Swift	Swift	Code upload or SCM integration	Supported	Swift Lock	Swift Lock • Files: Package.swift, Package.resolved	High
		Black Duck Bridge CLI (CI/CLI)	Supported	Swift Lock	Swift Lock • Files: Package.swift, Package.resolved	High
					Swift CLI • Files: Package.swift • Executables: swift	High
				Swift CLI	Swift CLI • Files: Package.swift • Executables: swift	High
Xcode	Swift	Code upload or SCM integration	Not Supported			
		Black Duck Bridge CLI (CI/CLI)	Supported	Xcode Workspace Lock	Xcode Workspace Lock • Directories: *.xcworkspace	High
					Xcode Project Lock • Directories: *.xcodeproj • Files: Package.resolved	
				Xcode Project Lock	Xcode Project Lock • Directories: *.xcodeproj • Files: Package.resolved	
Yarn	Node.js	All	Supported	Yarn Lock	Yarn Lock • Files: yarn.lock, package.json	High

SCA Package Manager Versions

SRM supports the following SCA package manager versions:

Note: Package manager version requirements are only applicable to tests created with Black Duck Bridge CLI (when testing relies on/requires access to executables). "N/A" in the table below indicates buildless capture is used to test projects that depend on the package manager.

Table 5:

Package Manager	Latest Supported Version
Apache Ivy	N/A
Bazel	4.2.0
BitBake	2.6.0 (Yocto 4.3.2)
Cargo	N/A
Carthage	N/A
CocoaPods	N/A
Conan	2.0.14
Conda	4.10.3
CPAN	Cpan Script 1.678 CPAN.pm 2.36 Cpanm 1.7047
CRAN	N/A
Dart	Dart 3.1.2 Flutter 3.13.4
Go	1.20.4
Go Dep	N/A
Gogradle	N/A
Go Modules	1.20.4
Go Vendor	N/A
Gradle	8.2.1
Hex	Rebar 3.20.0
Lerna	6.6.2
Maven	3.8.1
npm	Node 20.5.1 npm 9.8.1
NuGet	nuget 6.2 .NET runtime is not required with 7.13.0

Package Manager	Latest Supported Version
Packagist	N/A
PEAR	1.10.12
pip	23.1.2
pnpm	N/A
Poetry	N/A
RubyGems	2.0.0
SBT	1.5.0
Swift	5.6.1
Xcode	N/A
Yarn	4.1.0

Running a Basic Analysis with Sample Data

To provide an overview of how to use Software Risk Manager to run an analysis, the following is an outline of the process:

1. Make sure SRM has been installed and configured.
2. Launch the app and log in.
3. Create a project.
4. Configure the parameters for a new analysis.
5. Upload the source files.
6. Manually start a New Analysis for the project.
   SRM will begin analyzing the code. (The analysis will run in the background until complete.)
7. Inspect the findings from the project Findings page or the Dashboard.

Sample Data Sets

If you would like to use sample code for testing purposes, the following are some datasets that are all intentionally vulnerable applications used for educational and training purposes. They're referenced by their primary language, although some of them are multi-language.

- Java - WebGoat
  We recommend you use one of the WebGoat released war files directly as the input for Software Risk Manager since those tend to package everything, including the source, bytecode, and third-party dependencies. For instance, try this release.
- .NET - WebGoat.NET
  Since the Software Risk Manager .NET scanners require compiled assemblies, you will need to download the WebGoat.NET source and build it on your machine. Instructions for how to do so are at the link above.

For the following datasets, you can configure your new project's *Git Config* to fetch the source directly from GitHub using their git URL.

- Ruby on Rails - RailsGoat
  git URL: https://github.com/OWASP/railsgoat.git
- JavaScript - NodeGoat
  git URL: https://github.com/OWASP/NodeGoat.git

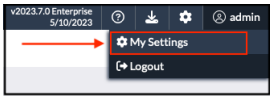
For other datasets, we recommend that you browse GitHub for different projects and scan some of them for testing purposes. Here are some queries to get you started:

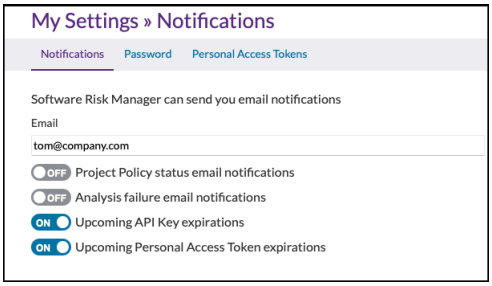
- Java
- C
- C++
- PHP
- Scala
- Python
- Ruby on Rails
- JavaScript

User Configuration Settings (My Settings)

User Configuration settings, or "My Settings," allows you to set notifications, manage passwords, and configure personal access tokens, which can be used to access the Software Risk Manager REST API.

Click your username in the upper right corner of the page and select My Settings to open the My Settings page.





For more information on user configuration settings, see the following topics:

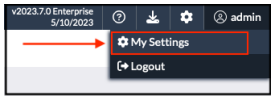
- Configuring email notifications
- Changing your password
- Managing personal access tokens

Configuring Email Notifications

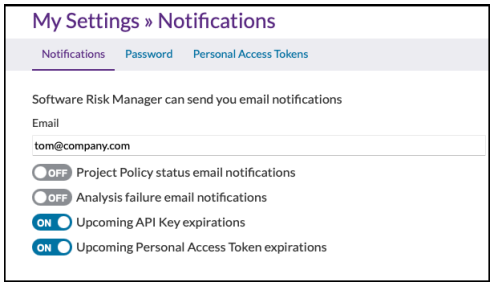
Email notifications allow you to receive emails when a policy status changes or when there's an analysis failure. (For more information on polices and policy status, see the Policies Overview section.)

To configure email notifications:

1. Click your username in the upper right corner of the page and select My Settings from the dropdown menu.



2. Select Notifications from the top menu to open the Notifications page.



3. Enter your email address in the Email field.
4. Use the toggles to enable email notifications.
   There are four notification options:
   - **Project Policy status email notifications.** Sends an email when there's a policy status change for a project.
   - **Analysis failure email notifications.** Sends an email when there is an analysis failure for a project.
   - **Upcoming API Key expirations.** Sends an email when an API Key is due to expire.
   - **Upcoming Personal Access Token expirations.** Sends an email when a Personal Access Token is due to expire.

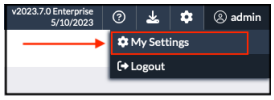
   Changes are saved automatically.

Changing Your Password

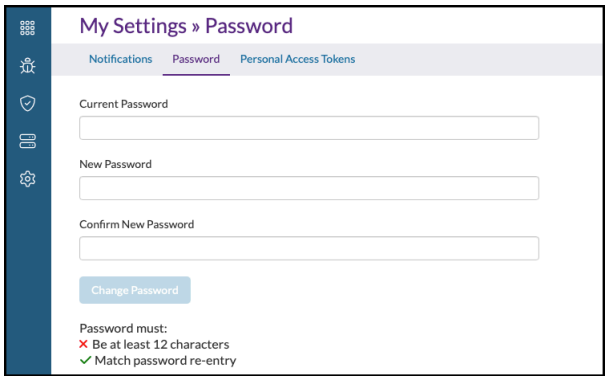
Users can change their own password from the My Settings page. (Admins can also change user passwords. See Changing a User Password.)

To change your password:

1. Click your username in the upper right corner of the page and select My Settings from the dropdown menu.



2. Select Password from the top menu.



3. Enter your current password.
4. Enter and confirm your new password.
   Passwords must be at least 12 characters.
5. Click Change Password.

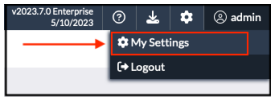
Managing Personal Access Tokens

The Personal Access Tokens page displays a list of users and usage data and allows you to generate a new token or delete an existing one.

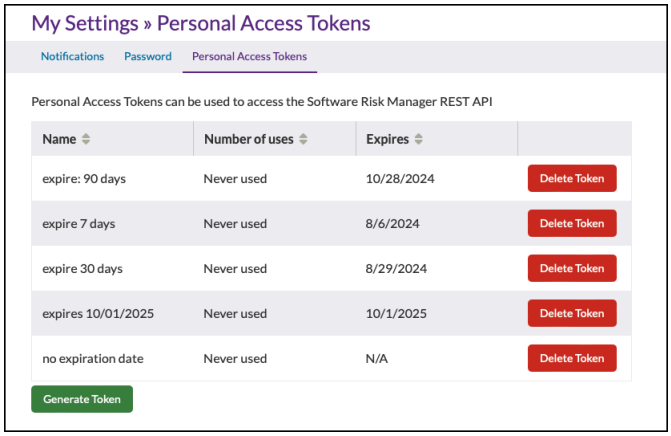
Viewing Existing Tokens

To view personal access tokens:

1. Click your username in the upper right corner of the page and select My Settings from the dropdown menu.



2. Select Personal Access Tokens from the top menu.

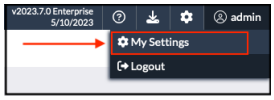


This page displays the existing tokens and usage data. Click the column headers to sort the list.

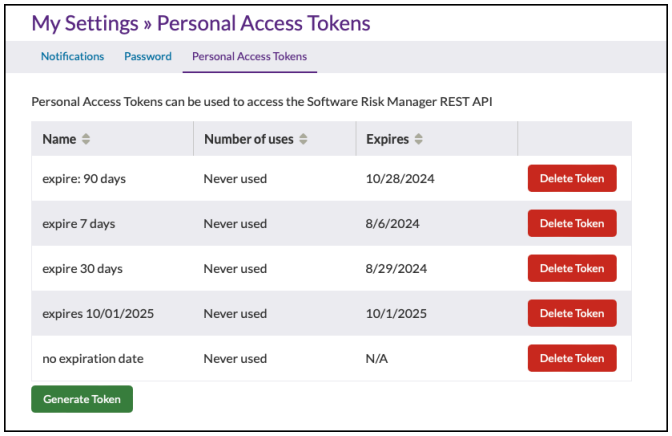
Generating Personal Access Tokens

To generate a personal access token:

1. Click your username in the upper right corner of the page and select My Settings from the dropdown menu.

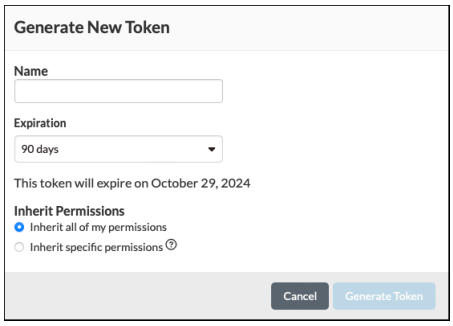


2. Select Personal Access Tokens from the top menu.



This page displays the existing tokens and usage data.

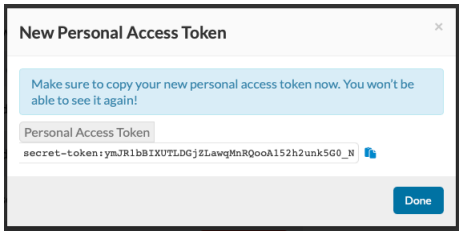
3. Click Generate Token to create a new personal access token.



4. Enter a name for the token.
5. Select an expiration. (The default is 90 days.)
6. Select permission options:
   - **Inherit all of my permissions.** The token will include all of your configured roles.
   - **Inherit specific permissions** (Read, Create, Update, Manage). The token will inherit the selected roles only. For example, selecting "Read" will allow the token to read any project where you have the "Read" permission set.

   Note: This setting will not override or grant permissions that a user doesn't already have.

7. Click Generate Token.

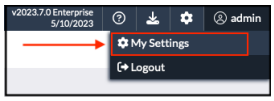


8. Copy and save your new token.
   Once this window is closed, the token cannot be redisplayed.
9. Click Done to close the window.

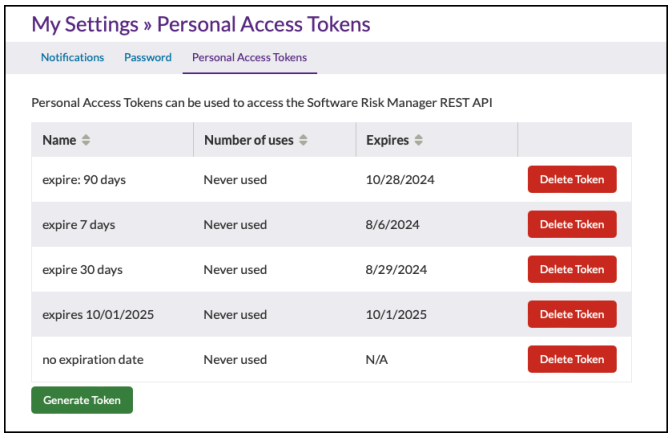
Deleting a Token

To delete a personal access token:

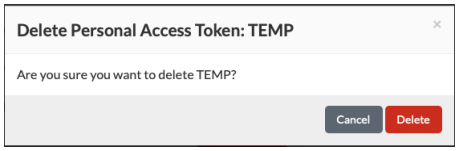
1. Click your username in the upper right corner of the page and select My Settings from the dropdown menu.



2. Select Personal Access Tokens from the top menu.



3. Locate the token you want to delete and click Delete Token.

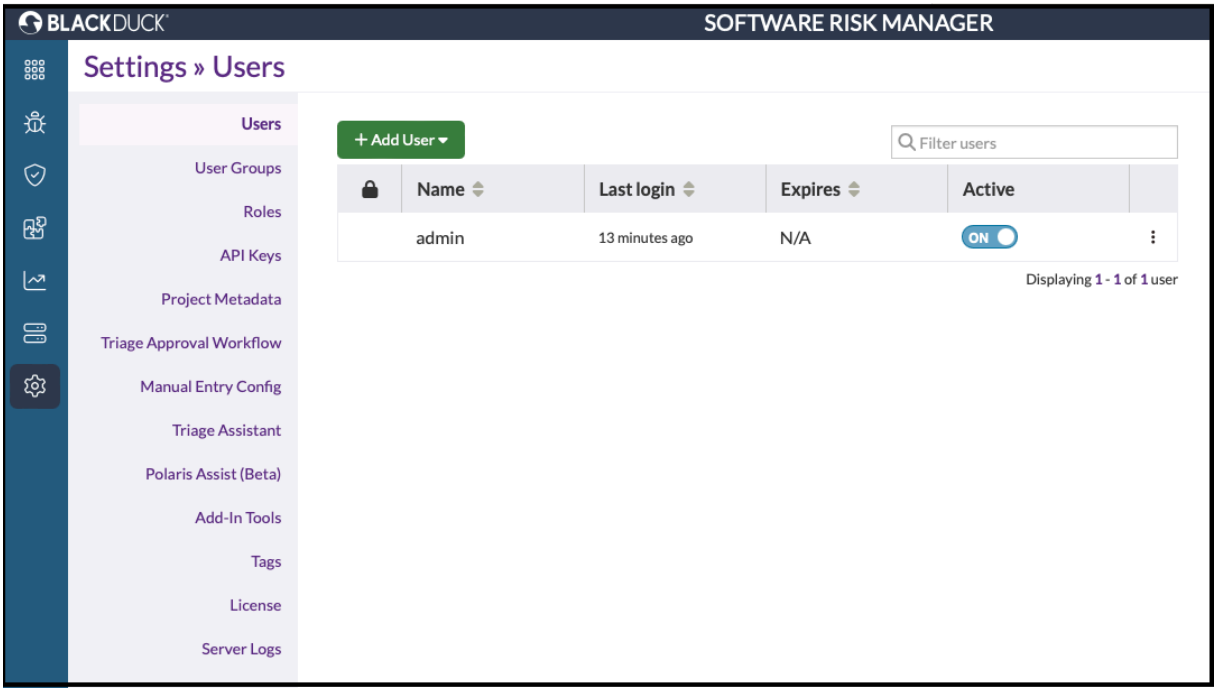


4. Click Delete to confirm.

Configuring Software Risk Manager

The Settings pages are used to configure Software Risk Manager.

Click the Settings icon in the left navigation bar to open the Settings page.



Select a settings option from the left menu to open the corresponding Settings page. The page currently being displayed is shown in the top left corner.

For detailed information on each page, see the following topics:

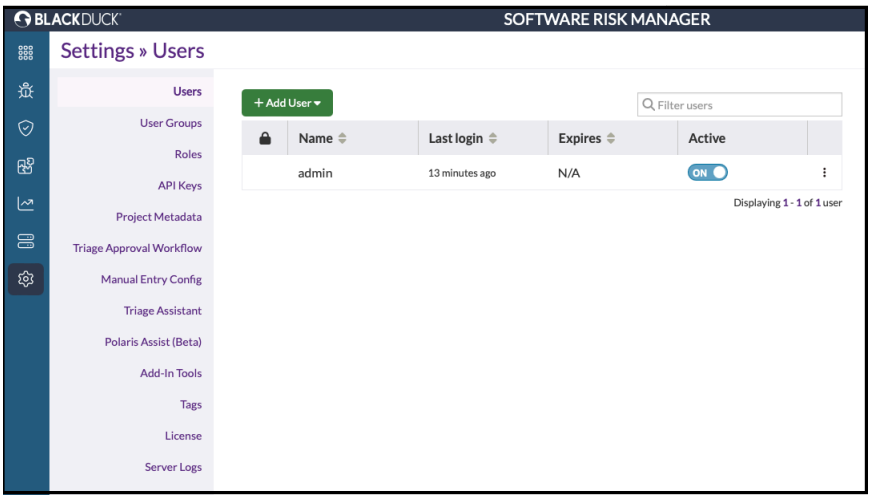
- Users. Add new users, configure user roles, manage existing user profiles, and so on.
- User Groups. Create and manage groups of users.
- Roles. Create and Manage Roles.
- API Keys. Generate and configure API keys.
- Project Metadata. Create, define, and configure custom metadata for projects.
- Triage Approval Workflow. Configure how status change requests are handled.
- Manual Entry Config. Define custom values that can be entered into selected fields and added as manual results in a findings report.
- Triage Assistant. Configure SRM to use previous triage determinations to make predictions about future findings.
- Polaris Assist (Beta). Generate AI Insight information.
- Add-In Tools. Configure additional tools that can be used to analyze code.
- Tags. Create and manage tags that can be assigned to findings.
- License. View license information with an option to update the current license.
- Server Logs. View events and errors associated with an analysis.

User Administration

Before running an analysis, an admin needs to configure user profiles, which includes user roles, permissions, group associations, and so on.

Note: Users can be managed individually or in groups. (To manage user groups, see Managing User Groups.)

Click the Settings icon in the navigation bar and select Users from the left menu to open the Users page.



This page displays a list of current users, the date the user last logged in, and whether that user is active. Clicking the column headings will re-sort the list.

For more information on configuring user profiles, see the following topics:

- Viewing existing users
- Adding a user
- Configuring user profiles
- Changing a user password
- Deleting a user profile

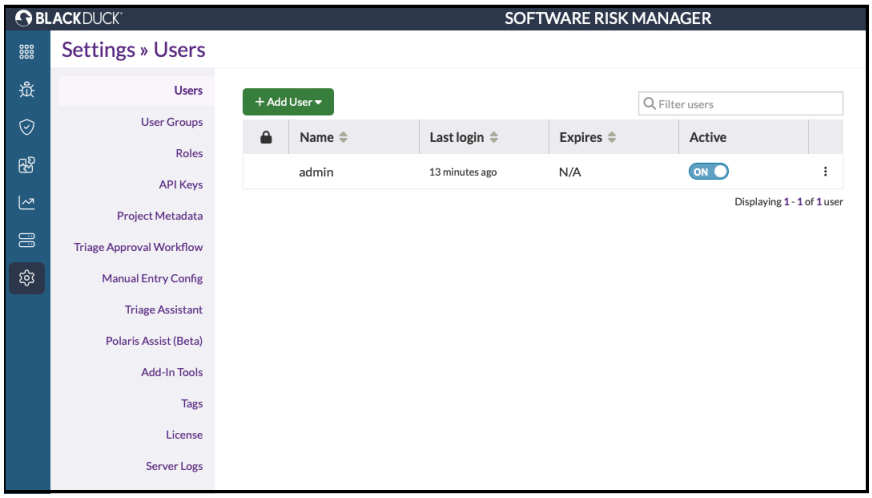
Viewing Existing Users

The User page allows you to view a list of existing users, the date of their last login, and whether they are active.

Note: The superuser's admin and active states may not be modified.

To view a list of existing users:

1. Click the Settings icon in the navigation bar and select Users from the left menu.



This page lists each user, when they last logged in, and whether they are "active."

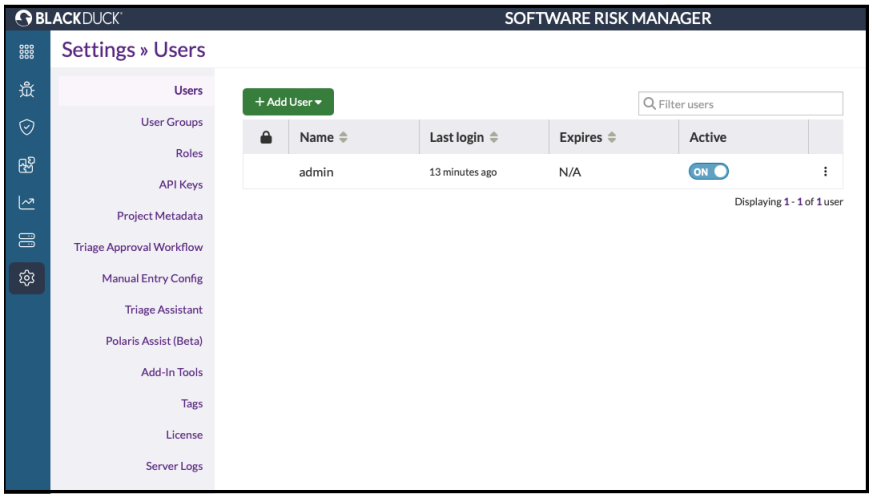
2. Use the filter field to search for a specific user, or click the column headings to re-sort the list.

Adding a User

Admins can add three different types of users: Local, LDAP, and SAML (depending on your configuration).

To add a user:

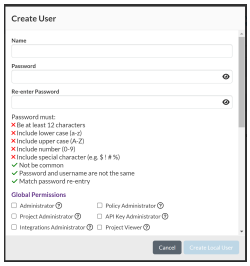
1. Click the Settings icon in the navigation bar and select Users from the left menu.



2. Click Add User and select a user type.
   There are three user types:

   - **Local Users.** Local Users exist only within Software Risk Manager. You pick a username and password for them. Software Risk Manager keeps their credentials in its database.
   - **LDAP Users.** LDAP Users can be added to Software Risk Manager by their username, but their password is managed by an external LDAP server. When an LDAP user logs in, Software Risk Manager will send their credentials to that server in order to authenticate the user.
   - **SAML Users.** SAML Users can be added to Software Risk Manager by their username, but authentication is handled by an external SAML provider. When a user reaches the Login page, Software Risk Manager will redirect them to the Sign On Portal of your SAML provider in order to

authenticate the user. They may see the standard Software Risk Manager Login page with a link to sign in via SAML, depending on your configuration.



3. Enter a name, password, and confirm the new password.
4. Use the toggles to set global permissions for the user.
   - **Administrator.** Grants user admin privileges. Admin users inherit all roles.
   - **Project Administrator.** Allows the user to create a new project.
   - **Integrations Administrator.** Allows user to manage centralized project configuration.
   - **Policy Administrator.** Allows user to create polices.
   - **API Key Administrator.** Allows user to manage API Keys.
   - **Project Viewer.** Allows user to view all projects.
5. Select which roles the user will have for each project.
   - **Read.** The user or user group can see the specified project and all of its contents. If a user doesn't have the *Read* role for a particular project, that project will not appear in the Projects page for that user.
   - **Update.** The user or user group can change the finding status and comment on findings for the specified project.
   - **Create.** The user or user group can create new analyses for the specified project
   - **Manage.** The user or user group can manage the specified project's configuration (e.g., Git, Issue tracker, etc.). The *Manage* role also allows the user to delete the specified project.
6. Click Create Local User.

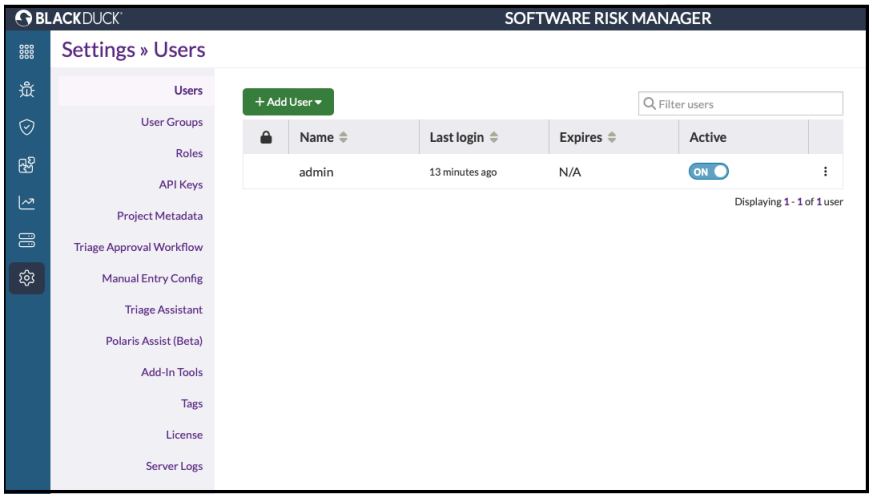
Configuring a User Profile

Configuration settings in a user profile determines how a user interacts with Software Risk Manager globally and what the user will be able to do on a project-by-project basis. Global permissions (or roles) include Administrator, Project Administrator, and Integrations Administrator. Next, users can be assigned specific roles for individual projects: Read, Update, Create, and Manage.

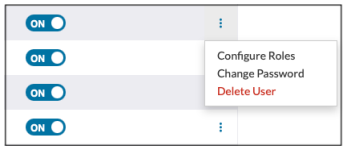
Note: The Super User's admin and active states may not be modified.

To edit an existing user profile:

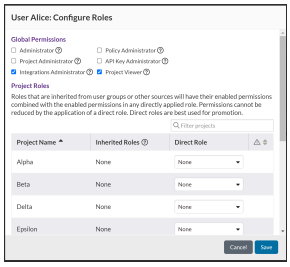
1. Click the Settings icon in the navigation bar and select Users from the left menu.



2. Click the dropdown configuration icon to the right of the user name and select Configure Roles.



This opens the "Configure Roles" window.



3. Configure user roles as needed.
Click on a role to select it. Click Clear to remove all selections.

Global Permissions. Select global permissions for the new user.

- **Administrator.** Grants user admin privileges. Admin users inherit all roles.
- **Project Administrator.** Allows the user to create a new project.
- **Integrations Administrator.** Allows user to manage centralized project configuration.
- **Policy Administrator.** Allows user to create polices.
- **API Key Administrator.** Allows user to manage API Keys.
- **Project Viewer.** Allows user to view all projects.

Project Roles. Select permissions for individual projects.

- **Read.** The user or user group can see the specified project and all of its contents. If a user doesn't have the *Read* role for a particular project, that project will not appear in the Projects page for that user.
- **Update.** The user or user group can change the finding status and comment on findings for the specified project.

- **Create.** The user or user group can create new analyses for the specified project
- **Manage.** The user or user group can manage the specified project's configuration (e.g., Git, Issue tracker, etc.). The *Manage* role also allows the user to delete the specified project.

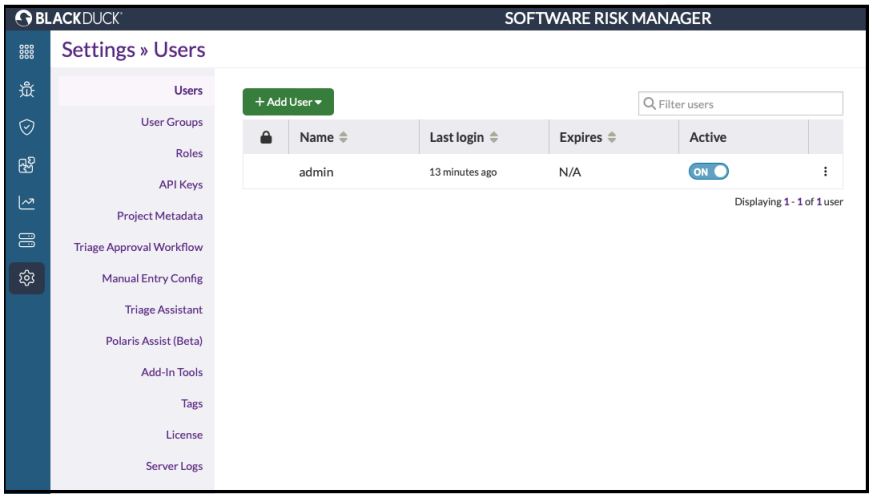
4. Click Save.

Changing a User Password

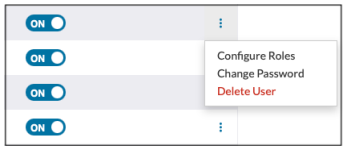
Admins can change the password of any user. (Users can change their own passwords on the My Settings page. See Changing Your Password.)

To change a user's password:

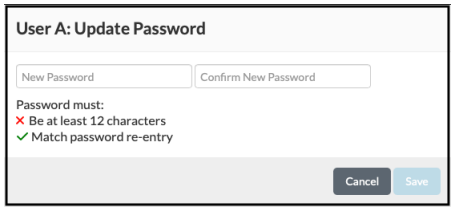
1. Click the Settings icon in the navigation bar and select Users from the left menu.



2. Click the dropdown configuration icon to the right of the user name and select Change Password.



This opens the change password window.



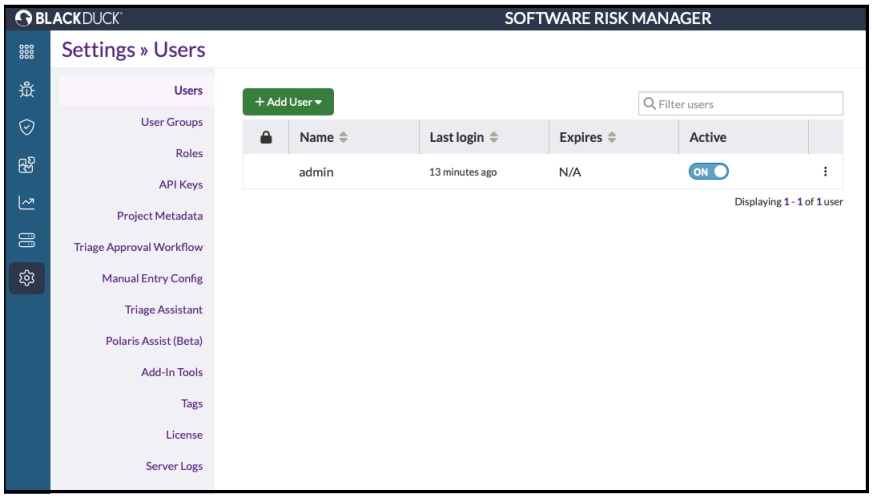
3. Enter and confirm the new password.
   Passwords must be at least 12 characters.
4. Click Save.

Deleting a User Profile

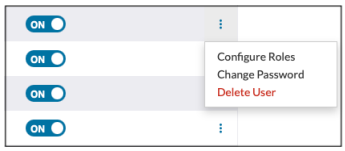
Admins can delete one or more users.

To delete a user:

1. Click the Settings icon in the navigation bar and select Users from the left menu.



2. Click the dropdown configuration icon to the right of the user name and select Delete User.

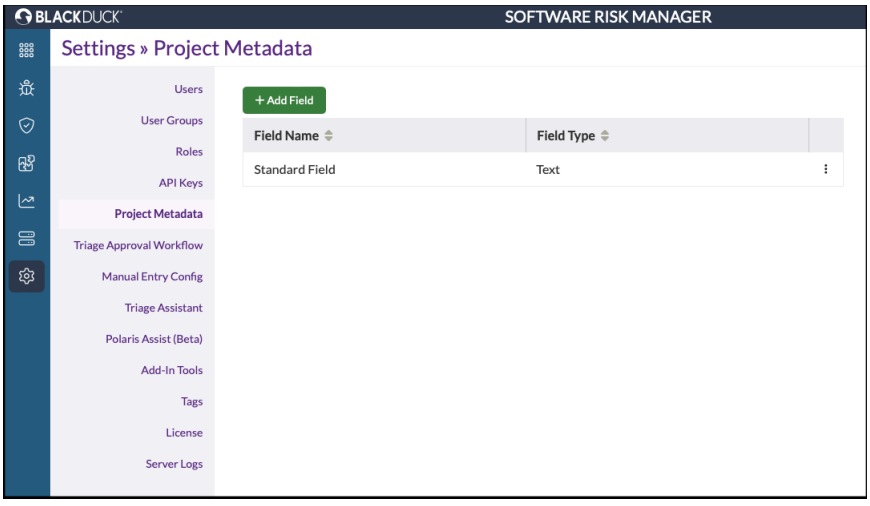


3. Click Delete to confirm.

Adding and Configuring Project Metadata Fields

Project Metadata Fields allow Software Risk Manager users to create and configure custom metadata for specified projects. Once a field has been defined, you can use that field to enter metadata values for specified projects. For more information on adding metadata values to projects, see Configuring Project Metadata.

Click the Settings icon in the navigation bar and select Project Metadata Fields from the left menu to open the Metadata Fields page.



The Project Metadata Fields page lists currently defined field names and types. Clicking on the column header will re-sort the list.

There are four field types:

- **Text.** A regular text input that allows a single line of text.
- **Multiline.** A larger text input that allows for multiple lines of text.
- **Dropdown.** Text input that a user can select from a dropdown list.
- **Tags.** A special input that behaves similarly to Text, but each individual word is converted to a "tag." As you type, pressing the space bar will convert whatever text you already had into a tag. (A space is required after the last tag.)

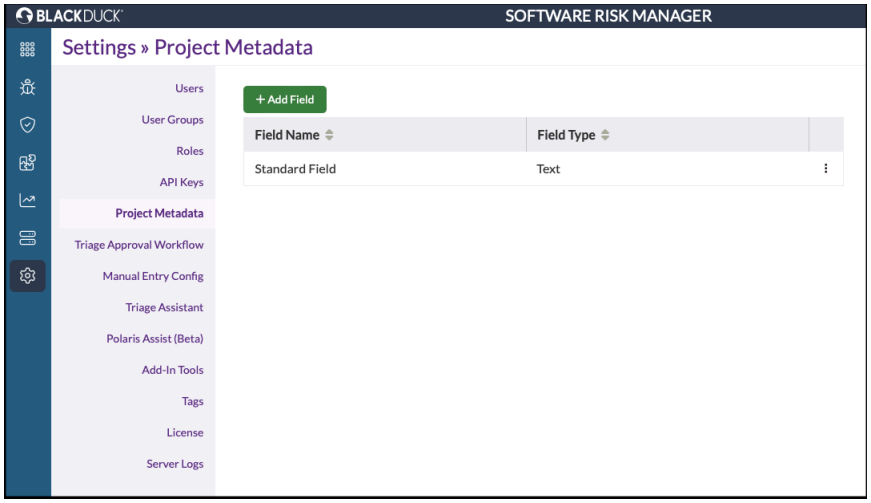
For more information on project metadata fields, see the following topics:

- Viewing existing project metadata fields
- Adding a project metadata field
- Editing a project metadata field
- Deleting a project metadata field

Viewing Existing Metadata Fields

To view a list of existing metadata fields:

1. Click the Settings icon in the navigation bar and select Project Metadata Fields from the left menu.



   This page shows the existing metadata fields and field type. There are four field types:

   - **Text.** A regular text input that allows a single line of text.
   - **Multiline.** A larger text input that allows for multiple lines of text.
   - **Dropdown.** Text input that a user can select from a dropdown list.
   - **Tags.** A special input that behaves similarly to Text, but each individual word is converted to a "tag." As you type, pressing the space bar will convert whatever text you already had into a tag. (A space is required after the last tag.)

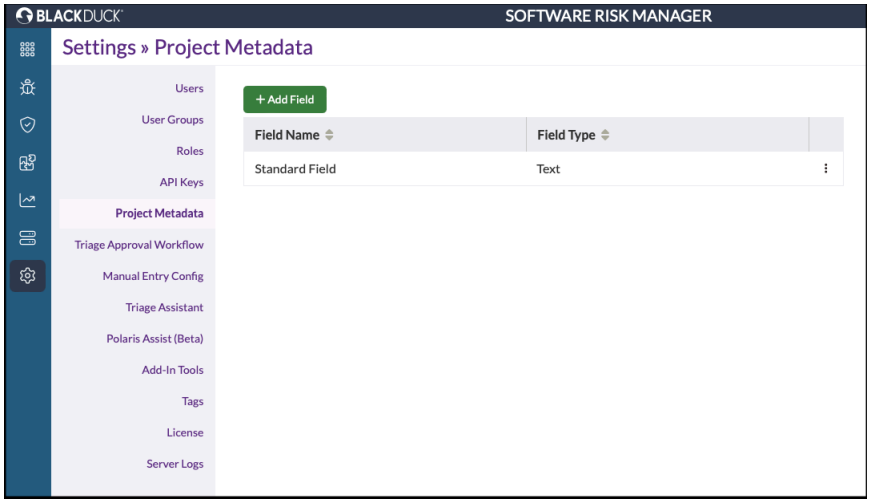
2. Click on the column headers to re-sort the list.

Adding a Metadata Field

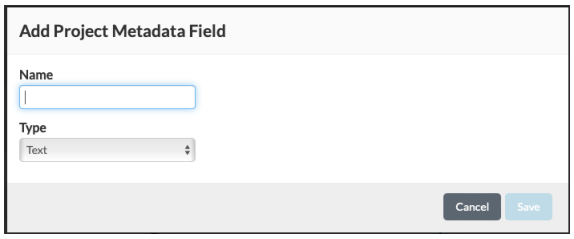
Metadata fields are defined by name and type.

To add a metadata field:

1. Click the Settings icon in the navigation bar and select Project Metadata Fields from the left menu.



2. Click Add Field.



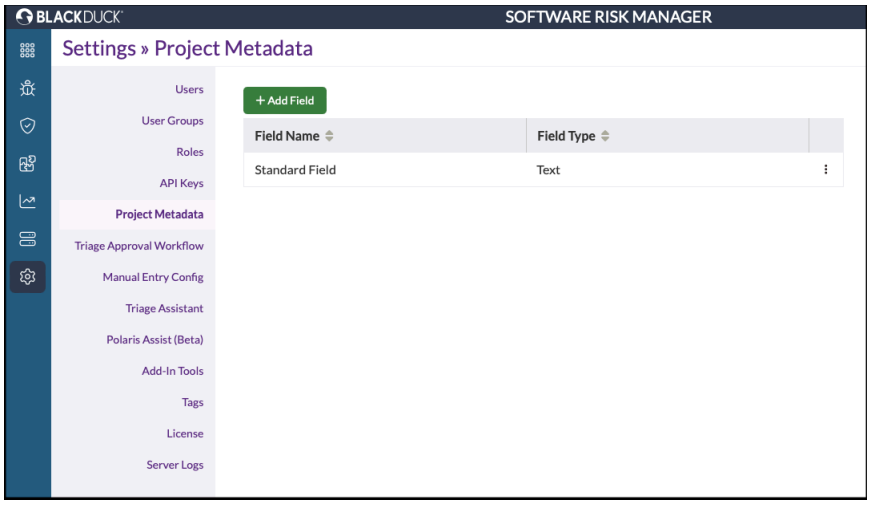
3. Enter a name for the field and select a field type.
   There are four field types:
   - **Text.** A regular text input that allows a single line of text.
   - **Multiline.** A larger text input that allows for multiple lines of text.
   - **Dropdown.** Text input that a user can select from a dropdown list.
   - **Tags.** A special input that behaves similarly to Text, but each individual word is converted to a "tag." As you type, pressing the space bar will convert whatever text you already had into a tag. (A space is required after the last tag.)
4. Click Save.

Editing a Metadata Field

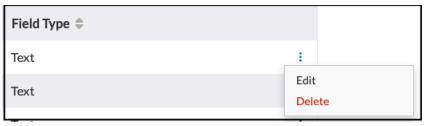
For existing metadata fields, both the field name and field type can be edited.

To edit a metadata field:

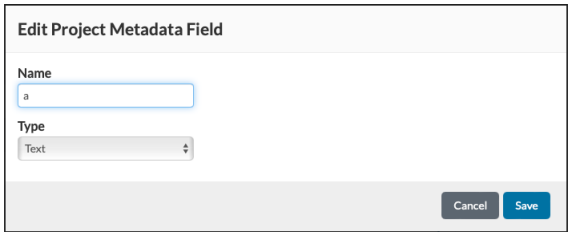
1. Click the Settings icon in the navigation bar and select Project Metadata Fields from the left menu.



2. Click the field's dropdown configuration icon and select Edit.



This opens the Edit Project Metadata Field window.



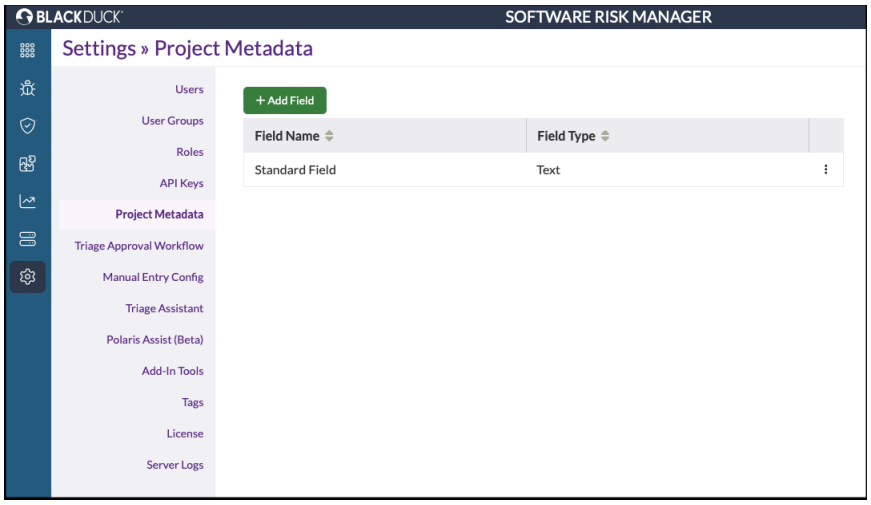
3. Make changes to the fields as needed.
4. Click Save.

Deleting a Metadata Field

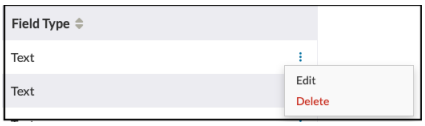
Once a metadata field is no longer needed, it can be deleted.

To delete a metadata field:

1. Click the Settings icon in the navigation bar and select Project Metadata Fields from the left menu.



2. Click the field's dropdown configuration icon and select Delete.

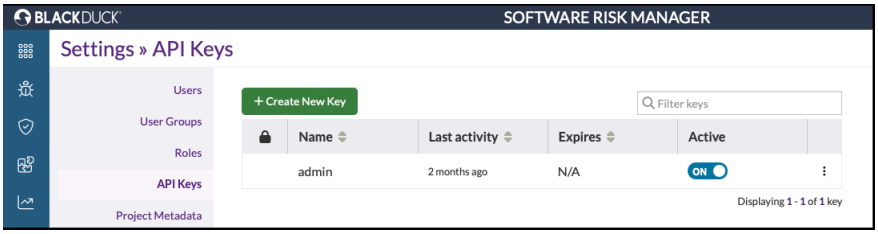


3. Click Delete to confirm.

API Keys Administration

API Keys can be generated for use with the Software Risk Manager API. Typically one key would be generated for a specific purpose, such as integrating with a specific tool or plugin. This would allow for fine-grained control over each API key's active/inactive state, as well as setting specific user roles for each key. In addition, API keys can be subject to expiration (default is 90 days).

Click the Settings icon in the navigation bar and select API Keys from the top menu to open the API Keys page.



This page lists the currently assigned API keys, shows when they were last active, and if they are currently active.

Note: For more information on Software Risk Manager API capabilities, please refer to the *Software Risk Manager API Guide*.

For more information on administering API keys, see the following topics:

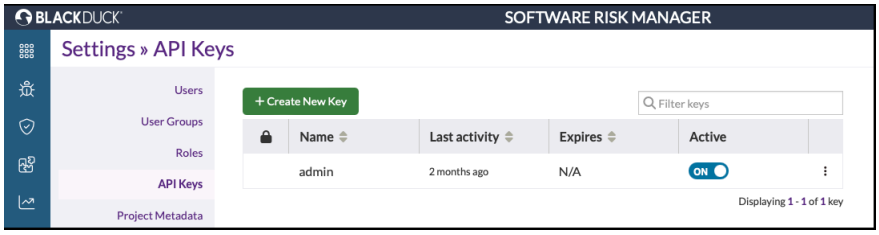
- Viewing existing API keys
- Creating an API key
- Editing an API key
- Regenerating an API key

- Deleting an API key

Viewing Existing API Keys

To view a list of existing API keys:

1. Click the Settings icon in the navigation bar and select API Keys from the left menu.



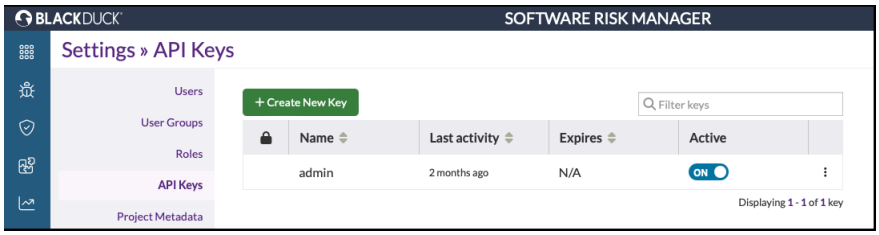
   This page a list of existing API keys, the date of the most recent activity, and whether the key is active.

2. Use the filter field to search for a specific API key or click the column headings to re-sort the list.

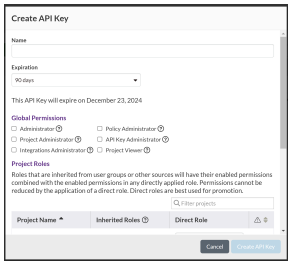
Creating an API Key

To create an API key:

1. Click the Settings icon in the navigation bar and select API Keys from the left menu.



2. Click Create New Key.



3. Enter a name for the new API Key.
4. Select an expiration. (The default is 90 days.)
5. Configure global permissions and project roles as needed.
   Click on a role to select it. Click Clear to remove all selections.

   Global Permissions. Select global permissions for the new user.

   - **Administrator.** Grants user admin privileges. Admin users inherit all roles.
   - **Project Administrator.** Allows the user to create a new project.
   - **Integrations Administrator.** Allows user to manage centralized project configuration.
   - **Policy Administrator.** Allows user to create polices.
   - **API Key Administrator.** Allows user to manage API Keys.

- **Project Viewer.** Allows user to view all projects.

Project Roles. Select permissions for individual projects.

- **Read**. The user or user group can see the specified project and all of its contents. If a user doesn't have the *Read* role for a particular project, that project will not appear in the *Project List* page for that user.
- **Update**. The user or user group can change the finding status and comment on findings for the specified project.
- **Create**. The user or user group can create new analyses for the specified project
- **Manage**. The user or user group can manage the specified project's configuration (e.g., Git, Issue tracker, etc.). The *Manage* role also allows the user to delete the specified project.

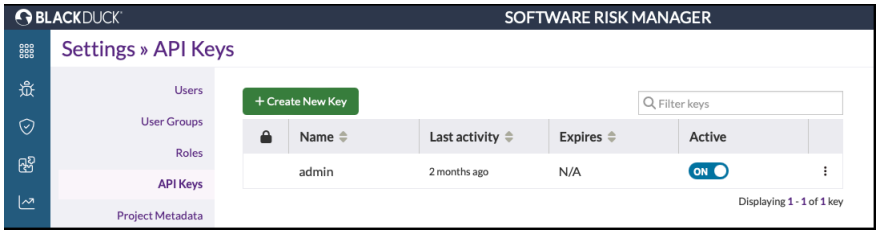
6. Click Create API Key.

Editing an API Key

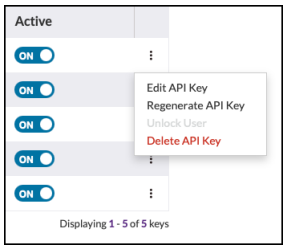
To edit an API key:

Note: The API token expiration cannot be edited; however, regenerating an API key allows you to set a new expiration.

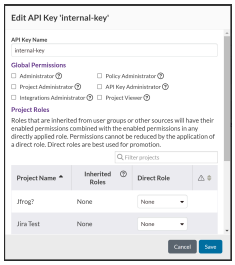
1. Click the Settings icon in the navigation bar and select API Keys from the left menu.



2. Click the dropdown configuration icon and select Edit API Key.



This opens the "Edit API Key" window.



3. Configure global permissions and project roles as needed.
Click on a role to select it. Click Clear to remove all selections.

Global Permissions. Select global permissions for the new user.

- **Administrator.** Grants user admin privileges. Admin users inherit all roles.
- **Project Administrator.** Allows the user to create a new project.
- **Integrations Administrator.** Allows user to manage centralized project configuration.
- **Policy Administrator.** Allows user to create polices.
- **API Key Administrator.** Allows user to manage API Keys.
- **Project Viewer.** Allows user to view all projects.

Project Roles. Select permissions for individual projects.

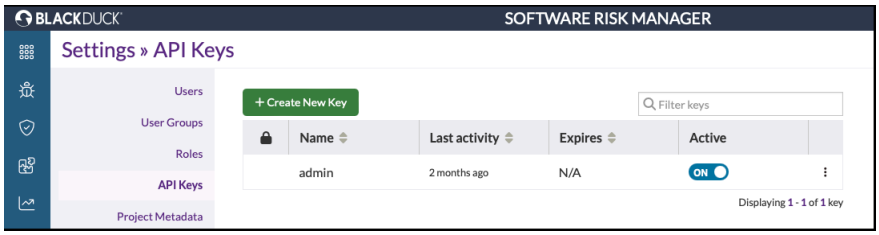
- **Read**. The user or user group can see the specified project and all of its contents. If a user doesn't have the *Read* role for a particular project, that project will not appear in the Projects page for that user.
- **Update**. The user or user group can change the finding status and comment on findings for the specified project.
- **Create**. The user or user group can create new analyses for the specified project
- **Manage**. The user or user group can manage the specified project's configuration (e.g., Git, Issue tracker, etc.). The *Manage* role also allows the user to delete the specified project.

4. Click Save.

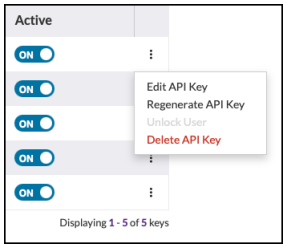
Regenerating an API Key

To regenerate an API key:

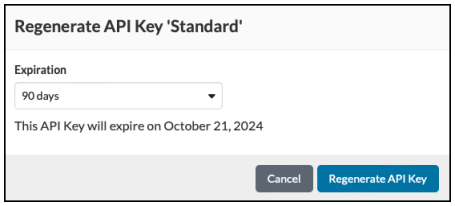
1. Click the Settings icon in the navigation bar and select API Keys from the left menu.



2. Click the dropdown configuration icon and select Regenerate API Key.



This opens the "Regenerate Key" window.



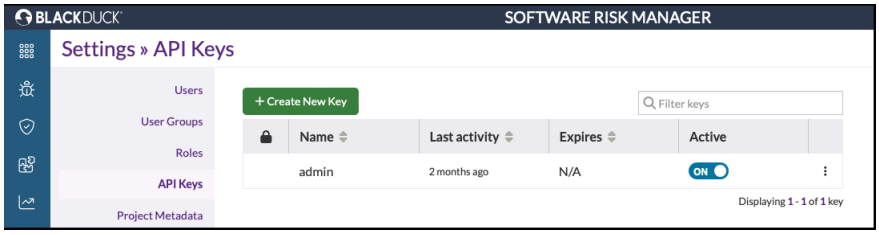
3. Select an expiration period and click Regenerate API Key. (The default is 90 days.)

4. Copy and save the new API key and click Close.

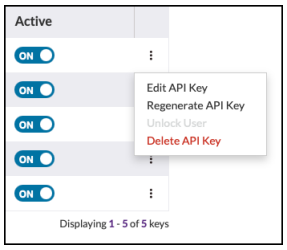
Deleting an API Key

To delete an API key:

1. Click the Settings icon in the navigation bar and select API Keys from the left menu.



2. Click the dropdown configuration icon and select Delete API Key.

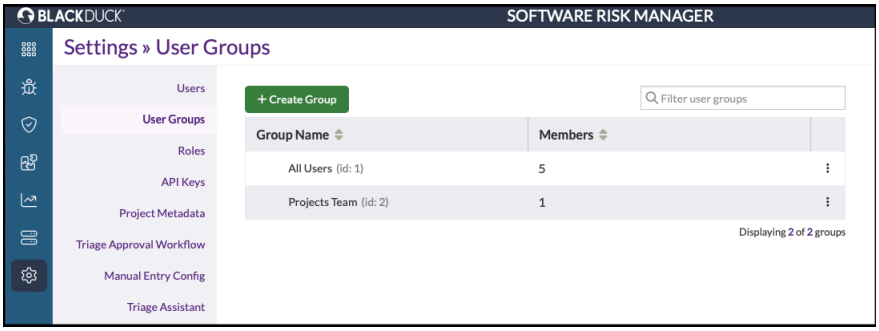


3. Click Delete to confirm.

Managing User Groups

To simplify user administration, users can be managed as groups. Roles and permissions assigned to a group will apply to every member of that group. Admins can set up multiple groups with multiple permissions, including nested groups.

Click the Settings icon in the navigation bar and select User Groups from the left menu to open the User Groups page.



The User Groups page shows a list of existing groups and the number of members in each. Clicking the column headers will re-sort the list.

For more information on managing user groups, see the following topics.

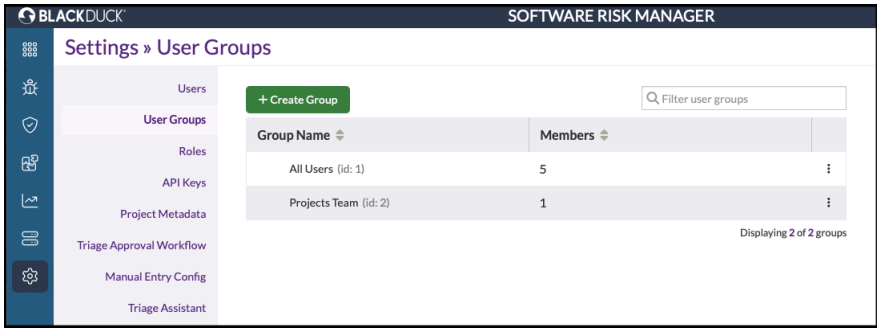
- Viewing existing user groups
- Creating a user group
- Configuring user group roles
- Editing user group settings

- Deleting a user group

Viewing Existing User Groups

To view a list of existing user groups:

1. Click the Settings icon in the navigation bar and select User Groups from the left menu.



   This page shows a list of existing groups and the number of members in each.

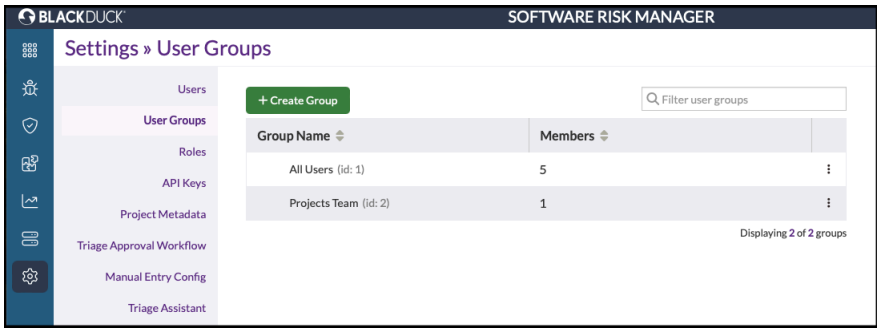
2. Use the filter field to search for a specific user group or click the column header to re-sort the list.

Creating a User Group

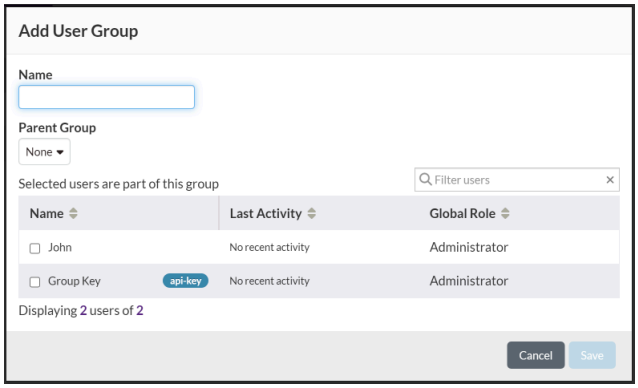
In addition to single groups, groups can be nested ("parent" group).

To create a user group:

1. Click the Settings icon in the navigation bar and select User Groups from the left menu.



2. Click Create Group.



3. Enter a group name and select group members from the list.
4. (Optional) Select a "Parent Group" to create a nested group.

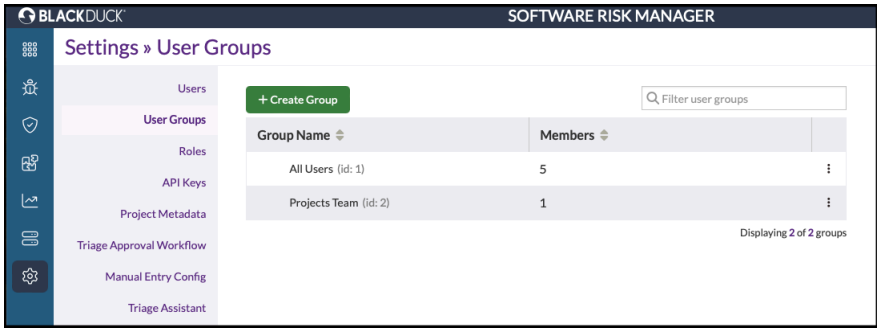
5. Click Save.

Configuring User Group Roles

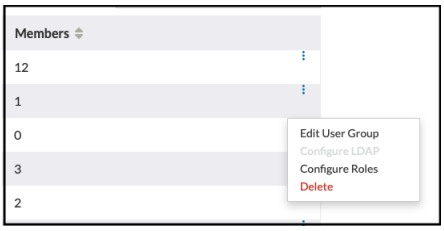
Assigning roles to a group will apply those roles to every member of that group. However, roles assigned to members in the group will not replace or overwrite the roles assigned individually to a user.

To configure group roles:

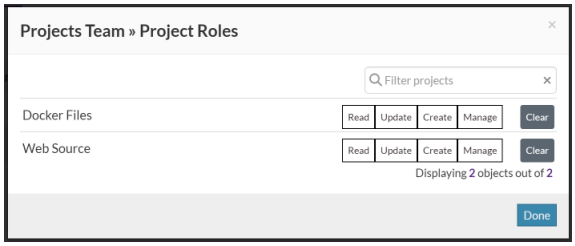
1. Click the Settings icon in the navigation bar and select User Groups from the left menu.



2. Click the dropdown configuration icon and select Configure Roles.



This opens the Configure Roles window.



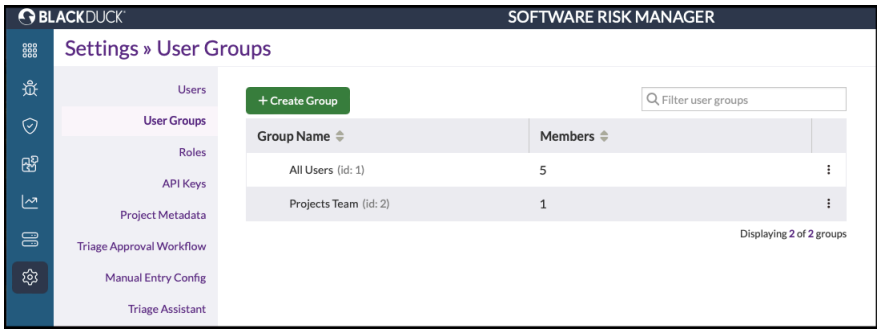
3. Make the necessary configuration changes.
4. Click Done.

Editing User Group Settings

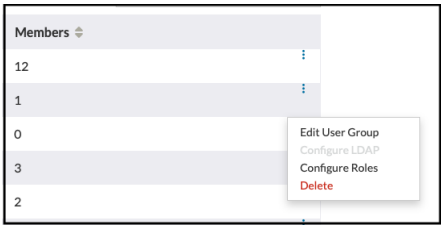
Editing group settings includes changing the group name, reassigning the parent group, and adding or removing users.

To edit group settings:

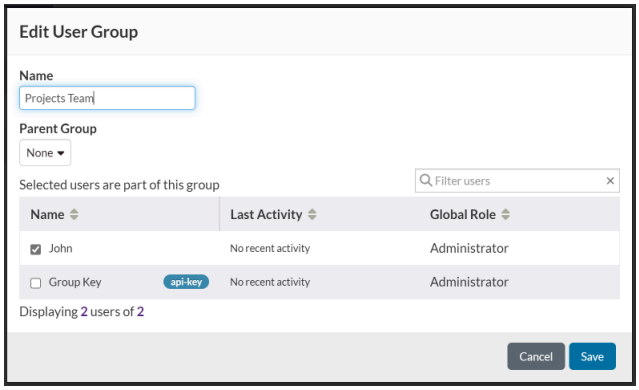
1. Click the Settings icon in the navigation bar and select User Groups from the left menu.



2. Click the dropdown configuration icon and select Edit Group Settings.



This opens the Edit User Group window.



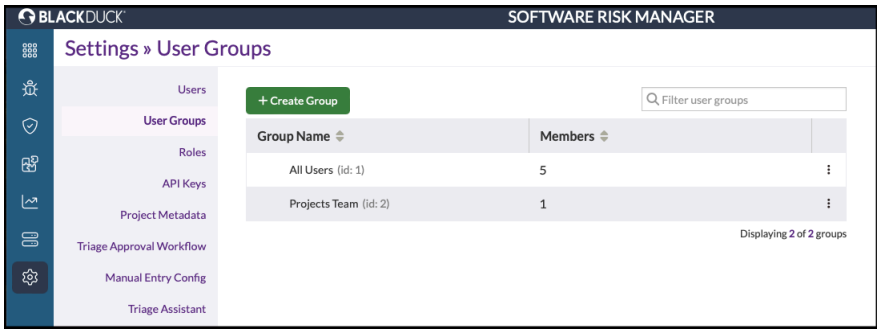
3. Make the necessary changes.
   Use the filter field to search for individual users or click the column headers to re-sort the list.
4. Click Save.

Deleting a User Group

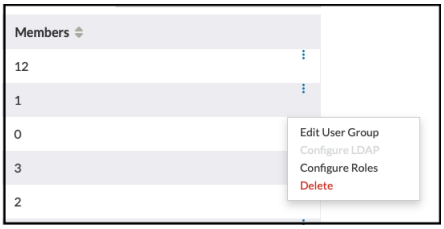
Deleting a user group will not delete individual users or remove any roles previously assigned to a user individually

To delete a user group:

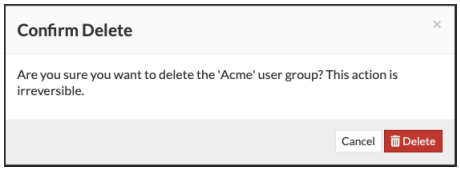
1. Click the Settings icon in the navigation bar and select User Groups from the left menu.



2. Click the dropdown configuration icon and select Delete Group.



This opens the Confirm Delete window.



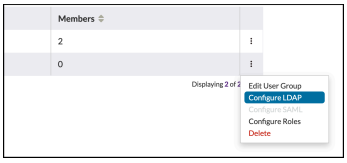
3. Click Delete to confirm.

Configuring LDAP for User Groups

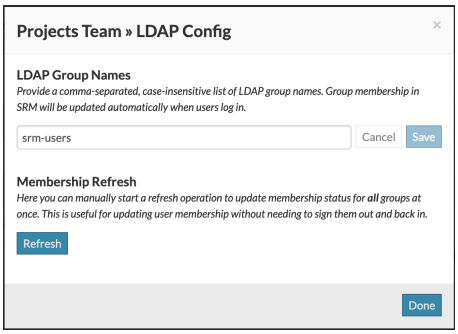
Note: To take advantage of LDAP with User Groups, there must be a valid LDAP configuration in the Software Risk Manager properties file. For more information, see the *SRM Install Guide*.

This feature may be presented when LDAP is configured but is only compatible with providers that create a user attribute containing group membership for SRM to check, such as Active Directory.

To manage mapping of LDAP groups to SRM user groups, pick the LDAP Config option from the group's config menu.



From here, you can enter a comma-separated list of LDAP group names.



The names for your LDAP groups are pulled from the DN path attribute specified in the SRM properties file. For more information, see LDAP group mapping in the *SRM Install Guide.*

When an LDAP user signs in, their LDAP groups will be checked against your mappings configured on this page. The user will be added to the SRM group if they are a member of any of the listed LDAP groups; otherwise, they will be removed.

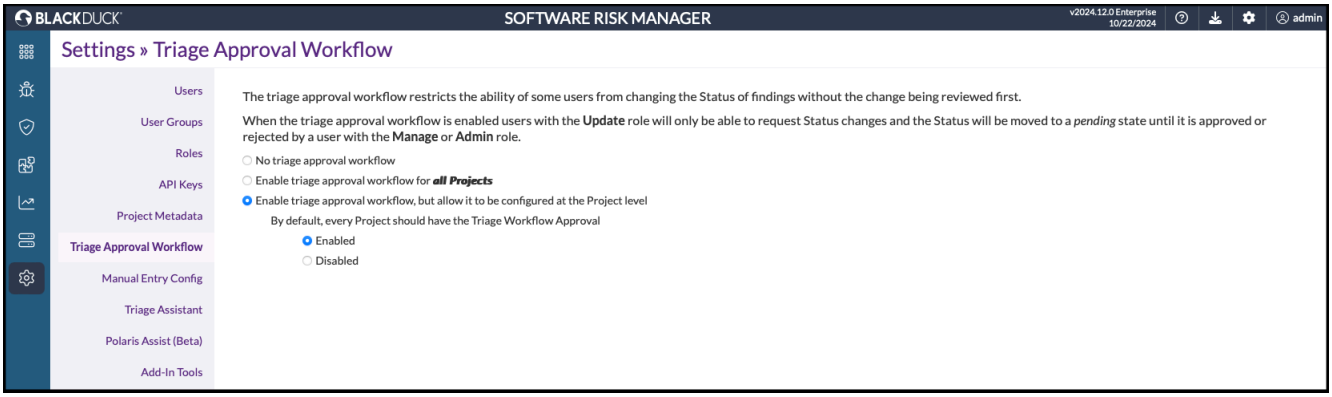
If users are added or removed from an LDAP group, their membership will not be updated until they sign out and back in. You can force a manual refresh of all group's LDAP members by clicking the "Refresh" button on this page.

Note: If the LDAP Group Names textbox is empty, a membership refresh will have no effect on that group.

Triage Approval Workflow

Triage settings associated with findings can be changed by users with an `admin` or `manage` role. Users who are unable to change the triage status directly can submit a change for approval. Admins can configure the triage workflow from the Settings menu and define what users are allowed to do by configuring user roles. Requests for changing triage status can be approved by admins.

Click the Settings icon in the navigation bar and select Triage Approval Workflow from the left menu to open the configuration page.



Managing the Triage Approval Workflow consists of the following tasks:

- Configuring the triage workflow
- Configuring the workflow for all projects
- Configuring the workflow for specified projects
- Approving requests for status changes

Configuring Triage Workflow

This page provides three options for triage approval configuration:

- No triage approval workflow. Select this option to restrict the ability to change status to admins and users with "update" or "manage" roles.
- Enabled triage approval workflow for all projects. Select this option to enable the approval workflow for all current and future projects.
- Enable triage approval workflow, but allow it to be configured at the Project level. Select this option to enable the workflow option on a project level.

Use the radio buttons to select an option. The selection will be saved automatically.

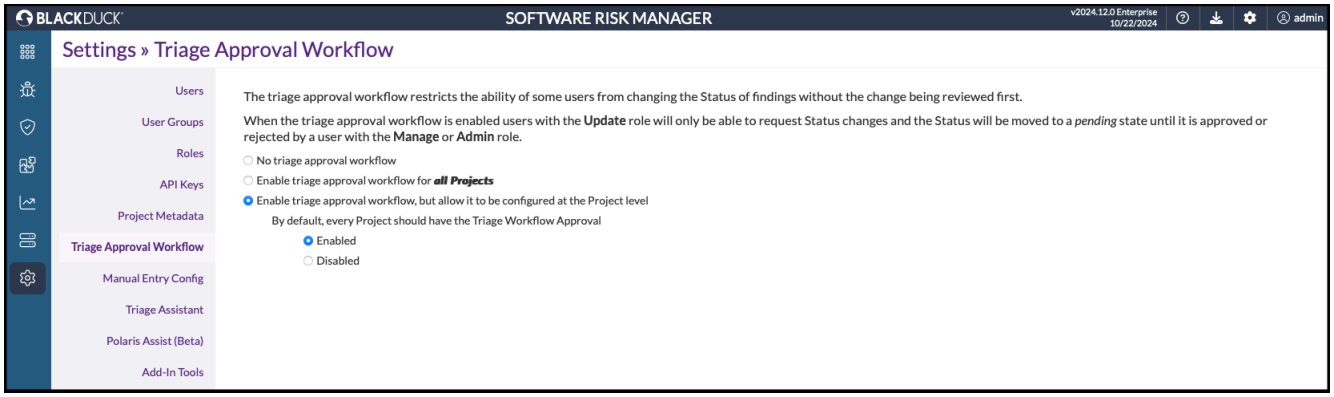
For more information on changing the status of a finding, see the Change Status section.

Configuring Triage Workflow for All Projects

Admins can configure the triage approval workflow based on a specified project.

To configure the triage workflow for a project:

1. Click the Settings icon in the navigation bar and select Triage Approval Workflow from the left menu to open the configuration page.



2. Select the "Enable triage approval workflow for all projects" to include all projects.

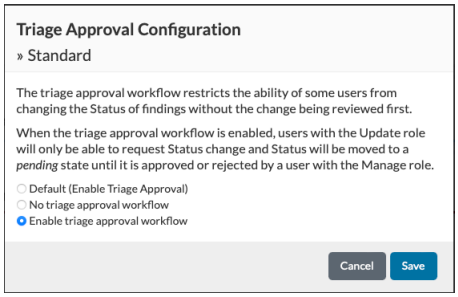
Configuring Triage Workflow for Specified Projects

Admins can configure the triage approval workflow based on a specified project.

To configure the triage workflow for a project:

1. Click the Settings icon in the navigation bar and select Triage Approval Workflow from the left menu to open the configuration page.
2. Select the "Enable triage approval workflow for all projects" to include all projects, or select "Enable triage approval workflow, but allow it to be configured at the Project level" for individual projects.
3. Open the Projects page, click the project's dropdown configuration icon, and select Triage Approval Config.

4. Select one of the following options:
   - Default (See Configuring Triage Workflow for All Projects.)
   - No triage approval workflow
   - Enable triage approval workflow

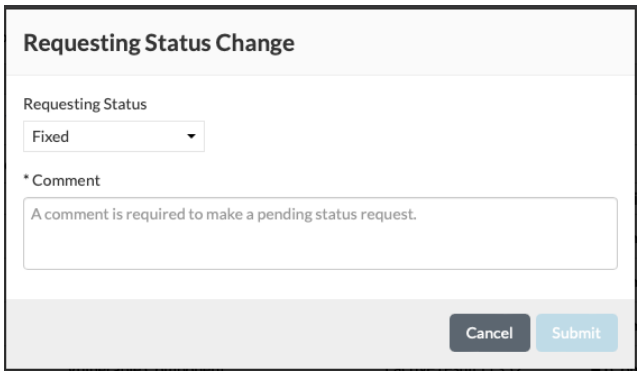


5. Click Save.

Creating a Status Change Request

To create a triage status change request:

1. Open the Findings page.
2. Locate the finding whose status you want to change and click the Status dropdown menu to open a list of Status options.
3. Select a new triage status.

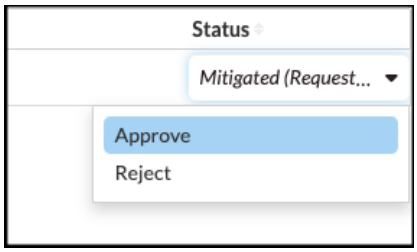


4. Enter a comment and click Submit.

Approving a Status Change Request

To approve a triage status change request:

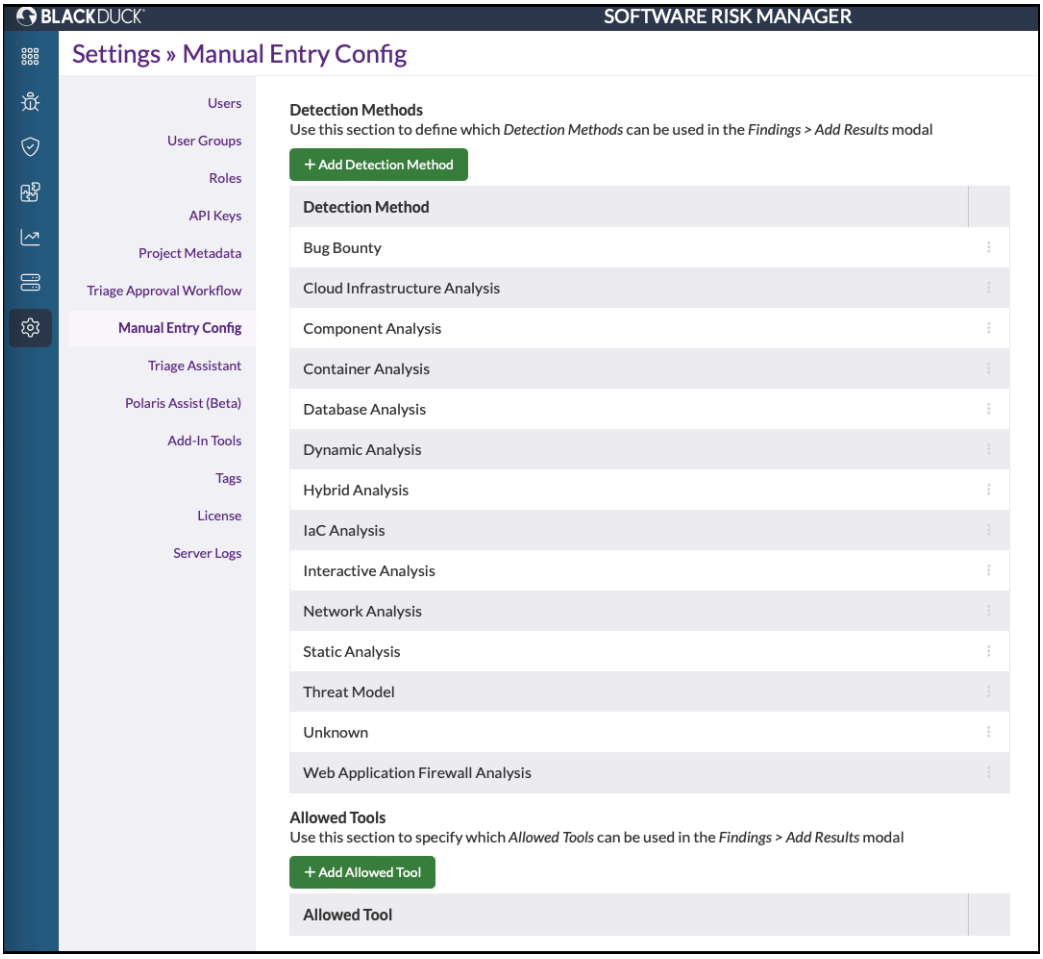
1. Open the Findings page.
2. Use the Pending Status filter to list status change requests.
3. Click the requested status for the finding, then click Approve.



Manual Entry Configuration

The Manual Entry Configuration page allows Software Risk Manager administrators to define custom values which can be entered into certain fields in the Manual Results.

Click the Settings icon in the navigation bar and select Manual Entry Configuration from the left menu to open the Manual Entry Configuration page.



This page displays a list of existing detection methods and allowed tools. From here, you can add, delete, or rename detection methods, and add or remove allowed tools.

For more information on detection methods and allowed tools, see the following topics:

- Managing Detection Methods
- Managing Allowed Tools

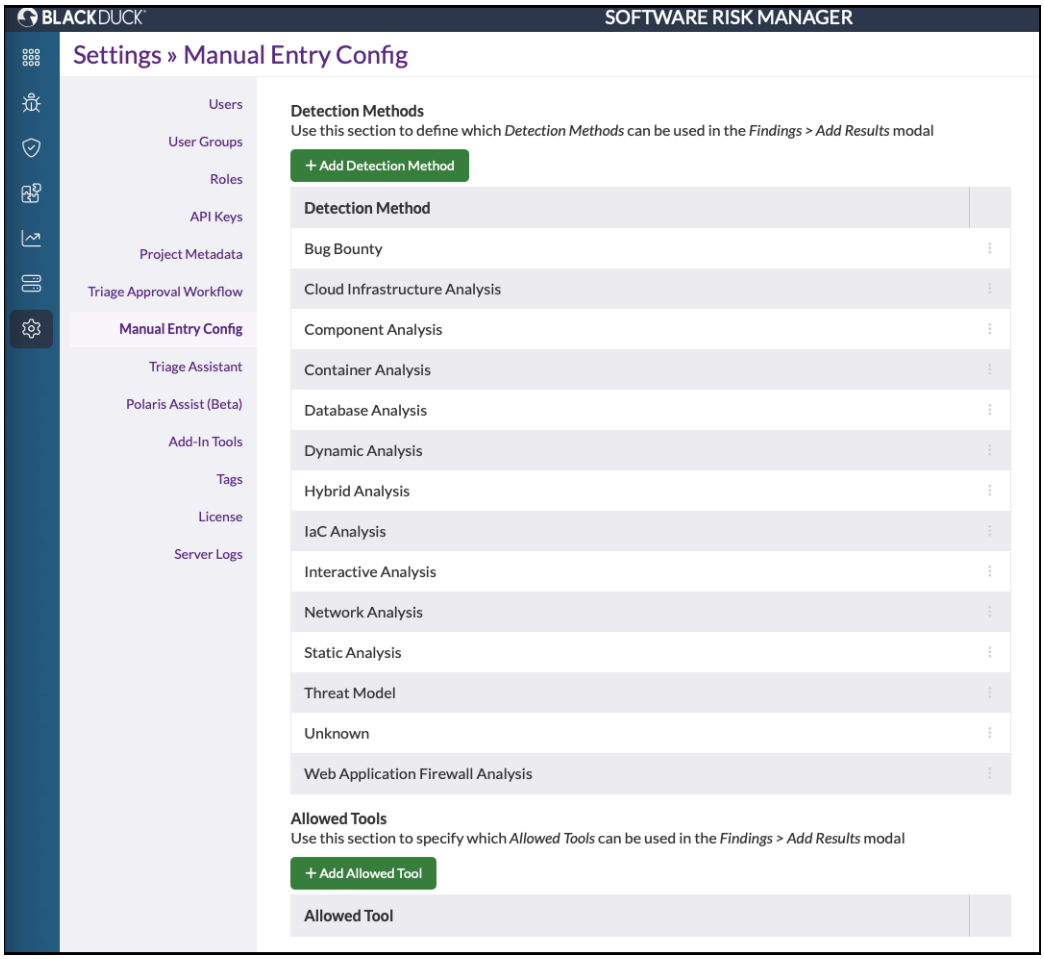
Managing Detection Methods

Software Risk Manager provides built-in detection methods, which explains how a finding was discovered. These built-in detection methods reflect the types of tools currently supported by Software Risk Manager. For manual entry, you may wish to specify your own custom detection methods.

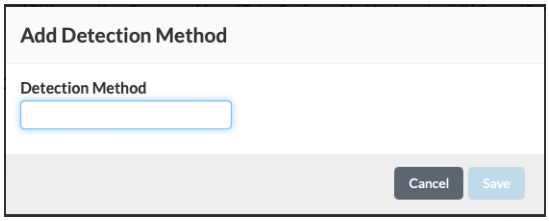
Adding a Detection Method

To add a new detection method:

1. Click the Settings icon from the navigation bar and select Manual Entry Config from the left menu.



2. Click Add Detection Method.



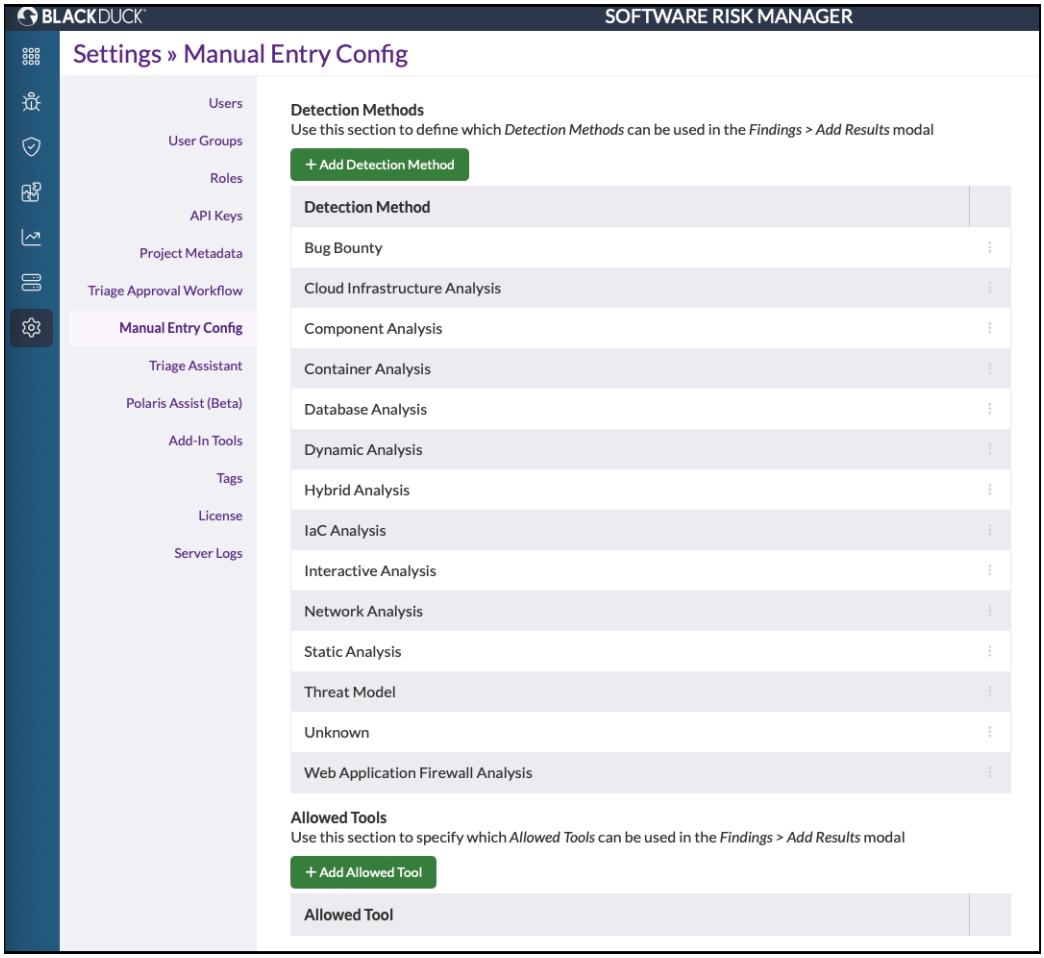
3. Enter a name for the detection method.
4. Click Save.

Renaming a Detection Method

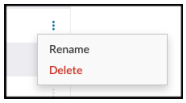
You can rename your custom detection methods as needed. However, the built-in detection methods cannot be edited in any way (indicated by the lock icon next to their edit/delete buttons).

To rename a detection method:

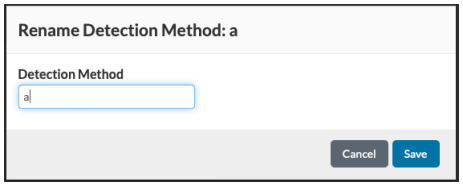
1. Click the Settings icon from the navigation bar and select Manual Entry Config from the left menu.



2. Click the dropdown configuration icon to the right of the detection method and select Rename.



This opens the Rename Detection Method window.



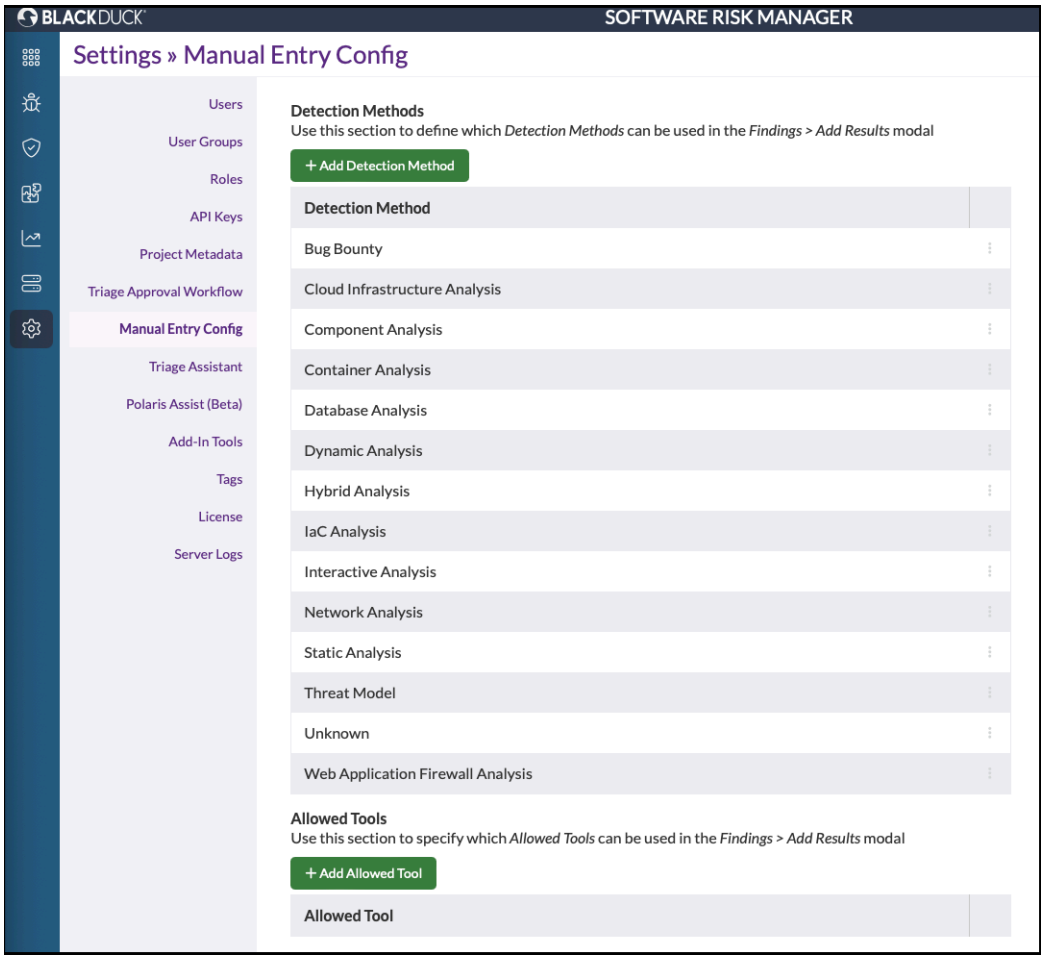
3. Enter a different name for the detection method.
4. Click Save.

Deleting a Detection Method

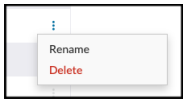
You can delete any unused custom detection methods (i.e., as long as no manually entered results use it as their own detection method). If your custom detection method is in use when you try to delete it, you will have to choose a replacement. All results using that detection method will be edited to use the replacement detection method instead. This will likely trigger the recorrelation prompt, since detection method is one of the correlation criteria.

To delete a detection method:

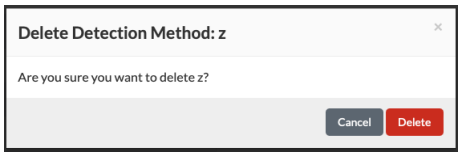
1. Click the Settings icon from the navigation bar and select Manual Entry Config from the left menu.



2. Click the dropdown configuration icon and select Delete.



This opens the Delete Detection Method window.



3. Click Delete to confirm.

Managing Allowed Tools

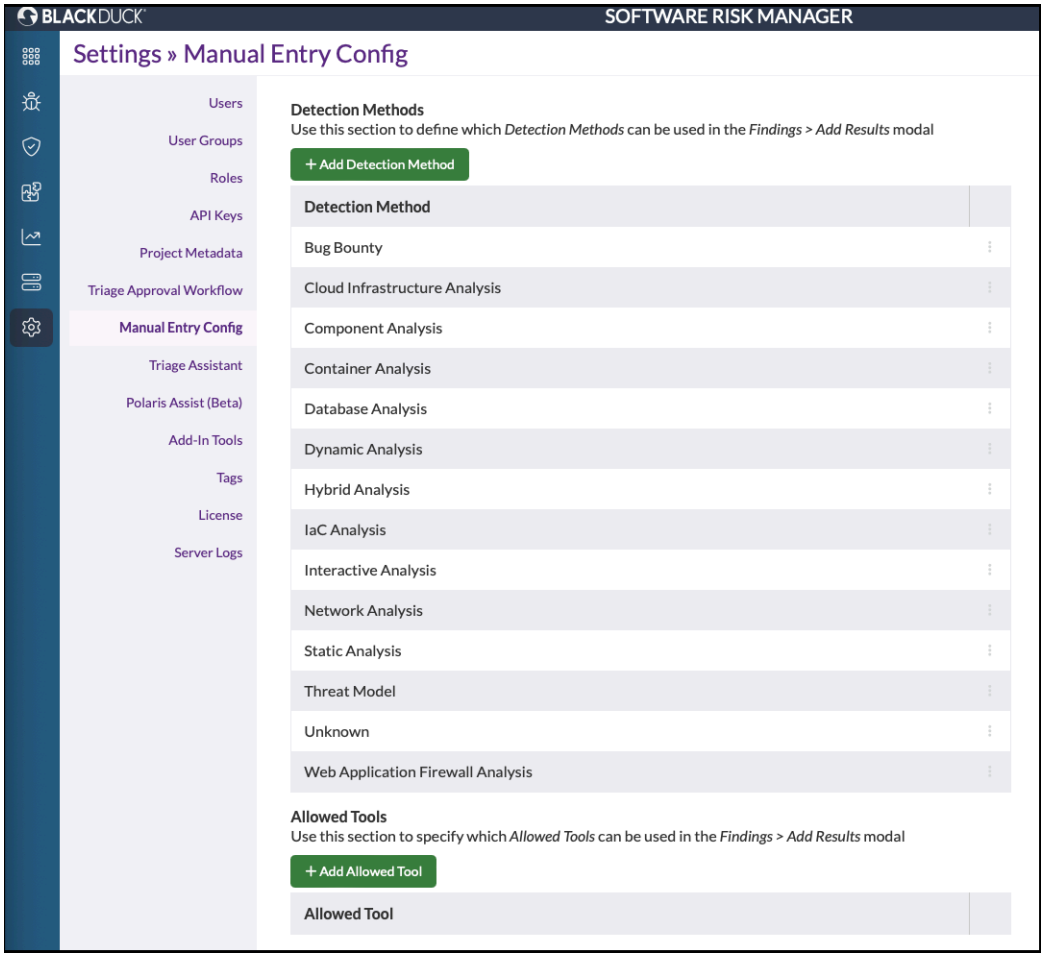
The Allowed Tools section lets you define which tools will appear in the Tool dropdown in the Manual Result form. The names you add to this list do not necessarily need to correspond to a tool known to Software Risk Manager, or even a real tool, for that matter: you can enter any tool name you want.

Note: Unlike the Detection Methods delete, you don't need to pick a replacement to delete an item in this list; this list only controls which tools are available when creating a new manual result, not which tools exist.

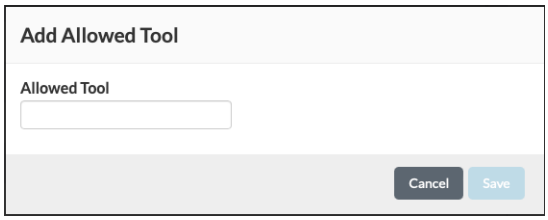
Adding an Allowed Tool

To add an allowed tool:

1. Click the Settings icon from the navigation bar and select Manual Entry Config from the left menu.



2. Click the Add Allowed Tool button.

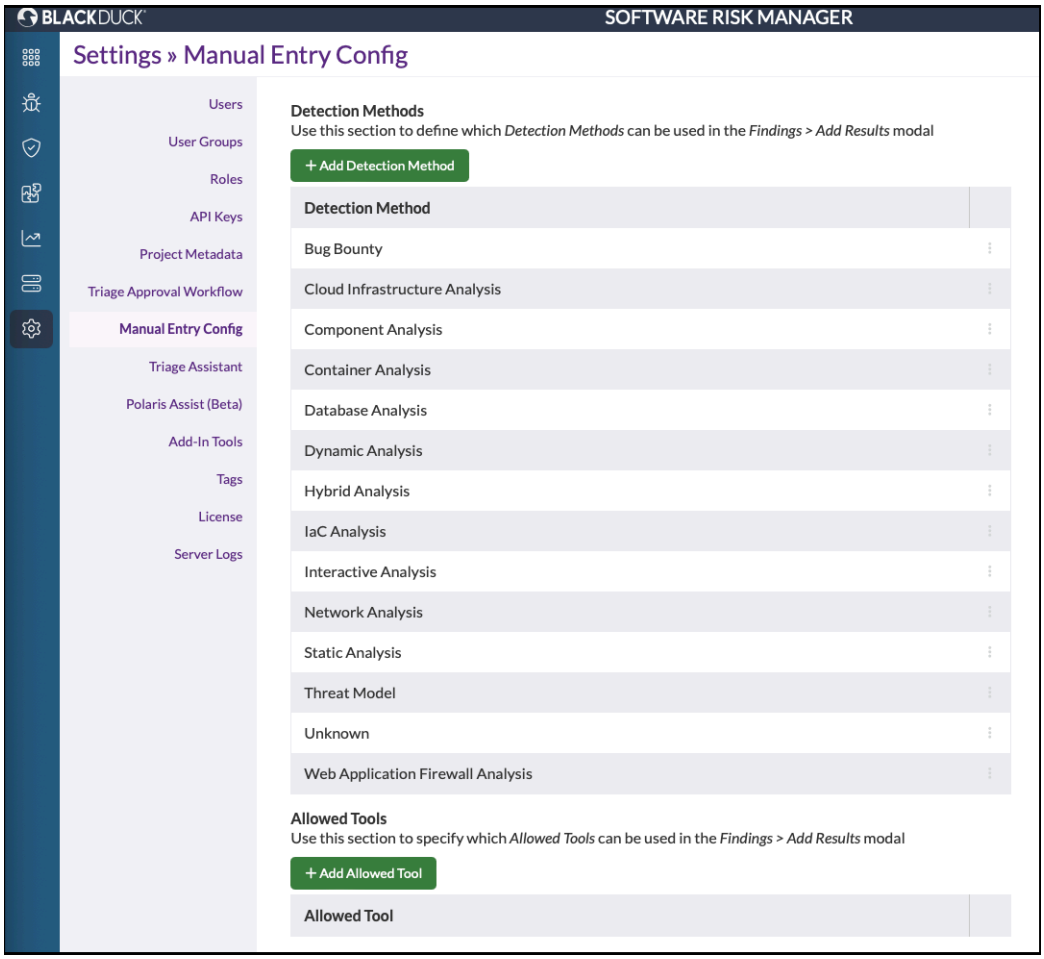


3. Enter the name of the tool.
4. Click Save.

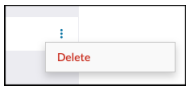
Deleting an Allowed Tool

To delete an allowed tool:

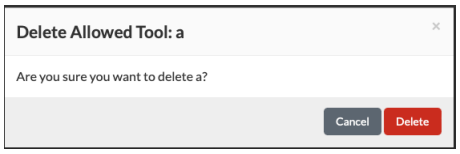
1. Click the Settings icon from the navigation bar and select Manual Entry Config from the left menu.



2. Click the dropdown configuration icon to the right of the allowed tool and select Delete.



This opens the Delete Allowed Tool window.

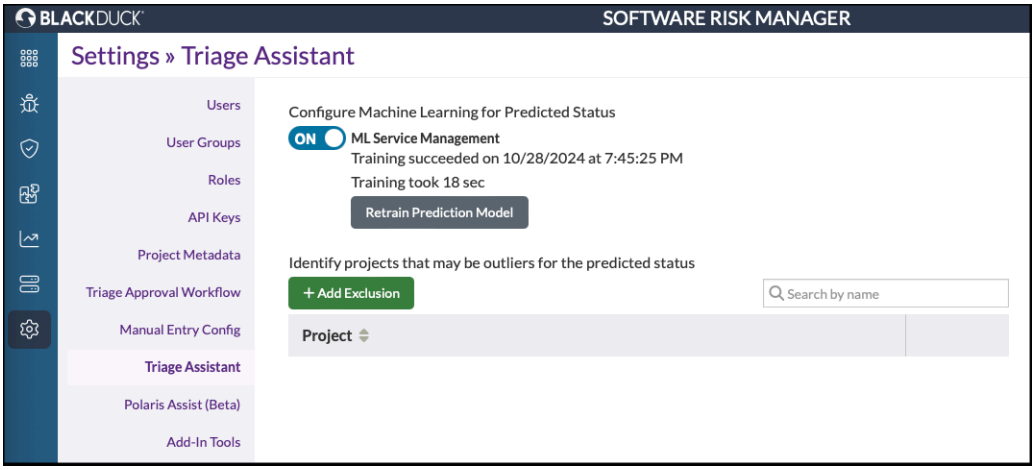


3. Click Delete to confirm.

Triage Assistant

If machine learning is enabled, Software Risk Manager is capable of automatically (re)training a prediction model so that it does not have to be done manually. The ML Service Management section will automatically transition from an *Idle* state to a *Working* state when Software Risk Manager automatically initiates a training session. Whether automatic updates of the prediction model occur at all, and the time at which they occur, are configurable through the SRM props file (see the Install Guide for more information.)

Click the Settings icon in the navigation bar and select Machine Learning (ML) Control Panel from the top menu to open the Machine Learning Control Panel page.



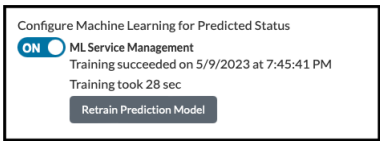
There are two parts to the Machine Learning Control Panel:

- **Configure Machine Learning for Predicted Status.** This part involves switching on the service management and retraining the prediction model.
- **Identify projects that may be outliers for the predicted status.** This part involves adding exclusions.

Configuring Machine Learning for Predicted Status

Machine learning configuration allows admins to manually trigger the training of the prediction model. To make use of machine learning, Software Risk Manager requires that at least 100 findings have been actively triaged. These triaged finding can come from multiple projects. If you have not met this requirement, then you will be presented with a statement detailing how many findings you have actively triaged and a statement detailing the minimum requirements.

The ML Service Management section of the control panel transitions back and forth between two states. The first is an *Idle* state, which means that Software Risk Manager is not currently training a prediction model. The second is a *Working* state, which means that Software Risk Manager is currently training a prediction model. If the section is in an *Idle* state, then either a Build Prediction Model button or an Update Prediction Model button will be present. Specifically, the Build Prediction Model button will be present if a prediction model has not been trained, and the Update Prediction Model button will be present if a prediction model has been trained.



When ML Service Management is off, Software Risk Manager will begin training a prediction model when the Retrain Prediction Model button is clicked. This will transition the section into a *On* state.

Once training has completed, the section will transition back to an *Idle* state and will present the user with when the last training session completed, how long it took, and whether or not it succeeded.

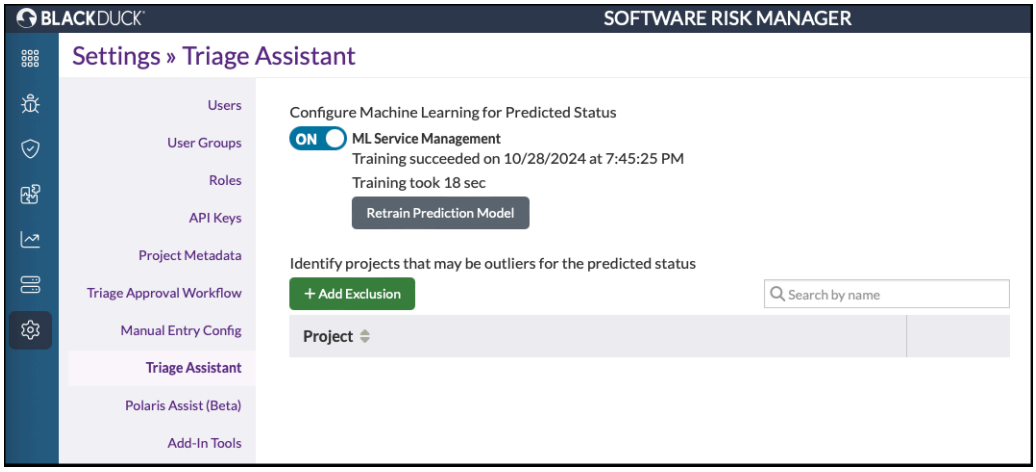
Managing Exclusions

The Excluded Projects section allows admins to configure which projects should not be considered when Software Risk Manager trains a prediction model. All excluded projects are listed.

Viewing Existing Exclusions

To view an existing exclusion:

1. Click the Settings icon in the navigation bar and select Machine Learning (ML) Control Panel from the top menu.



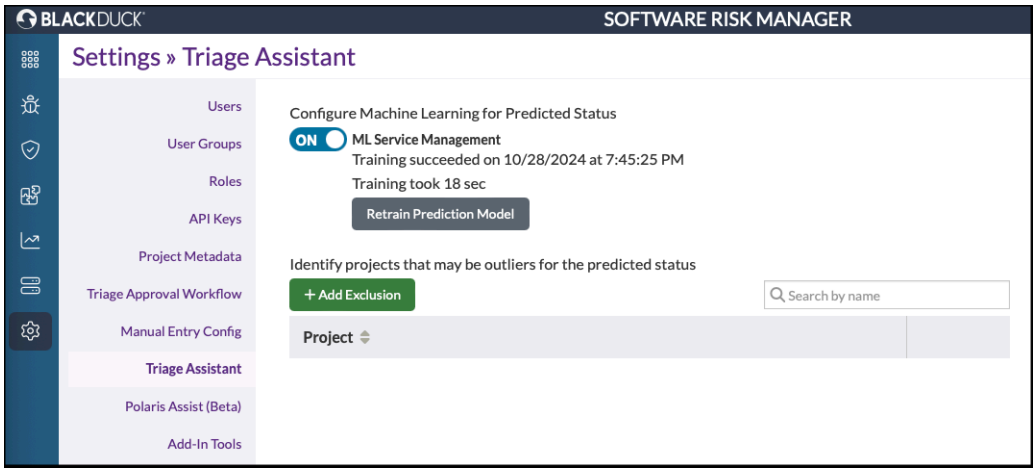
2. Use the search field to search by name or click the column header to re-sort the list.

Adding an Exclusion

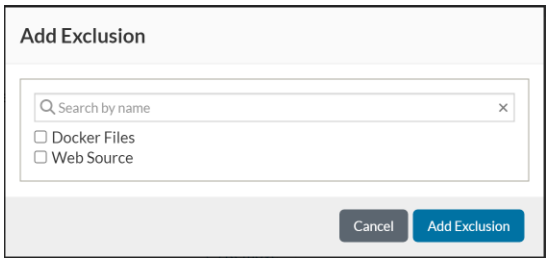
Adding exclusions will exclude the selected project from prediction models that Software Risk Manager trains.

To add an exclusion:

1. Click the Settings icon in the navigation bar and select Machine Learning (ML) Control Panel from the top menu.



2. Click Add Exclusion.



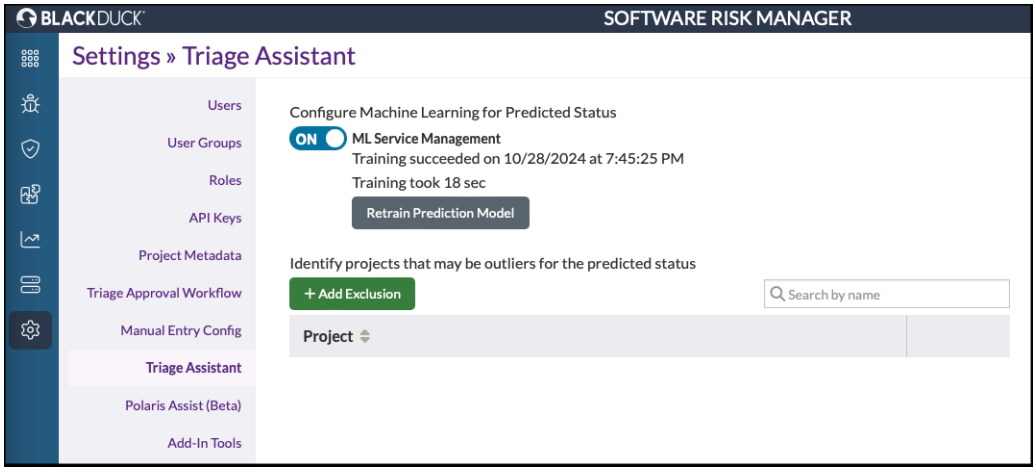
3. Select from the list of exclusions or use the search field to search by name.
4. Click Add Exclusion.

Removing an Exclusion

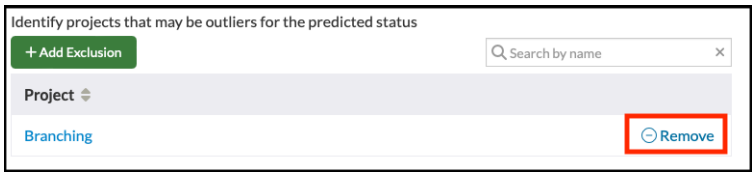
You can also choose to re-include an excluded project so that it may be considered during training by removing it.

To remove an exclusion:

1. Click the Settings icon in the navigation bar and select Machine Learning (ML) Control Panel from the top menu.



2. Locate the exclusion you want to delete.
   You can use search field to search by name or click the column header to re-sort the list.



3. Click Remove.

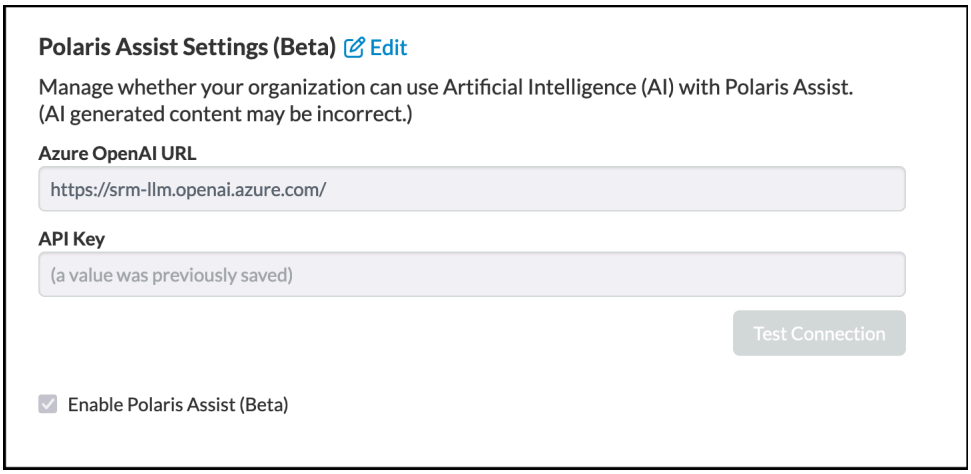
Polaris Assist (Beta)

Use the "Polaris Assist (Beta)" tab on the Settings page to configure and enable Polaris Assist. SRM uses Azure OpenAI APIs, namely the Chat Completion API, and the Azure OpenAI URL must refer to a service which exposes this endpoint.

Warning: Polaris Assist generates results created by artificial intelligence (AI) or other automated technologies. Such results are provided for informational purposes only and should not be relied upon for any specific purpose without verification of its accuracy or completeness.

SRM will use the model `gpt-4o` in its requests, which requires that a model with that name be available in the provided API.

Note: The model `gpt-4o` is the default. To change this model, see Polaris Assist in the *Install Guide*.



Enter your Azure OpenAI URL and API Key, then click Test Connection. Use the checkbox to enable Polaris Assist.

The "Test Connection" button will perform a test request using the configured `model-id`.

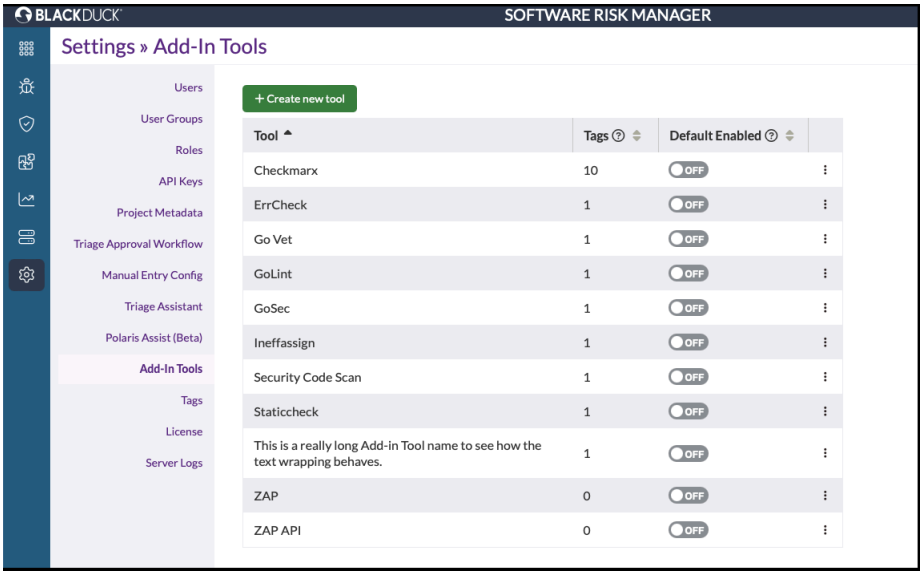
For more information on configuring Polaris Assist and AI Insight, please refer to the Install Guide.

Add-In Tools

An add-in tool is based on a scan request file that you define and register with Software Risk Manager. A scan request file contains the instructions that the tool service needs to invoke an application security testing tool on a Kubernetes cluster and ingest its output into Software Risk Manager.

The Add-In Tools section appears when the Tool Orchestration Service is enabled. (See Tool Orchestration in the *Software Risk Manager Install Guide* for instructions on how to enable this feature.)

Click the Settings icon in the navigation bar and select Add-In Tools from the top menu to open the Add-In tools page.



The Add-In Tools page allows you to manage the list of application security testing tools that can run on your cluster.

Add-In tools must be enabled on a per-project basis, and a registered tool starts in a disabled state. See the Customize Add-In Tools section to learn how to enable a tool for a specific project. You can also use the *Default enabled* toggle to enable a tool for every project, excluding those where it was explicitly disabled. Avoid enabling tools by default when they include project-based settings.

Some add-in tools, such as DAST tools, do not require an analysis input. Software Risk Manager will offer to run them with each new analysis. Others require an input file, and Software Risk Manager will scan a file to build a list of tags describing its contents. Tool registration data lets Software Risk Manager select appropriate add-in tools to run.

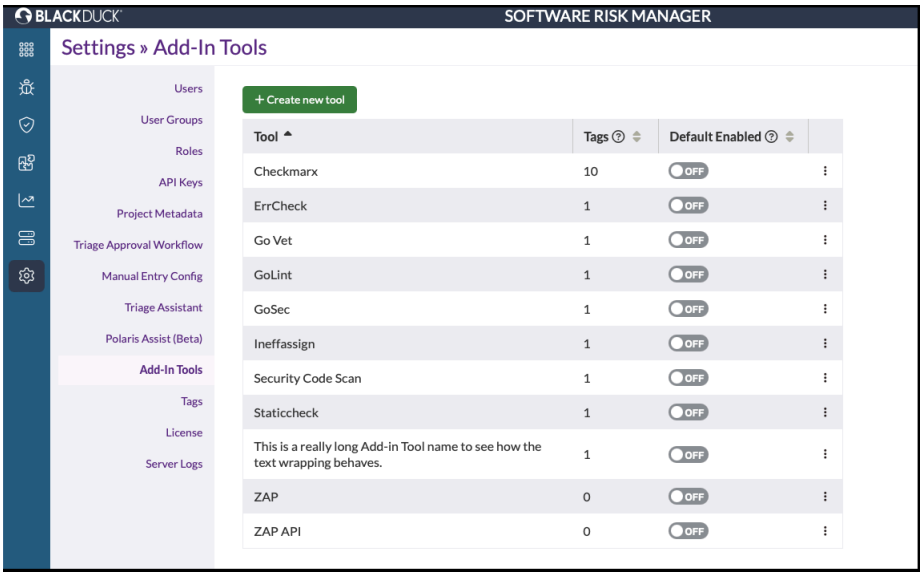
The Matched Tags section lets you associate content tags with an add-in tool. Select the *Tag type* and specify the associated criteria for the content tag. For *Language*, *Runtime*, and *Meta*, select from the options in the dropdown menu. For *Extensions*, specify any number of extensions to associate with the add-in tool as either a comma or space-delimited list (e.g., `zip, msi, pkg` or just `zip`). Click Add Tag to link a tool with a content type.

The *Language* and *Runtime* tags are detected based on the presence of files with the appropriate extension for the language or runtime. The *Meta* tags are based on the presence of other files:

- OpenSSL. An `openssl.v.h` file
- NuGet Manifest. Any `.nuspec` file
- npm Package. A `package.json` file
- .NET Core, Framework, Standard. Any `.csproj` or `.vbproj` file (contents are inspected to determine framework type)

Viewing Existing Add-In Tools

To view existing add-in tools, click the Settings icon in the navigation bar and select Add-In Tools from the left menu.



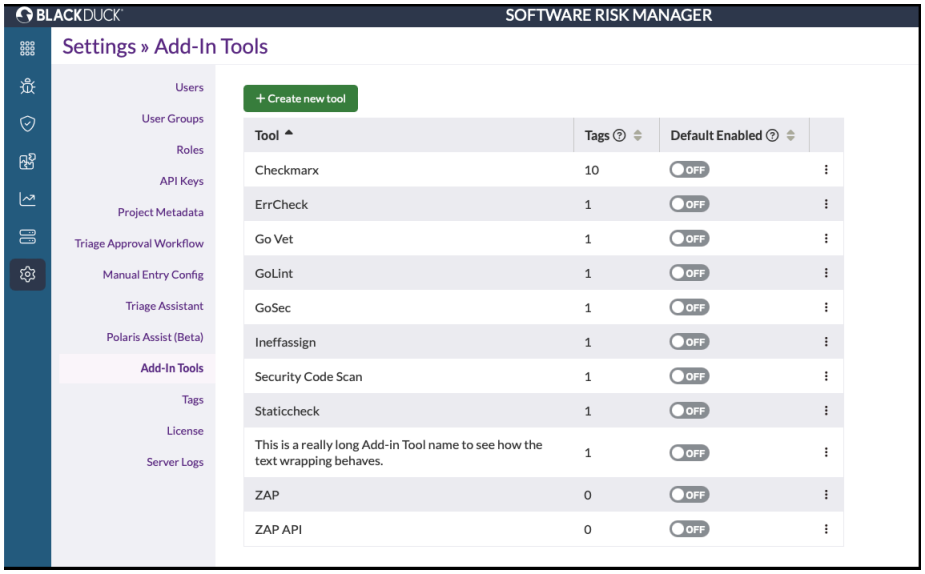
This list shows all the existing add-in tools along with information about how many tags have been assigned and whether the tool has been enabled.

Creating a New Add-In Tool

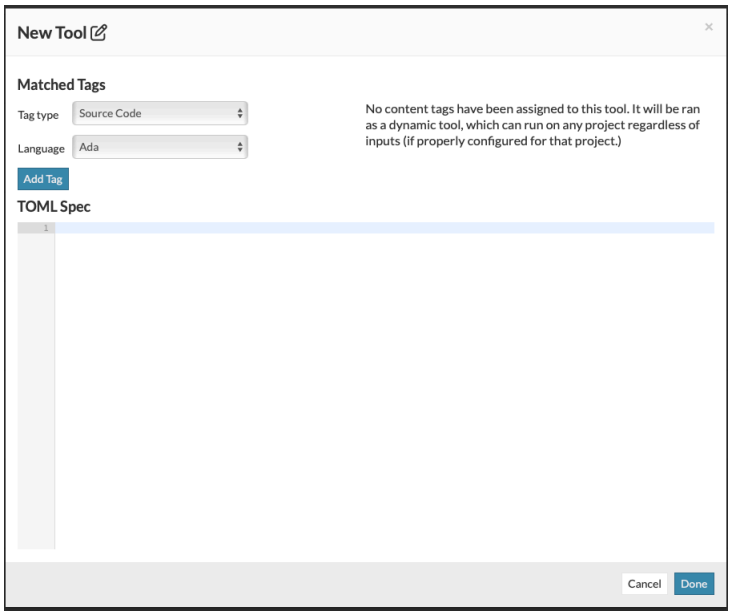
The Create New Tool feature allows you to add a tool registration.

To create an new add-in tool:

1. Click the Settings icon in the navigation bar and select Add-In Tools from the left menu.



2. Click Create New Tool.

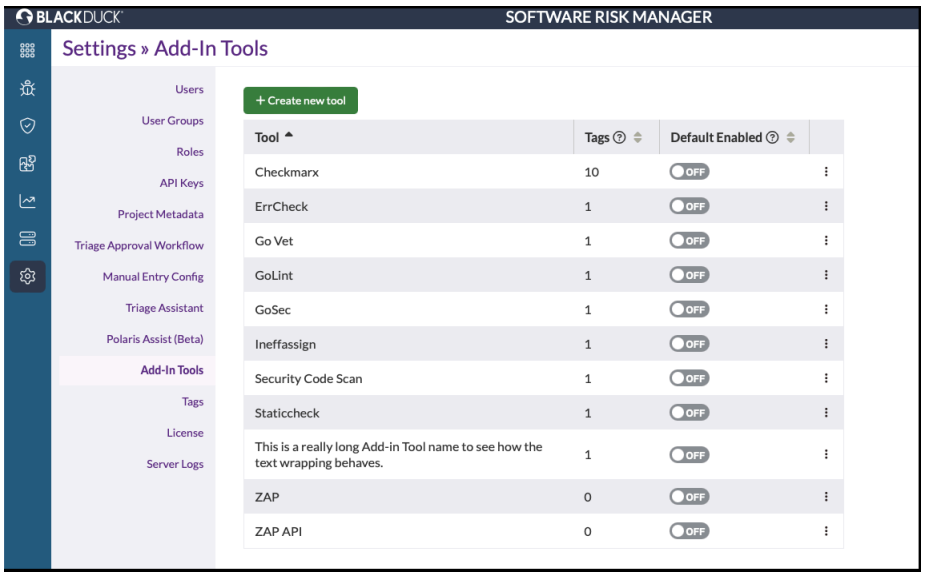


3. Select a tag type and language from the dropdown list.
4. Add a tag.
5. Enter a TOML Spec in the blank field.
   The *TOML Spec* includes the scan request file content that defines an add-in tool. (See the Scan Request File section to learn more about scan request files.)
6. Click Done.

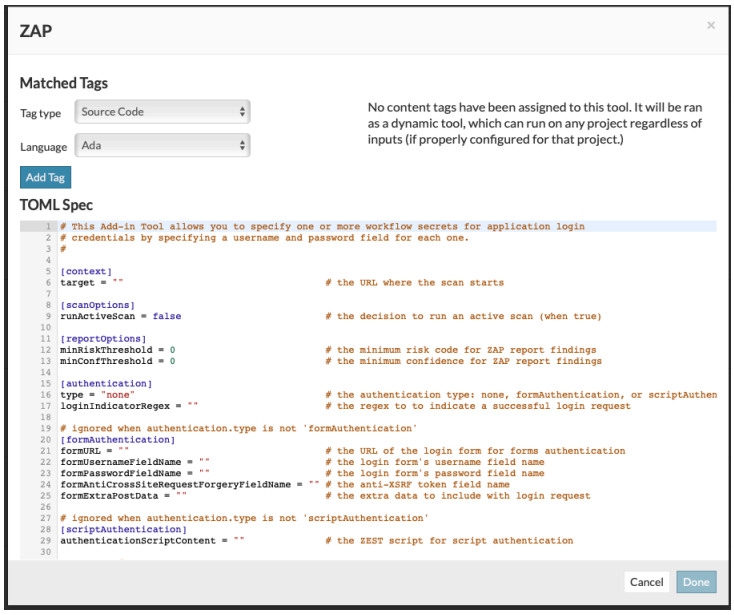
Configuring an Add-In Tool

To configure an add-in tool:

1. Click the Settings icon in the navigation bar and select Add-In Tools from the left menu.



2. Click the tool's dropdown configuration icon.



3. Make changes as needed.
4. Click Done.

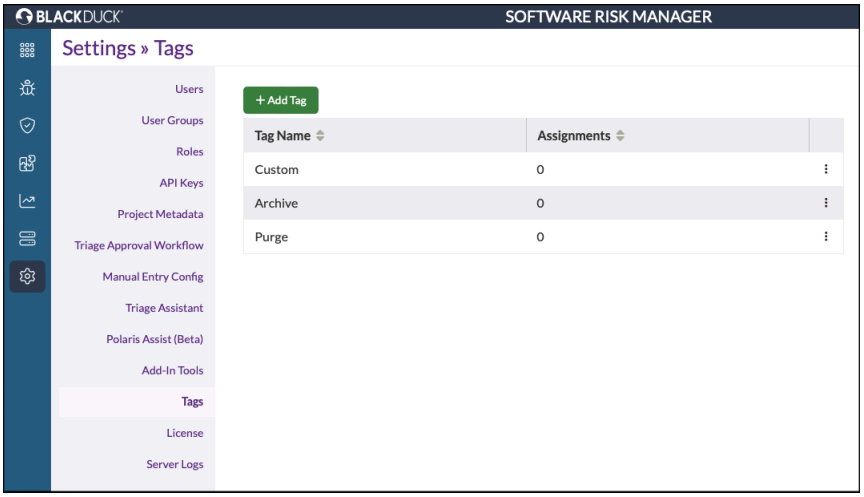
Renaming an Add-In Tool

You can change the tool's name by editing the window title and clicking OK, but you must click Done to save a tool name change. Tool names must be unique, and bundled add-in tools cannot be renamed.

Assigning Tags to Findings

Tags are part of a project's metadata, which can be configured by users with a "manage" role. (For more information, see Configuring Project Metadata.)

Click the Settings icon in the navigation bar and select Tags from the left menu to open the Tags page.



This page lists all tags (alongside the number of findings to which each tag has been assigned) that can be assigned to findings.

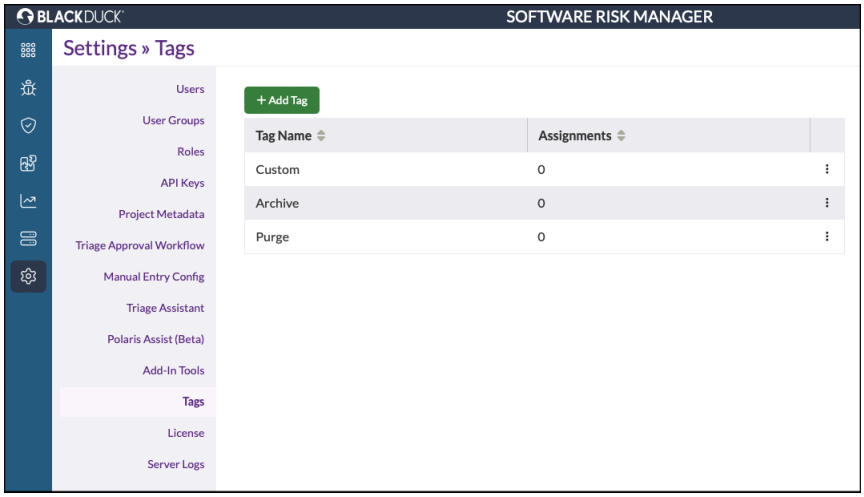
For more information, see the following topics:

Viewing Existing Tags

The Tags page shows a list of existing tags and the number of associated assignments.

To view a list of existing tags:

1. Click the Settings icon in the navigation bar and select Tags from the left menu.

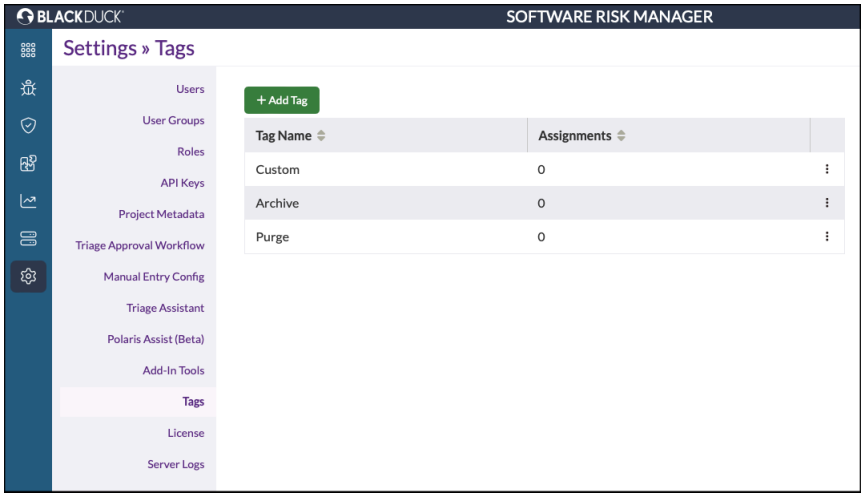


2. Click the column headers to re-sort the list.

Adding a Tag

To add a tag:

1. Click the Settings icon in the navigation bar and select Tags from the left menu.



2. Click Add Tag.

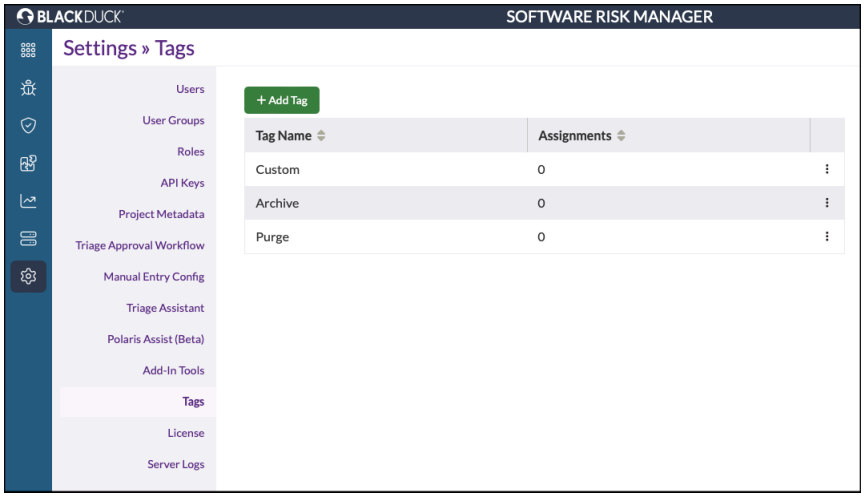
3. Enter a name for the tag.
4. Click Save.

Renaming a Tag

Note: If you attempt to rename a tag to a name that already belongs to a tag that has been assigned to at least one finding, then you will be prompted with a dialog asking you to confirm the operation. If you confirm the operation, then the tag will be renamed, and the finding assignments between the involved tags will be merged. Note that the number of assignments is not necessarily equal to the sum of the finding assignments between tags once they have been merged, as it is possible for tags to have overlapping finding assignments. If you do not confirm the operation, then the operation will be cancelled.

To rename a tag:

1. Click the Settings icon in the navigation bar and select Tags from the left menu.



2. Click the dropdown configuration icon to the right of the tag name and select Rename.

This opens the Rename Tag window.

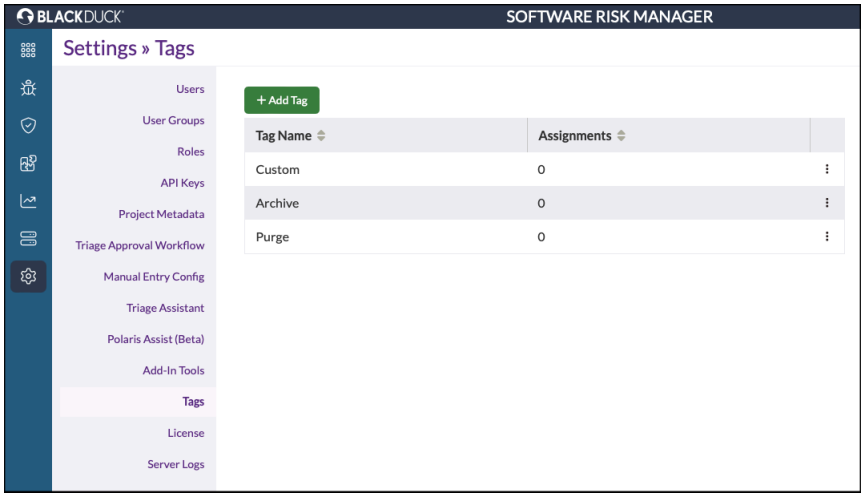
3. Enter a new name for the tag.
4. Click Save.

Deleting a Tag

Note: If you attempt to delete a tag which has been assigned to at least one finding, then you will be prompted with a dialog asking you to confirm the operation. If you confirm the operation, then the tag will be deleted and, as a consequence, the tag will be unassigned from all findings to which the tag had been assigned. If you do not confirm the operation, the operation will be cancelled.

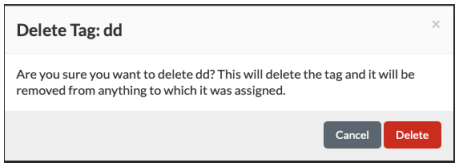
To delete a tag:

1. Click the Settings icon in the navigation bar and select Tags from the left menu.



2. Click the dropdown configuration icon to select Delete.

This opens the Delete Tag window.

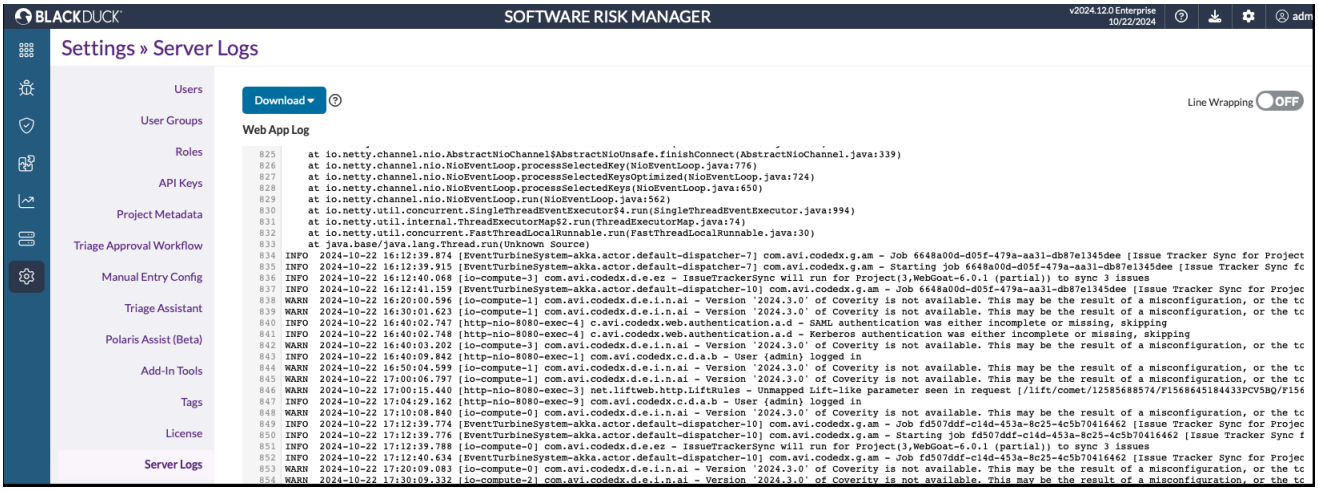


3. Click Delete to confirm.

Server Logs

The Server Logs page displays the contents of the Web App log, which includes details related to analysis progress, user changes, issue tracker updates, and warnings and errors. The Web App log can be useful in troubleshooting issues within Software Risk Manager as well as providing context when creating a support ticket.

Click the Settings icon in the navigation bar and select Server Logs from the top menu to open the Server Logs page.



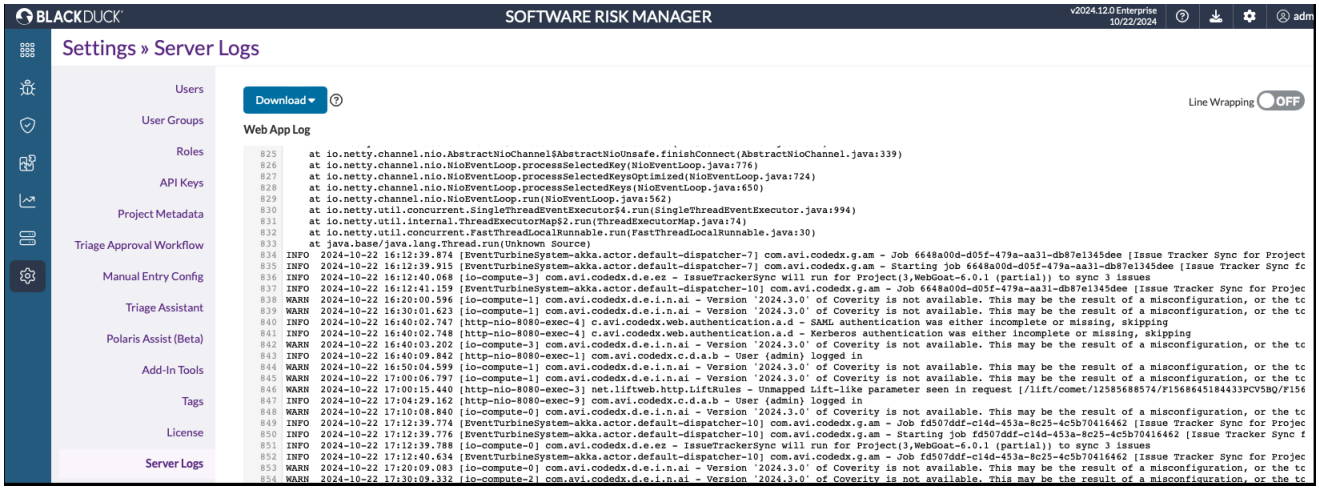
Software Risk Manager maintains additional log files that provide detailed information about runtime operations for various other tools bundled with Software Risk Manager. These files are located in the `log-files` folder in the Software Risk Manager `appdata` directory.

(For more information about the appdata directory, see "Understanding the AppData Directory" in the *Software Risk Manager Install Guide*.)

Viewing and Downloading the Server Log

To view and download the server log file:

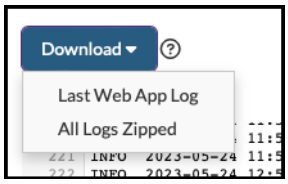
1. Click the settings icon in the navigation bar and select Server Logs from the left menu.



   Use the Line Wrapping toggle for easier viewing.

2. Click the Download button and select one of the following options:

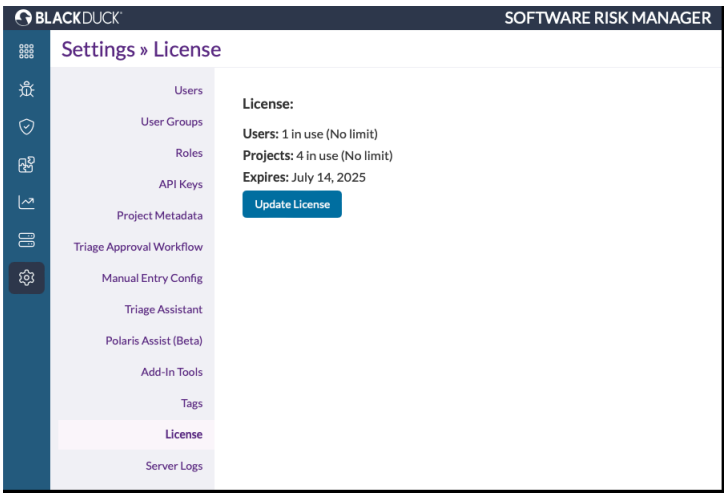
   - **Last WebApp Log.** Downloads the most current web app log. The WebApp Log is the most common server log used for troubleshooting.
   - **All Logs Zipped.** Downloads all of the server logs in one .zip file, including all webapp logs, ML_Triage logs, and logs with supporting data from analyses.



License Information

The License page provides information pertaining to the current SRM license and includes an option for upgrading your license.

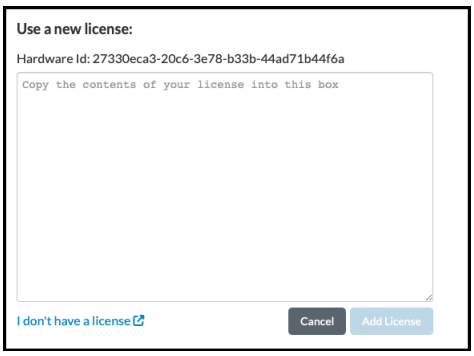
Click the Settings icon in the navigation bar and select License from the left menu to open the License page.



In addition to the SRM license ID, the following information is also displayed:

- Current number of active SRM users.
- Number of projects.
- License expiration date.

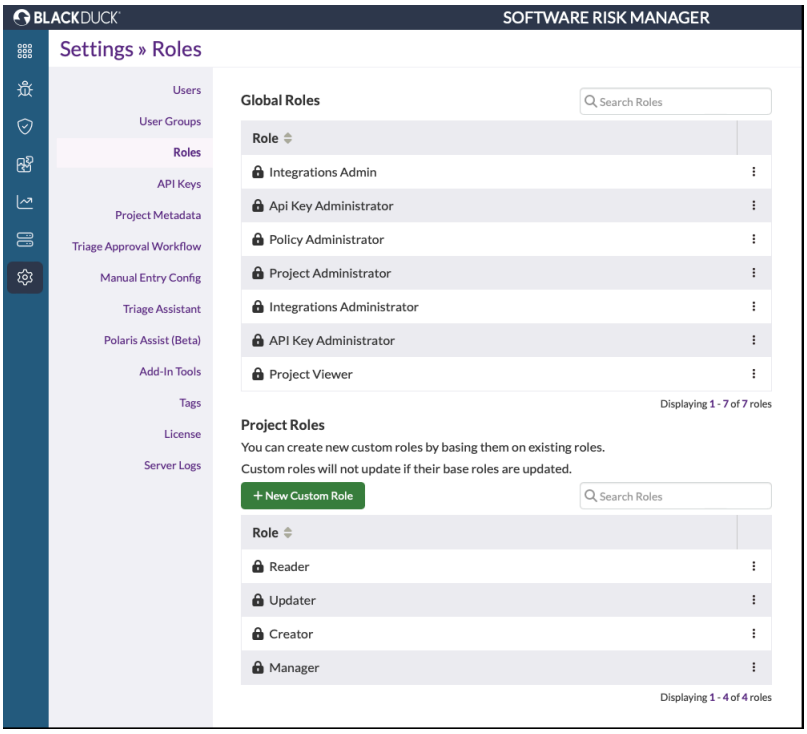
Click Update License to enter a new license key.



Roles

Roles enable users to interact with SRM in ways permitted by the set of permissions included in their definition. Role definitions can be viewed, edited, and created on the Roles page.

Click the Settings icon in the navigation bar and select Roles from the left menu to open the Roles page.



This page displays both Global Roles and Project Roles. SRM comes preconfigured with a collection of built-in roles, each of which is denoted by a padlock symbol next to its name.

Built-in roles can only be viewed, while all other roles can be edited, deleted, and replaced. Only custom project roles can be created.

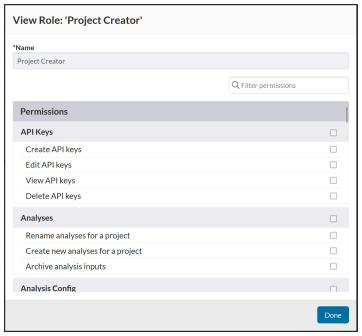
View Role

A built-in role's name and its set of associated permissions can be viewed on the Roles page.

To view a role's definition:

1. Click the Settings icon in the navigation bar and select Roles from the left menu.
2. Open the context menu for the role and select the View Role option.

   This opens the View Role window.

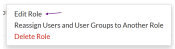
3. Review the role's name and its set of associated permissions.
4. Click Done.

Edit Role

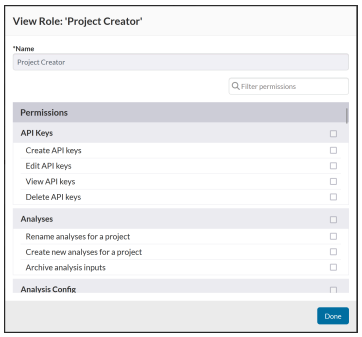
A custom project role's name and its associated set of permissions can be edited on the Roles page.

To edit a custom project role's definition:

1. Click the Settings icon in the navigation bar and select Roles from the left menu.
2. Open the context menu for the role and select the Edit Role option.



This opens the 'Edit Role' window.



3. Review the role's name and set of associated permissions and make changes as necessary.

> **Note:** Role names are unique. An error will be presented in the event that a role's name is changed to one that already exists.

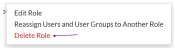
4. To save the changes made, click Save, otherwise click Cancel to discard them.

Delete Role

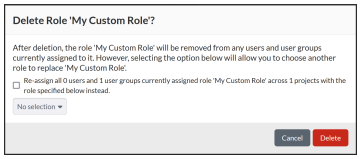
Custom project roles can be deleted on the Roles page.

To delete a custom project role:

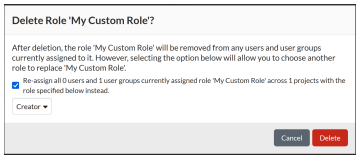
1. Click the Settings icon in the navigation bar and select Roles from the left menu.
2. Open the context menu for the role and select the Delete Role option.



This opens the Delete Role window.



3. If the role to be deleted has been assigned to users or user groups, a replacement can be selected by checking the "Reassign all..." checkbox and then selecting a replacement role.



4. To delete the role, click Delete, otherwise click Cancel.

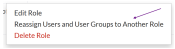
Replace Role Assignments

On the Roles page, users and user groups who have been assigned a role can have their assignment replaced with a different role.

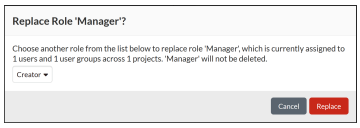
To reassign users and user groups who have been assigned a role with a different role:

1. Click the Settings icon in the navigation bar and select Roles from the left menu.
2. Open the context menu for the role and select the "Reassign Users and User Groups to Another Role."

   Note: The "Reassign Users and User Groups to Another Role" context menu option will be disabled if the role is not assigned to an users or user groups.

   This opens the Replace Role window.

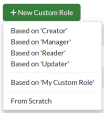
3. Select a replacement role.
4. To reassign users and user groups to the selected replacement role, click Replace, otherwise click Cancel.

Create Custom Project Roles

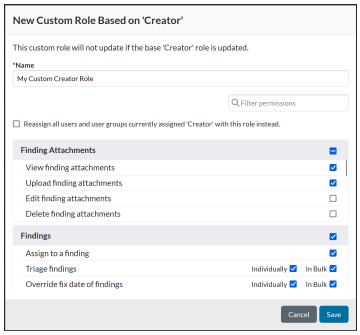
Custom project roles can be created on the Roles page.

To create a custom project role:

1. Click the Settings icon in the navigation bar and select Roles from the left menu.
2. Click on the New Custom Role button in the Project Roles section of the Roles page and choose to base the new role on an existing one or to build the new role from scratch.

   This opens the New Custom Role window.

3. Name the custom role and select any number of permissions.

   Note: Role names are unique. An error will be presented in the event that a role is attempted to be created with a name that already exists.

4. To create the new custom role, click Save, otherwise click Cancel.

For more information on how to assign roles, see the following topics:

- Configuring a User Profile
- Configuring User Group Roles

Project Management Overview

Before you can run an analysis on a file, you need to create a project. Once projects are configured, which includes policy associations, tool configurations, analysis settings, and so on, you can add files for analysis. SRM analyzes the files in that project and creates findings that can be viewed on the Findings page.

When working with projects, it's important to understand the following terms:

- **Project.** A collection of branches for a target software.
- **Branch.** A unique line of development containing a collection of scans over time. A project contains at least one branch, and each branch may contain any number of findings.
- **Analysis.** An individual scan, in which any number of tool results are taken into account in order to create or update findings.
- **Finding.** Information about some part of an application, generally a flaw or vulnerability. Findings are generated from an analysis, but can also be entered manually.
- **Tool Result.** Information about an application, as reported by a tool; tool results are correlated during analysis, becoming associated with findings.
- **Manual Result.** Information about an application which is entered into the system manually.
- **Result.** A generic term that includes both tool and manual results.

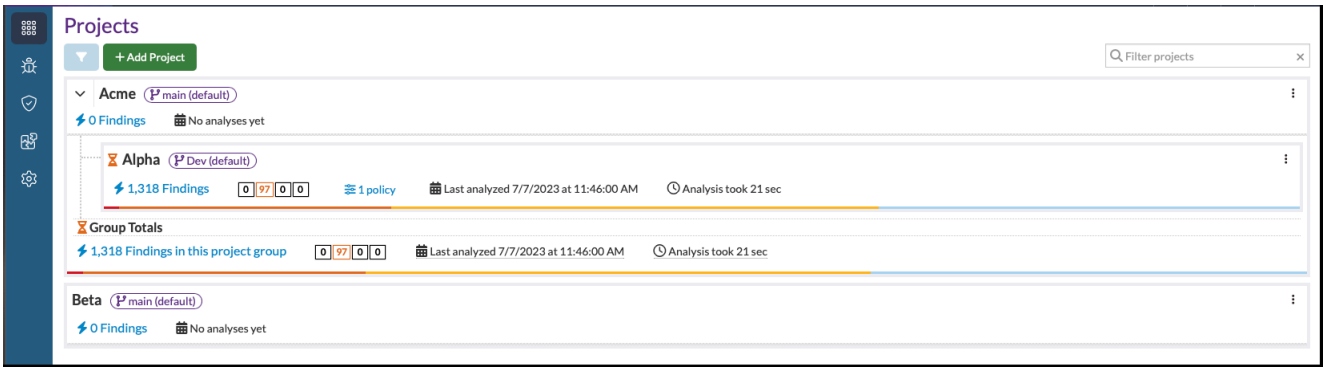
Project Management Tasks

For more information on Project Management tasks, see the following topics:

- Working with Projects
- Using Filters to Find Projects
- Adding a Project
- Working with Nested Projects
- Configuring a Project Analysis
- Configuring Tools for a Project
- Configuring Tool Connectors for a Project
- Analyzing Code in a Git Repository
- Issue Tracker Configuration
- Configuring Project Metadata
- Tool Service Configuration
- Orchestrated Analysis
- Working with Project Branches
- Intelligent Orchestration

Working with Projects

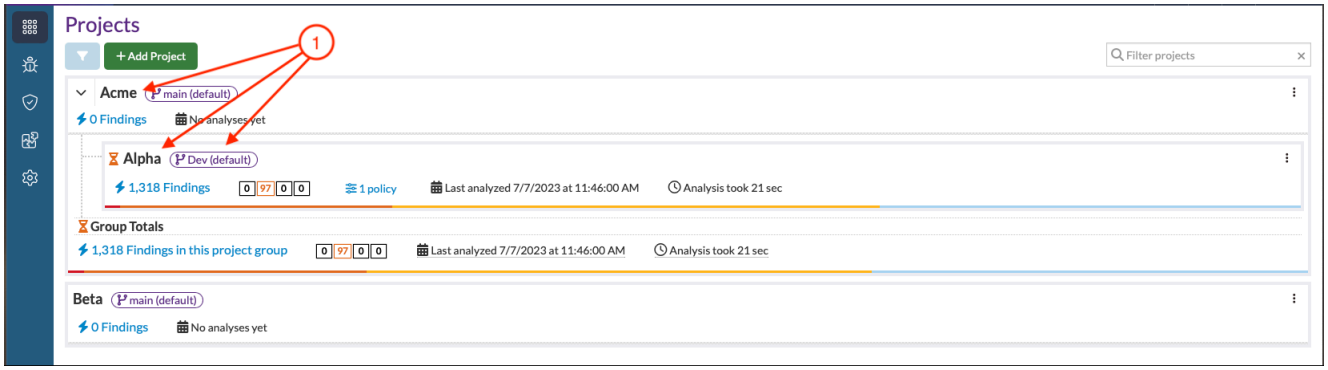
Click the Project icon in the navigation bar to open the Projects page.



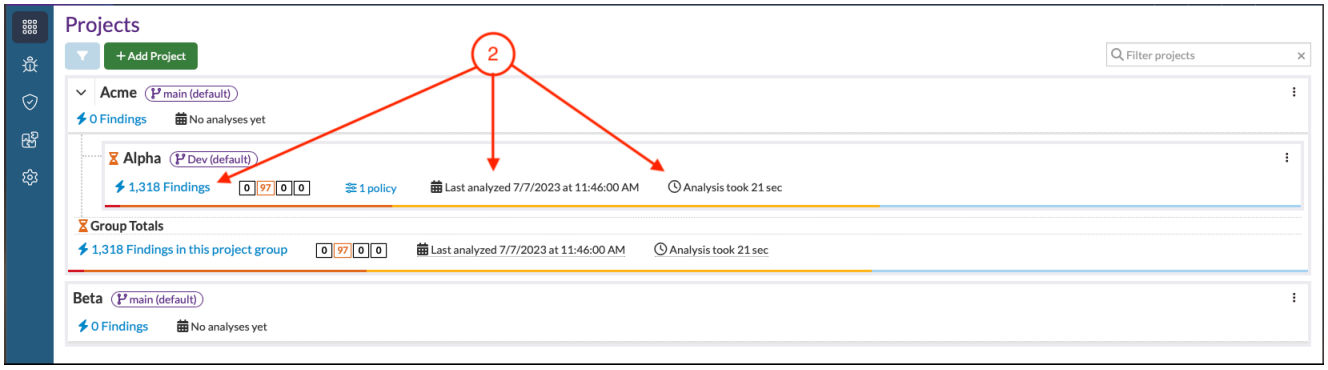
Viewing Project Information

The Projects page provides a variety of information and project options, as detailed below.

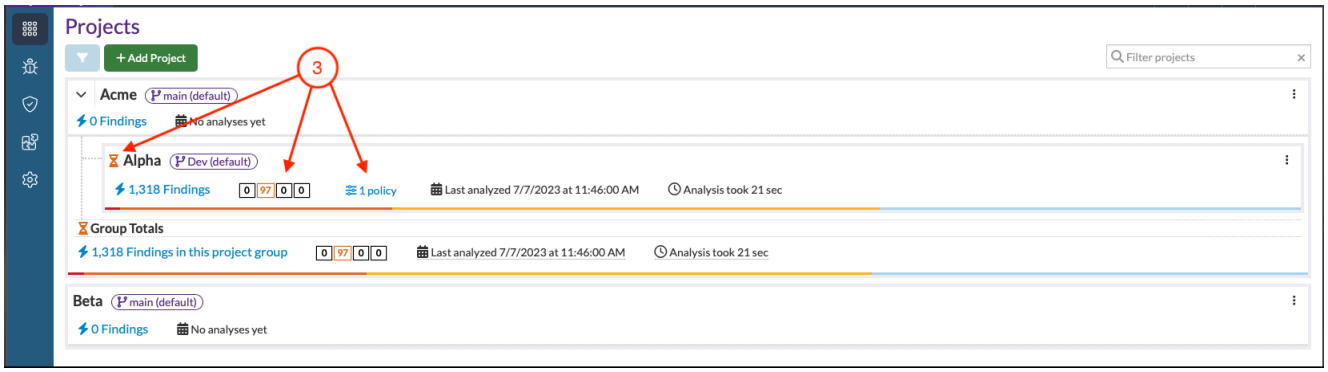
1. **Project name and branch.** This includes the project name, child projects, and associated branch. A paperclip "attachment" icon signifies that the project has linked attachments. (Clicking the icon displays the Attachments page with its list of attached files.)



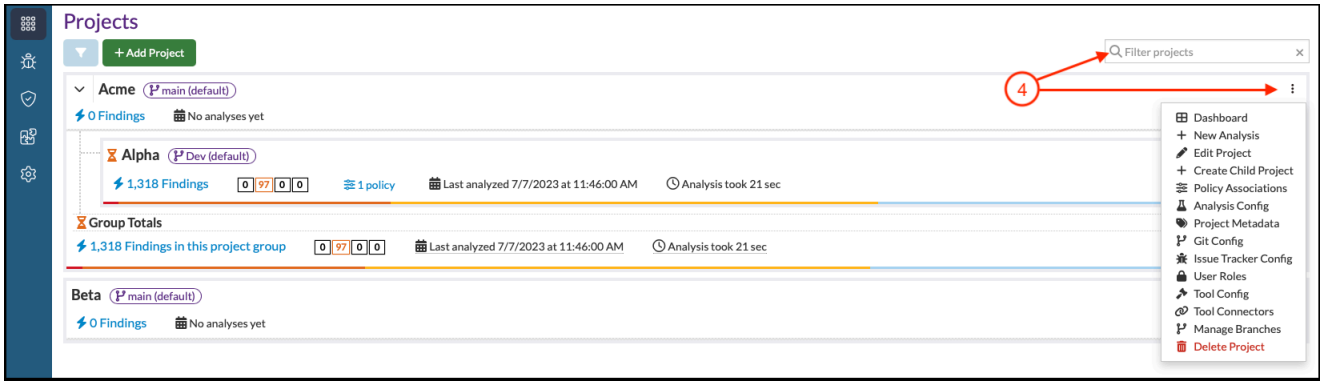
2. **Analysis information.** Shows the current number of findings, when the last analysis was completed, and how long the analysis took to complete. Click the Findings link to open the project's Findings page.



3. **Policy information.** This includes an icon next to the project name to indicate policy violation status, a tally of how many violations in each status category, and how many policies are associated with this project.



4. **Task options**. Click the dropdown configuration icon for a list of configuration options.



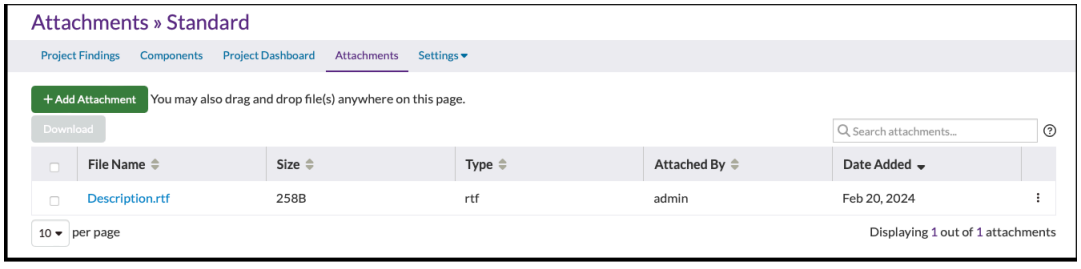
Adding an Attachment to a Project

File attachments can be added to a project from the Projects page or the project's Findings page.

Adding an Attachment from the Projects Page

To add an attachment to a project from the Projects page:

1. Click the Project icon in the navigation bar to open the Projects page.
2. Click the dropdown configuration icon for the desired project and select Attachments.
3. Use "drag-and-drop" to add a file or click Add Attachment.
   New files will be displayed in the attachment list, which includes the file name, file size, type, who attached the file, and when the file was added.

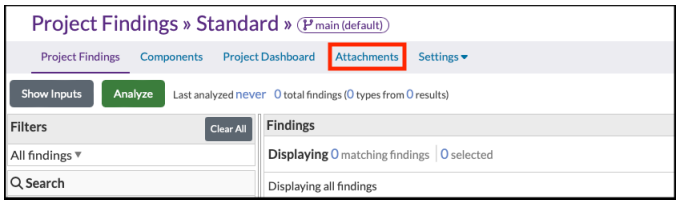


4. Use the dropdown configuration icon to download or delete an attachment.
   You can also click the filename to download the attached file. Or, you can download multiple files by selecting the file checkbox and clicking Download.

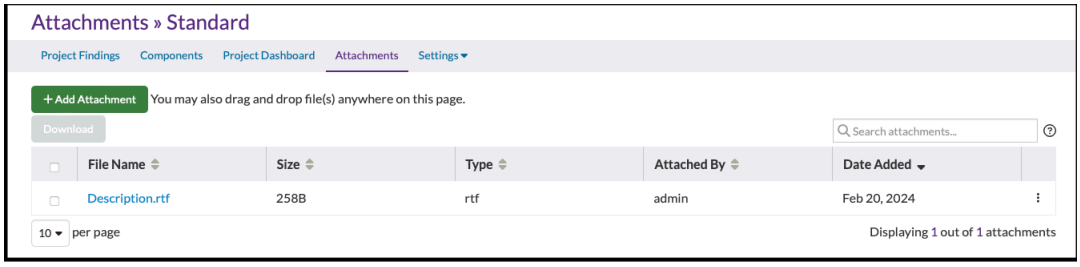
Adding an Attachment from the Findings Page

To add an attachment to a project from the project's Findings page:

1. Click the Project icon in the navigation bar to open the Projects page.
2. Locate the desired project and click the Findings link.



3. Select Attachments from the upper menu.
   New files will be displayed in the attachment list, which includes the file name, file size, type, who attached the file, and when the file was added.



4. Use the dropdown configuration icon to download or delete an attachment.
   You can also click the filename to download the attached file. Or, you can download multiple files by selecting the file checkbox and clicking Download.

Project Configuration Options

Clicking the project's dropdown configuration icon provides the following options:

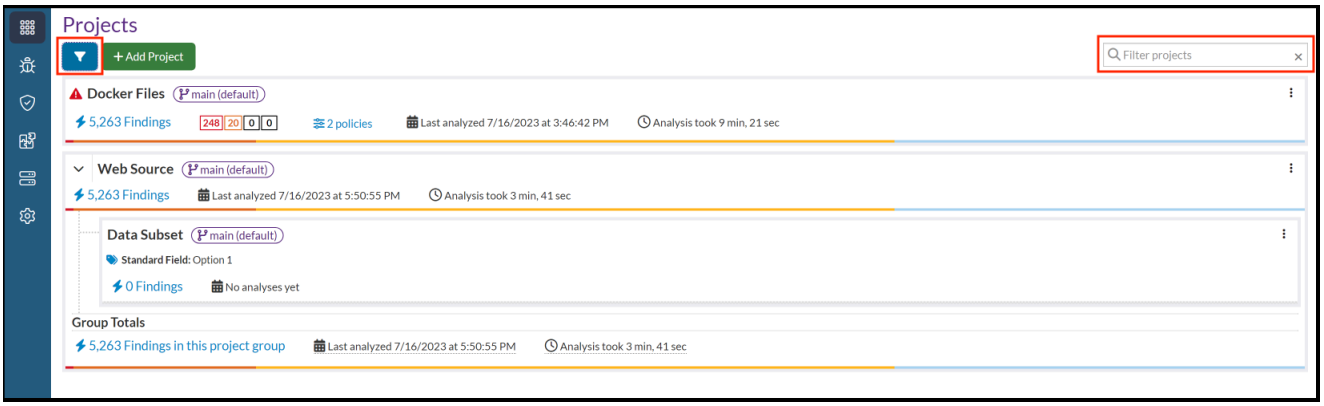
- **Dashboard.** Opens the project's analysis dashboard.
- **New Analysis.** Triggers an analysis.
- **Edit Project.** Opens an edit window where you can change the project name and parent project association.
- **Create Child Project.** Opens a window where you can create a new child project and branch association.
- **Policy Associations.** Lists current policy associations and provides options to add or delete policy associations.
- **Analysis Config.** Opens the configuration window where you can edit configuration settings.
- **Project Metadata.** Opens a window where you can add metadata to the project.
- **Intelligent Orchestration.** Allows you to enable or disable Intelligent Orchestration.
- **Git Config.** Allows you to enter a Git repository URL and assign a branch association.
- **Issue tracker Config.** Opens the Issue Tracker Configuration window where you can edit or delete configuration settings.
- **User Roles.** Lists current users and their assigned roles and provides options to edit roles according to project.

- **Tool Config.** Opens the Tool Configuration page where you can configure appsec tools.
- **Tool Connectors.** Opens a window where you can add or remove associated tool connectors.
- **Configure Tool Service.** Opes the Tool Configuration page where you can edit configuration settings.
- **View Orchestrated Analyses.** Opens the Orchestrated Analyses page where you can view analysis data.
- **Manage Branches.** Opens the Manage Branches page where you change branch associations.
- **Attachments.** Allows you to add attachments (files) to a project.
- **Delete Project.** Allows you to delete a project.

Using Filters to Find Projects

Filters can make it easier to find a specific project without having to scroll through the entire project list.

Click the Projects icon in the navigation bar to open the Projects page. The Filter field is located at the top of the page on the right side.

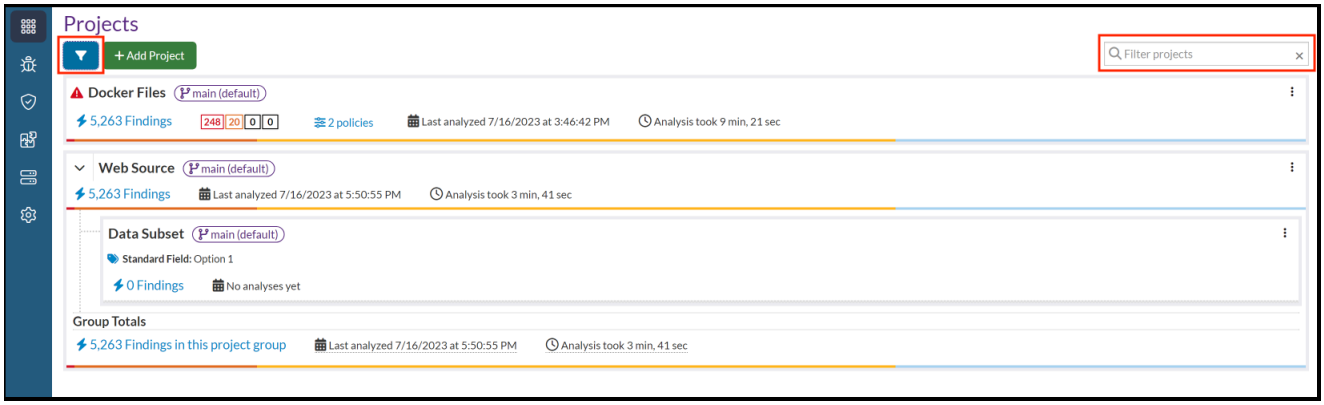


If the project includes metadata, advanced filters are available by clicking the filter icon next to the Add Project button.

Filtering Projects

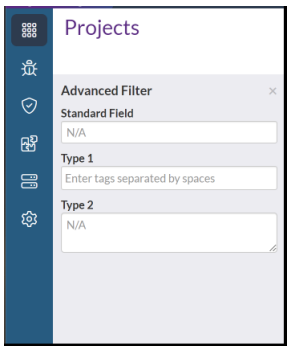
To filter the project list:

1. Click the Projects icon in the navigation bar to open the Projects page.



2. Enter a search term in the filter projects field and the list will filter automatically.
   Use the "x" to clear the search field.

3. (For projects including metadata) Click the Filter icon to open the advanced filter options.



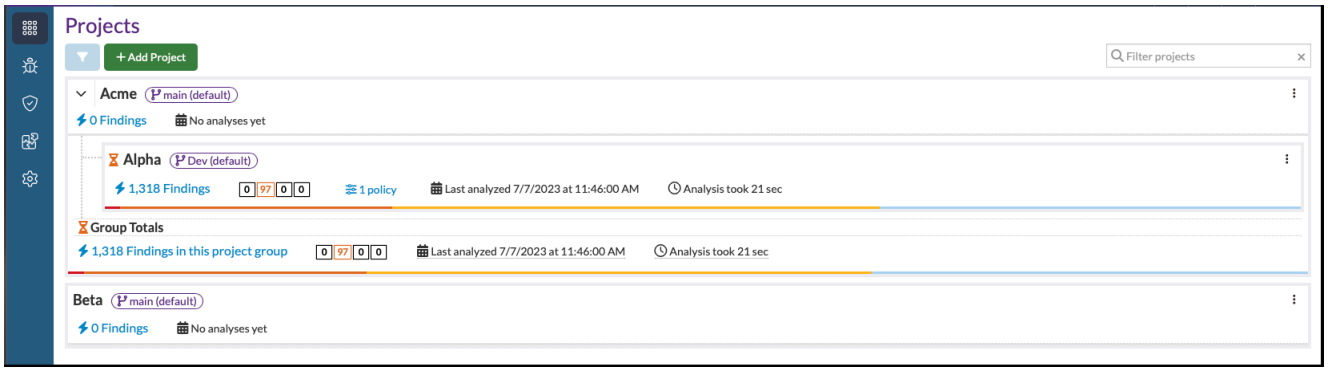
4. Enter search terms in the relevant metadata fields.
   The list will sort automatically. Clicking the Filter icon again will close the filter window.

Adding a Project

Before you can run an analysis, you need to create a project.

To add a project:

1. Click the Projects icon in navigation bar to open the Projects page.



2. Click Add Project.

3. Enter a unique project name.
4. Select or enter the default branch.
   For more information on project branches, see Project Branches below.
5. Click Save.

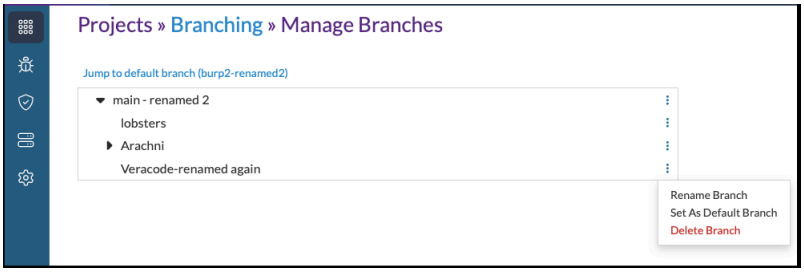
Project Branches

A *Branch* is a unique line of development containing a collection of scans over time. Each project contains at least one branch.

New branches can only be created by running an analysis. When creating a new branch for an analysis, a parent branch needs to be chosen. Conceptually, this represents the development line that this new branch was forked from. The new branch will be created by cloning the *Findings* and *Results* from the chosen parent branch, and then running the new analysis. The newly created branch will have the same contents as if the analysis had been run on the parent branch. However, since this is a new branch, the parent branch will be left untouched, and thus the two branches will be able to be tracked independently of each other over time.

The Branch Management page is used to manage the branches for a project and is accessible by users with the `manage` role.

Click the Projects icon in the navigation bar, then click the project's dropdown configuration icon and select Manage Branches.



From the Manage Branches page, you can view all branch hierarchies for a project as well as rename and delete individual branches. To navigate to a project's Branch Management page, click the project's dropdown configuration icon and select Manage Branches.

When viewing the finding and result information for a branch, a contextualized view of that information is given. Some information is shared globally across all branches containing the finding, while other information may differ based on the branch.

Globally shared finding information will be visible on every branch where that finding is present and will not change based on which branch is being viewed.

Information that is global includes the following:

- Comments added to findings by the user
- Issue Tracker Associations
- Tags

Contextual information will be tailored to the selected branch. Contextual information includes the following:

- **Finding Status.** Each branch includes a finding status, which includes "new," "existing," and "gone."
- **Severity Override.** Each branch has its own severity override for each finding.
- **Associated Results.** Different results may be present on each branch, depending on the analyses performed.
- **Location.** Note that line numbers may differ between branches.
- **Source Code.** Source code mapping will be based on the latest copy of source code uploaded for the branch.

The Activity Stream will be tailored for the branch that is being viewed and will include the global information as well as the contextual information (listed above) that is relevant to the branch. Contextual information is inherited from the parent branch at the time the branch is created. After the analysis for a new branch begins, any changes to the parent branch will diverge from the new branch and will not be visible in the child branch.

Project Groups

Projects may be repositioned in a hierarchy, where one project may become the parent (or group) containing another project.

Once you move one or more projects into a parent project, the parent project can be considered as a "project group." The Projects page displays project groups as a summary of all findings for all projects in that group, including the group project itself.

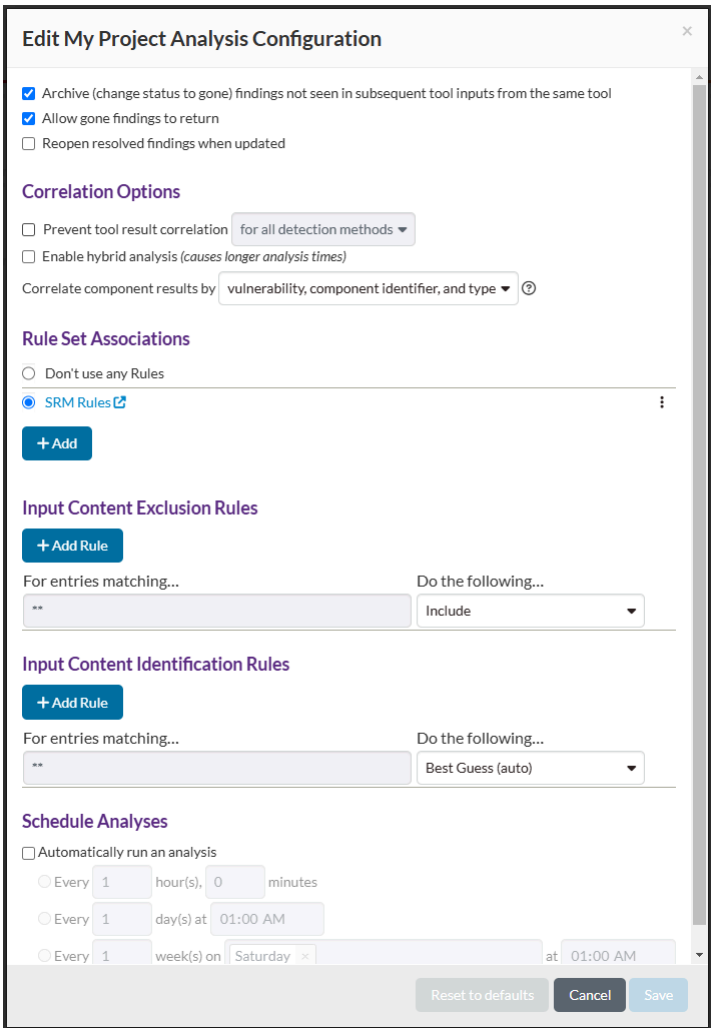
Note: A project group is still a project, and can still have findings of its own. The summary of findings specific to the parent project will appear above the child projects when you expand the group. There is no inherent limit to how deeply-nested projects can be. A child project can have its own child projects, and so on.

Configuring a Project Analysis

Before files can be analyzed, the project needs to be configured for an analysis. Configuration settings include auto-archiving, hybrid analysis, rule set associations, and so forth.

To configure a project analysis:

1. Click the Project icon in the navigation bar to open the Projects page.
2. Select a project and click the project's dropdown configuration icon and select Analysis Config.
3. Refer to the information below to complete each section.



Analysis Configuration Options

Select general configuration options. General configuration includes the following settings:

- **Archive (change status to gone) findings not seen in subsequent tool inputs from the same tool**
  As files are analyzed with Software Risk Manager, each one is remembered as an *analysis input*. As more and more analyses are performed, the number of *analysis inputs* can become very large. The *Auto-Archival* setting controls how old analysis inputs are handled.

  By default, auto-archival is enabled. As new inputs are analyzed, old inputs of the same type will be archived. For example, two analyses are performed in series on a project, both supplying a SpotBugs results file. In this scenario, the SpotBugs results file provided for the second analysis is perceived as "newer," so it will replace the SpotBugs results from the first analysis. The *analysis input* for SpotBugs results in the first analysis will be archived. Any findings that were present in the first file but not the second will have their statuses changed to *Gone* as a part of this process.

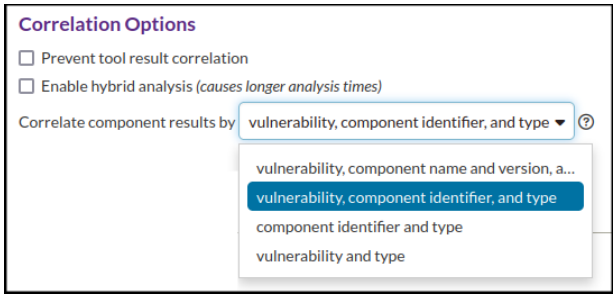
  With auto-archival disabled, the two SpotBugs result files will both remain present. This can be useful if you wish to provide one SpotBugs results file for a part of your application, but a different SpotBugs results file for a different part of your application. Both files may be analyzed without interfering with each other. However, keep in mind that without manual management of the *analysis inputs*, inputs will begin to pile up, potentially degrading the performance of filters and other interactions. **Note:** You can manually archive old inputs from the Analysis Inputs List.

- **Allow gone findings to be reopened**
  If the *Allow gone findings to be reopened* option is checked, then findings will be reused and have their status set to *Reopened* if they reappear later at the same location. With this option disabled, a new finding will be created instead.
- **Reopen resolved findings when updated**
  If the *Reopen resolved findings when updated* option is checked, then findings set to a resolved status (i.e., *Ignored*, *False Positive*, *Fixed*, *Mitigated*) will have their status changed to *Reopened* if new data is brought in from a tool (not matching previously seen data). Findings set to *Fixed* will be changed to *Reopened* if reported, regardless of if the data is new or not (since this signals that the issue has not been fixed).

Analysis Correlation Options

The correlation options are as follows:

- **Prevent tool result correlation**
  If the *Prevent tool result correlation* option is checked, then multiple tool results will not be added to a finding. This will give you a separate finding for every issue reported by a tool. Tool results will still be associated with rules according to the selected *Rule Set*; however, when multiple instances of the same issue occur at the same location, they will not be merged.
- **Enable hybrid analysis (causes longer analysis times)**
  If the *Enable hybrid analysis* option is checked, then additional steps will be performed during analysis to enable hybrid analysis. If you upload files that have Java Source or Java Binary files in them, Software Risk Manager will analyze the structure of these files (gathering information about their classes and methods), which will later be used to perform hybrid correlation. Note that this extra analysis is time-consuming; the larger the project, the longer the analysis. Because of this, the *Enable hybrid analysis* option is unchecked by default.
- **Correlate component results by [mode]**



  Modes control how Software Component Analysis (SCA) tool results are correlated to findings (for more information, see Understanding Component Correlation). The options are as follows:

  - vulnerability, component name/version, and type
  - vulnerability, component identifier, and type
  - component identifier and type
  - component name/version and type
  - vulnerability and type
- **Exclude host when correlating infrastructure security-related findings**
  If the *Exclude host* option is checked, then host information will be excluded from the correlation criteria when processing security-related findings. This will produce one finding with tool results from all hosts for each applicable rule. When this option is not checked, one finding will be produced for each host for each applicable rule.

Rule Set Associations

The Rule Set Associations section allows you to select the Rule Set that will be used to correlate similar tool results into Findings.

Three options are available:

- Don't use any rules
- SRM rules
- Clone of SRM rules

By default, new projects will use the built-in "Software Risk Manager Rules" set. The "Don't use any Rules" option is available if you don't want tool results to be mapped to rules. More information on Rule Sets can be found in the *Rule Sets* section of this guide.

Users with the `admin` role can use this section to manage Rule Sets by creating, cloning, or deleting them.

Click the dropdown configuration icon to open, copy, or delete the Rule Set.

Adding a Rule Set using the Add button will initialize a blank Rule Set.

A cloned Rule Set will be initialized as a copy of the "parent" set. This can be useful if you want one project to use mostly the same correlation logic, but with a few alterations from another project. Note that the default *Software Risk Manager Rules* set is read-only.

To make modifications to a rule set, you need to create a clone first. Click the dropdown configuration icon and select Copy to create a clone, then make modifications to the clone.

To view or modify an existing Rule Set, click the dropdown configuration icon and select Open.

Reminder: When making changes in the Analysis Configuration window, make sure to click OK to save any changes.

Reminder 2: Since a project's configured Rule Set determines the manner in which results are correlated, changing that configuration necessitates an update of the correlation. This happens when the configured Rule Set for a project is modified in any way, or if the Analysis Configuration is changed to use a different Rule Set. When this happens, the Findings page will display a notification prompting users to do so.

Host Scope Associations

Note: This section is only applicable to Software Risk Manager users with the InfraSec add-on.

Host Scopes are sets of projects that share host information with each other. They allow the Host Normalization process to determine which hosts are actually the same hosts within a Host Scope. Selecting any of the Host Scopes in the associations list will associate the current project with that Host Scope, which implies that the current project's host information belongs to the selected Host Scope. Click the Manage Host Scopes link to open the Hosts page.

Input Content Rules

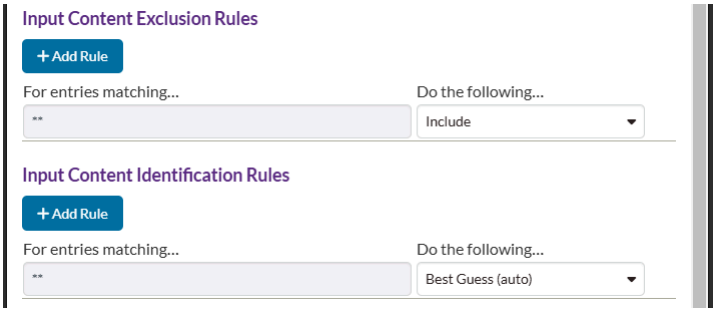
When a zip-like file (e.g. Zip, Jar, War, etc) is uploaded to a Software Risk Manager project, that project's *Input Content Exclusion Rules* and *Input Content Identification Rules* determine how entries in that zip file (and possibly entries in other zip-like files nested within the main zip file) will be treated by bundled tools using that file as input.

Exclusion Rules determine which zip entries will be ignored by bundled tools.

Identification Rules determine the perceived source of the zip entries, as either "library code" or "custom code" (third-party or first-party, respectively). Many tools will only be interested in custom code, and others (like component analysis tools) will only be interested in library code.

Proper configuration of these rules can drastically reduce the number of unwanted findings, for example, by avoiding analyzing files from a third-party library whose code you cannot directly modify.

By default, all entries in a zip-like file will be included, and their role (library code or custom code) will be automatically guessed by Software Risk Manager.



The two Input Content Rule sections in the Analysis Configuration window share a common format. Each row represents a rule, where files matching that rule's pattern will be subject to the decision chosen from its respective dropdown menu. Later rules (further down the list) take precedence over earlier rules. The first rule will always use `**` as its pattern, since it is the fallback for *all* zip entries. Its pattern may not be changed, but its decision *may* be changed. The patterns must be Glob Patterns, e.g., `**.java` matches any `.java` file in any folder. Patterns should use forward slashes (`/`) to denote directories instead of backslashes (\), even on Windows.

The `/` button at the right of the pattern input can be clicked to add a nested pattern that will apply to files nested in a zip-like file matched by the first pattern. For example, in a project where a `.war` file is typically uploaded, you could configure a pattern to match a particular `.jar` file inside that `.war` file, then click the `/` button to configure a pattern to match certain `.class` files inside that particular `.jar`. To undo adding a nested pattern, mouse over the > icon to the left of its text input. The icon will become a delete button, which can be clicked to remove the nested pattern.

To remove an entire rule, click the (-) icon to the right of the rule.

Input Content Exclusion Rules

Exclusion rules can be used to allow tools to completely ignore certain entries in an uploaded zip file. A typical use-case for this is to avoid analyzing test files. For example, entering `src/test/java/**.java` and selecting "Exclude" will exclude all `.java` files in any subdirectory of the `src/test/java` directory in the main zip file. (Note that since there is no leading `**` before `src`, it will only match if the `src` folder is at the "top level" of the zip. For example, the pattern won't match a file like `other/src/test/java/Foo.java`, but it will match a file like `src/test/java/Foo.java`.)

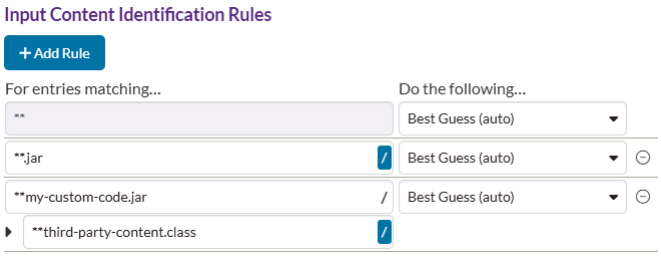
Input Content Identification Rules

Identification rules can be used to direct the attention of bundled tools to the correct sections of an uploaded zip file. For example, many projects will contain a mix of first- and third-party code, and without insider knowledge, there generally isn't a way for Software Risk Manager to know which is which.

One example of this is when uploading a `.war` file to be analyzed. The typical internal structure of a `.war` file includes many third-party libraries as `.jar` files, and often the custom code (first-party) is compiled into another `.jar` file and placed alongside the third-party `.jars`. In this case, Software Risk Manager has no way to distinguish whether each individual `.jar` file is first- or third-party, so it will typically assume that all of them are first-party. This can lead to analysis tools becoming overwhelmed and running out of

memory and causing the analysis to fail, or if the tool doesn't fail, it will have produced a large number of unactionable results due to it analyzing code from those third-party `.jars`.

In the example below, a configuration has been made to address a particular case like the one described above. It starts by assuming any `.jar`file is "Library Code", by combining the `**.jar` pattern with the *Mark as "Library Code"* decision. The next rule uses a more specific pattern to match a `my-custom-code.jar` and identify it as "Custom Code". This rule overrides the previous one because it comes after. Next, the user realized that they had some third-party library classes embedded in their "custom code" `.jar` file, so they configured a rule to mark those specific files as "Library Code," This was done by first entering `**my-custom-code.jar` as the pattern, then clicking the `/` button to add a nested pattern, then entering `**third-party-content.class` as the nested pattern.



Schedule Analyses

You can set your project to schedule an analysis for its Tool Orchestration, Tool Connectors, and Git configurations. Scheduling intervals include hours and minutes, number of days with a specified time of day, and number of weeks with a specified day and time. Click the checkbox "Automatically run an analysis," then use the radio buttons to select how often to run the analysis. Click Save to keep your settings.

Using Tool Connectors with Analysis Scheduling

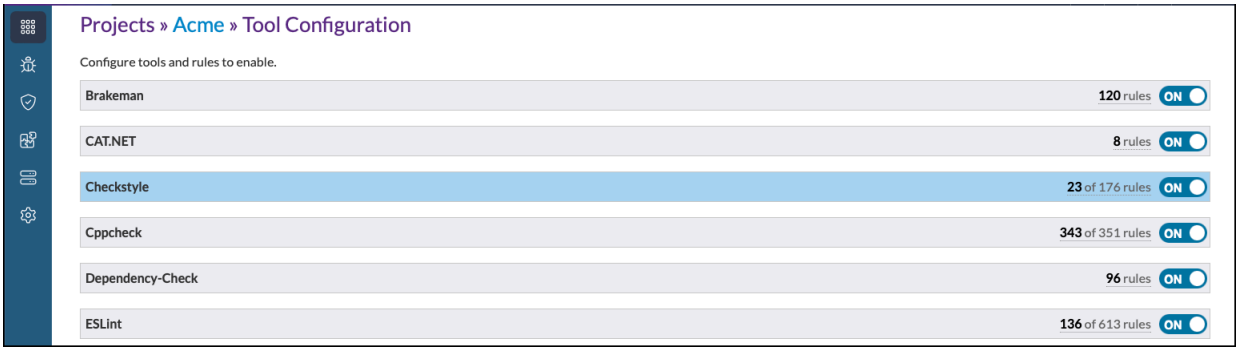
This section describes what you should do to have your tool connector(s) included in the analysis.

1. From the Projects page, click a project's dropdown configuration icon and select Tool Connectors.
2. The Tool Connectors window shows your existing tool connectors. Click the add icon (+) for each tool connector that you want to include in every analysis.
3. Check Run this connector during normal analyses. This will let the tool connector run during an analysis.

Configuring Tools for a Project

During an analysis, tool results are identified by a *tool result type*, which is a combination of the tool's name, any number of "groupings" (e.g., categories), and a name. The Tool Configuration page allows users with the `manage` role on a project to enable and disable tool result types for that project. Results whose tool result types are disabled by configuration will be ignored during an analysis.

Click the Projects icon in the navigation bar to open the Projects page, then click the Project's dropdown configuration icon and select Tool Config.



Tool Result Types

Tool result types are organized in a hierarchy, grouped by tool, then category(ies), then name. Tool result types can be enabled and disabled at any level of their hierarchy by clicking its respective on/off switch. For example, it is possible to completely disable a tool by clicking the switch next to that tool on the Tool Configuration page. Some entries will be disabled by default. The default enabled state is carefully selected to provide optimal results. However, this can be overridden at any time from this page by re-enabling the desired tool result types. Clicking on an entry (aside from its toggle switch) will expand (or collapse) it, showing all of its sub-entries.

Note: Any changes made on this page are project-wide, impacting all users of the project.

Software Risk Manager comes with a large set of predefined tool result types, based on the results generated by a collection of open-source tools. When Software Risk Manager encounters a new type of tool result, it will create a corresponding entry based on the result's raw tool code. These entries are referred to as "observed," and are marked with an eye icon.

If a change to the tool configuration would cause existing tool results to be disabled, it does not immediately remove those results. Instead a notification will appear, indicating the number of results that would be affected, prompting the user to purge those results. Clicking the Purge button in the notification will remove any tool results that still exist despite being disabled. Doing so is highly recommended, as having fewer tool results will improve the performance and responsiveness of Software Risk Manager. If you do not purge disabled tool results, they will remain present in the project and will continue to appear in filters and affect future analyses.

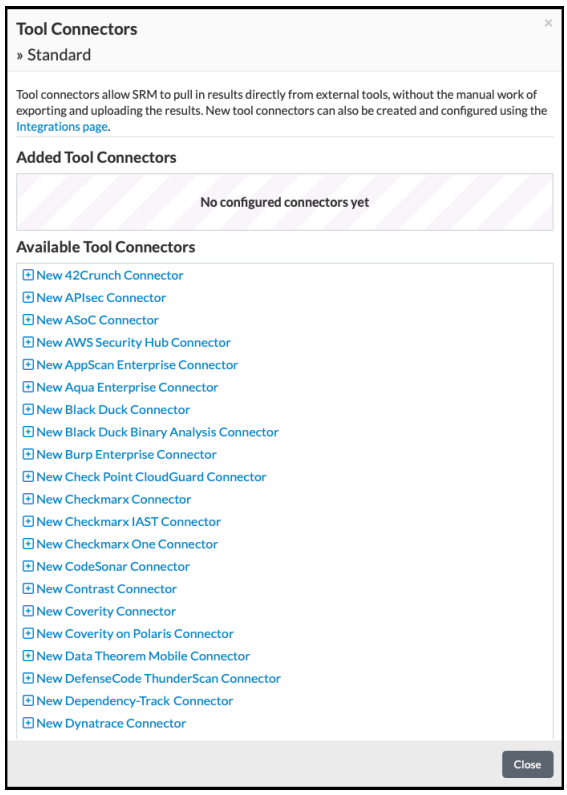
Configuring Tool Connectors for a Project

Tool Connectors allow Software Risk Manager to pull results directly from external tools without manually exporting the results from those tools and uploading them. Software Risk Manager provides tool connectors for a variety of app-sec tools. For a complete list, click here.

Note: Users with the manage role need only configure a connection to their tools once.

The Tool Connectors window for a project can be accessed from the project's dropdown configuration icon on the Projects page.

This window shows any tool connectors that have already been added, along with any tool connectors configured via central configuration, as well as a list of available tool connectors.



On a new project, no tool connectors will be configured.

Clicking a link in the bottom section will open a form to configure a connector for the link's respective tool.

Each tool connector configuration form includes a set of fields and tabs, starting with a Config Name field, which can be anything you choose. (The name won't affect the connector's functionality; it simply identifies this particular configuration. Since users may configure multiple connectors for a single tool, it may be useful to choose different names for each connector.)

Configuration tabs and fields are as follows:

- **Connection tab**
  - **Tool-Specific Fields.** To connect to a tool's API, SRM requires valid credentials for that tool. Different tools will require different fields; however, typically there will be a URL in combination with user authentication, such as a username/password combination or API Token.
- **Project tab**

    - **Project (or similar).** Different tools may use different terminology, such as *Build* or *Application*. This field is used to tell Software Risk Manager which of the tool's projects to pull from. This tab will be disabled until valid credentials have been entered.
- **Version tab**
    - **Target SRM Branch.** This specifies the name of an SRM branch that the user wants to run analyses with the selected tool connector.
    - **Derive version information based on SRM branch name during analysis.** This checkbox, when available, disables other fields from the integration. For example, if there was a "branch" dropdown from the integration, and you selected this checkbox, the "branch" dropdown would be disabled. This allows SRM to determine the rest of the configuration based on the SRM branch where you are running the analysis.
    - Fields defined by the integration.
- **Options tab**
    - Available options depend on the specific tool connector.
- **Schedule tab**
    - **Auto-update.** Selecting Auto-update along with one of the options below it tells Software Risk Manager to automatically perform an analysis using the configured connector at a regular interval. Selecting the first option will tell Software Risk Manager to auto-update at a fixed time interval, e.g., *every 12 hours*. Selecting the second option will tell Software Risk Manager to auto-update at a specified time each day.
    - **Run this connector during normal analyses.** This selection will cause the tool connector to appear on the New Analysis page as if it were a bundled tool, allowing the tool connector to run during a normal analysis, alongside any other files you might want to upload for analysis.

Note: Credentials entered for tool connector configurations will be stored (encrypted, but still reversible) by Software Risk Manager. For added security, it is recommended that users create one-off accounts in a tool, with the sole purpose of connecting Software Risk Manager to that tool.

Some tools support a "branch" abstraction (though each tool may have its own name for it, such as "Stream" or "Version"), allowing you to choose different incarnations of a selected project. When supported by Software Risk Manager, the corresponding field in that tool's connector config form will include a checkbox that allows you to opt in to sync that tool's "branch" with a corresponding *Software Risk Manager Branch.* The sync setting affects the auto-update behavior of the tool connector, as well as setting the default behavior of the *Run Now* form when you manually run the tool connector. When enabled, Software Risk Manager will try to run the tool connector on a *Software Risk Manager Branch* whose name corresponds to the selected value in the tool connector configuration dropdown.

Once all of the fields are completed, click OK to save the configuration and return to the connectors list.

Each configured connector has three buttons:

- **Run Now.** This can be used to start an analysis using a particular tool connector. This process is independent of the auto-update setting: it can be done regardless of whether or not the connector is configured to auto-update and will not interrupt the auto-update schedule. Users with the create role (specifically, the `project:manage-tool-connectors` and `analysis:create` permissions) for the project will be able to interact with this button.
- **Edit.** This reopens the configuration form for an individual tool connector. Only users with the manage role will be able to interact with this button.
- **Delete.** This deletes an individual tool connector. Only users with the manage role (specifically, the `project:manage-tool-connectors` permission) will be able to interact with this button.

After clicking Run Now in the list of configured connectors, a form will appear, allowing you to choose the *Software Risk Manager Branch* in which the analysis will be run. If you configured one of your tool connector's fields to sync with a *Software Risk Manager Branch*, the form will default to the corresponding branch. If the configured sync target branch does not exist, you must also select a parent branch so that Software Risk Manager can create a new branch, forked from your selected parent branch, to correspond with the sync target from your connector configuration. Once you complete the branch selection in the form, submit it by clicking Run Now. This will initiate a new analysis to run the connector in the background, close the Tool Connectors dialog, and display a notification.

Qualys VM Tool Connector and Scheduling Auto Update

For more information, see the following sections:

- Qualys VM Tool Connector
- Scheduling Auto Update

Supported Tool Connectors

Tool Connectors allow Software Risk Manager to pull results directly from external tools, without the manual work of exporting the results from those tools and uploading the results into Software Risk Manager. Users with the `manage` role can configure a connection to their tools one time and have Software Risk Manager take care of the rest.

Software Risk Manager currently provides connectors for the following tools:

- 42Crunch
- Aqua Enterprise
- Acunetix 360
- APIsec
- AWS Security Hub
- Black Duck SCA
- Black Duck Binary Analysis
- Burp Enterprise
- CAST Highlight

  Note: The CAST Highlight tool connector requires the user associated with the access token to have the "Portfolio Manager" role.
- Check Point CloudGuard
- Checkmarx
- Checkmarx One
- Checkmarx-IAST
- Checkmarx OSA
- CodeSonar
- Continuous Dynamic (formerly WhiteHat)
- Contrast
- Coverity Connect
- Coverity on Polaris

- Data Theorem Mobile Secure
- DefenseCode ThunderScan
- Dependency-Track
- Dynatrace
- Faraday
- Fortify Software Security Center
- GitHub Advanced Security

  Note: The GitHub Advanced Security tool connector requires the user associated with the access token to have permission to access the repositories, scopes public_repo for Dependabot and/or scopes security_events for Code Scanning.
- Google SCC

  Note: The credentials file for a Google Service Account with the roles "Service Usage Consumer", "Service Account User", "Security Center Assets Viewer", "Security Center Sources Viewer", and "Security Center Findings Viewer" in the desired projects. The file can be generated by creating a new Key for the service account (JSON format).
- Hacker One
- HCL AppScan Enterprise
- HCL AppScan on Cloud (ASoC)
- Imperva
- Invicti Enterprise (formerly Netsparker Enterprise)
- IriusRisk
- JFrog Xray

  Note: The JFrog Xray tool connector requires the user associated with the access token to have the "Manage Reports" role.
- Mend
- Microsoft Defender For Cloud
- NeuVector
- NowSecure
- Orca Security

  Note: The Orca Security connector requires an API token with the following permissions:
  - Authorization - Integration API tokens Read
  - Shift Left - CLI (All)
  - Shift Left - Scan Logs (All)
  - Shift Left - Projects Read
- Prisma Cloud (Redlock)
- Prisma Cloud Compute (Twistlock)
- Polaris
- Q-MAST

- Qualys VM (InfraSec add-on)
- Qualys VMDR
- Qualys WAS
- Rapid7 InsightAppSec

  Note: The Rapid7 InsightAppSec tool connector requires the user associated with the access token to have the "InsightAppSec Admin" role.
- Rapid7 InsightVM
- SD Elements
- Seeker

  Note: The Seeker tool connector requires an API key with "Manage Projects," "View Reports," and "View Vulnerabilities" permissions.
- Semgrep
- Snyk
- SonarQube/SonarCloud
- Sonatype Nexus
- Black Duck Managed Services Platform
- Tenable.io
- Tenable.sc
- Tenable.io Web App Scanning
- Tinfoil API
- Tinfoil Web
- Trustwave App Scanner
- Veracode
- Wiz

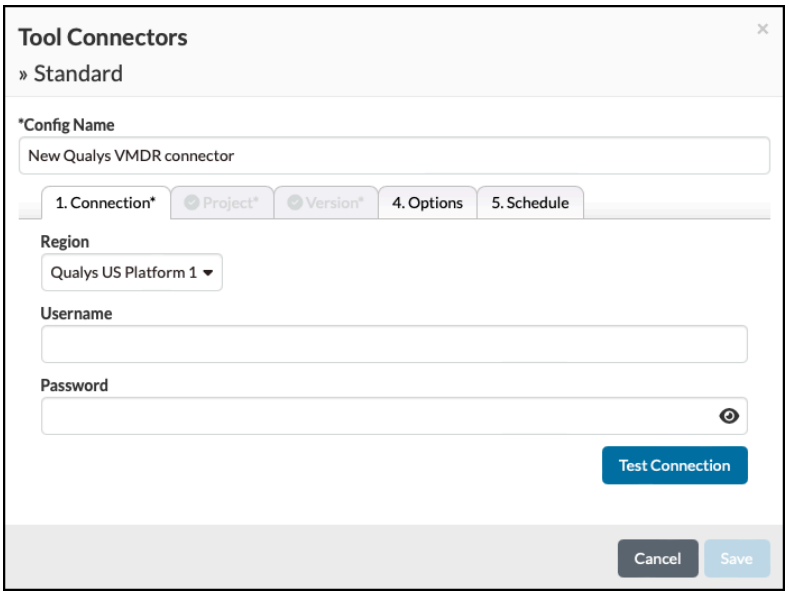
Qualys VM Tool Connector

Note: This section is only applicable to Software Risk Manager users with the InfraSec add-on.

The Qualys VM connector has two unique form configurations to choose from. The default form configuration has customization options including severity types, *Asset Group Titles*, *IP Ranges*, and *Include findings last seen* field. *Include findings last seen* is a required field and determines how far back to consider vulnerabilities that will be pulled into Software Risk Manager. *Asset Group Titles* and *IP Ranges* are optional fields and act as filters. For example, if you provide an IP range, only that information will be pulled into Software Risk Manager. Additionally, if both fields are left blank, all vulnerability information in Qualys will be pulled into Software Risk Manager. Multiple IPs can be specified by separating them with a comma, and IP ranges can be specified by separating them with a hyphen.

To access the second form configuration, select the *Import data using a Report Template* option. This form will present you with a *Report Template* dropdown and *Check on report every* field. Both fields are required for this configuration. The *Check on report every* field determines how often Software Risk Manager will interface with Qualys to get the status of the report being analyzed. The *Report Template* dropdown is populated with report templates that have been configured for your Qualys VM subscription. Software Risk

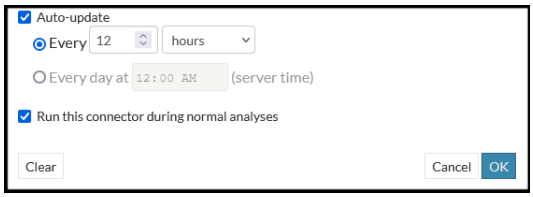
Manager will request that Qualys generate a report using the selected report template; once the report has been generated, it will be imported into Software Risk Manager.



Scheduling Auto Update

To set an auto-update schedule for your tool connector:

1. Click the Projects icon in the navigation bar to open the Projects page.
2. Click the Project's dropdown configuration icon and select Tool Connectors.
3. Select a tool connector, then click the dropdown configuration icon and select Edit.



   - To choose a daily time to update, select *Every day* at and fill in the time of day to run the analysis.
   - To choose a specific schedule to update, select *Every* and enter the number of analyses to run and the time frame (minutes, hours, days).
4. Select *Run this connector during normal analyses* if you want your tool connector to run during a scheduled analysis.

Note: If you configured one of your tool connector's fields to sync with a Software Risk Manager branch which does not yet exist, the connector will not be able to run. This will be indicated with a warning icon next to the tool connector's name in the list of configured tool connectors. The appropriate Software Risk Manager branch can be created by running a normal analysis or by clicking the Run Now button and submitting the subsequent form. Once the appropriate Software Risk Manager branch has been created, auto-update can continue.

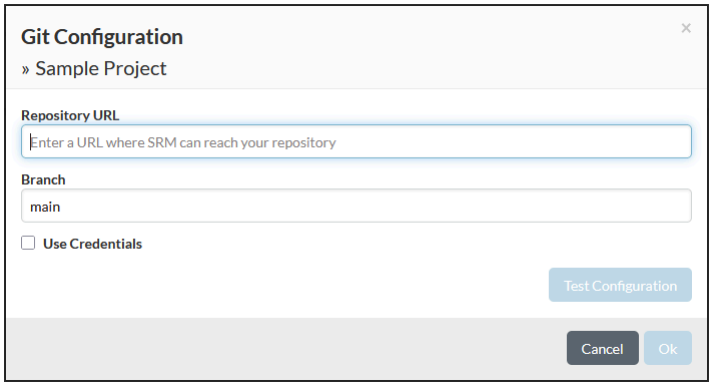
Analyzing Code in a Git Repository

Software Risk Manager can be used to analyze code stored in a Git repository.

Configuring a Project to Use a Git Repository

To configure a project to use a Git repository:

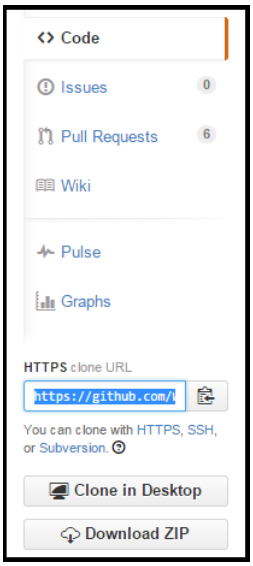
1. Click the Projects icon in the navigation bar to open the Projects page.
2. Click the project's dropdown configuration icon and select Git Config.
3. Enter the Repository URL and branch name.
4. If the repository requires credentials to access, click *Use Credentials* and enter the credentials.
5. Click *Test Configuration* to validate the configuration.
6. Click *Ok* to save the configuration.
   The *Git Configuration* popup will appear.



The form inside is used to tell Software Risk Manager to use a Git repository as the subject of analysis for this project. Once configured, Software Risk Manager will automatically include the contents of the configured repository as an input for each analysis with this project.

The Git Configuration window has two fields: *Repository URL* and *Branch*. The *Repository URL* should be filled out with the URL that you would use to clone the repository. The *Branch* field should be filled with the name of the branch in that repository that you want Software Risk Manager to analyze. By default, the branch is set to "main", which is the main branch for most Git repositories.

For many projects, setting up a Git configuration is as easy as copying the repository's URL into the text field and clicking *Test Configuration*. For example, if you wanted to analyze the contents of the open-source WebGoat repository, you would find the clone URL on the side of the GitHub repository page, then copy it into the *Repository URL* field in the *Git Configuration* window.



Software Risk Manager will verify the repository's existence and determine whether it needs credentials to connect. Click *Ok* to save and close the form.

Note: For public (open-source) repositories, credentials are not required. However, if credentials are required, see Working with Git Credentials for more information on how to manage credentials.

Once you have entered a URL and branch, and entered whatever credentials are necessary, click *Ok* to save the configuration. Doing so will close the window and tell Software Risk Manager to obtain a local clone of the configured repository. Depending on the size of the repository, the length of its history, and your network connection, the clone operation may take anywhere from seconds to hours. Once started, a progress bar will be displayed underneath the project's title on the Projects page.

The "cloning" job has several subtasks, so you will see the progress bar fill up several times. When the job is complete, the progress bar will turn blue, remain for a couple of seconds, then slide out of view.

Once the clone is ready, the New Analysis page will automatically include the latest contents of the configured branch of the configured repository as an input. See the Analysis Overview section for more detail.

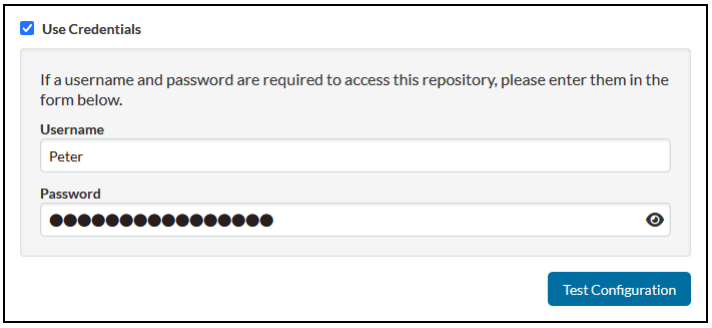
Working with Git Credentials

Some Git repositories are private and require credentials for access. Software Risk Manager supports two forms of authentication: HTTP and SSH. Depending on the URL in the Repository URL field, Software Risk Manager will automatically determine which type of credentials are required.

If you try to test the configuration for a private repository without entering credentials, Software Risk Manager will automatically enable *Use Credentials* and require that they are entered. Similarly, if credentials are provided when configuring a public repository, Software Risk Manager will automatically disable *Use Credentials*.

HTTP Credentials

HTTP credentials consist of a username and password. For GitHub repositories, these will generally be your GitHub account name and password. GitHub also supports creating "Personal access tokens," which can be used in place of a password.



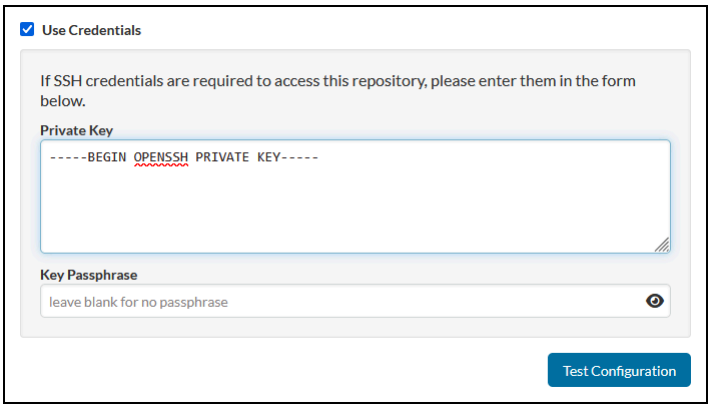
SSH Credentials

SSH uses a pair of files known together as a "keypair," or separately as a "private key" and "public key." For Software Risk Manager to connect to a repository through SSH, it needs your private key. The system in charge of the repository's security will also need your public key.

If you are trying to access a private GitHub repository, visit your SSH Keys page at https://github.com/settings/ssh to register your SSH key with GitHub. GitHub also provides help with SSH-related issues at https://help.github.com/categories/ssh/.

Some users will already have an SSH keypair on their computer. The two files are generally located in `<userhome>/.ssh/` and will be named `id_rsa` for the private key and `id_rsa.pub` for the public key. It is possible to use this pair, but you may want to generate a separate pair for use with Software Risk Manager.

Once you have located or generated a keypair, copy the contents of the private key file into the Private Key field.



When generating a keypair, you have the option to provide a passphrase for the private key. If you do this, Software Risk Manager will need that passphrase to use your private key. Enter it in the Key Passphrase field.

Two-Factor Authentication with GitHub

If you need to connect to a GitHub repository, and your GitHub account has two-factor authentication set up, you cannot use your regular username and password to authenticate. To connect over HTTP (e.g., `https://github.com/user/repo`), you will have to set up a Personal Access Token and use it in place of your regular password. You can still connect over SSH (e.g., `git@github.com:user/repo.git`) as usual.

Issue Tracker Configuration

Software Risk Manager allows you to associate findings with issues or work items in an issue tracker, either by creating a new issue or work item, or by identifying an existing issue or work item.

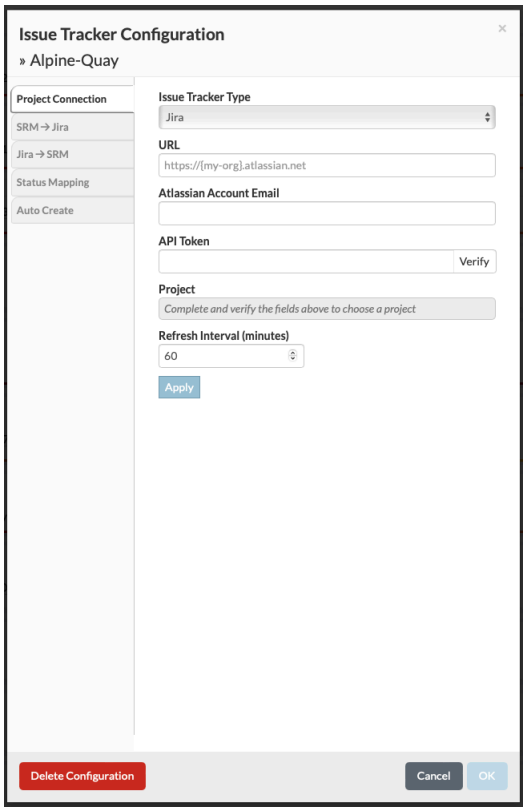
Software Risk Manager currently supports the following issue trackers:

- Azure DevOps (requires "Read" permission for "Graph" and "Project & Team" scopes, and "Read, Write, Manage" permissions for "Work Items" scope)
- GitLab (requires "api" access token scope)
- Jira (requires "Browse projects" project permission, "issue-level security" permission if issue-level security is configured, and the "read:jira-work" OAuth scope if using OAuth)
- ServiceNow (requires admin role or custom ACL rules granting access to the following ServiceNow tables - `incident`, `sys_user`, `sys_user_group`, `sys_choice` and `sys_dictionary`)
- GitHub and GitHub Enterprise (requires "repo" access token scope)

Configuring an Issue Tracker

To configure an Issue Tracker for a project:

1. Click the Projects icon in the navigation bar to open the Projects page.
2. Click the project's dropdown configuration icon and select Issue Tracker Config.



3. Enter the URL for your Issue Tracker server (including the "http://" or "https://"—even if you're using an IP address) as well as the credentials required for the user in whose name the issues or work items will be created.
4. Click Verify.
   Software Risk Manager will connect with the server and retrieve a list of projects the user can access.
5. Select the project you want to use from the dropdown menu.
   Software Risk Manager will periodically query the issue tracker server to refresh the status for all of the issues or work items associated with a given project. The Refresh Interval specifies the number of minutes between refreshes (the default is 60 minutes).
6. Click OK to save your configuration.
   If you delete the issue tracker configuration for a given project, all of the issue or work item associations tied to the findings in that project will be deleted. However, none of the issues or work items on the issue tracker server itself will be affected.

Additional Configuration Options

To perform additional configuration options, see the following sections:

- Advanced Field Configuration
- Issue Tracker Two-Way Sync
- Automatic Status Updating
- Automatic Issue Creation

Advanced Field Configuration

When creating an issue or work item from Software Risk Manager, several standard fields are provided (e.g., summary, description). However, many issue trackers provide more than just a few fields for issues or work items and can be configured to require these additional fields when creating an issue or work item. Issue trackers can also allow the creation of custom fields on a per-project or per-server basis. Software Risk Manager provides for this situation through "Advanced Fields." Jira users should note that while Software Risk Manager supports all of Jira's "Standard" custom fields as well as many of Jira's "Advanced" custom fields, some are implemented via third-party plugins and are not fully supported. These fields will still appear and can be used if the correct format is known, but they should be left empty otherwise.

You can create template expressions for any of the available fields when creating an issue or work item for the configured issue tracker server. These expressions will be applied to the relevant Software Risk Manager finding (or findings) when you create an issue or work item, which allows Software Risk Manager to pre-populate the field with data from the finding, according to your specification. More technical users should be advised that the template language is the JavaScript Handlebars library and that all of the template expressions are Handlebars Expressions.

Software Risk Manager will use its own default values for the Summary (Jira), Title (Azure DevOps, GitLab), Short Description (ServiceNow), Description and Due Date (Azure DevOps, GitLab, Jira, ServiceNow) fields if none are specified.

Note: Users should note that by default, the Due Date fields are based on the Software Risk Manager finding Fix By Date. The finding Fix By Date value is just a date; some issue trackers like Azure DevOps and ServiceNow also expect a time component in the Due Date value. So, SRM sends its local midnight as the time component value for these trackers, which may be rendered differently on the respective UI depending on the instance's timezone settings.

Users should also note that because fields can be given template expressions, which won't be evaluated until a finding is available, the validation that can be done on the fields is limited. The issue tracker field mappings are an advanced feature, and it is up to the user to make sure that the default values and expressions entered will produce valid values for the relevant issue tracker field types.

For more information on field configuration, see the following topics:

- Expression Basics
- Expression Logic
- Helpers
- Enumerable Fields

Expression Basics

The template engine will use the text you provide as-is, but it will treat anything inside pairs of double braces {{ }} as an expression to be evaluated using the active finding or findings. Software Risk Manager defines five basic data objects that can be used in the template expressions:

- `allFindings` - An array of all of the active findings ordered first by the severity then by finding ID.
- `finding` - The first element in `allFindings` (i.e., for multiple-finding issues, this will be the finding with the highest severity and lowest finding ID); this field is most useful for single-finding issues.
- `common` - An abstract finding containing any field whose value is shared by all of the findings in `allFindings`, containing the same fields enumerated below for finding objects; any value not shared by all findings will be reported as `null`.
- `project` - An object containing information about the project.
- `trackerType` - The type of issue tracker that the template is being generated for (`"jira"`, `"azure"`, `"servicenow"`, or `"gitlab"`).

These objects can be used to construct expressions containing data from the active findings. For example,

```
Finding {{finding.id}} has {{finding.severity.name}} severity
```

will, when applied to a Software Risk Manager finding with ID 1 and High severity, produce the following text:

```
Finding 1 has High severity
```

Code Block Format

Each issue tracker has its own markup language and may require special syntax when you create pre-formatted code blocks.

Jira: Use `{code}` before and after the code block.

```
{code} {{{requestBody}}} {code}
{code} {{{responseBody}}} {code}
```

ServiceNow: Use `[code]<code>` before the code block and `</code>[/code]` after the code block.

```
[code]<code> {{{requestBody}}} </code>[/code]
[code]<code> {{{responseBody}}} </code>[/code]
```

Azure DevOps and **GitLab** (triple backticks): Use ``` ``` ``` before and after the code block.

````
```
{{{requestBody}}}
```
```
{{{responseBody}}}
```
````

Finding Objects

The following fields are available on all finding objects (each element in the allFindings array, finding, and common). Fields marked as being optional can be omitted or set to null, all other fields will be present. The only exception to this rule is the common object, where any value not shared by all findings will be set to null regardless of whether it is optional or not.

- `id` - The ID of the Software Risk Manager Finding.
- `link` - A fully qualified URL pointing to the Software Risk Manager details page for this finding; must be wrapped to prevent html-escaping, for example `{{{finding.link}}}`.
- `triageStatus` - The finding's Triage Status, e.g. "Fixed" or "Ignored".
- `findingStatus` - The finding's Finding Status, e.g. "New" or "Existing".
- `assignee` - The name of the user that is assigned to the finding
- `firstSeenOn` - The date the finding was first seen, as text in `MM/dd/yyyy` format.
- `firstSeenOnDate` - The date the finding was first seen in ISO 8601 extended format; suitable for use with the `{{formatDate}}` and `{{formatDateTime}}` template helpers.
- `triageTime` - The date and time the finding's triage status was updated in ISO 8601 extended format; suitable for use with the `{{formatDate}}` and `{{formatDateTime}}` template helpers.
- `closeTime` - The date and time the finding's triage status was set to a closed status (i.e., Ignored, False Positive, Fixed, Mitigated, or Gone), in ISO 8601 extended format; suitable for use with the `{{formatDate}}` and `{{formatDateTime}}` template helpers.
- `fixByDate` - The Fix By Date of the finding in `yyyy-MM-dd` format.
- `detectionMethod` - An object representing the manner in which the finding was discovered.

    - `id` - An identifier for the detection method.
    - `name` - The name of the detection method (e.g. "Static Analysis").
- `detection` - A helper object containing booleans for some pre-defined detection methods.
    - `isDast` - True if the finding is a DAST finding.
    - `isSast` - True if the finding is a SAST finding.
    - `isComponent` - True if the finding is a Component Analysis finding.
    - `isHybrid` - True if the finding is a Hybrid finding.
    - `isInteractive` - True if the finding is an IAST finding.
    - `isThreatModel` - True if the finding is a Threat Modeling finding.
    - `isNetwork` - True if the finding is a Network Security finding.
    - `isDatabase` - True if the finding is a Database Analysis finding.
    - `isContainer` - True if the finding is a Container Analysis finding.
    - `isCloudInfrastructure` - True if the finding is a Cloud Infrastructure Analysis finding.
- `detectedBy` - The list of tools that detected the finding, in text form.
- `descriptor` - An object describing the type of finding.
    - `id` - An identifier for the descriptor.
    - `code` - A unique identifier for the descriptor.
    - `name` - A human-friendly name for the descriptor.
    - `type` - The type of descriptor; possible values can include the following:
        - `rule` - This finding represents one or more results that matched a rule in the current *Rule Set*.
        - `tool-code` / `observed-tool-code` - This finding directly represents a result from a tool.
        - `manual-entry` - This finding was manually entered.
        - `cve-group` - This finding was created to represent a group of CVEs.
    - `hierarchy` - The hierarchy of the type of finding, corresponding with the nesting represented in the *Type* filter on the *Findings* page; this is an array of strings, with `name` as the last element.
- `cwe` - An optional object representing the CWE associated with the finding.
    - `id` - The CWE ID associated with the finding.
    - `name` - The name of the CWE.
- `location` - The location where the finding has been identified.
    - `lines` - An object representing the line number range on which the finding is present, if available; uses the format `{start: <Number>, end: <Number>}`.
    - `line` - The line number of the location, if available, in text form (e.g., `'3-5'` or `'100'`).
    - `columns` - An object representing the start and end columns of the finding's location, if available; uses the format `{start: <Number>, end: <Number>}`.
    - `column` - The column number of the location, if available (e.g., `'12-44'` or `'440'`).
    - `path` - An object representing the location's path:
        - `path` - The full path of the location.

    - `pathType` - The type of the path (e.g., `url` or `file`).
    - `shortName` - A shortened version of `path`; this is the value that is displayed on the findings table for the finding.
    - `hasSource` - A boolean value reflecting whether Software Risk Manager has a source file for the given location.
- `element` - An optional object representing the element impacted by the finding:
    - `name` - The name of the element.
    - `shortName` - An abbreviated version of the `name`.
    - `type` - The type of the element.
        - `keyword` - A computer-friendly description of the element type (e.g., "query-string" or "http-header").
        - `name` - A human-friendly description of the element type (e.g., "Query String" or "HTTP Header").
- `severity` - An object representing the effective Software Risk Manager severity value for the finding (`severityOverride` if specified, otherwise `severityDefault`):
    - `key` - A numeric representation of the severity; higher is more severe.
    - `name` - The name of the severity (e.g., "Critical" or "Info").
- `severityDefault` - An object (in the same format as `severity` above) representing the severity of the finding as calculated by Software Risk Manager; this is the severity that is used if an override is not specified.
- `severityOverride` - An optional object (in the same format as `severity` above) representing the user-specified severity override for the finding, if provided.
- `descriptions` - An object containing the general and contextual descriptions for the finding:
    - `general` - The general description for the finding; corresponds to the description shown at the top of the Details page.
        - `format` - An indication of the description's format (e.g., `'text'`, `'markdown'`, or `'html'`).
        - `content` - The content of the description in the specified format.
    - `contextual` - The contextual description for the finding, if one is specified; this is in the same format as the `general` description above.
- `metadata` - An object containing metadata available for the finding; each key in the object is the metadata field name, and the value is the value for that field.
- `trainingLink` - A fully qualified URL pointing to a Secure Code Warrior training module if available. NOTE: you are required to wrap this field in an extra pair of curly braces to prevent html escaping of the URL (`{{{trainingLink}}}`, for example).
- `mostRecentAnalysis` - An object representing the last successfully completed analysis that either generated or updated the finding.
    - `id` - The ID of the analysis.
    - `projectId` - The ID of the Software Risk Manager Project this finding is on.
    - `state` - The state of the analysis, can be one of: `Created`, `Queued`, `Running`, `Failed`, `Complete`.
    - `createdBy` - The object representing the user who created the analysis.

        - `id` - The ID of the user.
        - `name` - The name of the user.
    - `creationTime` - When an analysis was created.
    - `startTime` - When an analysis started.
    - `endTime` - When an analysis ended.
    - `name` - The name of the analysis (note that this value is blank by default unless explicitly set).
- `sourceSnippet` - An object representing a snippet of code the finding occurs in.
    - `lines` - A list that contains the lines of source code for the snippet.
    - `startLine` - The line number from the source file corresponding with the first element of the list.
- `branch` - An object containing information about the branch the issue is associated with.
    - `id` - The id of the Software Risk Manager branch.
    - `name` - The name of the Software Risk Manager branch.
- `branches` - An array containing the branches this finding appears on.
- `results` - An array of all of the results (ingested from tools and manually entered) on the finding, corresponding with the *Evidence* section of the details page, ordered first by the severity and then by result ID. Each entry contains:
    - `id` - The ID of the Software Risk Manager Result.
    - `firstSeenOn` - The date the result was first seen in ISO 8601 extended format; suitable for use with the `{{formatDate}}` and `{{formatDateTime}}` template helpers.
    - `isManual` - A boolean indicating if the result was manually entered.
    - `detectionMethod` - An object representing the result's detection method (in the same format as the Finding `detectionMethod` above).
    - `tool` - An optional string representing the tool name (always present for tool results, but optional for manual results).
    - `severity` - An object representing the result's reported severity (in the same format as the Finding `severity` above).
    - `cwe` - An optional object representing the result's reported CWE (in the same format as the Finding `cwe` above); note that the result's `cwe` may be different from the finding's `cwe`, due to correlation based on rule sets.
    - `descriptor` - An object describing the type of result (in the same format as the Finding `descriptor` above).
    - `location` - An optional object representing the raw location reported by the result:
        - `rawDisplayPath` - The full display path, as reported by the tool; this will be `null` for manually entered results.
        - `pathObject` - An object representing the result's reported path (in the same format as the Finding `location.path` above); this path represents the normalized version of the path as understood by Software Risk Manager, and therefore may be slightly different from `rawDisplayPath`.
        - `lines` - An optional `{ start: <Number>, end: <Number> }` object for the result's reported line numbers, if specified.

    - `columns` - An optional `{ start: <Number>, end: <Number> }` object for the result's reported column numbers, if specified.
- `descriptions` - An object containing the general and contextual descriptions for the result.
    - `general` - A description object describing general information about the result (in the same format as the Finding description objects above).
    - `contextual` - A description object containing specific contextual data reported by the tool or manual entry (in same format as the Finding description objects above).
- `metadata` - An object containing tool-specific metadata; keys in this object are a camel case version of the name shown in the *Evidence* section of the details page - some (non-exhaustive) examples are as follows:
    - `ContinuousDynamicVulnerabilityId` - The Continuous Dynamic (formerly WhiteHat) Vulnerability ID (for Continuous Dynamic tool results).
    - `cvssV3` - The CVSS V3 score (typically for component analysis results).
    - `cpe` - The CPE of the associated component (typically for component analysis results).
    - `veracodeFlawId` - The Veracode Flaw ID (for Veracode tool results).
    - `sonatypeThreatLevel` The Sonatype Threat Level (for Sonatype tool results).
    - `prismaCloudComputeTwistlockDistro` The distro of the image scanned (for Prisma Cloud Compute [Twistlock] results).
- `httpMetadata` - An object containing the values associated with the result's HTTP Activity; similarly to `metadata`, the keys in this object are camel case and corresponds with the values displayed in the *Metadata* section for each HTTP variant on each associated result - for example:
    - `ContinuousDynamicAttackVectorId` - `["123", "456", "789"]` for a result with three Continuous Dynamic Attack Vectors with the IDs 123, 456, and 789.
- `cves` - An array of CVEs associated with the finding, in text form (each element in the array is a string in the format `CVE-YYYY-NNNN`).
- `vulnerabilities` - An array of vulnerability IDs (e.g., CVE and BDSA) associated with the finding, in text form (each element in the array is a string in the vulnerability format, such as `BDSA-YYYY-NNNN` for BDSAs and `CVE-YYYY-NNNN` for CVEs).
- `variants` - An object containing the values associated with the result's request & response HTTP Activity.
    - `requestData` - Formatted info for an HTTP request as seen in the "Raw Request Data" for a Result (does not include request body).
    - `responseData` - Formatted info for an HTTP response as seen in the "Raw Response Data" for a Result (does not include response body).
    - `requestBody` - The body of the corresponding HTTP request.
    - `responseBody` - The body of the corresponding HTTP response.
- `hostInfo` - An object containing the values associated with the result's Host Info.
    - `formattedHostname` - A formatted list of Host Names.
    - `formattedFqdn` - A formatted list of FQDN.
    - `formattedIp` - A formatted list of IP Addresses.
    - `formattedMac` - A formatted list of MAC Addresses.

- `formattedNetBios` - A formatted list of NetBIOS Names.
- `formattedOs` - A formatted list of Operating Systems.
- `formattedPorts` - A formatted list of Ports in the form of `[Port][Protocol][State]`.
- `formattedEnvironment` - A formatted list of Environments.
- `formattedHostInfo` - Full formatted Host Info using all previously listed items.

Project Object

- `id` - The ID of the Software Risk Manager Project.
- `name` - The name of the project.
- `metadata` - An array of value objects entered via the Project Metadata dialog.
  - `name` - The name of the metadata field.
  - `value` - The value entered for the field.
  - `valueId` - For "dropdown" fields, the choice ID.

Expression Logic

Most Handlebars expressions can be used. Some basic examples are given here, but much more information is available in the Handlebars documentation. Specific sections of interest are Expressions, Block Helpers, and Built-in Helpers.

Boolean Expressions

You can add basic boolean logic to your expressions by using the helpers `if`, `ifeq`, and `ifneq`. Note that `ifeq` and `ifneq` are custom helpers provided by Software Risk Manager.

- **If**

```
{{#if finding.detection.isDast}}
  This finding is a DAST finding.
 {{else}}
  This finding is not a DAST finding.
 {{/if}}
```

will result

```
This finding is a DAST finding.
```

when

```
This finding is not a DAST finding.
```

- **ifeq**
  The `ifeq` helper allows you to test the equality between two string or number values for a boolean result. Comparing values of types other than strings or numbers is unsupported, and the block will always evaluate to false.

  Note that `else` can not be used with `ifeq`; you may use `ifneq` instead to simulate `else`.

```
{{#ifeq finding.statusName "New"}}
  This finding is new
 {{/ifeq}}
```

will result in

```
This finding is new.
```

when a finding's status is new.

- **ifneq**
  The `ifneq` helper behaves the same way as `ifeq` except it negates the boolean result of testing the equality between two `string` or `number` values.

```
{{#ifeq finding.severity.name "Critical"}}
  This finding is critical
 {{/ifeq}}

 {{#ifneq finding.severity.name "Critical"}}
  This finding is not critical
 {{/ifneq}}
```

  will result in

```
This finding is not critical.
```

  when a finding's severity is `not Critical`.

Iterating Lists

You can iterate over arrays by using the each helper.

For example, the expression

```
{{#each allFindings}}
 {{id}},
{{/each}}
```

will result in

```
1, 2, 3, 4,
```

when evaluated on a group of findings with the IDs of 1, 2, 3, and 4.

In this example, all Results of relevant Findings are iterated through and all formatted Host Info and Variant Request and Response data is returned

```
{{#each allFindings}}
 {{#each results}}
  {{{hostInfo.formattedHostInfo}}}
  {{#each variants}}
   Request:
    {{{request-data}}}
   Response:
    {{{response-data}}}
  {{/each}}
 {{/each}}
{{/each}}
```

Understanding and utilizing `{{#each}}` is important, because as you can see in the above summary of the properties of the finding objects, many of the properties are arrays and therefore can't simply be accessed directly—you need to iterate over them and access each property inside the loop.

Helpers

Software Risk Manager includes all the #if, #unless, #each, and #with helpers provided by Handlebars. Several other helpers are also provided to assist as well.

formatDate

Note: It is recommended to use the new `formatDateTime` helper which gives you more flexibility to define both custom input and output format strings.

The `formatDate` helper allows you to format a date by specifying a format string. For example:

```
{{formatDate finding.firstSeenOnDate 'YYYY-MM-DD'}}
```

will take the first-seen date for the finding and convert it into an ISO-8601 valid format. You can use the symbols below to create your format string:

- `M` month: 1, 2, 11, etc.
- `MM` month: 01, 02, etc.
- `MMM` month: Jan, Feb, etc.
- `MMMM` full month name
- `D` day of month: 1, 2, 11, etc.
- `DD` day of month: 01, 02, 11, etc.
- `ddd` day of week: Sun, Mon, etc.
- `dddd` day of week: Sunday, Monday, etc.
- `YY` year: 98, 99, 00, 01, etc.
- `YYYY` year: 1998, 1999, 2000, 2001, etc.

The `formatDate` helper uses Moment.js under the hood, so you can look at its documentation for more formatting symbols.

formatDateTime

The `formatDateTime` helper allows you to format a datetime by converting it from a specified input format to a specified output format. For example:

```
{{formatDateTime finding.firstSeenOnDate 'YYYY-MM-DD' 'MM/DD/YYYY'}}
```

will take the first-seen date for the finding and convert it into the specified output format. You can use the symbols below to create your format strings:

- `M` month: 1, 2, 11, etc.
- `MM` month: 01, 02, etc.
- `MMM` month: Jan, Feb, etc.
- `MMMM` full month name
- `D` day of month: 1, 2, 11, etc.
- `DD` day of month: 01, 02, 11, etc.
- `YY` year: 98, 99, 00, 01, etc.
- `YYYY` year: 1998, 1999, 2000, 2001, etc.
- `H` hour of day: 1, 2, 11, 21, etc.
- `HH` hour of day: 01, 02, etc.
- `h` clock hour of AM/PM: 1, 2, 11, 12, etc.
- `hh` clock hour of AM/PM: 01, 02, 11, 12, etc.
- `m` minute of hour: 1, 2, 11, 31, etc.
- `mm` minute of hour: 01, 02, 11, 31, etc.
- `s` second of minute: 1, 11, 21, 51, etc.

- `ss` second of minute: 01, 11, 21, 51, etc.
- S fraction of second: 1, 11, 211, 351, etc.
- SS fraction of second: 01, 11, 211, 351, etc.
- a AM/PM of day: AM, PM
- z timezone name: GMT, EST, PST, etc.

The `formatDateTime` helper uses `java.time` under the hood, so you can look at its documentation for more formatting symbols.

makeOxfordList

The `makeOxfordList` helper assists in generating an Oxford list from an array of elements. The body will be evaluated for every item in the list.

For example, to list all of the Software Risk Manager Finding IDs, the template

```
{{#makeOxfordList allFindings ',' 'and'}}{{this.id}}{{/makeOxfordList}}
```

will result in

```
39955, 39956, 39939, and 39940
```

when evaluated on a group of findings with the IDs 39955, 39956, 39939, and 39940.

formatLocation

The `formatLocation` helper formats a location object into a user-readable version similar to those displayed elsewhere in Software Risk Manager.

For example, the template

```
{{formatLocation finding.location}}
```

will result in

```
AbstractLesson.java:182
```

when evaluated on a finding located in `WEB-INF/classes/org/owasp/webgoat/lessons/AbstractLesson.java` on line 182 (columns 25-50).

By default, the short name of the location is used, and any column numbers are omitted. You may opt to show the complete location information by passing `true` as the second parameter to this helper. For example, the template

```
{{formatLocation finding.location true}}
```

will result in

```
WEB-INF/classes/org/owasp/webgoat/lessons/AbstractLesson.java:182:25-50
```

in the previous example.

formatCWE

The `formatCWE` helper creates a more informative representation of a finding's CWE (if one is available).

Note: When using this helper, the last two parameters are optional.

The true/false parameter determines if a link to MITRE will be available. The `trackerType` parameter will default to the issue tracker type currently being processed.

For example,

```
{{formatCWE finding.cwe true trackerType}}
```

will result in

```
CWE 78 - Improper Neutralization of Special Elements used in an OS Command ('OS Command
 Injection') ([MITRE|https://cwe.mitre.org/data/definitions/78.html])
```

when evaluated on a finding associated with CWE-78.

If the second argument provided is true, a MITRE link for the CWE will be included (formatted properly for the issue tracker being used).

If no CWE is present on the finding, this helper will evaluate to an empty string if the second argument is false, or to "No Common Weakness Enumeration information available" if the second argument is true.

stripHtmlMarkup

The `stripHtmlMarkup` helper takes an HTML string and returns a copy with all HTML tags removed, with newlines/spaces inserted as necessary while attempting to preserve native formatting. By default, HTML escape sequences will be converted in the result; use `false` for the second parameter (as seen below) instead to prevent this. Whitespace-equivalent escape sequences (e.g., ` `) will simply be replaced with a space regardless of the second parameter value.

For example, a finding with a description of the following

```
<h1>Explanation</h1>
<p>
 Cross-site scripting vulnerabilities occur when:
 <ol>
  <li>Data enters through an untrusted source</li>
  <li>The data is included in dynamic content without being validated for malicious code</li>
 </ol>
</p>
```

and a template with

```
{{{stripHtmlMarkup finding.descriptions.general.content true}}}
```

will result in

```
Explanation
Cross-site scripting vulnerabilities occur when:

- Data enters through an untrusted source
- The data is included in dynamic content without being validated for malicious code
```

The following are known issues and limitations:

- Ordered/unordered lists will produce the same `-` prefix on list items.
- Nested lists may be improperly formatted.
- Issue trackers that interpret plaintext as emojis (e.g., `:(` (e.g., Jira) will still interpret as emojis.
- The un-escaped HTML content may be re-converted to the escape sequences automatically by Handlebars. (e.g., HTML content with `&`is un-escaped to `&` by `stripHtmlMarkup`, but gets converted back to `&` in the output by Handlebars.) Use `{{{}}}` instead of `{{}}` to prevent this behavior.

makeMarkupFromHtml

The `makeMarkupFromHtml` helper takes an HTML string and reformats it into appropriate markup for the current issue tracker. The helper uses the following formats:

- **Jira.** Atlassian Wiki markup format.
- **Azure DevOps.** No processing/pass-through (HTML content supported by ADO).
- **ServiceNow.** Simple text extraction, acts as an alias for `stripHtmlMarkup` with unescaping enabled.

- **GitLab.** Markdown.

The helper takes three arguments: the first is required and the following two are optional: the HTML to convert and the name of the issue tracker type to target (use `trackerType` if not manually overriding; accepted values are `jira`, `azure`, `servicenow`, and `gitlab`), and a boolean flag (`true`/`false`), indicating whether to pre-clean the HTML of any non-textual elements (defaults to `false`). The pre-cleaning flag is usually not necessary, but if there are formatting issues, the pre-cleaning option may prevent these issues.

For example, a finding with a description of

```
<h1>Explanation</h1>
<p>
 Cross-site scripting vulnerabilities occur when:
 <ol>
  <li>Data enters through an untrusted source</li>
 </ol>
</p>
```

and a template with

```
{{{makeMarkupFromHtml finding.descriptions.general.content trackerType}}}
```

will approximately result in

```
(If configured tracker is Jira or using `jira` as the second argument value)
h1. Explanation

Cross-site scripting vulnerabilities occur when:

# Data enters through an untrusted source
```

```
(If configured tracker is Azure DevOps or using `azure` as the second argument value)
<h1>Explanation</h1><p>Cross-site scripting vulnerabilities occur when: <ol><li>Data enters
 through an untrusted source</li></ol></p>
```

```
(If configured tracker is ServiceNow, or using `servicenow` as the second argument value)
Explanation
Cross-site scripting vulnerabilities occur when:

- Data enters through an untrusted source
```

```
(If configured tracker is GitLab, or using `gitlab` as the second argument value)
# Explanation

Cross-site scripting vulnerabilities occur when:

1. Data enters through an untrusted source
```

Keep in mind that the use of `{{}}` vs `{{{}}}` will affect how Handlebars escapes any HTML-like content—typically, the `{{{}}}` literal format is appropriate.

The following are known issues and limitations:

- Jira/Atlassian conversion ignores `<pre>` code blocks (limitation of Atlassian Renderer).
- Super/sub-scripts are generally unsupported (Jira/Atlassian renderer supports it but has known bugs).
- Nested lists in any format may be improperly formatted.
- Issue trackers that interpret plaintext as emojis (e.g., `:(`) (e.g., Jira) will still interpret as emojis.
- Issue trackers whose formatting is whitespace-sensitive (i.e., Jira, GitLab) may have formatting issues due to trailing/leading whitespace (the use of ~ for whitespace control is recommended).

jiraMarkupFromMd

The `jiraMarkupFromMd` helper is intended to be used to translate descriptions formatted using markdown into Jira markup.

For example,

```
{{{jiraMarkupFromMd descriptions.contextual.content}}}
```

will transform the following description:

```
# Header

Here is a *description*

1. Number one
2. Number two
3. Number three
```

into:

```
h1. Header

Here is a _description_

# Number one
# Number two
# Number three
```

makeSource Snippet

The `makeSourceSnippet` helper takes the `finding.sourceSnippet` object and outputs a formatted source snippet.

For example,

```
{{makeSourceSnippet finding.sourceSnippet}}
```

will result in

```
function foo() {
 console.log('bar');
}
```

The following are known issues and limitations:

- Jira/Atlassian shows line numbers, but by default do not support configuring the line numbers. The line numbers will start at 1.
- Jira/Atlassian has an issue with preserving the first source line's level of indentation.
- GitLab, Azure, and ServiceNow do not show line numbers.
- Service now only supports code blocks in specific fields, like the `Additional Comments` field.

minBy and maxBy

The `minBy` and `maxBy` helpers will cause the provided expression body to be evaluated against the object with the lowest or highest value. The path to this value is provided as an argument.

For example,

```
{{#minBy allFindings "severity.key"}}
         The lowest severity is on finding {{id}} with severity of {{severity.name}}.
{{/minBy}}
```

will result in

```
The lowest severity is on finding 10 with severity of Info.
```

when evaluated on a group of findings, where the lowest severity finding has the ID of 10 and severity of info.

In the case of a tie (e.g., multiple findings or results with the minimum or maximum value), the first item with that value will be used. As outlined in Finding Objects, in the case of a tie, this will give the finding or result with the highest severity and lowest numeric ID.

minByDate and maxByDate

The `minByDate` and `maxByDate` helpers will cause the provided expression body to be evaluated against the object with the lowest or highest date value in `yyyy-MM-dd` format. The path to this value is provided as an argument.

For example,

```
{{#minByDate allFindings "fixByDate"}}
        The lowest date is on finding {{id}}.
{{/minBy}}
```

will result in

```
The lowest date is on finding 10.
```

when evaluated on a group of findings, where the lowest Fix By Date finding has the ID of 10.

In the case of a tie (e.g., multiple findings or results with the minimum or maximum date), the first item with that value will be used.

Nested Arrays

These helpers can also handle nested arrays in a more advanced use case. This is signified by adding a `.[]` at any point the helper should continue iterating over inner arrays. The expression body's context will be at the inner-most object, and the parent object(s) may be referenced if desired.

Here are two examples to illustrate these points:

```
{{#maxBy allFindings "results.[].severity.key"}}
 Finding {{../../id}}, Result {{descriptor.name}}, Severity {{severity.name}}
{{/maxBy}}
```

will result in

```
Finding 10, Result My Weakness, Severity Medium
```

when evaluated against a group of one or more findings where the highest severity result across all of the findings is a result on finding 10, with a descriptor name of "My Weakness" and medium severity.

```
{{#maxBy allFindings "results.[].metadata.cvssV3"}}{{metadata.cvssV3}}{{/maxBy}}
```

will result in

```
9.8
```

when evaluated against a group of findings where the highest CVSS V3 metadata entry on any of the findings' results is 9.8.

Notice that the finding ID is accessed using `{{../../id}}`. You may have been expecting `{{../id}}` to fetch the finding ID, since the finding is the parent of the result that was selected. However, the array of results itself is the immediate parent, and the parent of that is the finding, so `../..` is used.

Enumerable Fields

Custom fields that are represented as one of a set of enumerable values (e.g., a set of radio buttons or a dropdown menu) can be configured to be pre-populated by selecting the enumerable Software Risk Manager field from the available dropdown menu. The currently defined enumerable fields are as follows:

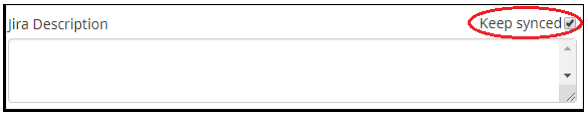
- Severity
- Triage Status
- Detection Method
- Static Value

Once you select a Software Risk Manager enumerable field, you'll see a table with a row for each possible value, along with a dropdown containing the possible values of your custom field. Choose which custom values from the dropdown menu that you want to use for each Software Risk Manager value. The `Static Value` option is available if you wish to define a single value for the Jira field regardless of the values in the Software Risk Manager finding.

Issue Tracker Two-way Sync

Software Risk Manager can be configured to automatically update issue or work item fields in response to any changes to a finding within Software Risk Manager. This is configurable on the "SRM -> *" tab.

Each field listed on the "SRM -> *" tab will have a "Keep synced" checkbox located to the right of the field's title. Enable this option to have Software Risk Manager push updates to editable fields for issues when the issue or work item's associated Software Risk Manager finding has changed. The values pushed to the issue tracker will be based on the branch associated with the issue at the time of creation. Currently, issues created via *Auto Create* can only be associated with a project's default branch.



Software Risk Manager can also be configured to watch specific issue or work item fields and update associated findings accordingly. This is configurable on the "* -> SRM" tab. Currently, only single select dropdowns and radio button fields can be mapped to affect Software Risk Manager finding Triage Status, Severity Override and/or Fix By Date[1]. Note that these changes will only take effect on the issue's associated branch; findings on other branches will be unaffected.

Note: 1. Only the Azure DevOps, GitLab, Jira and ServiceNow issue trackers support mapping of Software Risk Manager finding Fix By Date.

Automatic Status Updating

Note: This section is only applicable to Software Risk Manager users who are configuring a Jira integration.

Software Risk Manager can be configured to automatically update Jira issue statuses in response to status changes within Software Risk Manager. This is configurable on the "Status Mapping" tab.

Click the Projects icon in the navigation bar to open the Projects page, then select Issue Tracker Config from the project's dropdown configuration options.

When automatic status updating is enabled, a list of Software Risk Manager triage statuses will be shown, along with a dropdown list to pick the associated Jira status. These mappings are optional: if one is not selected, then no action will be taken on findings with that status.

After configuring status mappings, any time the status of a finding on the issue's associated branch is updated, the associated Jira issue will be updated according to the mapping (if applicable). If a transition is not available, then no action will be taken. If a transition requires some input for a field, Software Risk Manager will attempt to use any defined mappings in the "SRM -> Jira" tab that are marked "Keep synced" to satisfy those requirements. If multiple findings are associated with the same Jira issue, the Jira status will only be updated if all findings map to the same status.

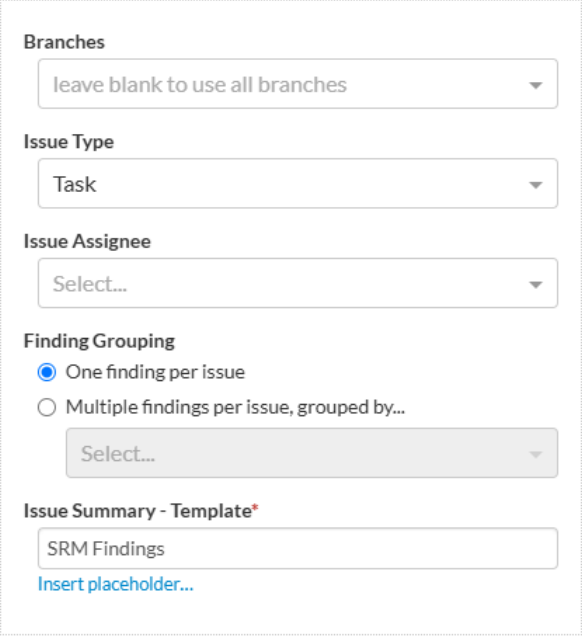
Automatic Issue Creation

Software Risk Manager can be configured to automatically create issues or work items based on a number of different criteria. This is configurable on the "Auto Create" tab of the Issue Tracker Configuration screen.

By default, Auto Create is disabled. To enable Auto Create, Jira and GitLab users should check the box labeled "Automatically create issues for findings," Azure DevOps users should check the box labeled "Automatically create work items for findings," and ServiceNow users should check the box labeled "Automatically create incidents for findings."

After enabling Auto Create, the rest of the form will be enabled and further configuration options are available.

Issue Configuration



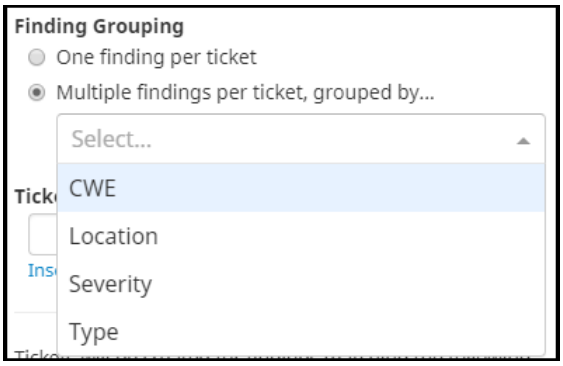
The following configuration settings are also required:

- **Branches.** What branches Software Risk Manager will use when creating issues or work items. If left blank, all branches will be used.
- **Issue Type.** What "Issue Type" Software Risk Manager will use when creating issues or work items.
- **Issue Assignee.** The user Software Risk Manager will assign these issues to (Jira and ServiceNow only; Azure DevOps and GitLab users can configure this on the "SRM -> *" tab).

- **Finding Grouping.** How Software Risk Manager should group findings (or not) when creating issues or work items.
- **Ticket Summary Template.** The summary or title Software Risk Manager will provide when creating issues or work items.

Finding Grouping

The "Finding Grouping" section allows users to either have Software Risk Manager create one issue or work item per finding, or group multiple findings together per single issue or work item. If *Multiple findings per ticket, grouped by...* is selected, the dropdown menu will be populated.



The selection(s) made here determines how findings are grouped. Multiple selections are allowed, but the order of the selections matters. For example, if "Location" is selected first, and "Severity" is selected second, Software Risk Manager will first group findings by their Location and then by their Severity. Therefore, if you had two findings at the same location but with different severities, these findings would be associated to different issues or work items.

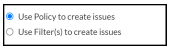
Ticket Summary

The *Ticket Summary - Template* field determines what Software Risk Manager uses for the summary or title when issues or work items are created. This field supports the same templates used on the field mappings tab (i.e., "SRM -> *" tab). For example, if you want Software Risk Manager to create issues or work items and have the summary display the finding's location and severity, you could configure a Ticket (Work Item) Summary as follows:

```
{{finding.location.path.path}} {{finding.severity.name}}
```

The *Insert placeholder...* control under the input will help in determining what kind of template expression to use.

Use Policy Rules or Filtering

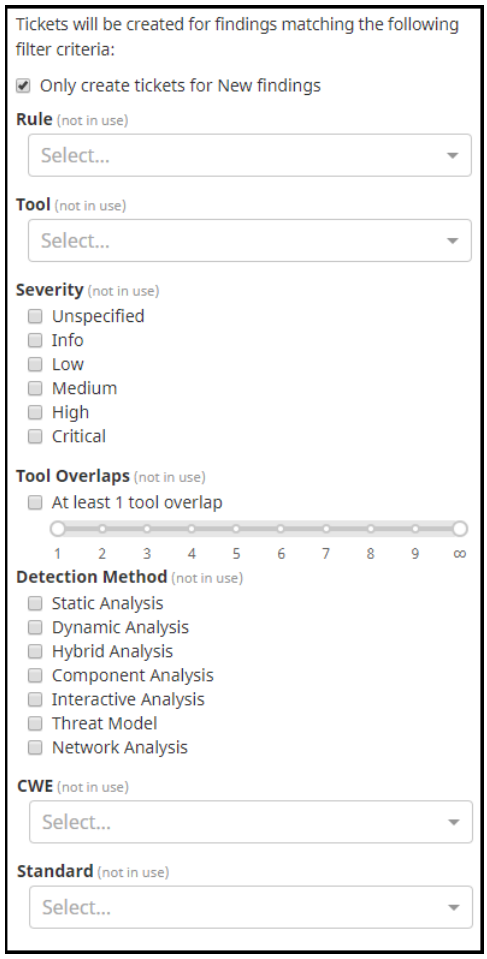


The option to "Use Policy" or "Use Filters" to create issues allows users to choose to either use configured Policy rules or the following filter options to determine which findings should have an issue automatically created.

Note: By default, "Use Policy to create issues" is selected. Users must assign a policy to this project, and that policy must have a rule where the action is "Create Ticket(s)." See Configuring Polices for more information.

Filtering

SRM provides a number of options you can use to filter which findings should have issues automatically created.



These filters include the following:

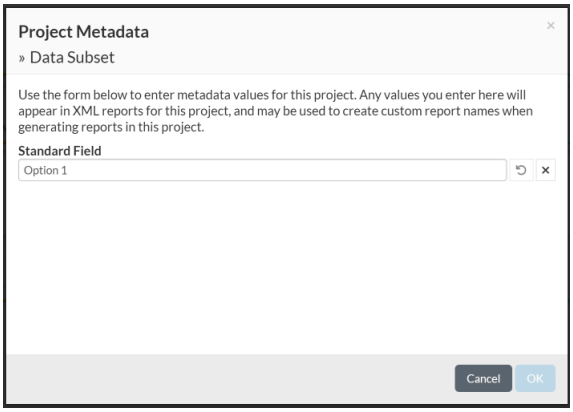
- **Only create tickets or work items for New findings.** If this is checked, only findings with a status of "New" will have issues or work items created for them.
- **Rule.** Select any number of Rules and only findings that match the selected Rules will have issues or work items created for them.
- **Tool.** Select any number of Tools and only findings that match the selected Tools will have issues or work items created for them.
- **Severity.** Check any number of Severities and only findings that match the checked Severities will have issues or work items created for them.
- **Tool Overlaps.** Select a number range and only findings that are in the selected Tool Overlap range will have issues or work items created for them.
- **Detection Method.** Check any number of Detection Methods and only findings that match the checked Detection Methods will have issues or work items created for them.
- **CWE.** Select any number of CWEs and only findings that match the selected CWEs will have issues or work items created for them.
- **Standard.** Select any number of Standards and only findings that match the selected Standards will have issues or work items created for them.

If a filter is left blank, that filter will not be used and all findings will be considered. For example, if you leave the Severity filter blank (i.e., nothing is checked), all severities will be considered.

Configuring Project Metadata

Project Metadata allows users of Software Risk Manager to enter values into Project Metadata Fields for any project they have the `manage` role for.

Click the Projects icon from the navigation bar to open the Projects page, then click the project's dropdown configuration icon and select Project Metadata.



It is up to an admin user to define the fields; once defined, they will be available to every project in your Software Risk Manager installation.

Adding Metadata to a Project

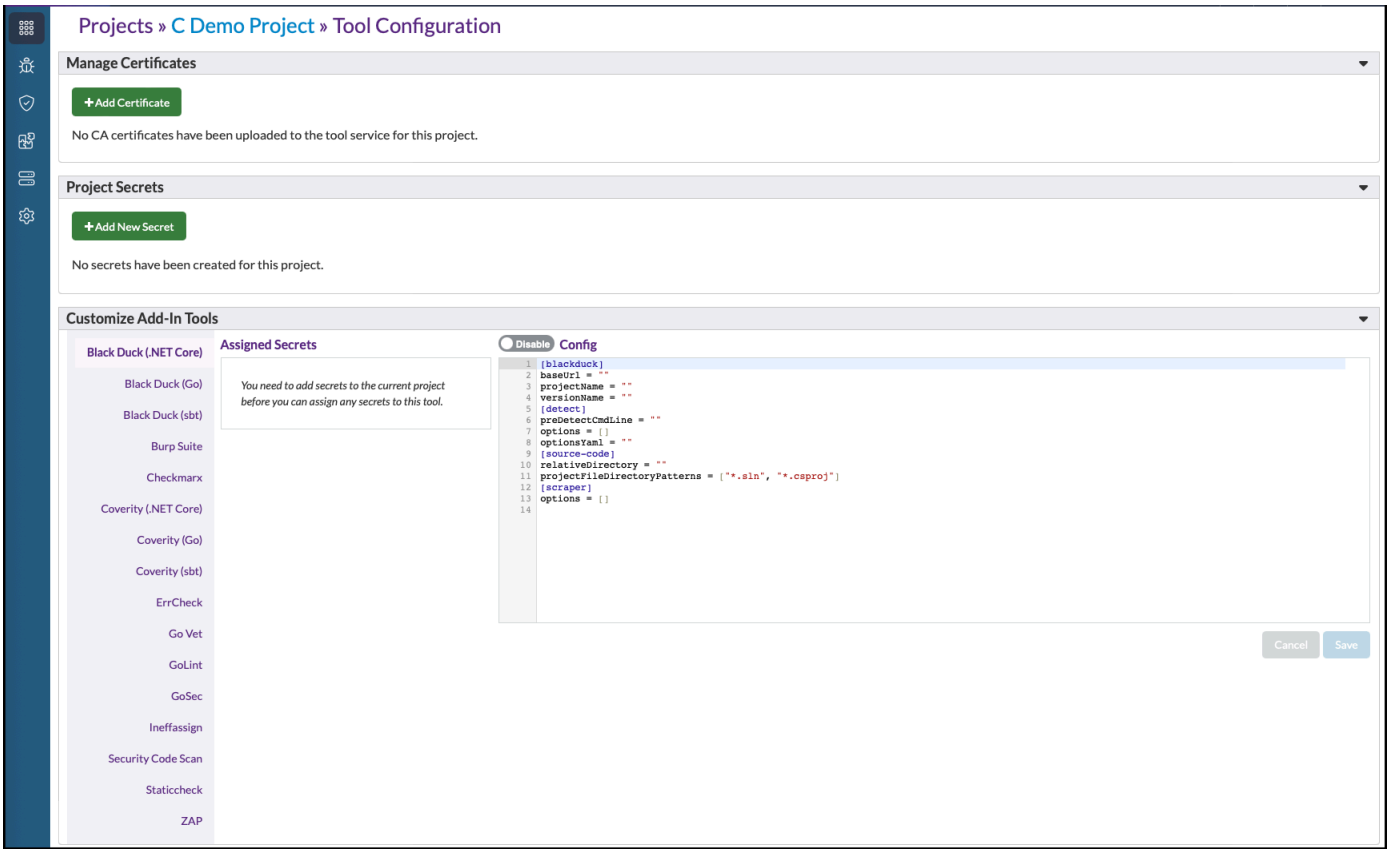
To add metadata to a project:

1. Click the Projects icon from the navigation bar to open the Projects page.
2. Click the project's dropdown configuration icon and select Project Metadata.
    - Click Reset (a circular arrow icon) to undo any changes and return to the last saved state.
    - Click Clear (an "X" icon) to clear all the fields.
3. Configure the fields as necessary.
   There are four of field types:
    - **Text.** A regular text input which allows a single line of text (e.g., *Project Owner* in the screenshot above).
    - **Multiline.** A larger text input which allows for multiple lines of text (e.g., *Description* in the screenshot above).
    - **Dropdown.** A dropdown select which allows users to pick one of the options (e.g., *Criticality* in the screenshot above).
    - **Tags.** A special input that behaves similarly to Text, but each individual word is converted to a "tag." As you type, pressing the space bar will convert whatever text you already had into a tag. Remember to enter a space after the last tag.
4. Click OK.

Tool Service Configuration

When the Tool Orchestration Service is enabled, the Tool Service Configuration page allows you to configure tool orchestration for an entire Software Risk Manager project.

Click the Projects icon in the navigation bar, then click the project's dropdown configuration icon and select Tool Config.



The Tool Service Configuration page has three sections:

- Manage Certificates
- Project Secrets
- Customize Add-In Tools

Manage Certificates

This section allows you to manage a list of certificates that tool orchestration components should trust.

The Manage Certificates section lets a tool or the Tool Orchestration Service handle applications that use a self-signed certificate or a certificate issued by a certificate authority that is not well-known. Click Add Certificate and specify a certificate file to update the list. You will see your certificate in the list after an upload completes. Software Risk Manager will give you the option to overwrite an existing certificate file or cancel your upload if you choose a file that is already in the list. The upload time will appear in the list under each certificate filename to help you manage your certificates.

Deleting a certificate removes it from the list and prevents future access by tool orchestration components. However, removing a certificate will not remove the certificate from any in-progress orchestration-related activity.

Project Secrets

This section allows you to manage data that you can share with one or more tool orchestration components that may require account credentials, keys, or other types of sensitive data.

Click Add New Secret to start generating data for your secret. Specify a name, and click OK to define your list of fields.

To add a field, click Add Field. To add a sensitive field, click Add Sensitive Field. Specify a name for the field, and click OK.

With sensitive data entry, your value is masked as you type, and you must confirm the correct value by entering it twice. Also, sensitive values are write-only and cannot be retrieved from the API of the Tool Orchestration Service. When you have finished specifying fields and field values, click Save to store your secret.

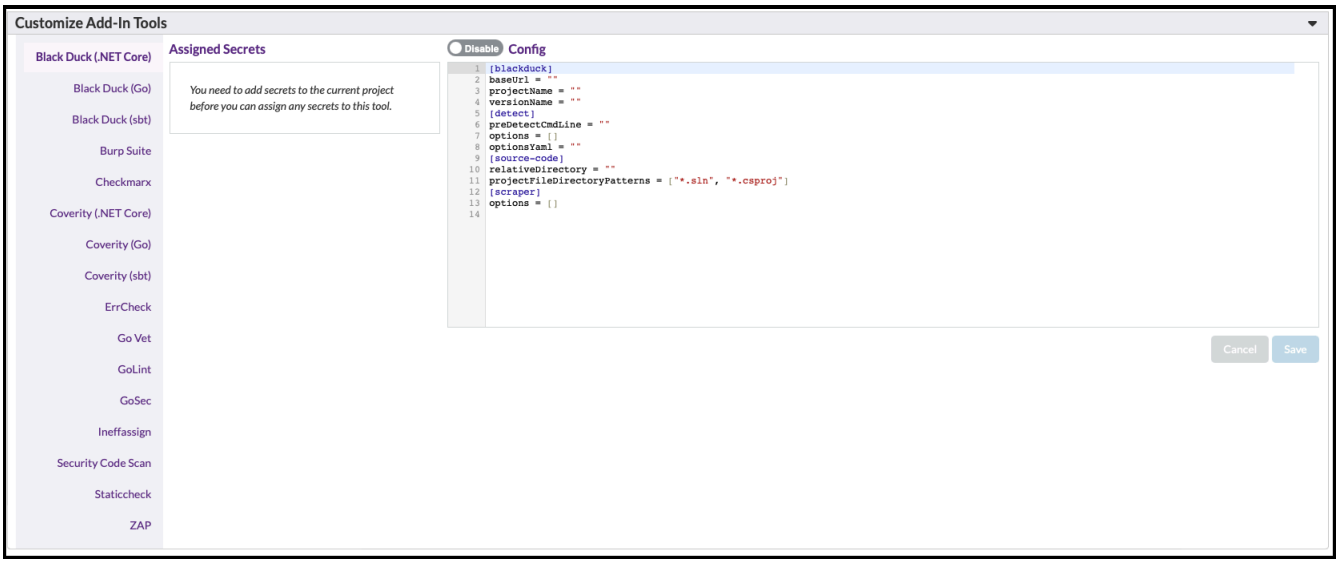
Project secrets get stored as Kubernetes Secrets, so it's recommended that you follow the Kubernetes guidance on encrypting secret data at rest (https://kubernetes.io/docs/tasks/administer-cluster/encrypt-data/).

You can edit field values at any time. Project secrets support limited editing: you cannot change a secret's name, add or remove fields, or change the data entry mode for a field value.

Project secrets are meant to be used with add-in tools, so Software Risk Manager will display a warning icon to highlight those that are not yet assigned to a tool. You can learn about assigning secrets to tools in the Customize Add-In Tools section below.

Customize Add-In Tools

This section lets you customize tools that you previously registered on the Add-In Tool page. Software Risk Manager shows your list of registered tools on the left, and you can select a tool to enable/disable it, assign one or more project secrets, or adjust tool behavior by changing any custom TOML configuration data the tool can read.



What you can customize will vary by tool. For example, the default Security Code Scan tool has no project-specific TOML configuration, and it does not read project secrets, but the Enabled/Disabled toggle lets you customize whether it is available for your project.

Alternatively, the ZAP tool allows you to use project secrets to make authenticated requests during scanning. The interpretation of project secret data is tool dependent—ZAP, for example, will ignore any secrets with missing username or password fields.

The Config box shows any project-specific TOML configuration for the tool you selected.

You should avoid using the *Default enabled* feature for tools with project-specific TOML configuration unless you have a process for specifying project configuration before a tool runs. Tools with insufficient configuration details typically fail when run.

Remember to click Save to keep changes, and you must save any changes you want to keep before selecting another tool.

Customize Checkmarx Add-In Tool

The Software Risk Manager Checkmarx Add-In has the following project-specific configuration:

```
[checkmarx]
baseUrl = ""
projectId = 0

[scan]
checkScanStatusDelay = 60
```

Use the Customize Add-In Tools feature to specify values for the above configuration before enabling the tool. Refer to the following table for an explanation of the configuration parameters.

Table 6:

Parameter	Description	Example
`baseUrl`	The base URL endpoint for the Checkmarx scanner (default="")	"https://cxprivatecloud.checkmarx.net"
`projectId`	The Checkmarx-assigned ID of a project created by the Checkmarx software at the base URL (default=0)	any integer value greater than 0
`checkScanStatusDelay`	The delay in seconds between requests to fetch scan status (default=60)	60

Note that you may have constant values for both `checkScanStatusDelay` and `baseUrl`, so you can specify values that will not vary by project.

You must also provide an account credential that authorizes use of the Checkmarx software at the base URL. The Checkmarx Add-In Tool expects to find a project secret named checkmarx-project-credential that includes both a username and a password field. The credential must grant permission to start a new scan in the configured Checkmarx project and to generate a Checkmarx report with scan findings.

Customize ZAP Add-In Tool

The ZAP Add-In has the following project-specific configuration.

```
[context]
target = ""

[scanOptions]
```

```
runActiveScan = false

[reportOptions]
minRiskThreshold = 0
minConfThreshold = 0

[authentication]
type = "none"
loginIndicatorRegex = ""

[formAuthentication]
formURL = ""
formUsernameFieldName = ""
formPasswordFieldName = ""
formAntiCrossSiteRequestForgeryFieldName = ""
formExtraPostData = ""

[scriptAuthentication]
authenticationScriptContent = ""
```

Use the Customize Add-In Tools feature to specify values for the above configuration before enabling the tool. Refer to the following table for an explanation of the configuration parameters.

Table 7:

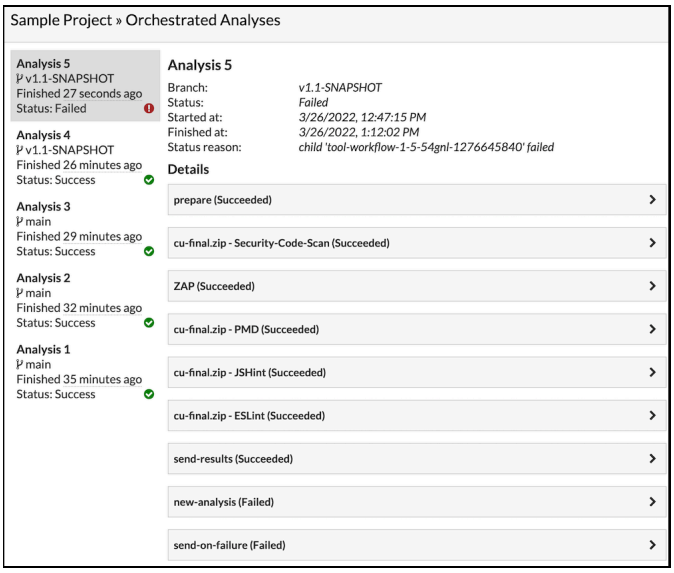
Parameter	Description	Example
target	The URL where the scan starts (default="")	the ZEST script for script authentication (default="")
runActiveScan	The decision to run an active scan (default=false)	true
minRiskThreshold	The minimum risk code for ZAP report findings (default=0)	1
minConfThreshold	The minimum confidence for ZAP report findings (default=0)	1
type	The authentication type: none, formAuthentication, or scriptAuthentication (default=none)	formAuthentication
loginIndicatorRegex	The regex to indicate a successful login request (default="")	'\QSet-Cookie: .AspNetCore.Identity.Application=\E'
formURL	The URL of the login form for forms authentication (default="")	"http://host.docker.internal/contosou/account/login"
formUsernameFieldName	The login form's username field name (default="")	"Email"
formPasswordFieldName	The login form's password field name (default="")	"Password"
formAntiCrossSiteRequestForgeryFieldName	The anti-CSRF token field name (default="")	"__RequestVerificationToken"

Parameter	Description	Example
formExtraPostData	The extra data to include with login request (default="")	"RememberMe=false"
authenticationScriptContent	The ZEST script for script authentication (default="")	See ZAP documentation

When you have ZAP authentication configured, you can provide account credentials by creating project secrets that include both a username and a password field. The ZAP scanner will send authenticated requests using each credential it finds. Be sure to specify the correct username and password with each credential so that ZAP can log on successfully.

Orchestrated Analysis

When the Tool Orchestration Service is enabled, the Orchestrated Analyses page can be used to view the portion of an analysis that's orchestrated on your Kubernetes cluster. Keep in mind that a Software Risk Manager analysis may include bundled tools for which Kubernetes support is unavailable—the Orchestrated Analysis page will not include information about those tools.



Every orchestrated analysis for a project will appear in a list on the left.

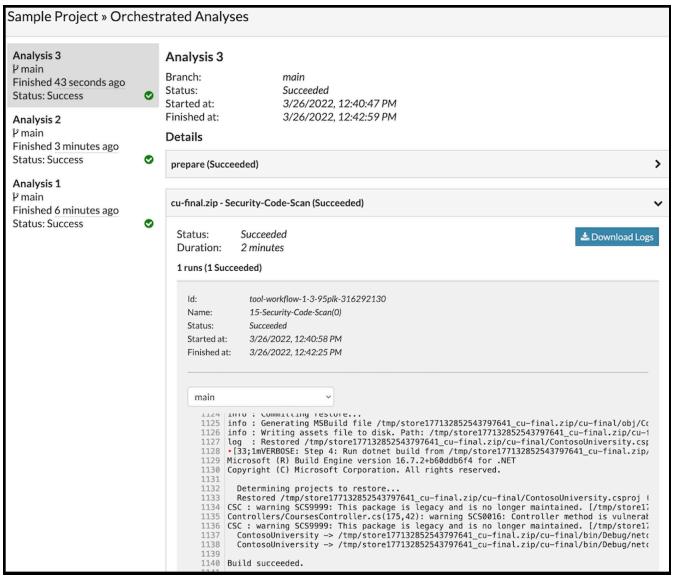
Software Risk Manager will automatically select the most recent analysis when you visit the page. Selecting an analysis lets you see summary information to include analysis status and start time. An orchestrated analysis has a unique, numerical identifier and executes as a multi-step workflow running on Kubernetes. Under the status information, you will find collapsed sections that represent each workflow step. Steps labeled with an ID and tool name represent tools running on your cluster. They differ from system steps, like prepare, which support the overall workflow.

An orchestrated analysis that completes successfully will show a *Success* status, also represented with a green checkmark icon in the analysis list. Failed analyses will show a *Failed* status and a red exclamation icon. The summary information for failed analyses will include a *Status reason* field that may provide further information. Failed steps may also include a *Message* field describing why a step failed to complete successfully.

Orchestrated analyses abandoned by previous Software Risk Manager instances continue to run to completion. Software Risk Manager will display a message when there's an orchestrated analysis whose results will be entered into Software Risk Manager as a brand new analysis.

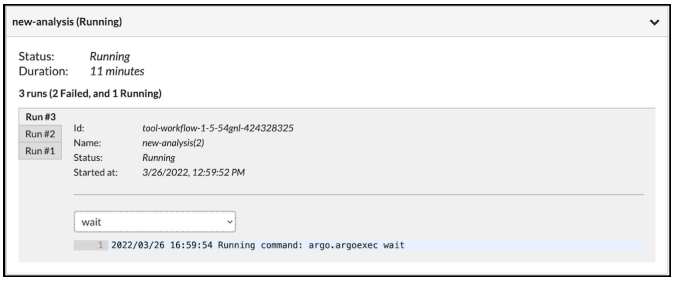
Viewing Logs

Every workflow step includes one or more logs. You can expand a step section to reveal a log viewer with support for live updates showing log data available from the Kubernetes API. Software Risk Manager shows you the *main* log by default, but you can view log data from other available sources using the dropdown shown below.



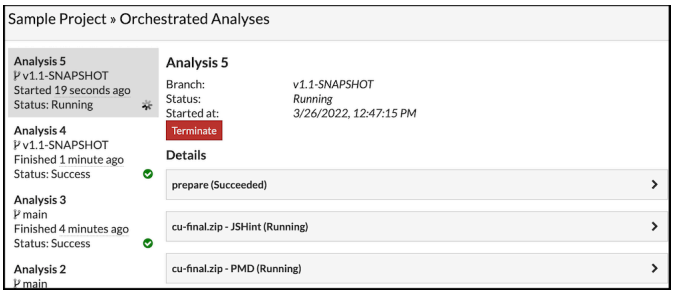
For tool completed steps, you can click Download Logs to fetch every tool log referenced by the tool's registration data. The Download Logs option will be unavailable when a tool run is in progress. Keep in mind that add-in tool authors may write log data that's unavailable via the Kubernetes API, so downloaded logs may include data that's not included in what's shown with live updates on the Orchestrated Analysis page.

Some steps of an orchestrated analyses may repeat in an attempt to recover from unexpected failures. How often they repeat and with what delay in between is step dependent. When log data is available for multiple tries, a "tabbed" log viewer will be displaed. Each tab will show you the log details for a specific attempt.



Termination

Software Risk Manager lets you stop orchestrated analyses from running to completion. Click Terminate to submit a request to cancel an analysis.



It may take a few moments before an analysis displays a terminated status, but you will see immediate feedback indicating that your termination request has been submitted, and you will not be able to submit additional termination requests.

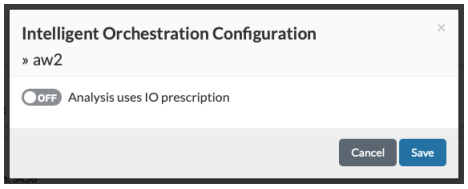
Intelligent Orchestration Overview

Intelligent Orchestration can determine which security tools are best suited for a particular analysis, depending on factors such as code changes, risk score, and security policies. In Software Risk Manager, Intelligent Orchestration is enabled by adding a pre-scan policy to a project.

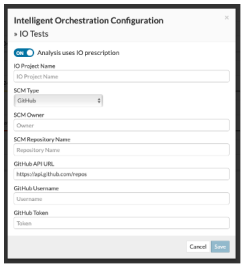
Note: Intelligent Orchestration requires a separate IO license and must be properly configured before use. For instructions, see "Enabling Intelligent Orchestration" in the *Software Risk Manager Install Guide*.

To configure Intelligent Orchestration for a project:

1. Click the Projects icon in the navigation bar to open the Projects page, then select Intelligent Orchestration from the project's dropdown configuration options.



2. Click the toggle switch to "on" to open the configuration window.



3. Enter the name of the project in the Project Name field (required).
   The IO Project Name refers to an existing project in your IO server.
4. Select the SCM Type.
   There are three options: GitHub, GitLab, and Bitbucket.
5. Enter the required configuration information according to the selected SCM type, as described below:
   Type: GitHub
   - SCM Owner
   - SCM Repository Name

- GitHub API URL
- GitHub Username
- GitHub Token

Type: GitLab

- SCM Owner
- SCM Repository Name
- GitLab API URL
- GitLab Token

Type: Bitbucket

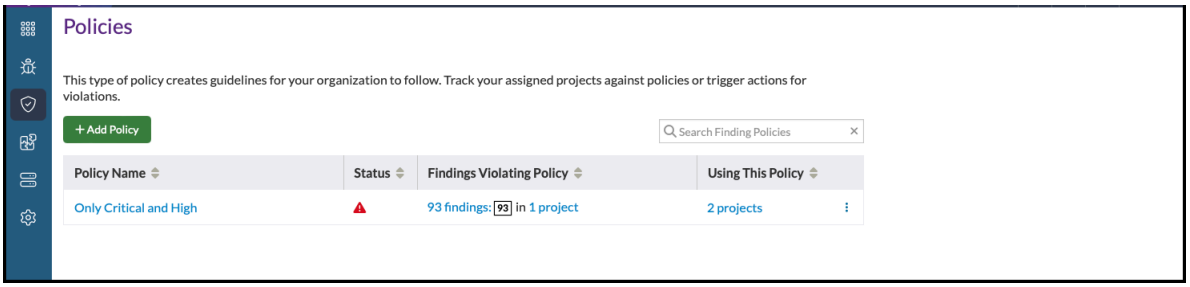
- SCM Owner
- SCM Repository Name
- Bitbucket API URL
- Bitbucket Host Type:
  Bitbucket supports three host types:
  - Cloud
  - Server
  - DataCenter
- BitBucket Username (Cloud only)
- BitBucket Password (Cloud only)
- Bitbucket API Version (Server/DataCenter)
- Bitbucket Project Key (Server/DataCenter)
- Bitbucket Username (Server/DataCenter)
- Bitbucket Password (Server/DataCenter)

6. Click Save.

Policies Overview

Policies in Software Risk Manager allow you to track compliance to specified requirements. Once defined, policies can be applied to projects, and policy violations can be monitored. In this way, polices can be used to prioritize which security issues need to be addressed, and so on.

Click the Policies icon in the navigation bar to open the Policies page.



The Policies page displays a list of all the currently defined policies, along with the following information:

- **Policy Name.** Lists the existing policies. Click the policy name to open the View Policy window, where you can view or edit the policy definition.
- **Status.** Provides a visual representation of the policy status. There are four status icons:
  - Red triangle: Fail (Overdue)
  - Orange hourglass: Warn (Due Soon)
  - Purple hourglass: Pass (On Track)
  - Green checkmark: Pass (No Violations)
- **Findings Violating Policy.** Provides the following statistics:
  - The total number of findings that violate the policy. Clicking the link opens the Findings page and lists all the findings violating the policy.
  - A color-coded breakdown of findings according to violation status. The numbers inside the colored boxes correspond to the number of findings for each category: Overdue (red), Due Soon (orange), On Track (purple), No Fix-by date (gray). Clicking a box opens the Findings page. Findings are sorted according to the corresponding Policy Violations and Policy Violation Urgency filters.
  - The number of assigned projects with policy violations. Clicking the link opens the Projects modal, which lists the projects using this policy.
- **Using This Policy.** Shows the number of projects using the selected policy. Clicking the link opens the View Policy Projects window, which lists all the projects associated with the policy.

Click the column headers to re-sort the list. You can also use the search field to search for a specific policy.

Working with Policies

For more information on policy management, see the following topics:

- Policy Configuration. How to view a policy's configuration.
- Creating and Editing Policies. How to create or edit a policy's configuration.
- Applying a Policy to a Project. How to apply a single policy to a project.
- Applying a Policy to Multiple Projects. How to apply a policy to multiple projects.
- Monitoring Policy Violations. How to track policy violations as they relate to projects, policy, and individual findings.

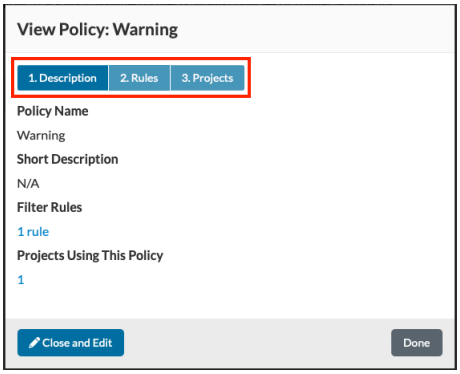
Policy Configuration

A policy consists of three parts:

- **Description.** The policy name and its purpose.

- **Rules.** The conditions that define the policy.
- **Projects.** The projects that will use the policy.

When creating, editing, or viewing policies, each part—Description, Rules, Projects—will be displayed in a separate window. Clicking the "button" tabs along the top will switch to that corresponding window. Click Done to close the window.

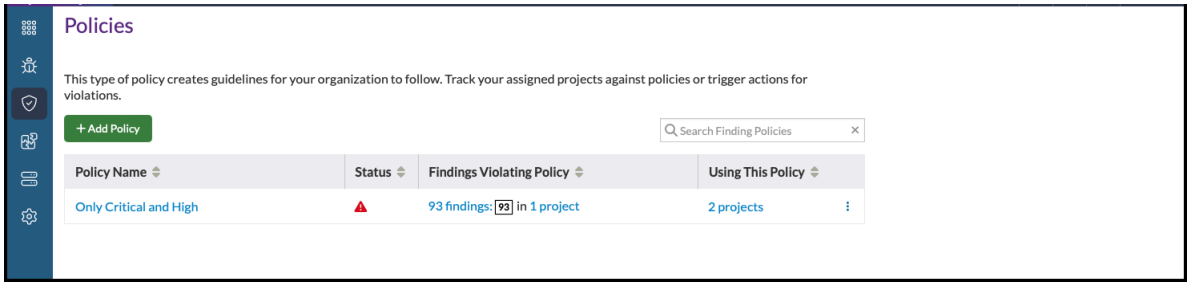


Viewing Policy Configuration

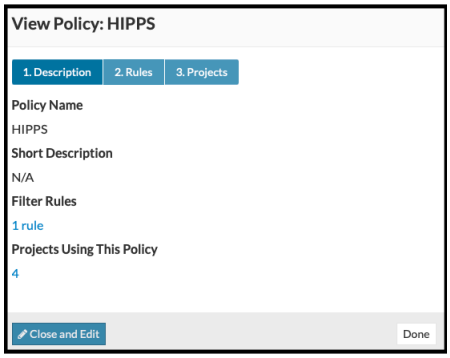
Policy definitions can be displayed from the Policies page.

To view the configuration of an existing policy:

1. Click the Policies icon in the navigation bar to open the Policies page.



2. Click a policy name to open the Policy Description window for that policy.



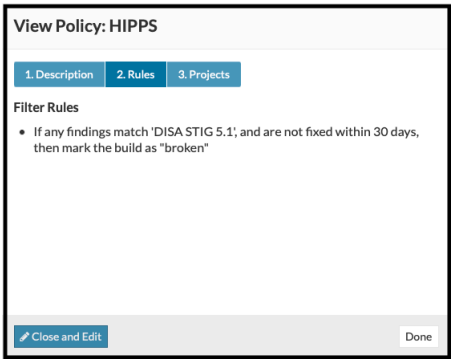
The Description window displays the following information:

- **Policy Name.** The name of the policy.
- **Description.** A description of the policy.
- **Filter Rules.** The number of rules used to define the policy. Click the link to display the filter rules. (This is the same as clicking the Rules tab.)

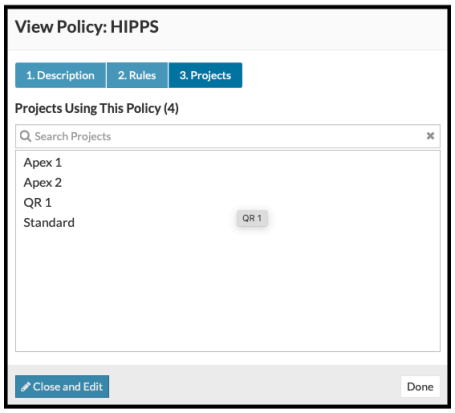
- **Projects Using this Policy.** The number of projects that use this policy. Click the link to display a list of projects that use this policy. (This is the same clicking the Projects tab.)

Note: Click Done to close the window. Click Close and Edit to open a window where you can edit the policy's configuration. For more information, see Creating a Policy.

3. Click the Rules tab to see a list of all the rules that define the policy.



4. Click the Projects tab to see a list of all the projects that use this policy.

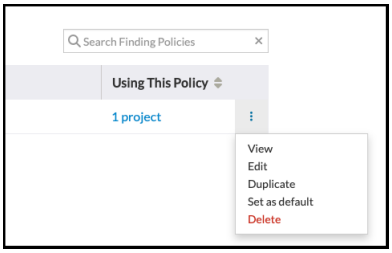


Use the search field to search for specific projects.

5. Click Done to close the window.

Additional Policy Configuration Options

Click the policy's dropdown configuration icon to display additional options:



- **View.** Opens the policy Description window (same as clicking a policy name).
- **Edit.** Opens the policy configuration page where you can edit the policy's configuration (see Creating a Policy).

- **Duplicate.** Creates a new policy with duplicate settings. This is useful when creating a new policy that includes many of the same rules as the original. Creating a policy with duplicate settings eliminates the need to recreate rules that have already been defined.
- **Set as default.** Specifies that the policy will be automatically assigned to all new projects (but not to existing ones). If a policy is a default policy, it will appear as part of the policy name. (A default policy must be "unset as the default" before it can be deleted.)
- **Delete.** Deletes the policy and all its associations with existing projects. Deleting a policy is irreversible. A warning pop-up window will list all the projects associated with the policy and ask for confirmation before deleting the policy.

Creating and Editing Policies

Policies are created and updated from the Policies page. (Note that there are often alternative ways to perform the same task. The most common are detailed below.)

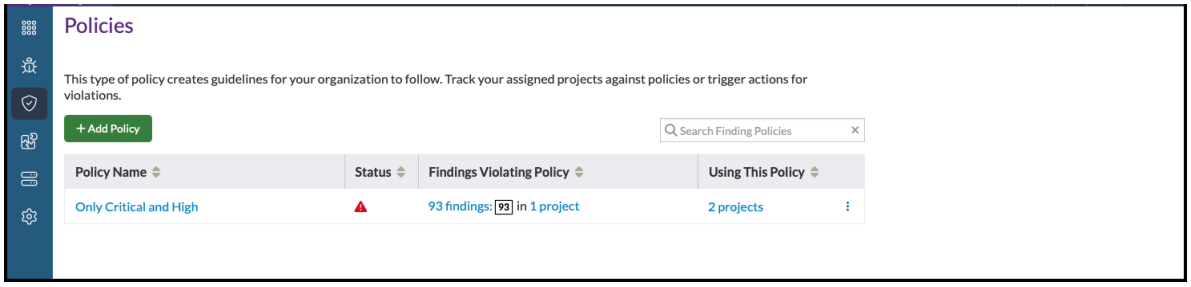
- Creating a Policy
- Editing a Policy
- Deleting a Policy

Note: Policy functionality is role-based: users assigned to the "Manager" role are limited to assigning and removing projects.

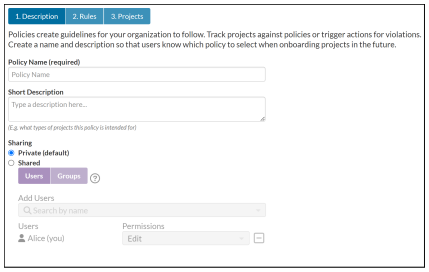
Creating a Policy

To create a new policy:

1. Click the Policies icon in the navigation bar to open the Policies page.

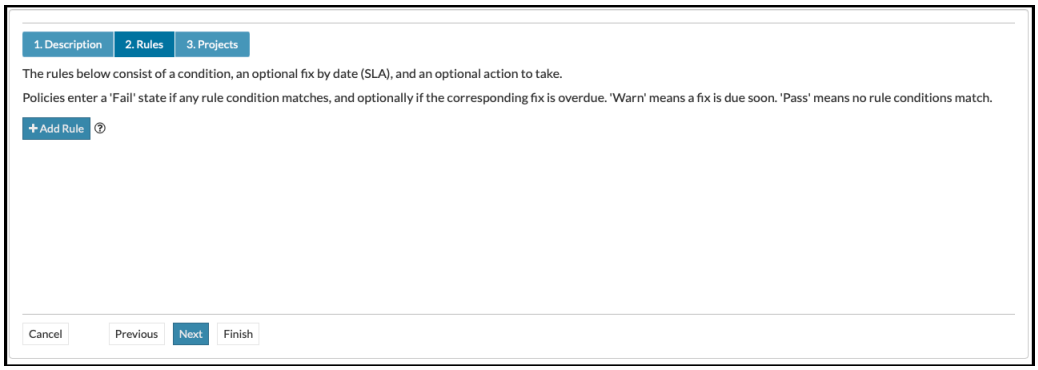


2. Click Add Policy to open the policy description window.

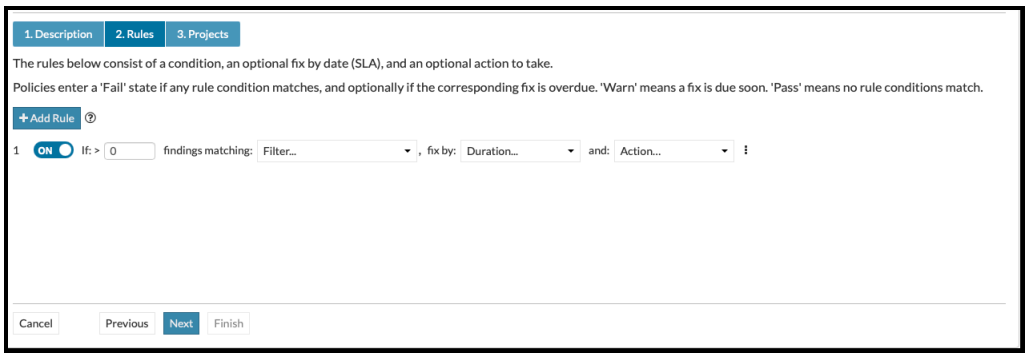


3. Enter a name (required) and description for the policy
   The policy name must be unique. A description is not required, but it is recommended.

4. Configure the sharing settings for the policy, then click Next.
   By default, a policy can only be viewed and edited by the user who creates it. Once projects are associated to the policy, any users who can view at least one of those projects will be able to view the policy.

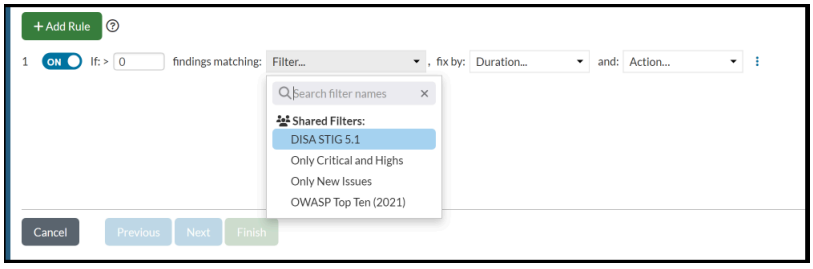


5. Click Add Rule to assign rules to the new policy.
Clicking Add Rule opens a dialog where you can configure new rules.



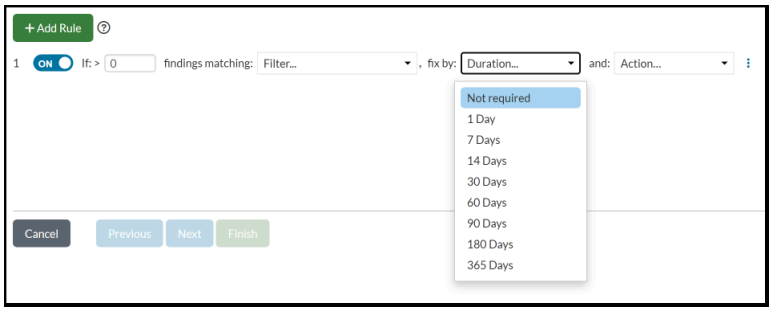
6. Use the dropdown options to configure the policy rule.
Creating a rule includes configuring the following elements:

- **On/Off toggle.** Allows individual rules to be temporarily deactivated.
- **Threshold.** Sets the minimum number of findings needed to trigger the rule.
- **Filter.** Defines which filters the policy will use. Select filters from the dropdown list. (**Note:** Private saved filters must be shared before they can be applied. For more information, see Saving Filters.) Use the search field to search for existing filters.



Note: Regarding the use of the Policy Violations and Policy Violation Urgency filters in saved private filters, if either (or both) of these filters are added to a saved filter, those options will be ignored when using that filter as a Policy rule.

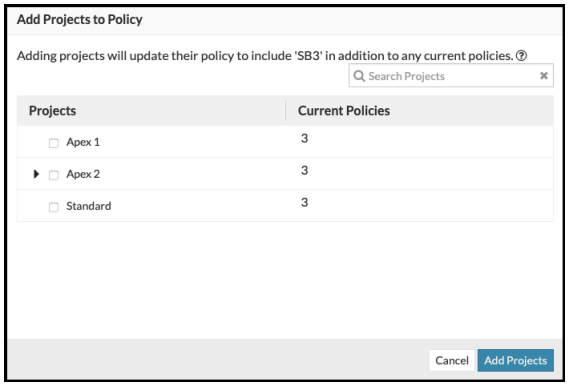
- **Fix by.** Sets the number of days before the violation status changes to overdue. Select the number of days from the dropdown list. Select "Not Required" if no date is necessary.



- **Action.** Specifies what action should be taken when the rule is violated. The options are "Nothing," "Create ticket(s)," and "Break Build."

7. Click Next to open the Projects window.

8. Click Add Project to open the Add Projects to Policy window.

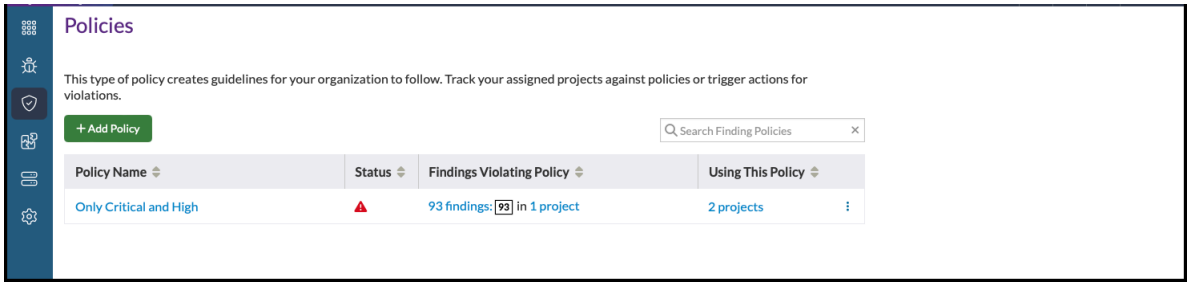


9. Select the projects to associate with this policy and click Add Projects.
   Use the checkboxes to select projects. You can search for projects using the search field. The Current Policies column shows the number of policies that have already been assigned to that project.
10. Click Finish.

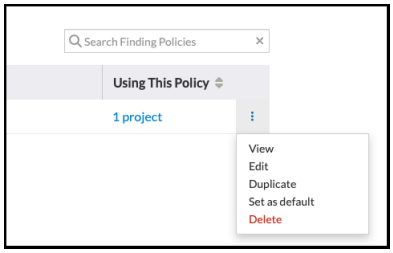
Editing a Policy

To edit an existing policy:

1. Click the Policies icon in the navigation bar to open the Policies page.



2. Click the policy's dropdown configuration icon and select Edit.

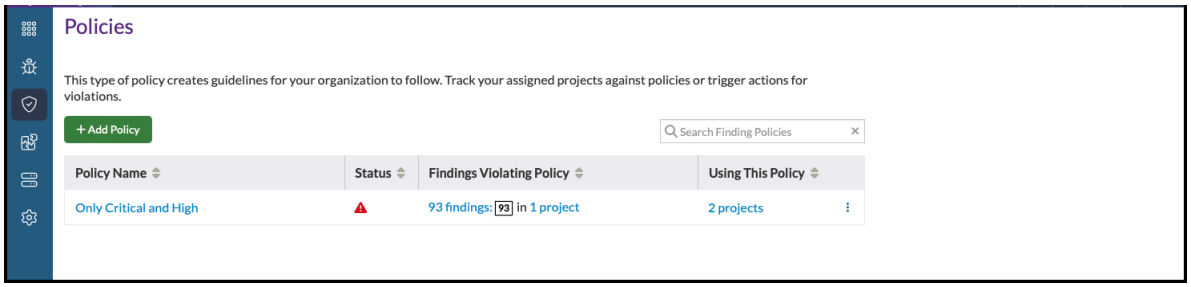


3. Make changes as necessary.
   For field definitions, see the descriptions in the Creating Policies section.
4. Click Finish.

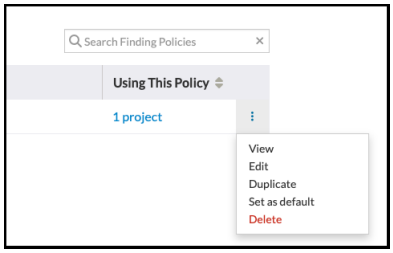
Deleting a Policy

To delete an existing policy:

1. Click the Policies icon in the navigation bar to open the Policies page.



2. Click the policy's dropdown configuration icon and select Delete.



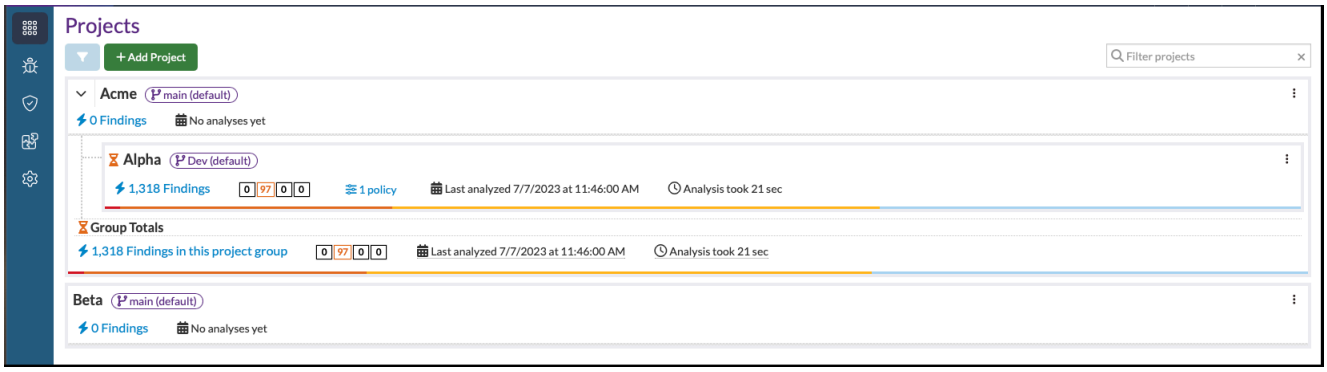
3. Click Delete to confirm.

Applying a Policy to a Project

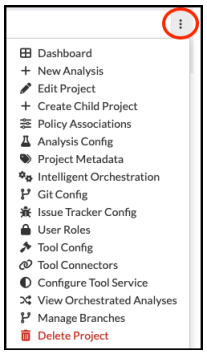
There is more than one way to apply a policy to a project. The instructions below detail the most common method of applying a single policy to a single project.

To apply a policy to a project:

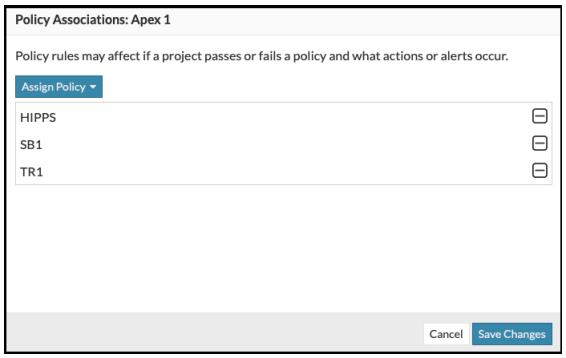
1. Click the Projects icon in the navigation bar to open the Projects page.



2. Click project's dropdown configuration icon and select Policy Associations.



This will open the Policy Associations window.



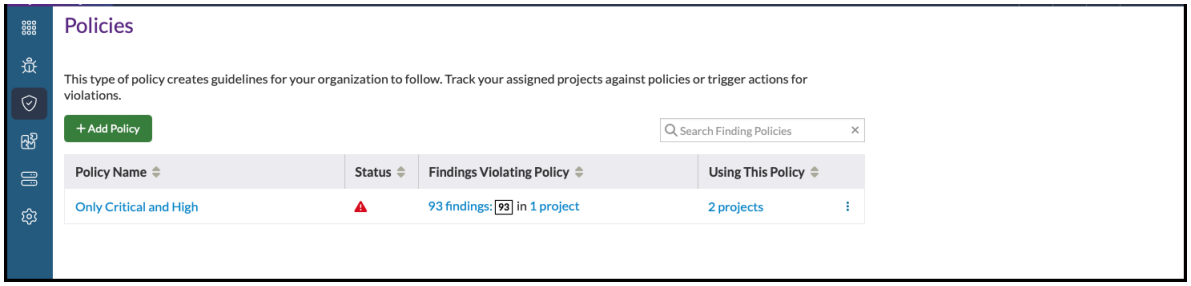
3. Click Assign Policy and select which policy(ies) to add from the list.
   Click the plus icon to select policies. You can search for policies using the search field. Clicking the minus icon next to an existing assignment will remove the policy association from the project. (For information on creating a policy, see Creating and Editing Policies.)
4. Click Save Changes.

Applying a Policy to Multiple Projects

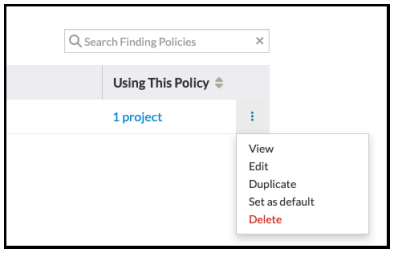
There is more than one way to apply a policy to multiple projects. The instructions below detail the most common method.

To apply a policy to multiple projects:

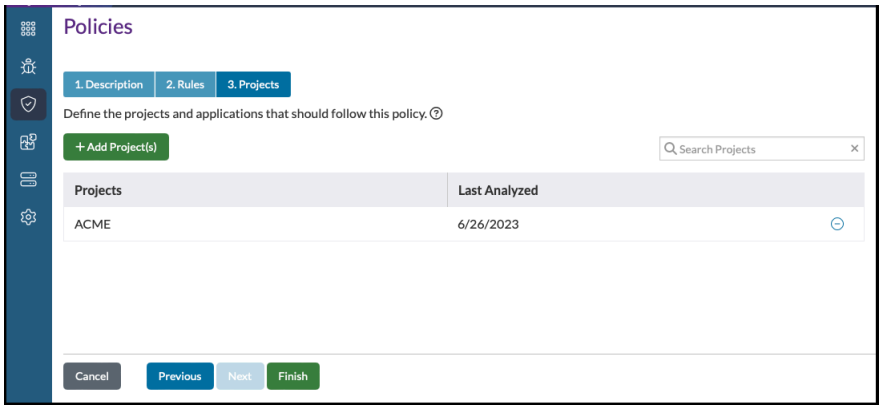
1. Click the Policies icon in the navigation bar to open the Policies page.



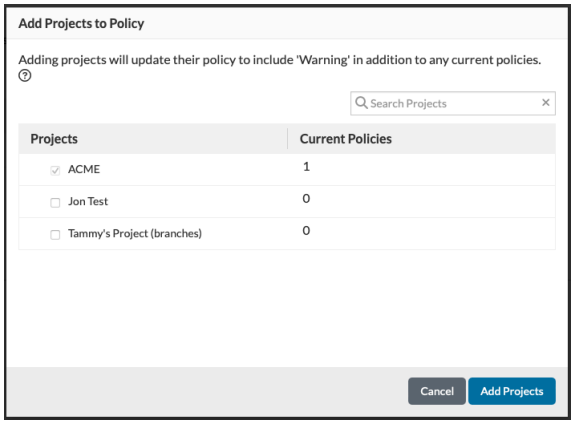
2. Click the policy's dropdown configuration icon and select Edit.



3. Click the Projects tab.



4. Click Add Projects and select the projects you want to apply this policy to.



Use the checkboxes to select projects. You can search for projects using the search field. The Current Policies column shows the number of policies that have already been assigned to that project. Clicking the minus icon removes the project association.

5. Click Add Projects.

6. Click Finish.

Monitoring Policy Violations

Once a policy has been defined and applied to a project, Software Risk Manager will track policy violations and provide violation status in a variety of ways. Policy information and violation status appears on the following pages:

- Projects page
- Findings page
- Policies page

Note: Email notifications for changes to policy violation status can be configured on the user configuration page. For more information, see User Configuration Settings. For information on customizing the policy email template, see Customizing the Policy Email Template.

Understanding Policy Violation Parameters

You can set the "duration" or number of days before a policy is violated when you create policy rules. Time-based violations are based on calendar days. Polices can be set to preselected periods, that is, 1 day, 7 days, 14 days, etc. The "fix-by" date is calculated based on the day the finding was created. For example, if a rule has been set to 7 days, the policy will show a violation 7 days after the finding was created.

Tickets will be created whenever a finding violates the `findings matching` condition; nevertheless, the threshold or fix-by date doesn't need to be reached. Consider the following example:

```
If > 100 findings matching Only Critical and Highs, fix by 14 days and Create
      Tickets
```

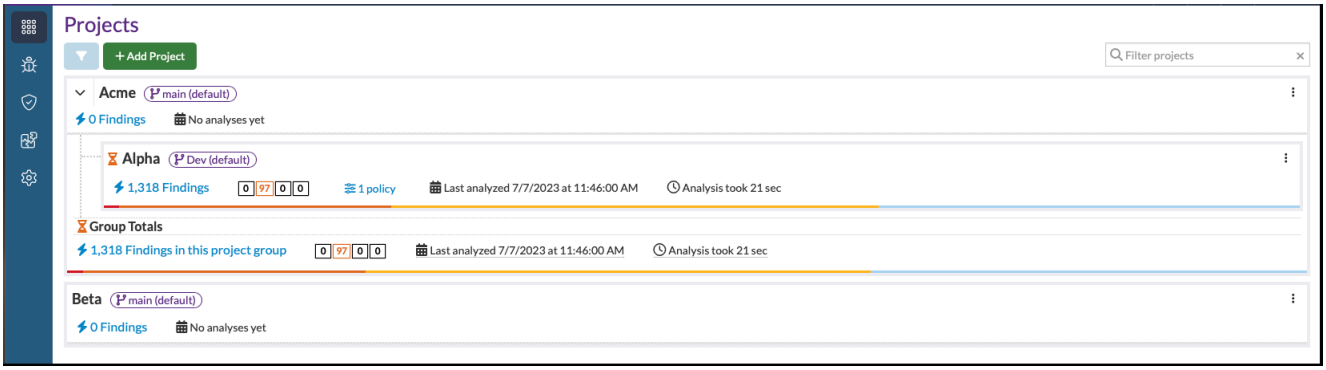
If your project only has one new critical or high finding, a ticket will be created for that finding even though the policy itself is still passing because it hasn't hit the threshold and hasn't gone over the fix-by date

When using the Policy filters, note the following range definitions:

- Due Soon is 0–7 days.
- On Track is anything over 7 days.
- Overdue occurs when the fix-by date has passed by at least one day.

Monitoring Policy Violations for Projects

Click the Projects icon in the navigation bar to view a summary of policy issues related to a specific project.



This page shows the number of policy violations for each project along with links to additional information. Policy information is displayed in the second and third columns to the right of the total number of findings

for that project. The total number of policy violations for a specific project is broken out by policy violation status, shown in color-coded boxes. Clicking a box takes you to the Findings page, where the findings have been filtered according to that status.

Policy violations are defined as follows:

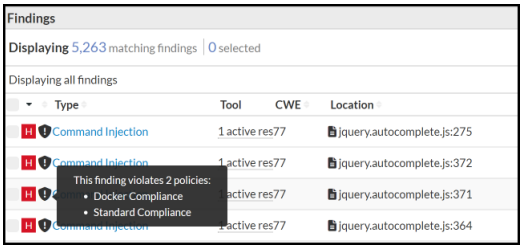
- Red: Overdue
- Orange: Due soon
- Purple: On track
- Gray: Unspecified "fix-by"

The third column shows the number of policies associated with that project. Clicking the link displays the policies associated with that project.

Monitoring Policy Violations for Findings

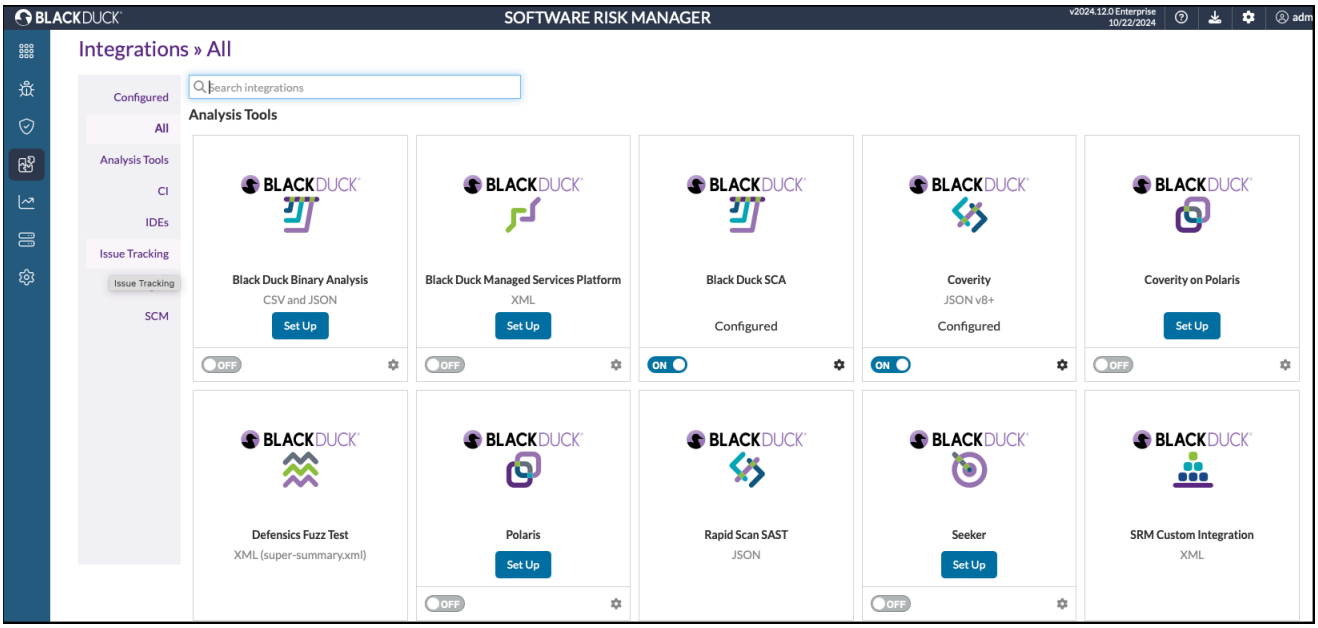
Policy violations for a single finding can be found on the Findings page. Click the Findings icon from the navigation bar to open the Findings page, then mouse over the shield icon next to the finding ID to see a summary of policy violations for that finding. The number of days specified to fix the issue is displayed in the "Fix By" column.

You can also use filters to sort findings based on policy violations. For more information, see Working with Filters.



Integrations Overview

Click the Integrations icon in the navigation bar to open the Integrations page.



The default view lists all the currently configured integration tools.

- Use the menu on the left to display supported tools arranged by tool type.
- Select All to display all the analysis tools, integration tools, IDEs, issue tracking systems, plugins, and version control tools supported by SRM.
- Select Configured to display all the currently configured integration tools.
- Enter a search term in the Search Integrations field to search for a specific tool.

Integration Tool Types

SRM supports the following types. (Click the link for additional information.)

- Analysis Tools
- Continuous Integration
- Integrated Development Environments
- Issue Tracking
- Plugins
- Source Code Management

Analysis Tools

SRM supports the following Analysis Tools:

- 42Crunch
- Acunetix Desktop (XML)
- Acunetix 360
- Anchore (JSON)
- Android Lint (XML)
- APIsec

- AppScan DAST (XML)
- AppScan Enterprise
- AppScan Source (OZASMT)
- AppSpider (XML)
- Aqua Enterprise
- Arachni (JSON and XML)
- Armorize CodeSecure (XML)
- ASoC (XML)
- AWS Security Hub (JSON)
- Azure Security Center (CSV)
- Black Duck SCA
- Black Duck Binary Analysis (CSV and JSON)
- Brakeman (JSON and ZIP of JSON outputs) (Built-in tool)
- Burp Enterprise
- Burp Suite (XML)
- C++test (XML)
- CAST Highlight
- CAT.NET (XML) (Built-in tool)
- Checkmarx (XML)
- Checkmarx IAST
- Checkmarx One
- Checkstyle (XML) (Built-in tool)
- Clang (ZIP of HTML outputs)
- Clang-Tidy (TXT: console log)
- Clippy (JSON and ZIP of JSON outputs) (Built-in tool)
- CodePeer (CSV)
- CodeSonar (CodeSonar-Scrape ZIP)
- Continuous Dynamic (formerly WhiteHat)
- Contrast
- Coverity (JSON v8+)
- Coverity on Polaris
- Cppcheck (XML v2) (Built-in tool)
- CycloneDX (JSON and XML)
- Data Theorem Mobile
- DefenseCode ThunderScan (JSON)
- Defensics Fuzz Test (XML: super-summary.xml)

- Dependency-Check (XML) (Built-in tool)
- Dependency-Track
- dotTEST (XML)
- Dynatrace
- ErrCheck (TXT: console.log)
- error-prone (TXT)
- ESLint (JSON) (Built-in tool)
- Faraday
- Fortify (FPR)
- Fortify Software Security Center
- FxCop (XML and ZIP of XML outputs) (Built-in tool)
- Gendarme (XML) (Built-in tool)
- GitHub Advanced Security
- GitLab Security (JSON) and Report (ZIP)
- GoCyclo (TXT: console log)
- Google SCC
- GoLint (TXT: console log)
- GoSec (JSON)
- Grype (JSON)
- Hacker One
- Harbor (JSON and CSV)
- Helix PRQA-QAC (CSV)
- Imperva
- IneffAssign (TXT: console log)
- Inviciti (XML)
- Inviciti Enterprise (XML: Vulnerabilities List)
- IriusRisk
- JFrog Xray (JSON)
- Jlint (TXT)
- JSHint (TXT) (Built-in tool)
- Jtest (TXT) (Built-in tool)
- Mend SCA
- Microsoft Defender For Cloud
- Microsoft Code Analysis (TXT: MSBuild log and TSV: errors table)
- Microsoft Threat Model (HTM and TM7)
- MobSF (JSON: Generate JSON Report endpoint)

- MobSF Scan (JSON)
- NDepend (XML)
- Nessus (NESSUS)
- NeuVector
- Nmap (XML)
- NowSecure
- NowSecure Workstation (JSON and ZIP of JSON outputs)
- OCLint (XML)
- Orca Security
- PHP_CodeSniffer (XML) (Built-in tool)
- PHPMD (XML) (Built-in tool)
- PMD (XML) (Built-in tool)
- Polaris
- Prisma Cloud (RedLock) (CSV and JSON: List Alerts V1 endpoint)
- Prisma Cloud Compute (Twistlock) (CSV and JSON)
- Pylint (JSON and ZIP of JSON outputs)
- Q-MAST
- Qualys CS (CSV)
- Qualys VM (XML)
- Qualys VMDR
- Qualys WAS
- Rapid Scan SAST (JSON)
- Rapid7 InsightAppSec
- Rapid7 Nexpose (XML)
- SafeSQL (TXT: console log)
- SARIF (JSON v2.1.0)
- SATE (XML)
- Scalastyle (XML) (Built-in tool)
- SCAP (XML)
- SciTools Understand (CSV)
- SD Elements
- Black Duck Seeker
- Semgrep (JSON)
- Snyk (JSON)
- SonarQube
- Sonatype Nexus

- SPDX (JSON and SPDX)
- SpotBugs (XML) (Built-in tool)
- SRM Custom Integration (XML)
- Staticcheck (JSON)
- STIG (CKL, CKLB)
- SWAMP (XML)
- Black Duck Managed Services Platform (XML)
- Tenable.io
- Tenable.io Web App Scanning
- Tenable.sc
- Black Duck Tinfoil API
- Black Duck Tinfoil Web
- Trivy (JSON: container image results)
- TruffleHog (JSON)
- Trustwave App Scanner
- Veracode (XML and ZIP)
- Vet (JSON)
- WebInspect (XML)
- Wiz
- WPScan (JSON)
- ZAP (XML)
- ZPA (JSON) (Built-in tool)

Continuous Integration

SRM supports the following CI tools (click the link for more information):

- Bamboo
- GitHub Action
- Jenkins
- Black Duck SRM API
- Black Duck Bridge CLI
- TeamCity

Integrated Development Environments

SRM supports the following IDEs (click the link for more information):

- Code Sight IntelliJ
- Code Sight Visual Studio Code
- Eclipse

- Visual Studio

Issue Tracking

SRM supports the following issue tracking tools (click the link for more information):

- Azure DevOps
- GitHub
- GitLab
- Jira
- ServiceNow

Plugins

SRM supports the following plugins (click the link for more information):

- Burp
- Splunk
- ZAP

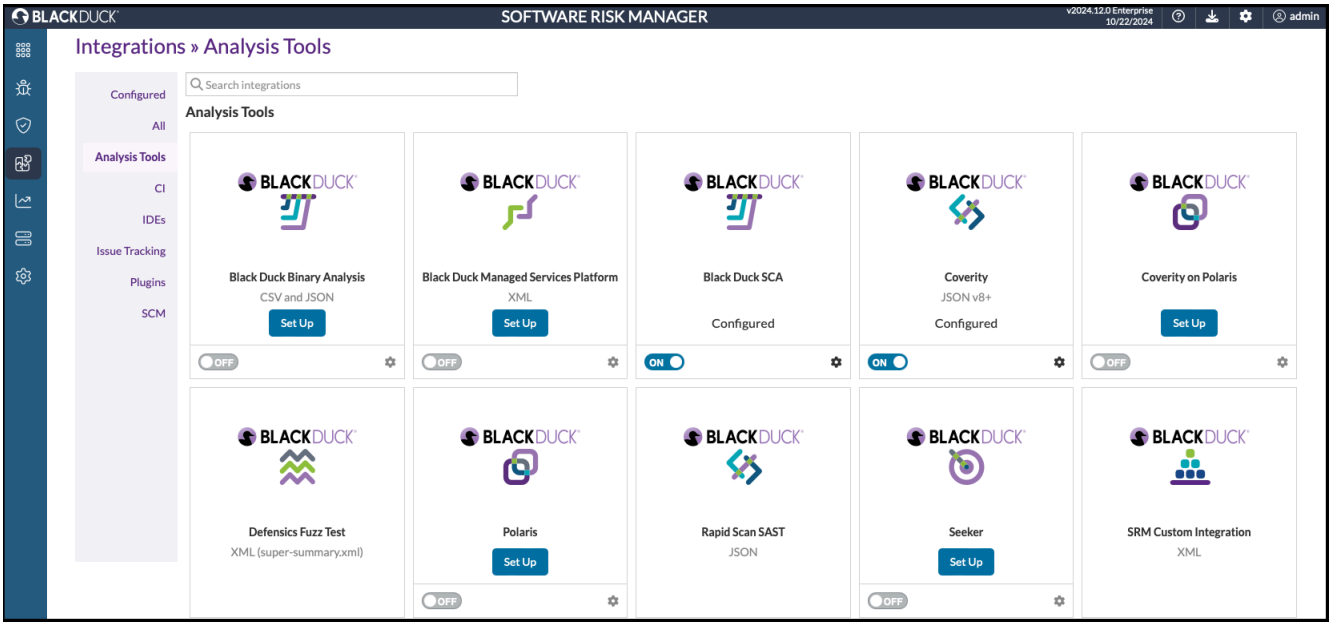
Source Code Management

SRM supports the following SCM systems (click the link for more information):

- Git

Analysis Tools

Click the Integrations icon in the navigation bar and select Analysis Tools to open the Analysis Tools page.

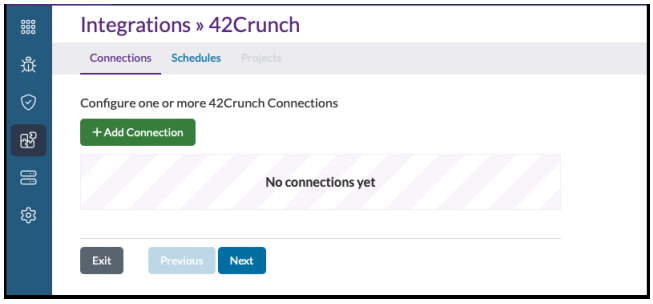


Use the Search field to search for a specific tool.

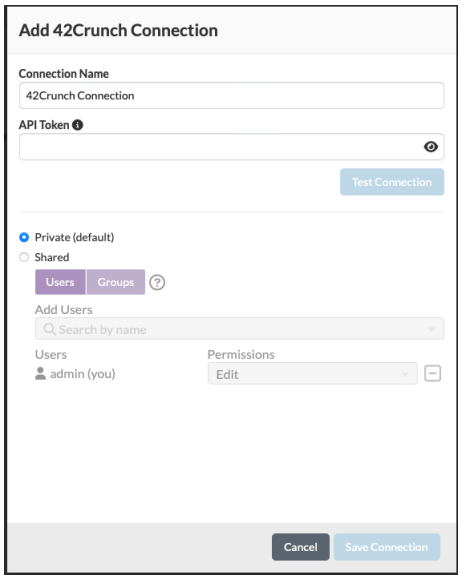
Configuring an Analysis Tool

To configure an analysis tool:

1. Click the Integrations icon in the navigation bar and select Analysis Tools to open the Analysis Tools page.
2. Click the Set Up button for the tool you want to configure.
   Not all tools are configurable.

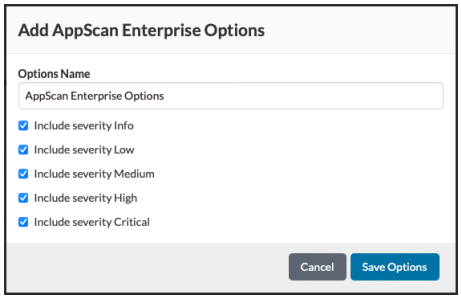


3. Click Add Connection.

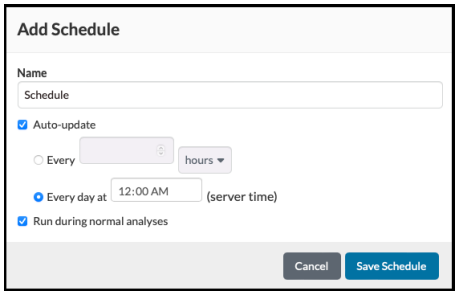


   a. Add an API token with "API Security Audit" rights in the API Token field.
      To test the connection, click the Test Connection button.
   b. Select sharing options with users or user groups.
   c. Select permissions.

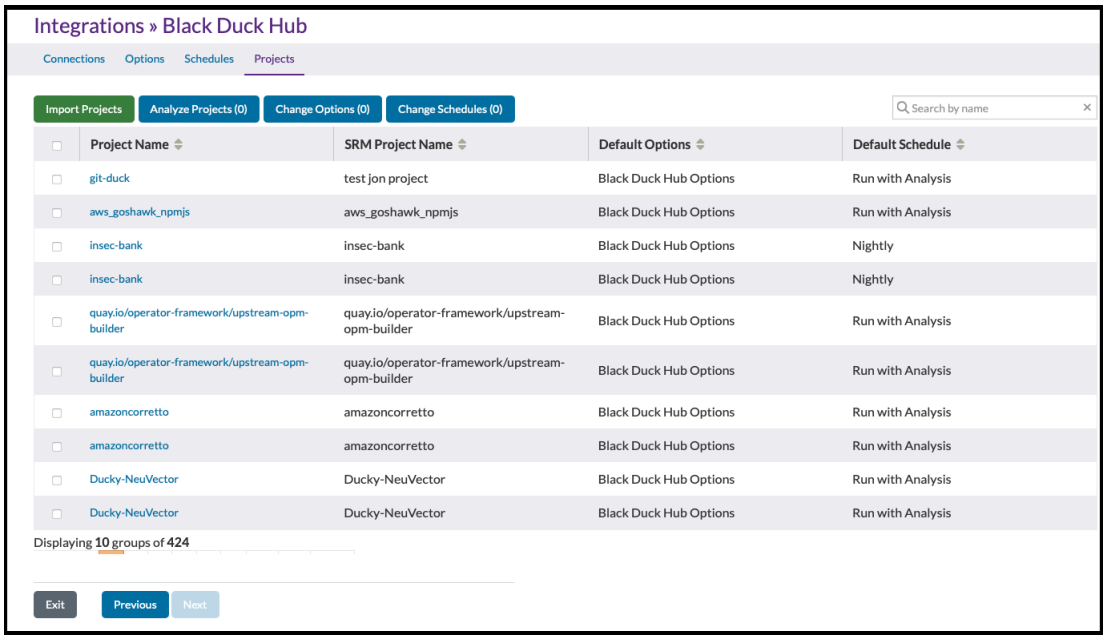
4. Click Add Options.



5. Enter a name in the Options Name field and select which options to include.
6. Click Save Options.
   Note that you can edit existing options.
7. Select Schedules from the upper menu.
8. Click Add Schedule.



9. Add projects.



Migrating Existing Tools to SRM

SRM's centralized configuration functionality allows Software Risk Manager to act as a hub for interacting with tool connectors. For Code Dx users migrating to Software Risk Manager, existing tool connectors can be converted to take full advantage of SRM's centralized configuration. (In an upcoming release of SRM, legacy tool connectors will be converted automatically, consolidating common information, such as connection URLs and credentials, into more-easily-managed shared entities.)

To migrate legacy tool connectors (Code Dx) to take advantage of centralized configuration, SRM provides two URLs that an administrator can use to perform the conversion:

- The first URL performs a "dry run" of the conversion, providing a report of all the decisions and consolidations that will be made during the actual conversion process.
- The second URL performs the conversion.

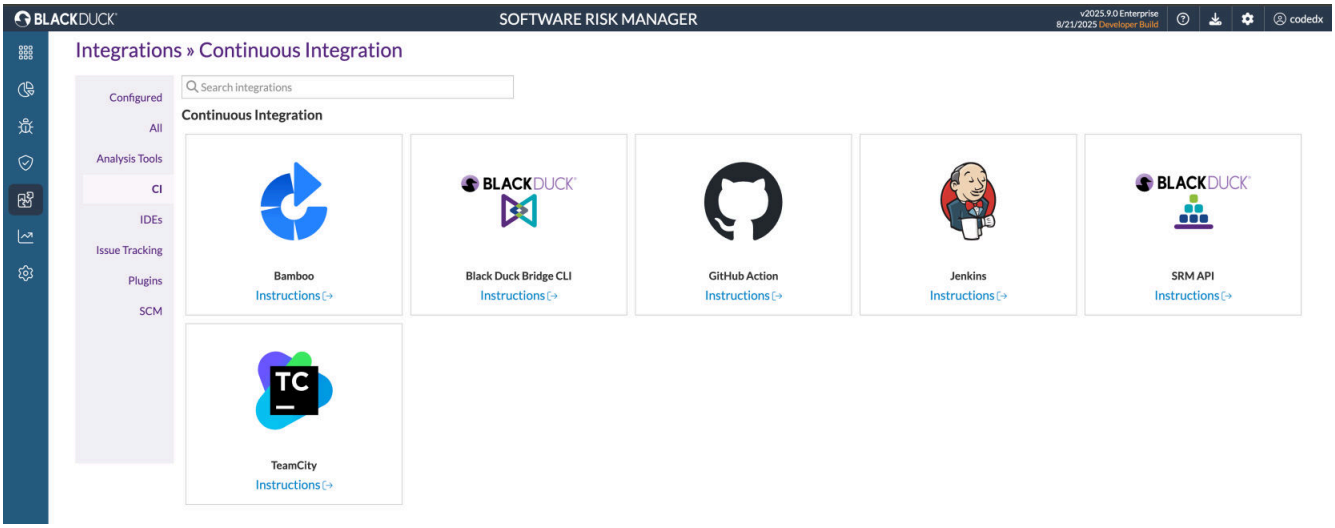
To perform the dry run and see a text-based report, log in as an admin and enter the following into your browser: `<SRM Base URL>/x/integrations/tool-connector/migration/dry-run`. You can also use your HTTP client-of-choice to send a `GET` request to that URL to obtain the report.

To trigger the actual conversion, use `<SRM Base URL>/x/integrations/tool-connector/migration/commit-run`. This URL responds with a `204 No Content` upon success.

⚠ **Warning:** Running the conversion is irreversible. Making a backup of your database prior to performing this action is highly recommended.

Continuous Integration

Click the Integrations icon in the navigation bar and select CI to open the Continuous Integration page.



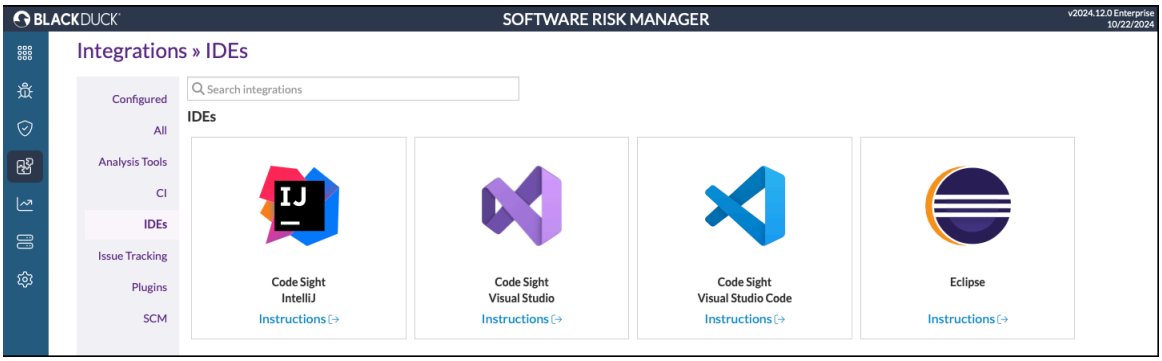
Use the Search field to search for a specific tool.

SRM supports the following CI tools (click the link for more information):

- Bamboo
- GitHub Action
- Jenkins
- Black Duck SRM API
- Black Duck Bridge CLI
- TeamCity

Integrated Development Environments

Click the Integrations icon in the navigation bar and select IDEs to open the IDEs page.



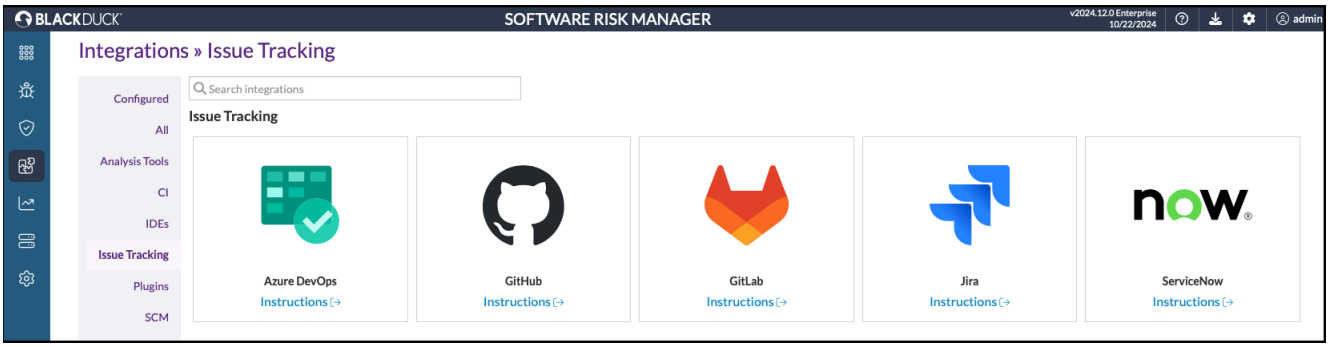
Use the Search field to search for a specific tool.

SRM supports the following IDEs (click the link for more information):

- Code Sight IntelliJ
- Code Sight Visual Studio Code
- Eclipse
- Visual Studio

Issue Tracking

Click the Integrations icon in the navigation bar and select Issue Tracking to open the Issue Tracking page.



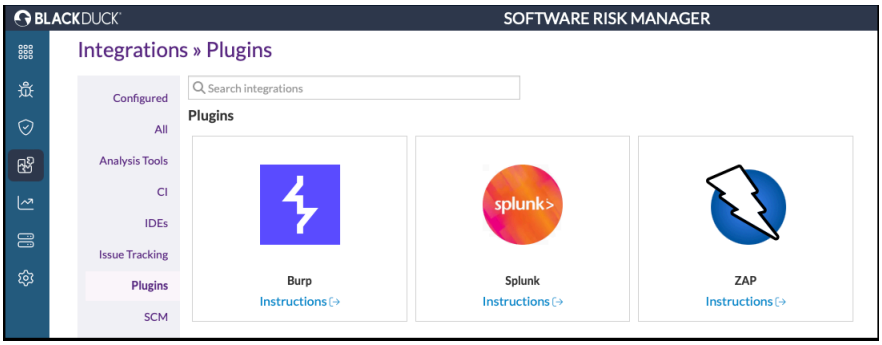
Use the Search field to search for a specific tool.

SRM supports the following issue tracking tools (click the link for more information):

- Azure DevOps
- GitHub
- GitLab
- Jira
- ServiceNow

Plugins

Click the Integrations icon in the navigation bar and select Plugins to open the Plugins page.



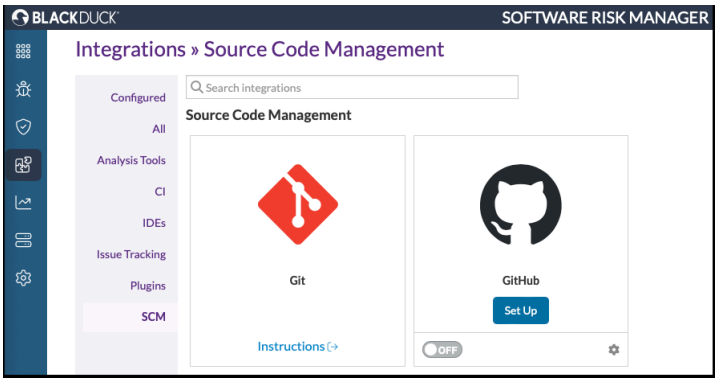
Use the Search field to search for a specific tool.

SRM supports the following plugins (click the link for more information):

- Burp
- Splunk
- ZAP

Source Code Management

Click the Integrations icon in the navigation bar and select SCM to open the Source Code Management options.



Use the Search field to search for a specific tool.

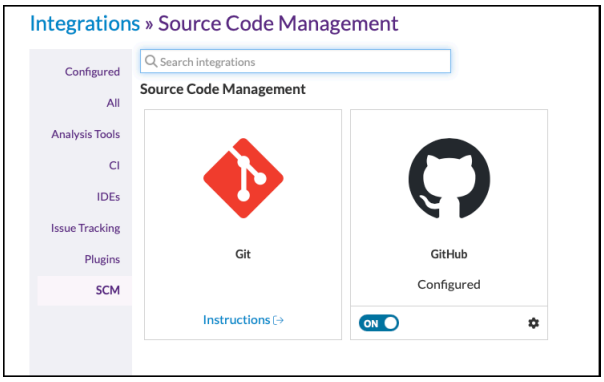
SRM supports the following SCM systems (click the link for more information):

- Git
- GitHub

Bulk Onboarding with GitHub Repositories

Software Risk Manager can automatically create projects based on existing GitHub repositories.

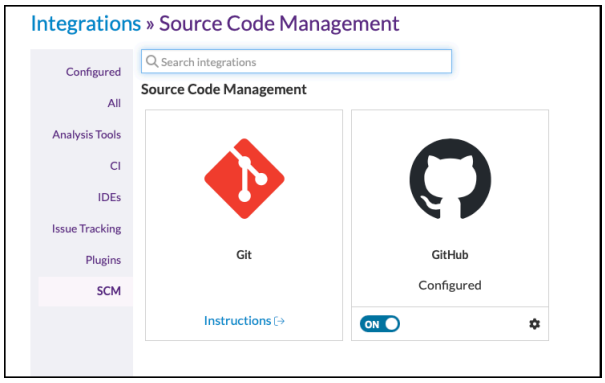
Click the Integrations icon in the navigation bar and select SCM to open the Source Code Management page.



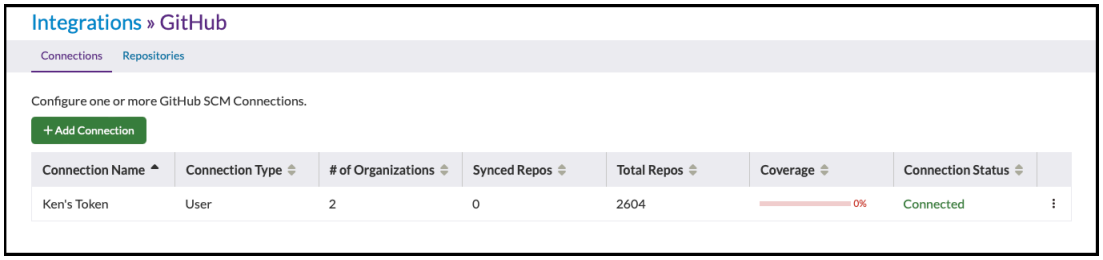
Configuring the Connection to a GitHub Repository

To configure the connection to a GitHub repository:

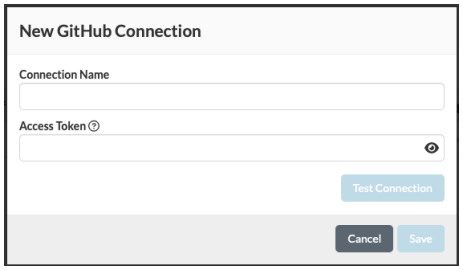
1. Click the Integrations icon in the navigation bar and select SCM to display the GitHub option.



2. Click the configuration icon for GitHub, then select the Connections tab from the top menu.



3. Click Add Connection.



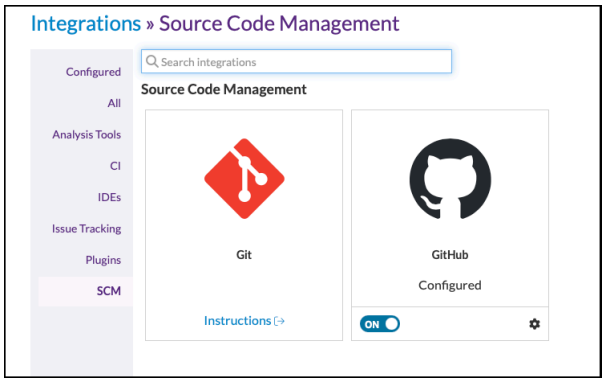
4. Enter a Connection Name and Access Token.
   Use the Test Connection button to verify proper configuration.

5. Click Save.
   With a saved connection, you can create projects from your GitHub repositories as well as associate GitHub repositories with existing projects.

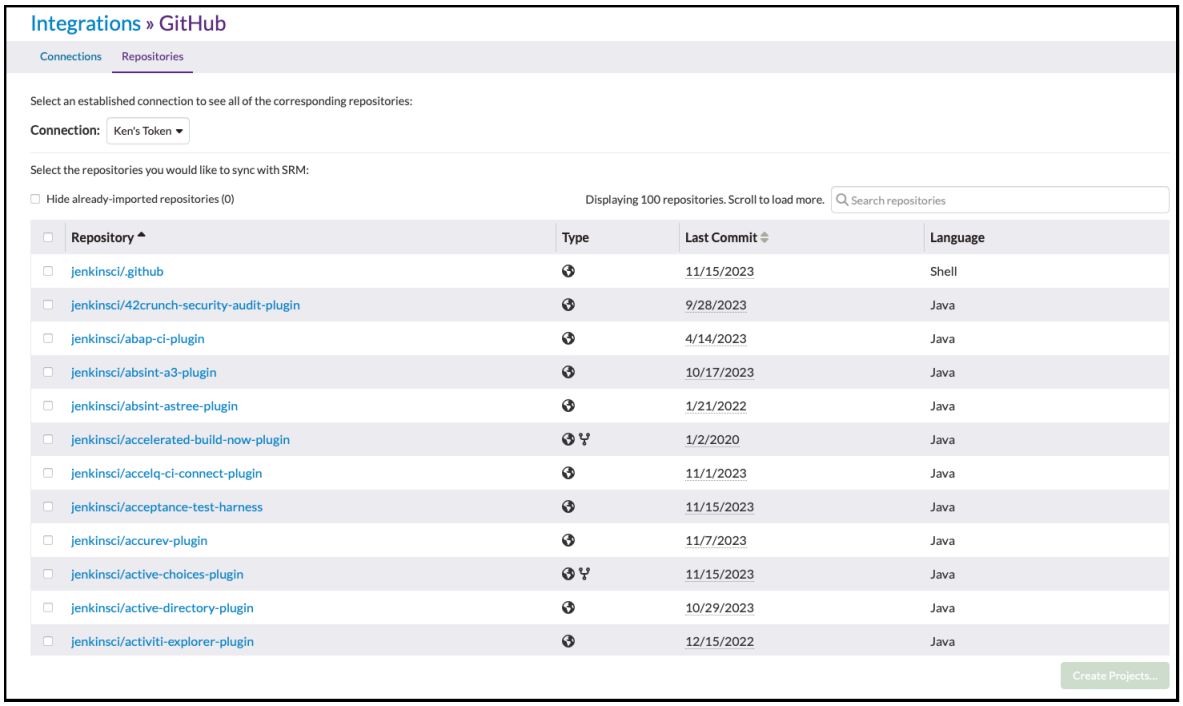
Synching GitHub Repositories with SRM

To sync GitHub repositories with SRM:

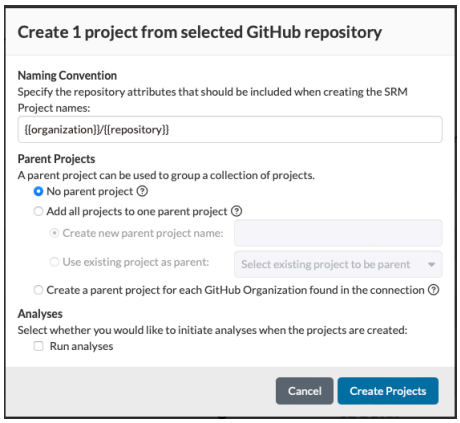
1. Click the Integrations icon in the navigation bar and select SCM to display the GitHub option.



2. Click the configuration icon for GitHub, then select the Repositories tab from the top menu.

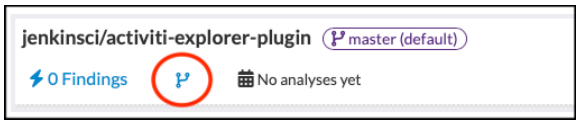


3. Select a connection from the dropdown menu.
4. Use the checkboxes to select which repositories to sync with Software Risk Manager.
5. Click Create Projects.

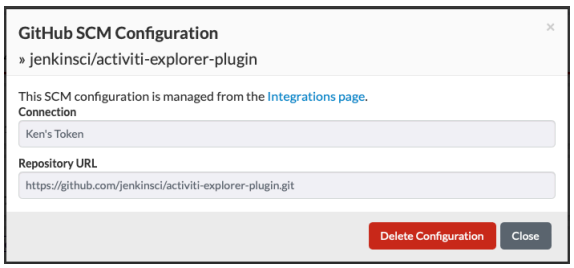


6. Specify the following parameters:
   - **Naming Convention.** The template is a https://handlebarsjs.com/ expression. The available fields are as follows:
     - `organization`. The organization name.
     - `repository`. The repository name.
     - `isPrivate`. Specifies whether the repository is private.
     - `isArchived`. Specifies whether the repository is archived.
     - `isFork`. Specifies whether the repository is a fork.
     - `languages`. A list of languages in the repository.

     Note: When the name (chosen by the naming convention) for a repository corresponds to an existing SRM Project, SRM will associate the repository with the existing project rather than creating a new project, but only if the user has the `manage` role for that project.
   - **Parent Projects.** Enables you to specify a parent project for the projects you are creating.
   - **Analyses.** Instructs SRM to run an analysis after creating the new projects.
7. Click Create Projects.
   The new projects will appear on the Projects page.

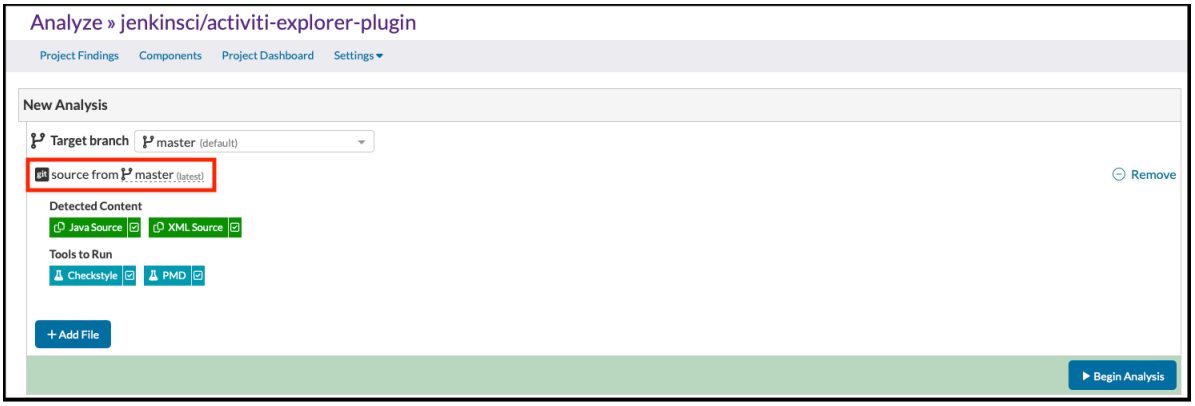


Clicking the git branch icon opens the SCM Configuration window, which displays the connection name and repository URL.



To delete the repository, click the Delete Configuration button.

For projects that are set up with a git configuration, SRM will automatically create a ZIP archive of the files from that git repo and include it in the analysis prep area as a "source from" item, as shown in the example below.



Analyses Overview

When the Scan Farm has been configured, Software Risk Manager can automatically run Coverity SAST and Black Duck SCA scans. Software Risk Manager also comes with bundled open source tools to scan a wide variety of applications. Supported languages and expected inputs for the built-in open source scanners are described in the Built-in Open Source Code Scanners and the Built-in Open Source Dependency Scanners sections. In addition to the bundled tools, Software Risk Manager can import the results of several commercial and open source tools. The supported tools and generic input formats are described in the Importing Scan Results section. There are a number of different options to configure and run analyses for Software Risk Manager: manually using the web interface, Jenkins plugins, or automatically (such as from your continuous integration server) using the API or using Black Duck Bridge CLI. These are all detailed in the Starting Analyses section.

Incremental Analysis

Software Risk Manager performs analyses incrementally. This means that as new analysis inputs (files) are added to a project, any findings associated with them are added to the project.

The life of a finding is tied to the inputs in which it was reported. When the last input contributing to a finding is archived, the finding itself is marked as "Gone" and hidden by default (see Findings View Options).

Analysis inputs can be archived manually or automatically. For more information on archival, see Auto-Archival.

Built-In Open Source Code Scanners

Software Risk Manager bundled open source code analyzers can analyze C/C++, Objective-C, Java, JavaScript, JSP, .NET (C#, VB), PHP, Terraform, Docker, Swift, Scala, Python, Ruby on Rails, and Rust applications. For all supported languages, Software Risk Manager will analyze the source using open source bundled tools built specifically for a target language. For applications built with any combination of the supported languages, Software Risk Manager will run the appropriate checkers on the provided source.

Note: The Software Risk Manager built-in open-source code scanners are turned off by default and can be turned on from the Integrations page. For more information, see the Integrations Overview section.

For Java applications, Software Risk Manager bundled open source code analyzers supports scanning compiled bytecode. In fact, the preferred approach for Java projects is to upload **both** source and bytecode to Software Risk Manager in the supported file format described in the bullets below. This yields the best coverage for issue detection.

For .NET applications, Software Risk Manager supports scanning compiled DLLs. It is also recommended that the source be uploaded. This will provide better source location information and will allow for viewing the source while looking at finding details.

Note: If you choose to upload an entire Visual Studio solution folder, there may be duplicates of the build DLLs and third-party DLLs. This will cause a longer analysis time and possibly incorrect results if some DLLs are stale. To achieve the best results, upload a zip that contains only the DLLs and PDB files for the binaries you wish to analyze. Upload the source as a separate zip.

Accepted zip Archive Formats

Software Risk Manager accepts application inputs in the following zip archive formats for running bundled open source tools:

- **C/C++.** `.zip` containing C/C++ source files that will be analyzed by Software Risk Manager bundled tools. Software Risk Manager will scan the `.zip` file for `.h`, `.c`, `.hpp`, and `.cpp` files. *(Note: If your project contains Objective-C source files then `.h` files will be treated as Objective-C rather than C/C++).*
- **Java source.** `.zip` containing Java source files – with a .java extension – to be analyzed by the Software Risk Manager bundled tools.
- **Java bytecode.** `.zip` containing `.class` or `.jar` bytecode files intended for the JVM.
- **.NET.** `.zip` containing C# or VB.NET source files – with a .cs or .vb extension.
- **.NET DLLs.** `.zip` containing compiled `.dlls`. You must also include the PDB files for `.dlls` you wish to scan. Software Risk Manager will only scan `.dlls` with corresponding PDB files – unless there are no PDB files, in which case Software Risk Manager will scan all `.dlls` but source location information may be sub-optimal.
- **iOS.** `.zip` containing `.ipa` files. *(Note: This detection is only for associating add-in tools with these files).*
- **Windows UWP.** `.zip` containing `.appx` files. *(Note: This detection is only for associating add-in tools with these files).*
- **Ruby on Rails.** `.zip` containing Ruby source files that are inside an `app/` directory.
- **PHP.** `.zip` containing PHP source files.
- **PL/SQL.** `.zip` containing PL/SQL source files.
- **Python.** `.zip` containing Python source files.
- **JavaScript.** `.zip` containing .js files; minified JavaScript will be ignored.
- **Scala.** `.zip` containing .scala files.
- **Swift.** `.zip` containing `.swift` files.
- **Objective-C.** `.zip` containing `.m`, `.mm`, `.M`, `.h` files. *(Note: '.h' will only be detected as Objective-C if there are other Objective-C file types in the '.zip').*
- **Terraform.** `.zip` containing `.tf` files.
- **Docker.** `.zip` containing `Dockerfile` files. Note that this has no extension.
- **Rust.**.`zip` containing Rust project files. Software Risk Manager will scan the `.zip` archive for the `Cargo.toml` file (which is the manifest for Rust projects) and `.rs` source files.

Note: Software Risk Manager enforces a single source .zip archive per analysis. Although Software Risk Manager supports multiple languages, the expectation is that they will all be packaged in a single `.zip`archive to enable consistent path correlation across all the checkers. And while source and bytecode inputs can be uploaded in separate files, they do not have to be split up. A single `.zip` file containing C/C++ source, Java source, Java bytecode, .NET DLLs, .NET source, PHP source, Scala source, Ruby on Rails source, Python source, JavaScript source and Rust source is perfectly acceptable.

Bundled Open Source Tool Versions

The bundled tool versions are listed in the table below.

Table 8:

Tool	Version	Release Date
Brakeman	5.2.0	12/16/2021
Checkstyle	10.26.1	07/28/2025
Clippy (user-installed)		
Cppcheck	2.9	8/28/2022
Dependency-Check	12.1.6	9/24/2025
ESLint	8.54.0	11/17/2023
FxCop (user-installed)	10+	Not available
Gendarme (not available in containerized deployments such as Docker Compose or Kubernetes)	2.11.0	Not available
JSHint	2.13.5	7/8/2022
PHP CodeSniffer	3.7.1	6/18/2022
phpcs-security-audit	2.0.1	8/5/2019
PHPMD	2.13.0	9/10/2022
PMD	7.15.0	07/24/2025
Pylint	2.4.4	11/13/2019
Scalastyle	2.12–1.0.0	8/20/2017
SpotBugs	4.9.3	07/25/2025
SpotBugs Find Security Bugs	1.14.0	07/25/2025
ZPA CLI	1.2.0	12/19/2021

Built-In Open Source Dependency Scanners

Software Risk Manager also scans input to check for dependencies with known vulnerabilities.

The following dependencies are checked:

- **Java:** `.jar` and `.war` files in Java projects.
- **.NET:** `.exe` and `.dll` files in .NET projects.
- **JavaScript** files are checked by name or a hash of the file (minified JavaScript incorporated into a different source file will not be checked).

Importing Scan Results

Software Risk Manager supports importing the results of commercial and open source application security testing tools as well as a couple of generic tool result listing formats. The list of supported tools for scan imports includes the built-in ones mentioned in the previous section. If one of the tools you want to import is not supported, please let us know. However, in the meantime, you can convert your data to the generic *SRM Input XML* format. The schema definition for this format and an example can be accessed via the download icon in the Software Risk Manager header.

Note: Some tools will output empty files if no results were found, which cannot be detected by Software Risk Manager as any particular format. For more information on empty or undetected tool results, see Empty/Undetected Tool Results.

Supported Tools

Software Risk Manager supports various tools and tool types, including the following:

- SAST tools
- DAST tools
- IAST tools
- Mobile tools
- InfraSec tools
- Threat Modeling tools
- Component tools
- Container tools
- Cloud Infrastructure tools
- Bug Bounty tools
- Infrastructure as Code (IaC)
- Web Application Firewall (WAF)
- Security Technical Implementation Guide (STIG)

Additional Support for Selected Tools

Software Risk Manager also provides additional support for the following tools:

- AppDetective Pro
- AppSpider
- Aqua
- CodeSonar
- Dynatrace

- Helix QAC
- Parasoft
- Prisma Cloud Compute (Twistlock)
- SARIF
- SBOM Files

SAST Tools

Software Risk Manager supports the following SAST tools and import formats:

- **42Crunch:** a *Tool Connector* for Security Audit scans.
- **Android Lint:** `.xml` and `.zip`.
- **Armorize CodeSecure:** `.xml`.
- **Brakeman** is a built-in scanner; `.json` is also supported.
- **Checkmarx:** `.xml` and a *Tool Connector*.
- **Checkmarx One:** a *Tool Connector*.
- **Checkstyle** is a built-in scanner; `.xml` is also supported.
- **Clang:** `.zip` containing one or more `.html` or `.plist.html` (CodeChecker) files. (Clang outputs one HTML file per checked source file.)
- **Clippy (user-installed)** is a built-in scanner; `.json` is also supported.
- **CodePeer:** `.csv` reports.
- **CodeSonar-Scrape:** see the *CodeSonar-Scrape utility* section for details.
- **Continuous Dynamic (formerly WhiteHat):** a *Tool Connector*.
- **CppCheck** is a built-in scanner; v2 `.xml` is supported as well.
- **Coverity:** `.json` using the `cov-format-errors` command line tool. For example: `cov-format-errors --dir /tmp/idir --json-output-v10 file.json`. When Scan Farm is configured, Coverity is available as a built-in tool.
- **Coverity on Polaris:** a *Tool Connector*.
- **Coverity Connect:** a *Tool Connector*.
- **DefenseCode ThunderScan:** `.json` report and a *Tool Connector*.
- **ErrCheck:** plain-text (e.g., `.txt`) with console output redirected to a file.
- **error-prone:** plain-text such as `.txt`.
- **ESLint** is a built-in scanner; `.json` is also supported.
- **Fortify:** `.fpr`.
- **Fortify Software Security Center:** a *Tool Connector*.
- **FxCop (user-installed)** is a built-in tool; `.xml` is also accepted.
- **Gendarme** is a built-in tool (not available in containerized deployments such as Docker Compose or Kubernetes); `.xml` is also supported.
- **GitHub Advanced Security (Code Scanning):** a *Tool Connector*.
- **GitLab Security:** `.json` and `Report .zip`.

- **GoCyclo:** plain-text (e.g., `.txt`) with console output redirected to a file; it may contain build errors.
- **GoLint:** plain-text (e.g., `.txt`) with console output redirected to a file; it may include build errors
- **GoSec:** `.json` when using the `-fmt json` flag.
- **HCL AppScan Source:** `.ozasmt`.
- **HCL AppScan on Cloud (ASoC):** `.xml` and a *Tool Connector*.
- **Helix QAC:** `.csv` report containing Helix QAC Rule Compliance results; see the *Helix QAC Support* section for more information.
- **IneffAssign:** plain-text (e.g., `.txt`) with console output redirected to a file.
- **JLint:** plain-text such as `.txt`.
- **JSHint** is a built-in tool; plain-text such as `.txt` is supported.
- **Microsoft Code Analysis** log files containing Roslyn analyzer results from MSBuild or Visual Studio as `.txt` files. Additionally, a copy/pasted `.tsv` output from the Error List table in Visual Studio with `Entire Solution` analysis enabled, which must contain the `Code`, `Description`, `Line`, and `File` (or `Path`) columns, and may optionally include the `Column` column. Furthermore, rules from the `Code Cracker` and `Security Code Scan` Roslyn analyzers, which are in the `.tsv` and `.txt` file formats as previously described.
- **MobSF:** `.json` where the JSON is generated by exporting a JSON file using the API, api/v1/report_json.
- **MobSF Scan:** `.json`.
- **NDepend:** `.xml` file containing NDepend Rule Results.
- **OCLint:** `.xml`.
- **Orca Security:** a *Tool Connector*.
- **Parasoft JTest/C++Test/dotTest:** `.xml`; please see the *Parasoft Support* section for more information.
- **PHP_CodeSniffer** is a built-in tool; `.xml` is also supported.
- **PHPMD** is a built-in tool; `.xml` is also supported.
- **PMD** is a built-in tool; `.xml` is also supported.
- **Pylint** is a built-in tool; `.json` is also supported.
- **Polaris:** a *Tool Connector*.
- **Rapid Scan SAST:** `.json`.
- **SafeSQL:** plain-text (e.g., `.txt`) with console output redirected to a file.
- **SARIF** `.json` format in compliance with SARIF v2.1.0 schema; please see the *SARIF Support* section for more information.
- **SATE:** `.xml` format for NIST's Static Analysis Tool Exposition V (SATE V).
- **Scalastyle** is a built-in tool; `.xml` is also supported.
- **SCARF:** `.xml` for SWAMP Common Assessment Result Format.
- **SciTools Understand:** `.csv` report containing SciTools Understand analysis results.
- **Semgrep:** `.json` and a *Tool Connector*.
- **Snyk Code:** a *Tool Connector*; `.json` is supported.

- **Software Risk Manager XML:** for cases where you have data from a custom tool or from a tool that isn't supported by Software Risk Manager, you can convert the output to the Software Risk Manager `.xml` format and input that directly for analysis. XML schemas and examples are provided via the download icon in the Software Risk Manager header.
- **SonarQube/SonarCloud:** a *Tool Connector*.
- **SonarQube Generic Issue Import Format:** `.json`.
- **SpotBugs/FindBugs** is a built-in scanner; `.xml` outputs are also accepted.
- **Staticcheck:** `.json` when using the `-f json` flag, with its console output redirected to a file.
- **TFLint:** `.json` file in compliance with SARIF format when using the `-f sarif` format option with `tflint` command. Software Risk Manager currently allows importing TFLint results in SARIF format only. The `-f` (or `--format`) option can be used to generate the SARIF formatted console output, which can then be redirected to a file, for example: `tflint -f sarif <file or directory>`.
- **TruffleHog:** `.json` file with repository scan results.
- **Veracode:** either the `.zip` files generated when exporting XML results, or the `.xml` files contained within them. Additionally, Veracode is a *Tool Connector*.
- **Vet:** `.json` from `go vet` by using the `-json` flag, with console output redirected to a file. It may include build errors.
- **ZPA:** `.json` using the `zpa-cli` command line tool, with the `sq-generic-issue-import` output-format.
- **Other source zip archives:** `.zip` (zipped source archives display contextual source for findings on the *Finding Details* page).

DAST Tools

Software Risk Manager supports the following DAST tools and import formats:

- **Acunetix Desktop:** `.xml` where the XML is generated by selecting Scans, then Select Scan, then WAF Export, and then XML.
- **Acunetix 360:** a *Tool Connector*.
- **AppSpider Vulnerability Summary:** `VulnerabilitiesSummary.xml`; see the *AppSpider Support* section for more information.
- **Arachni:** `.json`.
- **APIsec:** a *Tool Connector*.
- **Burp Suite:** `.xml`when the Base64 encoding option is selected; consider using our Burp Suite plugin to send results directly to Software Risk Manager.
- **Continuous Dynamic (formerly WhiteHat):** a *Tool Connector*.
- **Defensics Fuzz Test:** `super-summary.xml`.
- **Dynatrace:** a *Tool Connector* (Attack data only).
- **Fortify WebInspect:** `.xml` when these options are selected: File, then Export, then Scan Details. In the Settings section, choose *Full* from the "Details:" dropdown menu and click Export.
- **HCL AppScan Standard:** `.xml`.
- **HCL AppScan on Cloud (ASoC):** `.xml` and a *Tool Connector*.
- **Imperva:** a *Tool Connector*.

- **Invicti Standard** (formerly Netsparker): `.xml`.
- **Invicti Enterprise** (formerly Netsparker Enterprise): *Vulnerabilities List* `.xml` report and a *Tool Connector*.
- **OWASP ZAP:** `.xml`; consider using our OWASP ZAP add-on to send results directly to Software Risk Manager.
- **Qualys WAS:** a *Tool Connector*.
- **Polaris:** a *Tool Connector*.
- **Rapid7 InsightAppSec:** a *Tool Connector*.
- **Rapid7 InsightVM:** a *Tool Connector*.
- **Rapid7 Nexpose:** `.xml` generated with the XML Export or XML Export 2.0 reports. See Rapid7 Nexpose Working with report formats and Report templates and sections for more information.
- **Black Duck Managed Services Platform:** `.xml` report and a *Tool Connector*.
- **Tenable.io Web App Scanning:** a *Tool Connector*.
- **Tinfoil API:** a *Tool Connector*.
- **Tinfoil Web:** a *Tool Connector*.
- **Trustwave App Scanner:** a *Tool Connector*.
- **Veracode:** `.xml` and `.zip`. Additionally, Veracode is a *Tool Connector*.
- **WPScan:** `.json`.
- **sqlmap output** - Sqlmap does not provide a suitable output format; to that end we've developed a fork of sqlmap, which has flags for exporting in the Software Risk Manager Custom XML format.

IAST Tools

Software Risk Manager supports the following IAST tools and import formats:

- **Checkmarx:** a *Tool Connector*.
- **Contrast:** a *Tool Connector*.
- **HCL AppScan on Cloud (ASoC):** `.xml` and a *Tool Connector*.
- **NowSecure Workstation:** `.json`.
- **Q-MAST:** a *Tool Connector*.
- **Black Duck Seeker:** a *Tool Connector*.

Mobile Tools

Software Risk Manager supports the following Mobile tools and import formats:

- **Data Theorem Mobile Secure:** a *Tool Connector*.
- **HCL AppScan on Cloud (ASoC):** `.xml` and a *Tool Connector*.
- **MobSF:** `.json` where the JSON is generated by exporting a JSON file using the API, api/v1/report_json.
- **MobSF Scan:** `.json`.
- **NowSecure:** a *Tool Connector*.
- **NowSecure Workstation:** `.json`.

InfraSec Tools

Software Risk Manager configured with the InfraSec add-on supports the following Infrastructure tools and import formats:

- **AppDetective Pro:** `.xml` *Check Results* reports; please see the *AppDetective Pro Support* section for more information on report requirements.
- **Tenable Nessus:** `.nessus`.
- **Tenable.io:** a *Tool Connector*.
- **Tenable.sc:** a *Tool Connector*.
- **Rapid7 Nexpose:** `.xml`.
- **NMap:** `.xml` that contains vulnerability information associated with scripts written using the NMap Scripting Engine.
- **Qualys VM:** `.xml` generated with Scan-Based and Host-Based report templates and a *Tool Connector*. Before generating a report with a Host-Based report template, ensure that "Vulnerability Details" and at least one subsection are checked by navigating to the Display tab, in the "Edit Scan Report Template" window, and looking under "Include the following detailed results in the report."
- **Qualys VMDR:** a *Tool Connector*.
- **Qualys CS:** `.csv` of "images" or "container" scans.
- **SCAP:** `.xml` file containing the SCAP tool's scan results.

Threat Modeling Tools

Software Risk Manager supports the following Threat Modeling tools and import formats:

- **IriusRisk:** a *Tool Connector*.
- **Microsoft Threat Modeling Tool 2016:** `.htm` reports and `.tm7` files. Note: `.htm` reports will include images of the diagram and interaction for each finding.
- **SD Elements:** a *Tool Connector*.

Component Tools

Software Risk Manager supports the following Component tools and import formats:

- **Black Duck Binary Analysis:** a *Tool Connector*, `.csv` and `.json` are supported. Note: CSV files can be created via Export > Vulnerabilities as CSV > Comma separator, and JSON files are only available through API using the `GET api/product/{ productId }` endpoint.
- **Black Duck SCA:** When Scan Farm is configured, Black Duck is available as a built-in tool. Also accessible as a *Tool Connector.*
- **CAST Highlight:** a *Tool Connector.*
- **Checkmarx One:** a *Tool Connector.*
- **Checkmarx OSA:** a *Tool Connector.*
- **Continuous Dynamic (formerly WhiteHat):** a *Tool Connector*.
- **Dependency-Check** is a built-in scanner; `.xml` is also supported.
- **Dependency-Track** a *Tool Connector.*
- **Dynatrace:** a *Tool Connector* (Vulnerability data with no related container images only).

- **GitHub Advanced Security (Dependabot)** a *Tool Connector.*
- **GitLab Security:** `.json` and `Report .zip`.
- **HCL AppScan on Cloud (ASoC):** `.xml` and a *Tool Connector*.
- **JFrog Xray:** a *Tool Connector*, `.json` is supported
- **Mend:** a *Tool Connector.*
- **NeuVector:** a *Tool Connector.*
- **Orca Security:** a *Tool Connector*.
- **Polaris:** a *Tool Connector.*
- **Retire.js** is checked by Dependency-Check; if run externally, `.json` is supported.
- **Snyk Open Source:** a *Tool Connector*, `.json` is supported.
- **Snyk License Compliance Management:** a *Tool Connector*, `.json` is supported.
- **Sonatype Nexus:** a *Tool Connector.*
- **Veracode:** `.xml` and `.zip`. Additionally, Veracode is a *Tool Connector*.
- **WPScan:** `.json`.

Container Tools

Software Risk Manager supports the following Container tools and import formats:

- **Anchore:** a `.json` file generated using `anchore-cli image vuln {image-name} all`.
- **Aqua Enterprise:** a *Tool Connector*.
- **Check Point CloudGuard:** a *Tool Connector* (Vulnerability data only).
- **Dynatrace:** a *Tool Connector* (Vulnerability data with related container images only).
- **GitLab Security:** `.json` and `Report .zip`.
- **Grype:** a `.json` file with container image/filesystem results.
- **Google SCC:** a *Tool Connector*.
- **Harbor:** a `.json` or `.csv` Harbor vulnerability report.
- **NeuVector:** a *Tool Connector.*
- **Orca Security:** a *Tool Connector*.
- **Snyk Container:** a *Tool Connector*, `.json` is supported.
- **Prisma Cloud Compute (Twistlock):** a *Tool Connector*, a `.json` file generated with `twistcli`, or one of the downloadable Twistlock CSVs in the Images, Scans, or Hosts format (the Connector is strongly recommended); please see the *Twistlock Support* section for more information.
- **Trivy:** a `.json` file with container image results (other scan types are not yet supported).

Cloud Infrastructure Tools

Software Risk Manager supports the following Cloud Infrastructure tools and import formats:

- **Prisma Cloud (RedLock):** a *Tool Connector* (Alert data only), a `.csv` file of Alerts downloaded from Prisma Cloud UI, or a `.json` file from the Prisma Cloud REST API "List Alerts V1" endpoint.
- **AWS Security Hub:** `.json` and a *Tool Connector*.

- **Azure Security Center:** a `.csv` file by clicking 'Download CSV report' from the 'Recommendations' page in Microsoft Defender for Cloud.
- **Check Point CloudGuard:** a *Tool Connector* (Posture Finding data only).
- **Wiz:** a *Tool Connector.*
- **Google SCC:** a *Tool Connector*.
- **Microsoft Defender for Cloud:** a *Tool Connector.*

Bug Bounty Tools

Software Risk Manager supports the following Bug Bounty tools and import formats:

- **Hacker One:** a *Tool Connector*.

Infrastructure as Code (IaC) Tools

Software Risk Manager supports the following Infrastructure as Code tools and import formats:

- **Checkmarx One:** a *Tool Connector.*
- **Checkov:** `.json`.
- **Orca Security:** a *Tool Connector*.

Web Application Firewall (WAF) Tools

Software Risk Manager supports the following Web Application Firewall tools and import formats:

- **Imperva:** a *Tool Connector*.

Security Technical Implementation Guide (STIG) Tools

Software Risk Manager supports the following STIG tools and import formats:

- **STIG:** `.ckl` and `.cklb` format checklist results exported by any common STIG tool.

AppDetective Pro

When generating a *Check Results Report* in AppDetective Pro, you will be given options for which fields to include. For best results, we recommend including every field. However, at a minimum, the following fields are required:

- Check Category
- Summary
- Overview
- Fix Information
- CVE
- References
- Links
- Vulnerability
- Description
- Show Occurrences

If any of these required fields are excluded, you will receive an error when uploading the report to Software Risk Manager and analysis of the file will not be allowed.

AppSpider

Software Risk Manager accepts the `VulnerabilitiesSummary.xml` file from AppSpider. This file is output as part of the report generation process within AppSpider.

To generate a report and locate the summary XML file:

1. Run a new scan or open an existing scan in AppSpider.
2. Generate a report by clicking the Generate Report button on the scan toolbar.
3. Locate the generated report on disk (the default location is Documents/AppSpider/Scans). The `VulnerabilitiesSummary.xml` file in the report folder is the file that should be uploaded to Software Risk Manager for analysis.

Aqua SaaS Configuration

Software Risk Manager supports Aqua SaaS Configuration. For more information, click here.

CodeSonar

The preferred means of importing CodeSonar result into Software Risk Manager is to use the CodeSonar *Tool Connector*. However, in situations where the machine running Software Risk Manager and the machine running CodeSonar cannot communicate with each other, the *CodeSonar-Scrape* utility can bridge the gap.

CodeSonar-Scrape is a command-line utility that you can use to generate a Zip file that Software Risk Manager understands as CodeSonar results. You provide it the URL of your CodeSonar server, the name of the project you want to import into Software Risk Manager, and optionally your username and password. SRM will find all of the "warnings" associated with that project and download them into a Zip file, which you can then upload to Software Risk Manager. Results imported in this manner will include descriptions (tracing) information and links back to CodeSonar's hub for warning details and category documentation. Detailed instructions for this tool can be found in the *CodeSonar-Scrape User Guide*. If you need CodeSonar-Scrape or have questions, please contact us.

Dynatrace

Software Risk Manager supports data ingestion via the Dynatrace *Tool Connector*.

Connector authentication is performed with an access token. The user should have both the `Read security problems` and `Read attacks` scopes for this token.

Helix QAC

Software Risk Manager supports the importation of Helix QAC rule compliance reports (.csv). The instructions below show how to use the Helix QAC GUI to create a rule compliance report on your local machine that you can upload to SRM.

To generate a rule compliance report from the Helix QAC GUI:

1. Click Report from the main menu and use the dropdown list to select which project or files to use.
2. Click the "Report Type" field and select "Rule Compliance Report" from the dropdown list.
3. Confirm the output location and the name of the report in the location and name fields. You can either use the default settings or enter new values in the respective fields.
4. Click OK. The report will be generated and placed in the selected output folder.

You can now upload the Helix QAC rule compliance report to SRM.

For more information about Helix QAC, visit Perforce.com.

JFrog Xray Support

Software Risk Manager imports results from JFrog Xray using its built-in Reports feature, which does not include a list of the scanned artifacts. This may cause resolved vulnerabilities to still appear in SRM. For more information, click here.

Parasoft Support

Software Risk Manager accepts the XML SATE reports generated by Parasoft tools, which can be generated using both the GUI and CLI. For more information, click here.

Prisma Cloud Compute (Twistlock)

Software Risk Manager supports data ingestion via the Prisma Cloud Compute (Twistlock) *Tool Connector* and CSV/JSON files. The tool connector is **strongly recommended** due to limitations in CSV export[1] and JSON ingest[2].

Connector authentication is performed with a username and password. The user should have the `monitorImages` (Compute -> Monitor) permission at a minimum. The recommended built-in role is `DevSecOps`. The connector will validate that the given user has the necessary permission during configuration.

The connector offers several options for filtering images. Verifying any of these options will state the number of images matched. This result is based only on the filter being verified. During analysis, all filters will be combined.

The connector may ingest vulnerabilities from Deployed, Registry, and CLI image scans. Each particular image type can be toggled during configuration. Filters will be applied to all selected scan types, where applicable. Not all filters can be applied to CLI image scans. Fields inapplicable to CLI images will be noted in their descriptions within the connector configuration page.

[1]Downloading CSVs from the Prisma Cloud Compute UI will not download all results, only those visible on the current page. Full CSVs can be downloaded by interacting with the REST API at `api/v1/images/download`, which requires a separate API authentication step against `api/v1/authenticate`. (Navigating to this endpoint with a browser will typically fail, regardless of authentication within the Prisma UI.)

[2]Analyzing JSON results in Software Risk Manager will archive any previous Twistlock JSON results as a standard part of archival behavior. If multiple images are being scanned with Twistlock, the scans for all images must be uploaded together in order for all of their results to persist after analysis.

Base Image Scanning

Twistlock allows filtering (exclusion) of base image vulnerabilities when retrieving results, which is exposed as the option *"Exclude base image vulns"* in the connector config page.

Note: Twistlock requires the base image to be in one of its configured registries, and the base image must have been previously scanned:

To define your base images, go to **Defend > Vulnerabilities > Images > Base images**. The base images you define must reside in your registry and they must be scanned to exclude their vulnerabilities from scan reports.

Missing Image Data

Twistlock may temporarily report image data as "missing," even though it may reappear in subsequent analyses. Results for these images will be missing once the analysis completes. Such missing data will be reported as a warning on the Analysis page and in the Visual Log.

SARIF

Software Risk Manager strictly supports the v2.1.0 SARIF spec as outlined here and detailed here. New formats will be added explicitly; support for v2.1.0 does not imply support for v2.1.1, and so on.

Note: All ingested SARIF results will be detected as "SAST," regardless of whether a "Container Analysis" tool generated it.

Limitations

Software Risk Manager support for SARIF currently does not include the following notable features:

- External properties files
- Inline external properties
- JSON pointers/SARIF-URI schemes
- Suppressions
- Text/code/artifact snippets
- Localization data
- Details of tool/converter invocation
- Version control "provenance" information
- Location references to binary files
- Graph information
- Stack traces
- Tool notifications

Results with Multiple Locations

SARIF results containing multiple locations will be split into duplicate results, one for each location. If multiple `codeFlows` are specified and none have a sink matching the result location, the `codeFlow` sinks will be treated as the effective locations, and the result will similarly be split. This may cause a mismatch between the location reported by the SARIF result and the location used by Software Risk Manager.

SBOM Files

Software Risk Manager supports the following SBOM import formats:

- **CycloneDX:** `.xml` and `.json`
- **SPDX:** `.spdx` and `.json`

Note: Uploading and analyzing SBOM files will not result in any new findings. The purpose of importing SBOM files is to provide a single place to retrieve SBOMs and to store them.

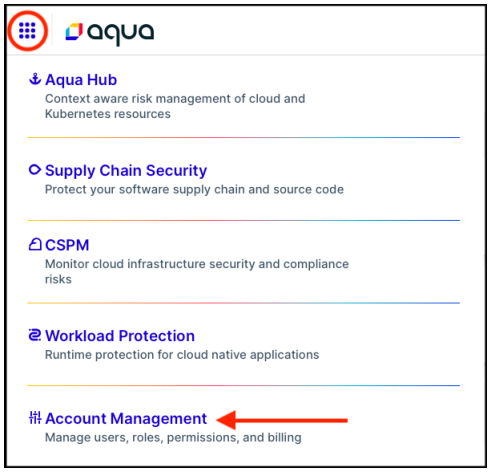
Aqua SaaS Configuration

For Aqua Enterprise on-prem deployment, the connector only requires giving credentials for an Aqua user with the necessary permissions. However, in an Aqua SaaS instance, the process of getting the necessary

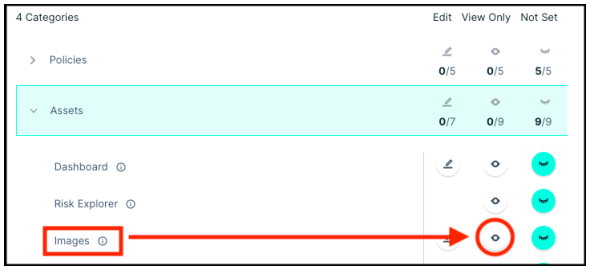
credentials for the connector is more complicated. In this case, Aqua SaaS requires creating a permission set, role, and an access token, as shown below.

To prepare credentials for the connector:

1. Log into cloud.aquasec.com.
2. Create a permission set:
   a. Click the applications icon located in the top left corner and select Account Management from the dropdown menu.



   b. Select User Management > Permission Sets from the left menu.
   c. Click "Add Permission Set" and enter a name in the Name field.
   d. Expand the Assets category, locate "Images," and enable "view" by clicking the corresponding icon. This is needed for access to images.



   e. Expand the Compliance category, locate "Vulnerabilities," and click the view icon. This is needed for access to vulnerability reports on a finer, per-image-layer basis.

      Note: (Optional) SRM will detect when this permission is unavailable and alter its logic appropriately.

   f. Click Save.
3. Create a role and assign the previously created permission set:
   a. Select User Management from the left navigation menu and click Roles.
   b. Click Add Role.
   c. Enter a name in the Name field, select a permission set, and select an application scope.
   d. Click Save.
4. Create an API Key:

a. Select Settings from the left navigation menu and click API Keys.

b. Click Generate Key and save the API Key and Key Secret.

c. Click the configuration icon to the right of the new API Key and select Edit.

d. Disable "Global Permissions."

e. Enable "roles:assign", "tokens:readwrite", permission (needed to authenticate with token).

Note: For "tokens:readwrite", give it permission to use the role created previously.

f. Click Save.

The role name from step 3 is used for the “Role Names” connector form field. The API Key and Key Secret from step 4 are used in the “API Key” and “API Key Secret” connector form fields.

JFrog Xray Support

Software Risk Manager imports results from JFrog Xray using its built-in Reports feature, which collects a list of vulnerabilities for all scanned artifacts that match the specified filters. While this data includes the list of vulnerabilities and the affected artifacts, it does not include a list of the scanned artifacts.

If a new version of an artifact is uploaded and scanned by JFrog Xray, and if it is found to have zero vulnerabilities, that artifact may not have any entries in the report and vulnerabilities from older versions of the artifact may still be present. While SRM *does* track the list of affected artifacts through the "JFrog Impacted Artifacts" field on individual results, the existence of findings in the scanned project gives a false impression of its state and makes it harder to tell which vulnerabilities are still present in the most recent artifacts of interest.

Requests by SRM for the Build report type automatically include a filter to only fetch results for the latest build. However, Repository reports do not have this option, which necessitates detection and filtering by SRM to determine the latest version and its associated results.

When the JFrog Xray connector is configured to use Repository reports, SRM can perform a variety of additional checks to discover the most recent version of each affected artifact. The types and number of checks done will depend on which options are selected in the connector configuration form.

Detecting the "Latest Version"

The strategy for detecting the "latest version" depends on the "Latest Version Search Method" dropdown field, which has the following options:

- **None**. All results from the report are ingested.
- **Report Only**. Finds the most recent artifact versions present in the Xray report.
- **Active Search**. Uses the JFrog Xray API to find the most recent version of each affected artifact.

The "None" option may be used if you are interested in vulnerabilities across all versions of the affected artifacts. The "Report Only" option may be used if you expect the scanned artifacts to always contain at least some vulnerabilities. The "Active Search" option may be used if you expect at least some scanned artifacts to be completely free of vulnerabilities.

When a "latest version" is detected, its artifact ID and file path are recorded and used as a filter on the list of vulnerabilities in the JFrog Xray report.

Finding the "Latest Version" within a Report

When using the contents of the Xray report to detect the most recent version, SRM will take each affected artifact ID and split it on its last : (colon) character. Text up to this character will be used as the package ID, and text after that character will be used as the package version.

Note: This logic is also used to filter for specific versions when "Version Filtering Mode" is set to "Text Pattern."

SRM will take each artifact ID, its file path, and scan time for the artifact, and group this information together based on the parsed package ID. For each group, the entry with the most recent scan time will be used as the "latest version" for the given package ID.

Finding the "Latest Version" via JFrog Xray API

When using the Active Search method, SRM will use the Scans List - Get Package Versions API to find the most recent version for each affected artifact. This requires detecting the correct package ID for the artifact, which is the most time-intensive part of this process.

For each file name found in the JFrog Xray report, SRM will make a request to the Scans List - Get Artifacts API, filtering to the file name and a creation time of ±1 hour of its scan date. If an entry is found with a matching file path, it will be associated with the entry's `packageId` for later reference.

For any artifacts whose package ID was not resolved, SRM will create a set of "candidate" package IDs based on the artifact ID. While JFrog artifact IDs do have a standard format, it's not guaranteed that the IDs will be "fully formed." For example, there may be an RPM artifact with an artifact ID of `rpm://7:rpm-python:7:4.11.3-43.el7`, or it may present the artifact as `rpm://rpm-python:4.11.3-43.el7` depending on the Artifactory configuration. The candidate package IDs for an artifact ID will be permutations of a subset of the artifact ID, for example `7:rpm-python:7, rpm-python:7, rpm-python`.

SRM combines the list of known package IDs and candidate IDs and uses the `Get Package Versions` API for each package ID. SRM collects pages of package versions and, for each artifact associated with the tested package ID, checks for an API response which matches the artifact ID and file path. This ensures that SRM is reading from the correct list of package versions for a given artifact. SRM will collect these pages until all associated artifacts are resolved, or the received versions are older than the oldest artifact associated with the tested package ID. For artifacts that were matched, their latest version is set to the most recent entry from the beginning of the list of versions provided by the JFrog Xray API.

SRM continues querying for package versions of each potential package ID, updating its list of associated unresolved artifacts, until a latest version is discovered for all artifacts or all options are exhausted. For remaining artifacts that have a known package ID but an unresolved version, SRM uses the scan date from the JFrog Xray report to find the most recent version. For artifacts with an unresolved version **and** do not have a known package ID, their results will be included without filtering.

This process gives SRM an updated list of "latest artifacts" to filter by. SRM then reads the JFrog Xray report and only ingests results for artifacts whose artifact ID and file path exist in this list.

Optimization Options for "Active Search" Detection

The Active Search detection method attempts to ensure that the latest detected version is correct. This can involve many requests to the JFrog Xray API and significantly lengthen analysis times. You can use the "Latest Version Search Optimizations" dropdown field to let SRM make certain assumptions, which will reduce the amount of API requests to JFrog Xray. This field has the following options:

- **Accurate**. The full search process is performed as previously described.
- **Pre-Filter**. SRM will use the "Report Only" filtering before making any API requests, reducing the number of requests to `Get Artifacts` and, potentially, `Get Package Versions`.
- **Optimistic**. In addition to the Pre-Filter optimization, this option assumes that each package ID candidate is a match for each associated artifact (without confirming their existence in the list of package versions) and immediately returns the first package version in the list. This reduces the number of requests to `Get Package Versions`.

It is typically safe to enable "Optimistic" filtering for the following package types:

- Docker

- Maven
- NPM
- Go
- Composer
- NuGet
- PyPi
- Conan

It is recommended to start with the "Accurate" optimization to get a baseline set of results and confirm that other optimization options are consistent with that baseline.

Parasoft Support

Software Risk Manager accepts the XML SATE reports generated by Parasoft tools, which can be generated using both the GUI and CLI.

Generating a Report from the GUI

To generate the report from the GUI:

1. Run a scan.
2. Click the Test Progress summary tab and click the Generate Report button in the toolbar.
   The Report and Publish dialog will open.
3. Click Preferences.
4. Click the Format dropdown and select XML SATE (Static Analysis Tool Exposition).
5. Click Apply, then click OK.
   The Report and Publish dialog will open.
6. Select the option to open the report in a browser.
7. Click OK.
   The generated report will appear above the Test Progress and summary tab; the location of the file on disk will display in the report tab.

Generating a Report from the Command Line Interface

To generate a report from the CLI and export the findings:

1. Create a file containing the proper report preferences, one setting per line.
   The file must contain at least these settings:
   - `report.custom.extension=xml`
   - `report.format=sate`
2. Select Parasoft -> Preferences from the toolbar.
3. Select Parasoft (root).
4. Click the "share" link in the "Configure settings" section.
5. Enter the filepath into the text box.
   This will be where your settings file will be located.
6. Check the "Reports" option, then click OK.
7. Run the CLI with the following options:

- `-localsettings path/to/settings/file`
- `-report path/where/report/should/go/filename.xml`

The report to upload to Software Risk Manager will be at `path/where/report/should/go/filename_report.xml` or `path/where/report/should/go/filename_sate.xml`

SonarQube Support

When using SonarQube, there are two potential issues to be aware of. The first deals with permissions; the second, listings.

Permissions

Non-admin tokens may see permission-related issues when importing projects or when running analyses. SRM performs permission checks during project import to prevent the selection of items that cannot be accessed. Consequently, you should note the following:

- These checks can greatly extend the runtime of project auto-import through the Integrations page.
- The additional requests can cause rate-limiting errors when accessing SonarQube.
- Projects that are successfully imported may later see analysis errors if the token has some permissions revoked at a later time.

However, if you are using an admin token, you can set `sonarqube.permission-checks.enabled = false` in the SRM props file to disable these permission checks during project import. (This will not affect permission checks done during analysis.)

Listings

SonarQube has an internal limit of 10,000 items when listing any sort of data from their API. This affects lists of projects, bugs, hotspots, and so on.

When listing projects, SRM will stop once it reaches this internal limit, which can prevent some projects from appearing. However, if an admin token is provided, SRM will use an alternative method that will bypass the 10,000 project limit, allowing SRM to show the full list of projects. Note: This ability to bypass the project limit does not apply to any other data requested from SonarQube.

When listing issues during analysis, SRM mitigates this limit by using specific lists like “critical bug issues in project X” instead of larger lists like “all issues in portfolio Y.” Nevertheless, it’s still possible for these “specific” lists to exceed the 10,000 project limit, in which case analysis will fail with an error.

Starting an Analysis

There are several ways to prepare and initiate an analysis with Software Risk Manager.

For detailed instructions on running an analysis, click one of the links below:

- Using the web interface
- Using the API
- Starting an analysis with Black Duck Bridge CLI

Starting an Analysis Using the Web Interface

Analyses can be prepared and initiated manually from the Software Risk Manager web interface.

Note: To start an analysis, you will need a defined project and a current analysis configuration.

To start an analysis:

1. Click the Projects icon on the navigation bar to open the Projects page.
2. Click the project's dropdown configuration icon and select New Analysis.
3. Select a target branch from the Target Branch dropdown menu.
   You can choose an existing branch or create a new one by typing a new (unique) name into the branch field. Entering text will also filter the existing branch names.
4. Click Add File to upload files for analysis.
   As you add files, they will be uploaded to the SRM server for identification. Once the server has identified the file contents, SRM will display the following information:
   - Detected Content
   - Tools to Run

   Use the checkbox on the tag to disable (or re-enable) that tag. Disabling a tag in the Tools to Run section will tell SRM not to run that tool, even though it is applicable to that file.
5. Click Begin Analysis.
   Analysis is conducted as a "job." The work order is placed in the job queue and will be executed once enough resources are free. Often, the time spent in the queue is negligible, but you might still see a message stating that the analysis has been queued. Once the analysis job is finished queueing, the analysis will begin. The page will display a timer to indicate the current duration of the analysis.

The actual duration of the analysis depends on several factors, including the following:

- *How big is your application?* An application's size is likely the most significant factor in determining the duration of an analysis. Smaller apps usually take around 30 seconds, medium-size apps can take tens of minutes, and large apps can take hours or more.
- *Is Software Risk Manager running tools for you?* If so, the analysis duration will include the time it takes to run these tools. The time it takes to run a tool on your application will usually grow as your application grows.
- *How much activity is going on in Software Risk Manager?* More activity by users of Software Risk Manager means more load on the database, which can slow down analysis to some degree.
- *How many findings can be discovered?* This is difficult to know ahead of time, but the number of tool results/findings in a project will also affect the analysis duration. In this manner, a small application with many vulnerabilities might take longer to analyze than a large application with very few vulnerabilities.

Once the analysis has been queued, it is safe to leave the page. The analysis will continue in the background. Keep the page open, however, is recommended in order to see any warnings or errors that might occur during the analysis.

If the analysis completes successfully, the analysis timer will become a link to the Findings page. Any currently-opened Findings pages will be updated to reflect the latest analysis results.

Deploying Intelligent Orchestration

If your Software Risk Manager implementation includes Intelligent Orchestration (IO), applying an IO policy to a project will generate *prescriptions* for the best tools to use in analyzing the code. Those prescribed tools will be automatically selected in the *Tools to Run* section of the New Analysis page. For more information on Intelligent Orchestration, see Intelligent Orchestration.

Inputs from Git Repositories

If you set up a *Git Configuration* for a project, the New Analysis page will automatically include the latest contents of the configured branch of the configured repository as an input.

Normally, Software Risk Manager will update the local clone and check out the appropriate branch before sending the files to the analysis. As development is done on that branch, analysis of that branch will change

along with the contents. But if you want to analyze a specific point in the repository, you can configure Software Risk Manager to use a specific branch, commit, or tag by clicking on the underlined section of the input. Select the branch, commit, or tag and click Use this.

Starting an Analysis Using the API

Software Risk Manager offers an expanding API to interface with the system's functionality programmatically. The ability to push files for an analysis by Software Risk Manager is exposed by the API. This enables automated integration scenarios such as continuous integration. In a continuous integration scenario, a post-build step can be added to the build jobs to automatically push the source and compiled artifacts to Software Risk Manager for analysis. This type of setup is strongly recommended for development teams to catch potential issues within their codebases early for quick remediation. (Software Risk Manager offer a Jenkins plugin to facilitate use in a continuous integration context.)

Before an API key can be used for automated analyses, the key must be assigned the `create` role for the project. The API call to push the files and initiate the analysis is documented in the Software Risk Manager API Guide.

Tool Orchestration

When the Tool Orchestration Service is enabled, Software Risk Manager can orchestrate analyses that run in whole or in part on your Kubernetes (k8s) cluster. (See the Tool Orchestration Configuration section in the *Software Risk Manager Install Guide* for instructions to enable this feature.)

A Software Risk Manager analysis may run one or more built-in code scanners. Many of those tools can run on your Kubernetes cluster when you enable the tool orchestration feature. Those that cannot, such as Dependency Check, will continue to run on the Software Risk Manager web server.

The following table shows which bundled tools Software Risk Manager can run on your k8s cluster.

Table 9:

Bundled Tool	Tool Orchestration Support
Brakeman	Yes
CheckStyle	Yes
CPPCheck	Yes
DependencyCheck	No
ESLint	Yes
FxCop (user-installed)	No
Gendarme	Yes
JSHint	Yes
PHP Code Sniffer	Yes
PHP MD	Yes
PMD	Yes
Pylint	Yes
Retire JS	No
ScalaStyle	Yes

Bundled Tool	Tool Orchestration Support
SpotBugs	Yes
ZPA CLI	Yes

Software Risk Manager also includes the following tool orchestration capabilities that run only on your k8s cluster:

- Checkmarx
- Security Code Scan
- ZAP

A single Software Risk Manager analysis can have tools running on both the webserver and on multiple nodes of your k8s cluster. All tool outputs will be combined into one analysis that either succeeds or fails as a whole, provided the Software Risk Manager web server remains online throughout the analysis.

If the Software Risk Manager web application unexpectedly restarts, a built-in fail-safe lets Software Risk Manager receive k8s analysis results from abandoned orchestrated analyses. Software Risk Manager will lose any results from bundled tools in this case, so a restart of the Software Risk Manager web application is one scenario where Software Risk Manager may process results from a partially completed analysis. When Software Risk Manager detects an orchestrated analysis that it is not tracking, a message will appear on the Orchestrated Analyses page.

You can configure Software Risk Manager to run additional tools by implementing other add-in tools.

For additional information, see the following sections:

- Resource Requirements
- Scanning a Request File
- Adding a Tool

Resource Requirements

When the Tool Orchestration Service is enabled, Software Risk Manager can create orchestrated analyses that run one or more application security testing tools where each tool has access to its host's memory and CPU resources. Using Kubernetes (k8s) tools, you can control the memory and CPU capacity available to analyses. You can also improve k8s scheduling outcomes by requesting CPU or memory capacity for specific tools or projects. Resource requirements can also include a node selector and pod toleration, with taint effects NoSchedule and NoExecute, to influence further where tools run on your cluster.

The resource requirements feature cannot be configured using the Software Risk Manager user interface, but you can use the k8s kubectl command to define configuration maps (configmaps) that cover specific scope determined by a special naming convention. The tool service will look for and read optional configmaps to determine how resource requirements apply to a specific tool run.

Resource requirements containing CPU and memory instructions translate to k8s resource requests and limits and fit with any other related k8s configuration, such as a resource limit defined for a k8s namespace. You can specify resource requirement data by using the following configmap field names:

- **requests.cpu** – k8s CPU request (e.g., 1).
- **requests.memory** – k8s memory request (e.g., 1G).
- **limits.cpu** – k8s CPU limit (e.g., 2).
- **limits.memory** – k8s memory limit (e.g., 2G).
- **nodeSelectorKey** – key portion of a label associated with a node selector (e.g., purpose).

- **nodeSelectorValue** – value portion of a label associated with a node selector.
- **podTolerationKey** – key portion of a taint with the NoSchedule and NoExecute taint effects (e.g., dedicated).
- **podTolerationValue** – value portion of a taint with the NoSchedule and NoExecute taint effects (e.g., tools).

There are four types of configmaps that can contain resource requirements:

- Global
- Global Tool
- Project
- Project Tool

Software Risk Manager deployment creates the Global Resource Requirement, which provides default resource requirements for tools across every Software Risk Manager project. Global Tool requirements override Global requirements for specific tools. Project requirements override both Global and Global Tool requirements by providing default resource requirements for tools associated with a given project. And Project Tool requirements override other requirements by specifying values for a specific tool in a specific project.

Scope Overlap

The following is an example of how the scopes can overlap:

Global Resource Requirement:

- CPU Request = 1
- CPU Limit = 4
- Memory Request = 11G
- Memory Limit = 12G

Global Tool Resource Requirement:

- CPU Request = 3

Project Resource Requirement:

- Memory Request = 13G

Project Tool Resource Requirement:

- Memory Limit = 14G

Here are the effective resource requirements resulting from the above:

Effective Resource Requirement:

- CPU Request = 3
- CPU Limit = 4
- Memory Request = 13G
- Memory Limit = 14G

The naming convention determines the scope of a resource requirement configmap:

- Global: cdx-toolsvc-resource-requirements
- Global Tool: cdx-toolsvc-**ToolName**-resource-requirements

- Project: cdx-toolsvc-project-**ProjectID**-resource-requirements
- Project Tool: cdx-toolsvc-project-**ProjectID**-**ToolName**-resource-requirements
  where **ProjectID** is the integer value representing the Software Risk Manager project identifier and **ToolName** is the tool name converted to an acceptable k8s resource name by the following rules:
  - Uppercase letters must be converted to lowercase.
  - Any character other than a lowercase letter, number, dash, or period must be converted to a dash.
  - An initial character that is neither a number nor a lowercase letter must be preceded by the letter "s."
  - A name whose length is greater than 253, must be truncated to 253 characters.

Example 1 - Project Resource Requirement

To create a resource requirement for all tool runs of a Software Risk Manager project represented by ID 21, create a file named `cdx-toolsvc-project-21-resource-requirements.yaml` and enter the following data:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: cdx-toolsvc-project-21-resource-requirements
data:
  requests.cpu: "1"
  limits.cpu: "2"
  requests.memory: "1G"
  limits.memory: "2G"
```

Note: You can find a project's ID at the end of its Findings page URL. For example, a project with the ID 21 will have a Findings page URL that ends with `/srm/projects/21`.

Run the following command to create the configmap resource (replacing the cdx-svc k8s namespace, if necessary).

```
kubectl -n cdx-svc create -f ./cdx-toolsvc-project-21-resource-requirements.yaml
```

Example 2 - Global Tool Resource Requirement

To create a Global Tool resource requirement for ESLint, create a file named `cdx-toolsvc-eslint-resource-requirements.yaml` and enter the following data:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: cdx-toolsvc-eslint-resource-requirements
data:
  requests.cpu: "2"
  limits.cpu: "3"
  requests.memory: "4G"
  limits.memory: "5G"
```

Run the following command to create the configmap resource (replacing the cdx-svc k8s namespace, if necessary).

```
kubectl -n cdx-svc create -f ./cdx-toolsvc-eslint-resource-requirements.yaml
```

Example 3 - Node Selector

To create a Global Tool resource requirement for running a tool named MyTool on cluster nodes labeled with `canrunmytool=yes`, create a file named `cdx-toolsvc-mytool-resource-requirements.yaml` and enter the following data:

```
apiVersion: v1
```

```
kind: ConfigMap
metadata:
  name: cdx-toolsvc-mytool-resource-requirements
data:
  nodeSelectorKey: canrunmytool
  nodeSelectorValue: yes
```

Run the following command to create the configmap resource (replacing the cdx-svc k8s namespace, if necessary):

```
kubectl -n cdx-svc create -f ./cdx-toolsvc-mytool-resource-requirements.yaml
```

Scan Request File

An add-in tool is based on a scan request file that you define and register with Software Risk Manager. A scan request file contains the instructions that the tool service needs to invoke an application security testing tool on the k8s cluster and ingest its output into Software Risk Manager. Scan request files use the TOML file format. You can specify any valid TOML content in your tool's scan request file provided you specify the `request` table, which is a reserved section with the following parameters.

Table 10:

Key	Description	Required?
`imageName`	The name of the Docker image containing your add-in tool	Yes
`workDirectory`	The work directory where your add-in tool can find tool inputs	Yes
`shellCmd`	The Bourne shell command to invoke your add-in tool	Yes
`resultFilePath`	The output of your add-in tool	Yes
`logFilePaths`	An array of log files produced by your add-in tool	No
`preShellCmd`	An optional command to run prior to invoking the shellCmd	No
`postShellCmd`	An optional command to run after invoking the shellCmd	No
`securityActivities`	The Intelligent Orchestration security activities supported by this tool (e.g., sca, sast, dast)	No

A tool run ends in an error when either shellCmd, preShellCmd, or postShellCmd return a non-zero exit code. When the tool service runs an add-in tool, it creates the following directory structure at the path specified by the value of the `workDirectory` key.

Table 11:

Content	Description
`/ca-certificates`	A directory containing zero or more certificates that should be considered trusted certificate authorities
`/config/request.toml`	A copy of the tool's scan request file, including any project-specific configuration
`/input`	A directory containing an optional input file
`/volume-secret`	A system directory required for storing tool outputs
`/workflow-secrets`	Zero or more workflow secrets associated with an add-in tool's project configuration

When the tool service invokes an add-in tool, it provides the tool with a copy of its scan request file, so the file is a convenient place to store configuration data. After you register an add-in tool, Software Risk Manager lets you edit TOML content outside the `request` table on a per-project basis, so you can have key values that vary by project. For example, a DAST tool might have a scan request file with a key whose value indicates the URL from which to start a scan; the URL can vary from one Software Risk Manager project to the next.

Adding an AppSec Testing Tool

You can add an application security testing tool to the list of tools that Software Risk Manager can run on a Kubernetes cluster by completing the following tasks:

1. Implement a command/script/application that automates a tool.
2. Package the capability into a Docker image that can be invoked from a Bourne shell.
3. Define a scan request file, specifying (at a minimum) the key values of the Software Risk Manager request table.
4. Register the add-in tool.
5. Enable the tool for specific projects, configuring any project-specific key values defined in the scan request file.

This walkthrough will show you how to create, register, and enable an add-in tool that automates Security Code Scan, a static code analysis tool for .NET. The Security Code Scan add-in tool is automatically installed when you enable the Software Risk Manager Tool Orchestration feature, but you can use this walkthrough to learn how to add a new add-in tool whose output must be transformed to the Software Risk Manager XML Schema.

Tool Automation

Your first task will be to create a PowerShell Core script that can automate Security Code Scan. We will use a script that defines two parameters: a path to an input archive containing C# source and a path to an output file with findings that Software Risk Manager can ingest.

Create a directory called SecurityCodeScan. Download SecurityCodeScan.ps1 to the directory. The PowerShell Core script you downloaded takes the following steps to automate Security Code Scan.

1. Unpack the source code in the input file provided by Software Risk Manager.
2. Add the SecurityCodeScan.VS2017 project reference to each source code project file.
3. Run dotnet build.
4. Translate the findings from the build results into the generic SRM XML format.

The last step is required because Software Risk Manager does not support ingesting Security Code Scan findings directly. If you were automating Checkmarx, a tool whose output Software Risk Manager can read, then Step 4 would be unnecessary. You will handle Step 4 with a separate script, so download SecurityCodeScan-Results.ps1 to your SecurityCodeScan directory.

Tool Packaging

To package the Security Code Scan automation, you must create a Docker image capable of running both PowerShell Core scripts and compilations of .NET Core 2 code. Adding PowerShell Core to a Docker image based on `microsoft/dotnet:2.2-sdk` creates a suitable environment.

Download Dockerfile.txt to your SecurityCodeScan directory, and run the following command from that directory to generate a Docker image that can automate Security Code Scan.

```
docker build -t codedx-securitycodescanrunner:v1.0 -f ./Dockerfile.txt .
```

Scan Request File

The docker build command from the previous section created a Docker image named `codedx-securitycodescanrunner:v1.0` that contains the SecurityCodeScanner.ps1 script in the `/opt/codedx/securitycodescan` directory. The following scan request file content describes how to run SecurityCodeScanner.ps1 on an input provided by Software Risk Manager.

```
[request]
imageName = "codedx-securitycodescanrunner:v1.0"
workDirectory = "/opt/codedx/securitycodescan/work"
shellCmd = '''
source=$(ls /opt/codedx/securitycodescan/work/input)
  pwsh /opt/codedx/securitycodescan/script/SecurityCodeScan.ps1 \
   "/opt/codedx/securitycodescan/work/input/$source" \
   /opt/codedx/securitycodescan/work/output/securitycodescan.output.xml
'''
resultFilePath = "/opt/codedx/securitycodescan/work/output/securitycodescan.output.xml"
securityActivities = ['sast']
```

The value of the imageName key is `codedx-securitycodescanrunner:v1.0`, the Docker image you created. The workDirectory key value is /opt/codedx/securitycodescan/work, a directory that already exists because your Dockerfile established a /opt/codedx/securitycodescan/work/output directory to store the result from SecurityCodeScan.ps1. Software Risk Manager uses the work directory to store add-in tool data.

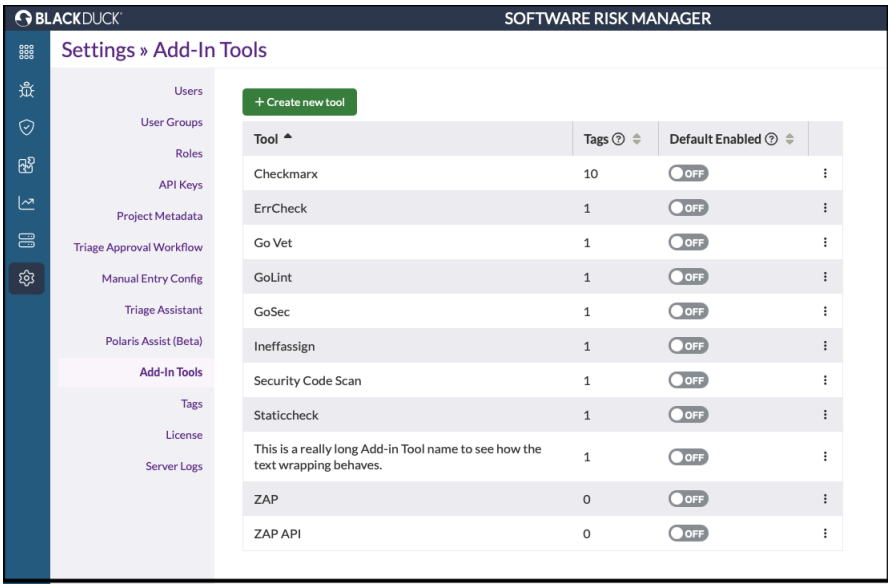
The shellCmd key value is the Bourne shell script Software Risk Manager will run to invoke your add-in tool. SecurityCodeScan.ps1 requires two parameters: an analysis input file and an output file. Software Risk Manager puts the analysis input file in the input directory, which is a sub-directory of the work directory. The analysis input file parameter will come from a search of that directory, and the output file will be `/opt/codedx/securitycodescan/work/output/securitycodescan.output.xml`. The value of the resultFilePath key directs Software Risk Manager to the add-in tool output and will match the script's output file parameter.

In this example, you did not use the optional scan request file keys. The logFilePaths key is unnecessary because SecurityCodeScan.ps1 writes log information to stdout, and the shellCmd does not require any pre or post commands that you could accomplish with preShellCmd and postShellCmd.

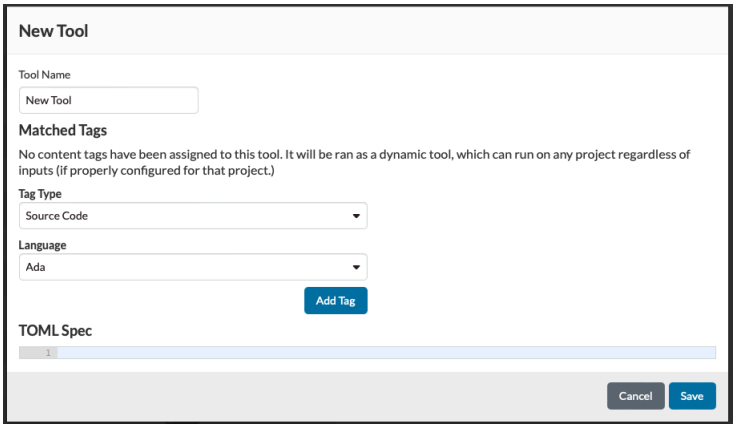
Note: `securityActivities` is used to define the add-in tool type for use in Intelligent Orchestration implementations.

Software Risk Manager Registration

Registering your add-in tool with Software Risk Manager is the next step. Log on to Software Risk Manager as an administrator, click the Settings icon in the navigation bar, then select Add-In Tools from the left menu.



Click Create New Tool to open the Add-In Tool Registration window.



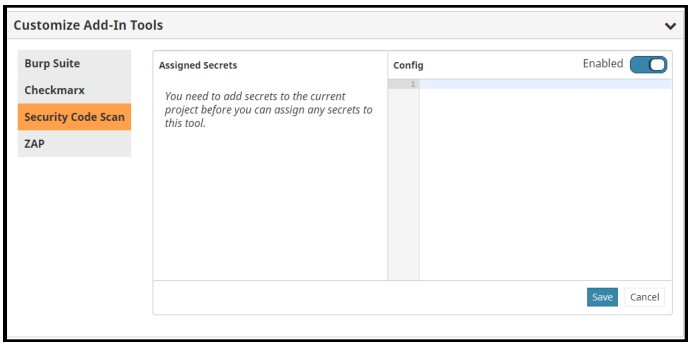
Click the *New Tool* label in the window title area, replace the text with a meaningful name, like Security Code Scan, if that name is not in use, and click OK.

Security Code Scan is a SAST tool requiring an analysis input, so you must associate your add-in tool with one or more types of content that Software Risk Manager can detect. Select Source Code for *Tag type*, C# for *Language*, and click Add Tag so that Software Risk Manager will offer to run Security Code Scan whenever it detects an analysis input file containing C# source. Lastly, specify the contents of your scan request file in the *TOML Spec* section. Click Done to save your add-in tool registration.

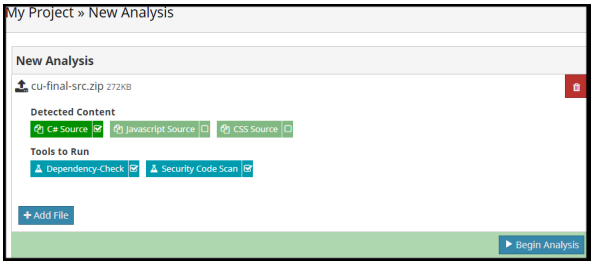
Enable Add-In Tool

Your add-in tool is now registered in a disabled state. To enable your add-in tool for a specific project, open the Tool Service Configuration page, find your add-in tool in the *Customized Add-In Tools* section, toggle the *Disabled/Enabled* switch to enabled, and click Save. You can also use the *Default enabled* toggle to enable a tool for every project, excluding those where it was explicitly disabled.

Below the *Disabled/Enabled* toggle is a box where you could edit any project-specific TOML content, which is scan request file content outside the `request` section (Security Code Scan has none).



Software Risk Manager will offer to run enabled add-in tools whenever it detects an analysis input containing C# source code in the project where you enabled the add-in.



Users will have the option to deselect your add-in tool when they start a new analysis. For example, Software Risk Manager does not distinguish between C# source code for .NET Core and .NET Framework, and since your add-in tool runs on Linux, it supports .NET Core code only. A user configuring a new analysis with .NET Framework source code (not .NET Core source code) could deselect your add-in tool in that case.

Tool Status and Severity Mapping

Not all of the tools supported by Software Risk Manager report risk in the same way. For this reason, the severity and status findings reported by each tool are converted or "mapped" to SRM risk ratings.

To see how each tool's findings are mapped, use the links below to locate the desired tool. Tools are grouped according to type.

- SAST Tools Mapping
- DAST Tools Mapping
- IAST Tools Mapping
- Mobile Tools Mapping
- InfraSec Tools Mapping
- Threat Modeling Tools Mapping
- Component Tools Mapping
- Container Tools Mapping
- Cloud Infrastructure Tool Mapping
- Infrastructure as Code (IaC) Tool Mapping
- WAF Tools Mapping
- STIG Tools Mapping

SAST Tools Mapping

The tables below show the severity and triage status mappings for all of the SAST tools that are supported by Software Risk Manager.

Tools are listed alphabetically. Tool results are mapped to the Software Risk Manager status shown at the top of each column. (A blank cell indicates that an equivalent status value is unavailable or undefined.)

Severity Mapping

Table 12:

SAST Tools	Critical	High	Medium	Low	Info	Unspecified
Android Lint[1]		Security	Correctness; Correctness: Messages, Fatal; Correctness: Messages, Error; Internationalization, Fatal; Internationalization, Error; Bi-directional Text, Fatal; Bi-directional Text, Error	Correctness: Messages Warning, Performance	Usability, Topography; Usability, Icons; Usability; Accessibility; Internationalization, Warning; Bi-directional Text, Warning	
Armorize CodeSecure		HIGH	MEDIUM	LOW		
Brakeman						
Checkmarx (SAST)	4	High / 3	Medium / 2	Low / 1	Info / 0, Information	Unspecified, Unknown
Checkmarx One (SAST)	CRITICAL	HIGH	MEDIUM	LOW	INFO	
Checkstyle						
Clang						
Clang (CodeChecker)	CRITICAL	HIGH	MEDIUM	LOW	STYLE	
Clippy		error	warning	note / failure-note	help	none
CodePeer		high	medium	low		
CodeSonar-Scrape[2]		Red	Yellow	Green		
Continuous Dynamic (formerly	urgent, critical	high	medium	low	informational, note	

SAST Tools	Critical	High	Medium	Low	Info	Unspecified
WhiteHat) - (Legacy Rating System)						
Continuous Dynamic (formerly WhiteHat) - (Advanced Rating System)	Critical	High	Medium	Low	Note	
CppCheck		error	performance, warning	portability, style	information	none
Coverity	Very High / Critical	Major / High	Moderate / Medium	Minor / Low	Audit, Very Low	
Coverity On Polaris	critical	high	medium	low	audit	
42Crunch	5	4	3	2	1	
DefenseCode ThunderScan	critical	high	medium	low	informational	
ErrCheck			all			
error-prone						
ESLint						
Fortify[3]	impact >= 2.5 and likelihood >= 2.5	impact >= 2.5	likelihood >= 2.5	likelihood < 2.5 and impact < 2.5		
Fortify Software Security Center[***]	impact >= 2.5 and likelihood >= 2.5	impact >= 2.5	likelihood >= 2.5	likelihood < 2.5 and impact < 2.5		
Gendarme						
GitLab Security	critical	high	medium	low	informational	
GoCyclo				all		
GoLint						
GoSec		HIGH	MEDIUM	LOW		
HCL AppScan Source	critical	high	medium	low	informational	
HCL AppScan	critical	high	medium	low	informational	

SAST Tools	Critical	High	Medium	Low	Info	Unspecified
on Cloud (ASoC)						
Helix QAC		7 (Undefined behavior), 8 (Language constraints)	3 (Important issue), 4 (Local criteria), 5 (Data flow analysis), 6 (Portability)	2 (Minor issue)	0 (Information), 1 (Obsolete message), 9 (Error)	
IneffAssign				all		
JLint						
JSHint				all		
Microsoft Code Analysis						
MobSF		dangerous, insecure, high	medium, warning		normal, signature, info, good	
MobFS Scan		ERROR	WARNING		INFO	
NDepend	Critical	High	Medium	Low	Info	
OCLint						
Orca Security (Secret Scans)	CRITICAL	HIGH	MEDIUM	LOW	INFO	
Parasoft JTest / C ++Test / dotTest		Level 1: Severe Violation; Level 2: Possible Severe Violation	Level 3: Violation	Level 4: Possible Violation; Level 5: Informational		
PHPMD		1, 2	3	4, 5		
PMD		1, 2	3	4, 5		
Polaris	critical	high	medium	low	informational	
Pylint						
Rapid Scan SAST	critical	high	medium	low	informational	
SafeSQL		all				
SARIF	severe / critical	high / error	medium / moderate	low / warning	note / info / informational	

SAST Tools	Critical	High	Medium	Low	Info	Unspecified
SATE		1, 2	3	4, 5		
Scalastyle						
Scan@Source	critical	high	medium	low	informational	
SCARF						
Semgrep		high	medium	low		
Snyk Code	critical	high	medium	low		
SonarQube / SonarCloud	BLOCKER / CRITICAL	MAJOR / HIGH	MEDIUM	MINOR / LOW	INFO	
SpotBugs / FindBugs		1	2	3		
Staticcheck						
TruffleHog	Verified = true; Verified = false AND Detector name = Oauth, AWS, or Heroku	Verified = false AND Detector Name = PrivateKey	Verified = false AND Detector Name = Generic Secret			Verified = false AND Detector Name = Unspecified
Veracode		4	3	2	1	
Vet						

1. Android Lint evaluates risk based on both a category and a severity level. Categories are indicated by an asterisk.

2. CodeSonar reports risk through a combination of a ranking formula and an analysis warning system (red, yellow, green). Software Risk Manager uses the red, yellow, and green statuses to map to high, medium, and low, respectively.

3. Fortify reports risk by creating scores for “impact” and “likelihood.” The combination of these scores is then mapped to the Software Risk Manager severity levels.

Triage Status Mapping

Table 13:

SAST Tools	Ignored	False Positive	To Be Fixed	Mitigated	Fixed	Reopened
Android Lint						
Armorize CodeSecure						
Brakeman						
Checkmarx (SAST)	NOT_EXPLOITABLE / 1	False Positive	URGENT / 3; CONFIRMED / 2			

SAST Tools	Ignored	False Positive	To Be Fixed	Mitigated	Fixed	Reopened
Checkmarx One (SAST)	NOT_EXPLOITABLE; PROPOSED_NOT_EXPLOITABLE		URGENT; CONFIRMED			
Checkstyle						
Clang						
Clang (CodeChecker)	intentional	false_positive, suppress	confirmed			
Clippy						
CodePeer	not a bug	false positive				
CodeSonar-Scrape**						
Continuous Dynamic (formerly WhiteHat)	accepted, out of scope	Invalid, false		open, mitigated		
CppCheck						
Coverity	Intentional, ignore	False Positive				
Coverity On Polaris	DISMISSED INTENTIONAL, DISMISSED OTHER	FALSE POSITIVE	TO BE FIXED			
42Crunch						
DefenseCode ThunderScan		false positive				
ErrCheck						
error-prone						
ESLint						
Fortify***	Suppressed, Not an Issue		Exploitable, Suspicious, Reliability Issue, Bad Practice			
Fortify Software Security Center***	Suppressed, Not an Issue		Exploitable, Suspicious, Reliability Issue, Bad Practice			
Gendarme						

SAST Tools	Ignored	False Positive	To Be Fixed	Mitigated	Fixed	Reopened
GitLab Security						
GoCyclo						
GoLint						
GoSec						
HCL AppScan Source		noise		passed	fixed	reopened
HCL AppScan on Cloud (ASoC)		noise		passed	fixed	reopened
Helix QAC						
IneffAssign						
JLint						
JSHint						
Microsoft Code Analysis						
MobSF						
MobFS Scan						
NDepend						
OCLint						
Orca Security (Secret Scans)						
Parasoft JTest / C ++Test / dotTest						
PHPMD						
PMD						
Polaris	dismissed (any other reason)	dismissed (false positive)	to-be-fixed			
Pylint						

SAST Tools	Ignored	False Positive	To Be Fixed	Mitigated	Fixed	Reopened
Rapid Scan SAST						
SafeSQL						
SARIF						
SATE						
Scalastyle						
Scan@Source						
SCARF						
Semgrep					fixed	
Snyk Code	ignored					
SonarQube / SonarCloud	WON'T FIX, SAFE	FALSE POSITIVE	ACKNOWLEDGED		FIXED	REOPENED
SpotBugs / FindBugs						
Staticcheck						
TruffleHog						
Veracode	Accept the Risk	Potential False Positive	Reported to Library Maintainer	Mitigate by Design, Mitigate by Network Environment, Mitigate by OS Environment		
Vet						

For SRM Triage Status definitions, click here.

DAST Tools Mapping

The tables below show the severity and triage status mappings for all of the DAST tools that are supported by Software Risk Manager.

Tools are listed alphabetically. Tool results are mapped to the Software Risk Manager status shown at the top of each column. (A blank cell indicates that an equivalent status value is unavailable or undefined.)

Severity Mapping

Table 14:

DAST Tool	Critical	High	Medium	Low	Info
Acunetix Desktop		high	medium	low	info

DAST Tool	Critical	High	Medium	Low	Info
Acunetix 360	CRITICAL	IMPORTANT, HIGH	MEDUIM	LOW	INFORMAT (BEST PRACTICE
AppSpider Vulnerability Summary		4	5	6	1, 0
APIsec	Blocker, Critical	Major, High	Medium	Minor, Low	Info
Arachni		high	medium	low	information
Burp Suite		high	medium	low	information
Continuous Dynamic (formerly WhiteHat)	urgent (critical)	high		low	note (informatio
Defensics		fail	warning		
Dynatrace					
HP WebInspect	4	3	2	1	0
HCL AppScan Standard (enterprise)	Critical	High	Medium	Low	Information
HCL AppScan on Cloud (ASoC)	Critical	High	Medium	Low	Information
Imperva*	CRITICAL	MAJOR	MINOR		
Invicti Standard (formerly Netsparker)		Critical, Important, High	Medium	Low	Information Practice)
Invicti Enterprise (formerly Netsparker Enterprise)	Critical	Important, High	Medium	Low	Information Practice)
OWASP ZAP		3	2	1	0
Polaris	critical	high	medium	low	information
Qualys WAS	5	4	3	2	1
Rapid7 InsightAppSec					
Rapid7 InsightVM	Critical	Severe	Moderate		
Rapid7 Nexpose	8-...-10	4-...-7	0-...-3		
Black Duck Managed Services Platform	Critical	High	Medium	Low	Minimal
Tenable WAS	blocker / critical	major / high	medium	minor / low	info
Tinfoil Web	critical	high	medium	low	information
Trustwave App Scanner		High	Medium	Low	all other va
Veracode		4	3	2	1
WPScan		all			

DAST Tool	Critical	High	Medium	Low	Info
Sqlmap output		all			

Triage Status Mapping

Table 15:

DAST Tool	Ignored	False Positive	To Be Fixed	Mitigated	Fixed	Re
Acunetix Desktop						
Acunetix 360	Accepted Risk	False Positive			Fixed	
AppSpider Vulnerability Summary						
APIsec						
Arachni						
Burp Suite						
Continuous Dynamic (formerly WhiteHat)	accepted, out of scope	Invalid, false		open, mitigated		
Defensics						
Dynatrace						
HP WebInspect						
HCL AppScan Standard (enterprise)		noise		passed	fixed	reo
HCL AppScan on Cloud (ASoC)		noise		passed	fixed	reo
Imperva*						
Invicti Enterprise (formerly Netsparker Enterprise)	Accepted Risk	False Positive			Fixed	
OWASP ZAP						
Polaris	dismissed (any other reason)	dismissed (false positive)	to-be-fixed			
Qualys WAS						
Rapid7 InsightAppSec	ignored	false positive	verified		remediated	
Rapid7 InsightVM						
Rapid7 Nexpose						
Black Duck Managed Services Platform		False Positive				
Tenable WAS						

DAST Tool	Ignored	False Positive	To Be Fixed	Mitigated	Fixed	Rec
Tinfoil Web						
Trustwave App Scanner						
Veracode	Accept the Risk	Potential False Positive	Reported to Library Maintainer	Mitigate by Design, Mitigate by Network Environment, Mitigate by OS Environment		
WPScan						
Sqlmap output						

For SRM Triage Status definitions, click here.

*Imperva only produces severities for API Attack Analytics results and not for API Risks or WAF Security Events. API Risk and WAF Security Event findings will only have Unspecified severity in SRM.

IAST Tools Mapping

The tables below show the severity and triage status mappings for all of the IAST tools that are supported by Software Risk Manager.

Tools are listed alphabetically. Tool results are mapped to the Software Risk Manager status shown at the top of each column. (A blank cell indicates that an equivalent status value is unavailable or undefined.)

Severity Mapping

Table 16:

IAST Tool	Critical	High	Medium	Low	Info	Unspecified
Checkmarx (IAST)	Critical / 4	High / 3	Medium / 2	Low / 1	Informational / 0	Unspecified, Unknown
Contrast	Critical	High	Medium	Low	Note	
HCL AppScan on Cloud	Critical	High	Medium	Low	Information	
NowSecure Workstation						
Q-MAST	CRITICAL	HIGH	MEDIUM	LOW		
Black Duck Seeker	critical	high	medium	low	informational	

Triage Status Mapping

Table 17:

IAST Tool	Ignored	False Positive	To Be Fixed	Mitigated	Fixed	Reopened
Checkmarx (IAST)		NOT_A_PROBLEM	CONFIRMED	REMEDIATED		
Contrast	URL access limited or internal security control	False Positive	Confirmed or Suspicious	Remediated		
HCL AppScan on Cloud		noise		passed	fixed	reopened
NowSecure Workstation						
Q-MAST						
Black Duck Seeker	Ignored / Won't Fix / Intentional, Archived	False Positive			Fixed	

For SRM Triage Status definitions, click here.

Mobile Tools Mapping

The tables below show the severity and triage status mappings for all of the Mobile tools that are supported by Software Risk Manager.

Tools are listed alphabetically. Tool results are mapped to the Software Risk Manager status shown at the top of each column. (A blank cell indicates that an equivalent status value is unavailable or undefined.)

Severity Mapping

Table 18:

Mobile Tool	Critical	High	Medium	Low	Info	Unspecified
Data Theorem Mobile Secure	critical	high	medium	low	information	
HCL AppScan on Cloud (ASoC)	Critical	High	Medium	Low	Information	
MobSF						

Mobile Tool	Critical	High	Medium	Low	Info	Unspecified
MobSF Scan						
NowSecure AUTO						
NowSecure INTEL						
NowSecure Workstation	critical	high	medium	low	info	unknown

Triage Status Mapping

Table 19:

Mobile Tool	Ignored	False Positive	To Be Fixed	Mitigated	Fixed	Reopened
Data Theorem Mobile Secure						
HCL AppScan on Cloud (ASoC)		noise		passed	fixed	reopened
MobSF						
MobSF Scan						
NowSecure AUTO						
NowSecure INTEL						
NowSecure Workstation						

For SRM Triage Status definitions, click here.

InfraSec Tools Mapping

The tables below show the severity and triage status mappings for all of the InfraSec tools that are supported by Software Risk Manager.

Tools are listed alphabetically. Tool results are mapped to the Software Risk Manager status shown at the top of each column. (A blank cell indicates that an equivalent status value is unavailable or undefined.)

Severity Mapping

Table 20:

InfraSec Tool	Critical	High	Medium	Low	Info	Unspecified
AppDetective Pro		high	medium	low	informational	
Tenable Nessus[1]	4	3 or Cat I	2 or Cat II	1	0 or Cat III	
Rapid7 Nexpose						
NMap[2]	9+	7-...< 9	4-...< 7	< 4		
Qualys VM	5	4	3	2	1	
Qualys CS	5	4	3	2	1	
SCAP		High	Medium	Low	Info	
Qualys VMDR	5	4	3	2	1	

1. Tenable Nessus reports risk through a "category" ranking (1-3) and a severity level (0-4).

2. NMap reports risk using a CVSS score.

Triage Status Mapping

Table 21:

InfraSec Tool	Ignored	False Positive	Fixed	Mitigated	Fixed	Reopened
AppDetective Pro						
Tenable Nessus						
Rapid7 Nexpose						
NMap						
Qualys VM						
Qualys CS						
SCAP						
Qualys VMDR						

For SRM Triage Status definitions, click here.

Threat Modeling Tools Mapping

The tables below show the severity and triage status mappings for all of the Threat Modeling tools that are supported by Software Risk Manager.

Tools are listed alphabetically. Tool results are mapped to the Software Risk Manager status shown at the top of each column. (A blank cell indicates that an equivalent status value is unavailable or undefined.)

Severity Mapping

Table 22:

Threat Modeling Tool	Critical	High	Medium	Low	Info	Unspecified
IriusRisk[1]	Critical (76-100)	High (51–75)	Medium (26–50)	Low (1–25)	Very Low (0)	
Microsoft Threat Modeling Tool 2016		high	medium	low		
SD Elements	9+	7–8	5–6	1–4		

1. IriusRisk threats are assigned a severity in SRM based on their Current Risk value. The risk ratings are mapped to SRM severities based on this mapping.

Triage Status Mapping

Table 23:

Threat Modeling Tool	Ignored	False Positive	To Be Fixed	Mitigated	Fixed	Reopened
IriusRisk						
Microsoft Threat Modeling Tool 2016	Not Applicable			Mitigation Implemented		
SD Elements						

For SRM Triage Status definitions, click here.

Component Tools Mapping

The tables below show the severity and triage status mappings for all of the Component tools that are supported by Software Risk Manager.

Tools are listed alphabetically. Tool results are mapped to the Software Risk Manager status shown at the top of each column. (A blank cell indicates that an equivalent status value is unavailable or undefined.)

Severity Mapping

Table 24:

Component Tool	Critical	High	Medium	Low	Info	Unspecified	None
Black Duck Binary Analysis[1] **CVSSv3 mapping **CVSSv2 mapping*	>=9*	>=7* , >=7**	<=4* , <=4**	>0* , <=0**	=0*	<0* , <0**	
Black Duck Hub	CRITICAL, BLOCKER	HIGH, MAJOR	MEDIUM, MINOR	LOW, TRIVIAL		UNKNOWN, UNSPECIFIED	
CAST Highlight	critical	high	medium	low	advisory		
Checkmarx One (SCA)	CRITICAL	HIGH	MEDIUM	LOW	INFO		
Continuous Dynamic (formerly WhiteHat) - (Legacy Rating System)	urgent, critical	high	medium	low	informational, note		
Continuous Dynamic (formerly WhiteHat) - (Advanced Rating System)	Critical	High	Medium	Low	Note		
Dependency-check	critical	high	medium or moderate	low	informational	unknown	none
Dependency-Track	critical	high / fail	warn / medium	low			none
Dynatrace[2]	CRITICAL	HIGH	MEDIUM	LOW			NONE
GitHub Security	CRITICAL	HIGH	MODERATE / medium				
GitLab Security	critical	high	medium	low	informational		
JFrog Xray	critical	high	medium	low			

Component Tool	Critical	High	Medium	Low	Info	Unspecified	None
NeuVector	critical	high / error	medium / warn	low / note			
Orca Security (Vulnerabilities Scan)	CRITICAL	HIGH	MEDIUM	LOW	INFO		
Polaris	critical	high	medium	low	informational		
Retire.js		high	medium	low			
Snyk Open Source	critical	high	medium	low			
Snyk License Compliance Management	critical	high	medium	low	informational		
Sonatype Nexus	critical	severe	moderate	low	no threat, none		
Veracode		4	3	2	1		
WhiteSource		high, Rejected by policy	medium	low, Multiple licenses, Multiple library versions, New library version	License results		

1. To use CVSS version 3 mapping for CVSS version 2 scores, set `cvss.use-cvss3-buckets = true` in the SRM props file.

2. Dynatrace only produces severities for Vulnerability results and not for Attack results. Dynatrace Attack findings will have no severity in SRM.

Triage Status Mapping

Table 25:

Component Tool	Ignored	False Positive	To Be Fixed	Mitigated	Fixed	Reopened
Black Duck Binary Analysis				FD (feature disabled)	VP (vendor patched)	
Black Duck Hub	Duplicate, Ignored	Not Affected	Affected	Mitigated	Remediation Complete	
CAST Highlight						

Component Tool	Ignored	False Positive	To Be Fixed	Mitigated	Fixed	Reopened
Checkmarx One (SCA)	NOT_EXPLOITABLE; PROPOSED_NOT_EXPLOITABLE		URGENT; CONFIRMED			
Continuous Dynamic (formerly WhiteHat)	accepted, out of scope	Invalid, false		open, mitigated		
Dependency-check						
Dependency-Track	not affected, suppressed	false positive				
Dynatrace					RESOLVED	
GitHub Security					CLOSED	
GitLab Security						
JFrog Xray						
NeuVector						
Orca Security (Vulnerabilities Scan)						
Polaris	dismissed (any other reason)	dismissed (false positive)	to-be-fixed			
Retire.js						
Snyk Open Source	Ignored				Patched	
Snyk License Compliance Management	Ignored					
Sonatype Nexus	Not Applicable		Confirmed			
Vericode	Accept the Risk	Potential False Positive	Reported to Library Maintainer	Mitigate by Design, Mitigate by Network Environment, Mitigate by OS Environment		

Component Tool	Ignored	False Positive	To Be Fixed	Mitigated	Fixed	Reopened
WhiteSource						

For SRM Triage Status definitions, click here.

Container Tools Mapping

The tables below show the severity and triage status mappings for all of the Container tools that are supported by Software Risk Manager.

Tools are listed alphabetically. Tool results are mapped to the Software Risk Manager status shown at the top of each column. (A blank cell indicates that an equivalent status value is unavailable or undefined.)

Severity Mapping

Table 26:

Container Tool	Critical	High	Medium	Low	Info	Unspecified
Anchore	critical	high	medium	low	negligible	
Aqua CSP	critical	malware, high	sensitive data, medium	low	negligible	unknown
Check Point CloudGuard	Critical	High	Medium	Low	Informational	
Dynatrace*	CRITICAL	HIGH	MEDIUM	LOW		
Grype Reader	Critical	High	Medium	Low		Unknown
GitLab Security	critical	high	medium	low	informational	
Harbor	Critical	High	Medium	Low	Negligible	Unknown
Google SCC	critical	high	medium	low		
Microsoft Defender for Cloud	Critical	High	Medium	Low		
NeuVector	critical	high / error	medium / warn	low / note		
Orca Security	CRITICAL	HIGH	MEDIUM	LOW	INFO	
Prisma Cloud Compute (Twistlock)	critical / important	high	medium / moderate	low		
Snyk Container	critical	high	medium	low		

Container Tool	Critical	High	Medium	Low	Info	Unspecified
Trivy	CRITICAL	HIGH	MEDIUM	LOW		UNKNOWN

*Dynatrace only produces severities for Vulnerability results and not for Attack results. Dynatrace Attack findings will have no severity in SRM.

Triage Status Mapping

Table 27:

Container Tool	Gone	New	Ignored	False Positive	To Be Fixed	Mitigated	Fixed
Anchore							
Aqua CSP							
Check Point CloudGuard							
Dynatrace*							RESOLVED
Grype Reader	Fixed	not-fixed, unknown	wont-fix				
GitLab Security							
Harbor							
Google SCC			MUTED				
Microsoft Defender for Cloud							
NeuVector							
Orca Security							
Prisma Cloud Compute (Twistlock)							
Snyk Container			ignored				patched
Trivy							

For SRM Triage Status definitions, click here.

Cloud Infrastructure Tool Mapping

The tables below show the severity and triage status mappings for all of the Cloud Infrastructure tools that are supported by Software Risk Manager.

Tools are listed alphabetically. Tool results are mapped to the Software Risk Manager status shown at the top of each column. (A blank cell indicates that an equivalent status value is unavailable or undefined.)

Severity Mapping

Table 28:

Cloud Infrastructure Tool	Critical	High	Medium	Low	Info	Unspecified
Prisma Cloud (RedLock)	critical	high	medium	low	informational	
AWS Security Hub*	critical, 80+	high, 60–...–79	medium, 40–...–59	low, 20–...–39	informational, 0–...–19	
Azure Security Center	critical	high	medium	low	informational	
Check Point CloudGuard	Critical	High	Medium	Low	Informational	
Google SCC	critical	high	medium	low		
Microsoft Defender for Cloud	Critical	High	Medium	Low		
Wiz	CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL	

*AWS reports risk through a ranking [1–100] and a severity level [low, medium, etc.]. Both are listed.

Triage Status Mapping

Table 29:

Cloud Infrastructure Tool	Ignored	False Positive	To Be Fixed	Mitigated	Fixed	Reopened
Prisma Cloud (RedLock)	Snoozed					
AWS Security Hub*	suppressed		notified		resolved	
Azure Security Center						

Cloud Infrastructure Tool	Ignored	False Positive	To Be Fixed	Mitigated	Fixed	Reopened
Check Point CloudGuard						
Google SCC	MUTED					
Microsoft Defender for Cloud						
Wiz						

For SRM Triage Status definitions, click here.

Infrastructure as Code (IaC) Tool Mapping

The tables below show the severity and triage status mappings for all of the IaC tools that are supported by Software Risk Manager.

Tools are listed alphabetically. Tool results are mapped to the Software Risk Manager status shown at the top of each column. (A blank cell indicates that an equivalent status value is unavailable or undefined.)

Severity Mapping

Table 30:

IaC Tool	Critical	High	Medium	Low	Info	Unspecified
Checkmarx One (IaC)	CRITICAL	HIGH	MEDIUM	LOW	INFO	
Checkov	CRITICAL	HIGH	MEDIUM	LOW		
Orca Security	CRITICAL	HIGH	MEDIUM	LOW	INFO	

Triage Status Mapping

Table 31:

IaC Tool	Ignored	False Positive	To Be Fixed	Mitigated	Fixed	Reopened
Checkmarx One (IaC)	NOT_EXPLOITABLE; PROPOSED_NOT_EXPLOITABLE		URGENT; CONFIRMED			
Checkov						
Orca Security						

For SRM Triage Status definitions, click here.

WAF Tools Mapping

The tables below show the severity and triage status mappings for all of the WAF tools that are supported by Software Risk Manager.

Tools are listed alphabetically. Tool results are mapped to the Software Risk Manager status shown at the top of each column. (A blank cell indicates that an equivalent status value is unavailable or undefined.)

Severity Mapping

Table 32:

WAF Tool	Critical	High	Medium	Low	Info	Unspecified
Imperva (WAF)						

Triage Status Mapping

Table 33:

WAF Tool	Ignored	False Positive	To Be Fixed	Mitigated	Fixed	Reopened	None
Imperva (WAF)							

For SRM Triage Status definitions, click here.

STIG Tools Mapping

The tables below show the severity and triage status mappings for all of the STIG tools that are supported by Software Risk Manager.

Tools are listed alphabetically. Tool results are mapped to the Software Risk Manager status shown at the top of each column. (A blank cell indicates that an equivalent status value is unavailable or undefined.)

Severity Mapping

Table 34:

STIG Tool	Critical	High	Medium	Low	Info	Unspecified
STIG		CAT I (high)	CAT II (medium)	CAT III (low)		

Triage Status Mapping

Table 35:

STIG Tool	Ignored	False Positive	To Be Fixed	Mitigated	Fixed	Reopened	None
STIG	Not A Finding, Not Applicable		Open				Not Reviewed

For SRM Triage Status definitions, click here.

Tool First Seen Date

The "first seen" date reflects the earliest date that a finding was imported (or *seen*) by SRM, which will either be the date the finding was first observed in SRM or the earliest date reported by a supported tool.

The following tools are supported and will provide a "first-seen" date:

- ApiSec
- Aqua
- ASoC
- AWS Security Hub
- CloudGuard
- Continuous Dynamic (formerly WhiteHat)
- Coverity
- Coverity on Polaris
- Dynatrace
- GitHub Security
- HackerOne
- Polaris
- Qualys WAS
- Seeker
- SonarQube
- Tenable.sc
- Twistlock
- Wiz

Correlation Overview

Correlation is the process Software Risk Manager uses to evaluate data returned by any combination of supported AppSec tools to determine which, if any, of the results reported by the various tools refers to the same issue. When matching results are found, Software Risk Manager correlates those results and creates a single finding for that issue. Software Risk Manager does this by looking at the data provided for each result; although, in some cases, the correlation process will factor in the detection method as well (e.g., Static vs. Dynamic results).

Correlation involves various processes that check for data within results that would suggest whether results should be correlated. The output is a per-result set of associations, where each association indicates whether to allow or deny correlation. At the end of this process, the correlation decisions for all results are used to determine which results have enough evidence to be grouped into the same finding. (For more information, see Analysis Correlation Options.)

The correlation process includes the following elements:

- Rule Sets
- Data Normalization
- Location-based Correlation

- Component Correlation
- Hybrid Correlation
- InfraSec Correlation
- Location-less Correlation

Understanding Rule Sets

In performing cross-tool analysis, results that do not have matching data can still refer to the same issue. In this case, correlation depends on rule sets. A rule set consists of multiple rules (e.g., specific tools and tool codes, categories of results in a specific tool, CWEs, etc.) that are used to evaluate results and create findings. For more information, see the section on Rule Sets.

Understanding Data Normalization

In broad terms, data normalization can be understood as the process of ordering, structuring, and simplifying the reported data that makes up a result. The purpose is to make it easier to determine if different data points refer to the same item. In other words, the correlation process does not evaluate the raw data provided by a result; instead, the evaluation is based on a normalized representation of that data.

Data that is displayed on the Findings page is the normalized representation of the available data. To view the raw data, you can open the specific finding and inspect the results attached to it.

Software Risk Manager performs the following normalizations:

- File paths*
- URL paths: Query parameters and anchors are stripped from URLs.*
- Component info: Component (package) names and versions are collected and stored as-is. CPE strings and certain package format strings (e.g., maven, npm) are parsed to collect this information, if available.
- Hosts: IP addresses, FQDNs, MAC addresses, NetBIOS names, and hostnames are collected when available, and the existence of a result that ties any of these values will lead to the values being associated as the same host. (For example, if a single result reports a vulnerability and specified `10.0.0.9` and `PRODENV` hostname, these two identifiers would be combined to the same normalized host.)

* Normalization involving paths will compare the structure of the available paths to discover overlaps and determine the correct location of a file with respect to the known structure. If a user uploads source code, the paths from the source code are used as the normalized path. If the source code isn't uploaded, the normalized path becomes the most specific path shared by all of the paths. For example, the paths `src/main/test.java` and `main/test.java` would be normalized to the same, most-specific path, which is `main/test.java`, because `src/` is not shared between them. Base paths that are common across all inputs may be stripped.

Understanding Location-based Correlation

Location data is essential for correlation. Results that have the same location type and value might be candidates for correlation, depending on the location type.

Location types are as follows:

- File, Logical paths: Line ranges must at least overlap. Results without a line range will only be correlated with other results without a line range, and vice-versa. Column ranges are ignored.
- URL paths: The "Element" for both results must match exactly. An "Element" refers to the type and name of the part of the request that was indicated as vulnerable. If a result indicates that the query

parameter "user" is vulnerable, that result will only correlate with other results with an Element of "Query Parameter ('user')".

Understanding Component Correlation

There are five ways to correlate components (for more information, see Analysis Correlation Options). Component correlation modes control how Software Component Analysis (SCA) tool results are correlated to findings. SRM can be configured to correlate component results using different combinations of the following data:

- vulnerability (e.g., BDSA-2021-0069, CVE-2021-24122)
- component name and version (e.g., Spring Framework version 3.2.4)
- type (e.g., Vulnerable Component)
- component identifier (e.g., org.springframework:spring-aop:3.2.8.RELEASE)

You will get one finding per unique set of values for each option. For example, selecting "vulnerability, component name/version, and type" will result in a finding for each vulnerability for each component and type that you have. However, if "vulnerability and type" is selected, you will only get a finding per vulnerability and type that covers all components. The default mode is "vulnerability, component identifier, and type."

Understanding Hybrid (SAST/DAST) Correlation

When Hybrid Correlation is enabled, URL-located results may be correlated with file-located results by mapping result URLs to a set of source code locations. Results with file and URL locations may be correlated if the file location overlaps with any of the discovered file locations for the given URL. (Data flows are also checked for overlaps.)

Source code is analyzed (for supported languages and frameworks) to determine the specific files and line ranges that declare the endpoint used by the URL path. If binaries are also provided, Software Risk Manager will automatically build a call graph from the indicated source location to collect additional locations to compare against. (Call graph generation is only supported for JVM and CLR binaries.)

Understanding InfraSec Correlation

In InfraSec correlation, results with matching Host information may be correlated if the list of CVEs is an exact match between the available results. Results with differing host information will be marked with a "do not correlate" flag, which will prevent correlation by any other process.

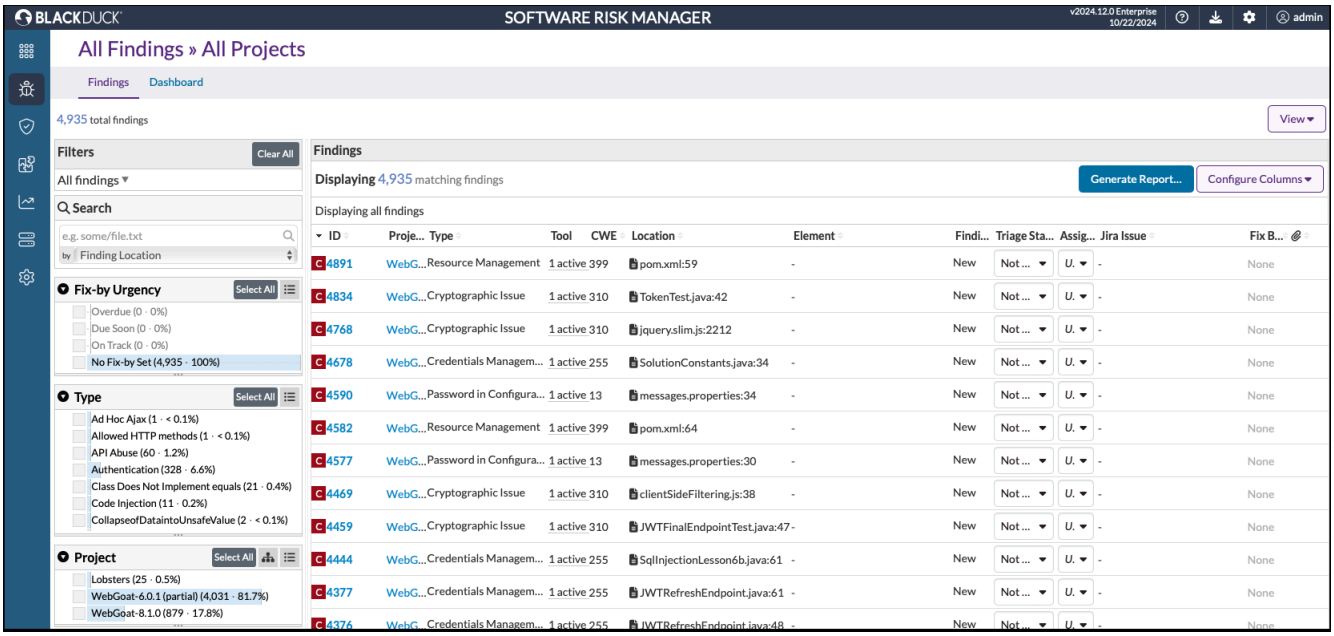
Understanding Location-less Correlation

Results that do not include location information can be correlated if the results have matching descriptors and detection method.

Findings Overview

The findings page displays all the findings associated with a particular project or projects, along with all the relevant supporting data.

Click the Findings icon in the navigation bar to open the Findings page.



You can generate a report of the findings by clicking the Generate Report button.

Findings View Options

The Findings page provides a variety of information and project options, as detailed below. You can customize what information is displayed by using the Configure Columns and View buttons or clicking the column headings.

- Click Configure Columns to select which columns to display. Click to add or remove checkmarks.

- Click View to select which findings to display. Use the toggles to display or hide information.

The viewing options are as follows:

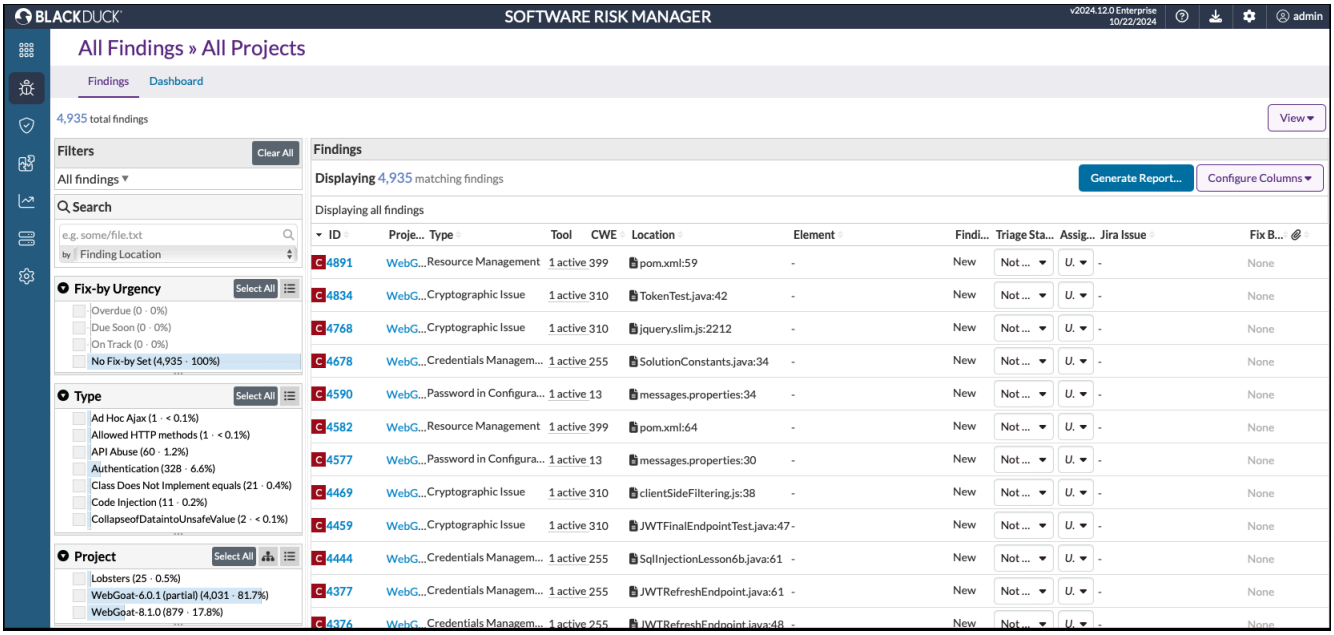
- Hide archived results.
- Hide findings marked as "Gone."
- Hide findings marked as "Fixed."
- Hide findings marked as "Mitigated."

- Hide findings marked as "Ignored."
- Hide findings marked as "False Positive."
  The purpose of "hiding" or "un-hiding" findings is to exclude or include the associated findings from the Findings page. The "completed" triage statuses in these options are "Gone," "Fixed," "Mitigated," "False Positive," and "Ignored." When turned ON, each setting will cause the findings marked with the particular triage status to be excluded from the page. This affects the table, filters, and counts throughout the Findings page. When turned OFF, the findings associated with that status will be included on the page. The default setting is ON.

- Click the column heading to re-sort the list.

Findings List

The Findings page displays all the findings from the project analysis and provides analysis data as well as links to additional information.



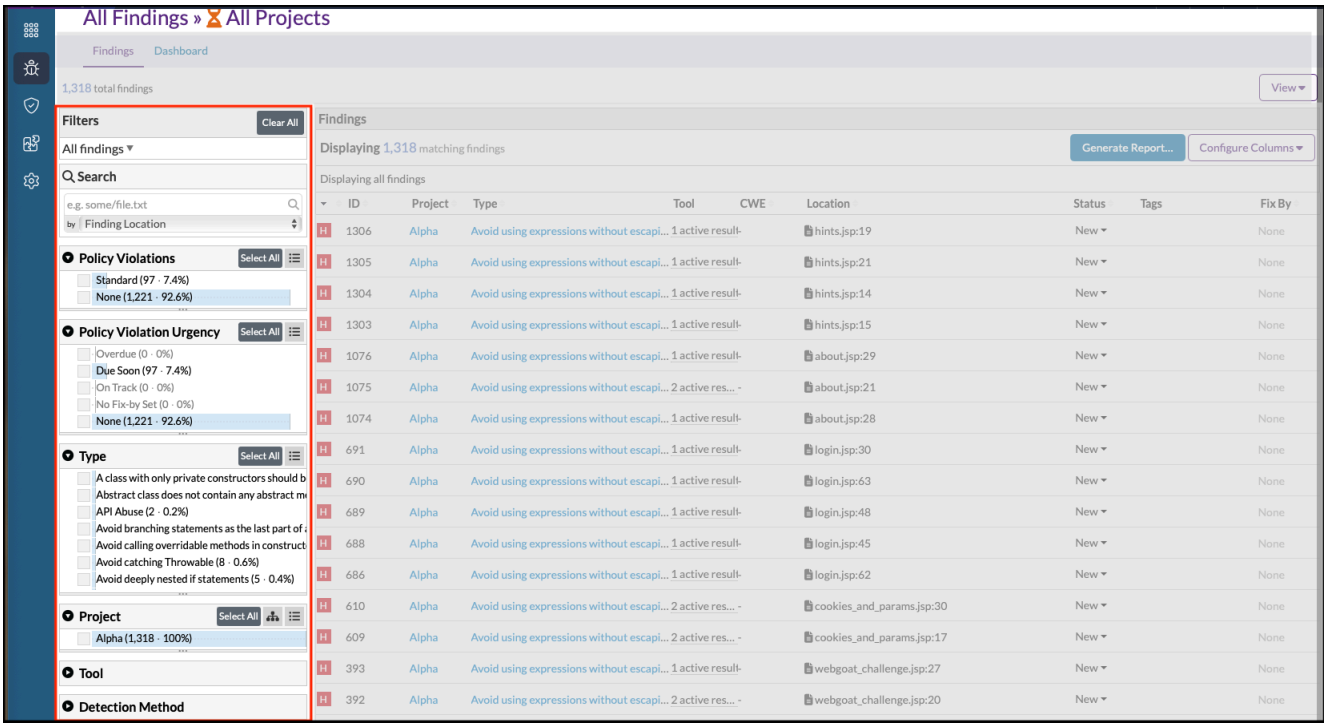
The analysis data is displayed in columns and includes the following information:

- **Severity** (icon). Displays the findings severity level.
- **Policy violation** (icon). Indicates a policy has been violated.
- **ID.** The ID of the finding.
- **Project** (displayed when viewing a list of findings for all projects). The name of the associated project. Click the link to open the Findings page for that specific project.
- **Type.** The finding type. Click the link to view details for that specific finding.
- **Tool.** The tool used to discover the finding.
- **CWE.** The finding's CWE data.
- **Location.** The location of the finding.
- **Finding Status.** The current status of the finding.
- **Triage Status.** The current triage status for the finding.

- **Assignee.** The person assigned to the finding.
- **Predicted Status.** Indicates the "predicted" status based on existing machine learning configuration settings.
- **Tags.** Any associated tags.
- **Fix By.** The fix-by date.
- **Attachments (icon).** Shows if there are any attachments (files) attached to the finding.

Using Search/Display Filters

Filters allow you to determine which findings to display and provide information about that subset of findings.



For more information on filters, see Searching Using Filters.

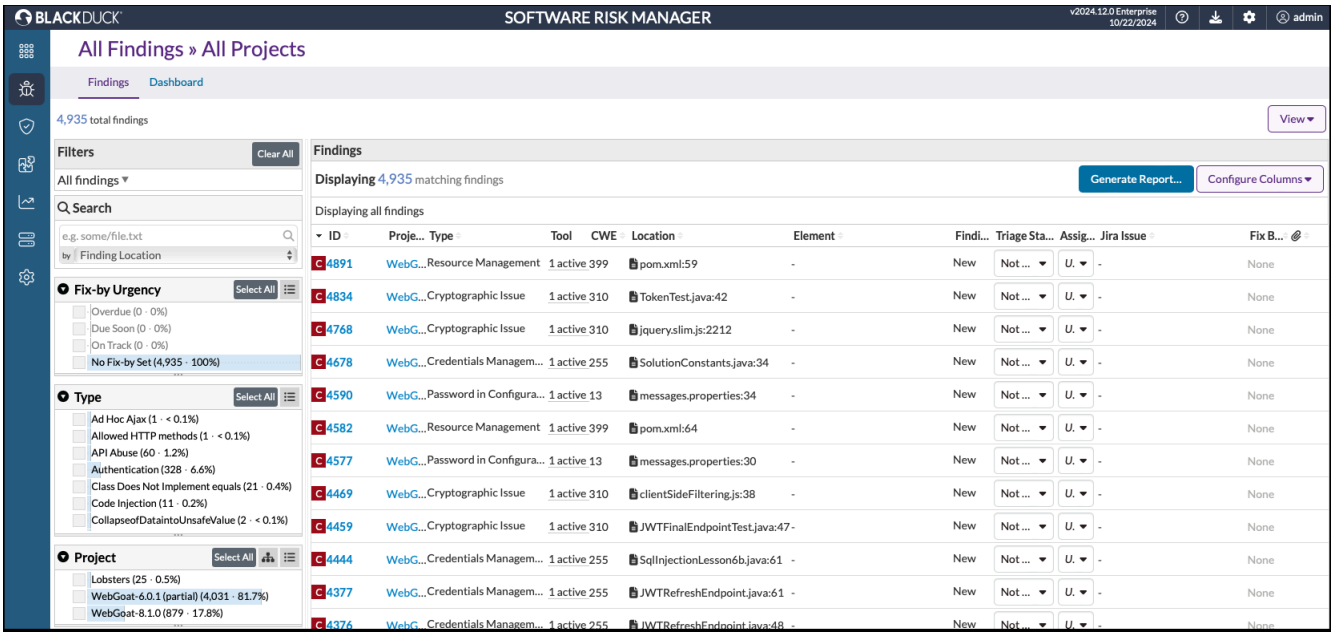
Additional Findings Options

The process of generating and viewing finding data involves a variety of tasks. For more information, see the following:

- Searching for Findings.
- Working with Filters.
- CWE Support.
- Performing Bulk Operations.
- Findings Table.
- Analysis Inputs List.
- Adding Manual Results.
- Using Machine Learning with Project Findings.

Searching for Specific Findings

Click the Findings icon in the navigation bar to open the Findings page.



You can search for a specific finding using the search field. Search options are located in the upper left corner of the page.

To search using the Search field:

1. Enter a search term in the search field.
2. Open the dropdown option menu and select the type of search you want to run. Options include the following:
   - Finding Location
   - Finding ID
   - CVE
   - CWE
   - Type/Tool
   - Host
   - Black Duck Component Policy Violations
   - Black Duck Exploit Available
   - Black Duck Project
   - Black Duck Solution Available
   - Black Duck Workaround Available
   - Brakeman Confidence
   - CPE
   - CVSS v2

- CVSS v2 Vector
- CVSS v3
- CVSS v3 Vector
- Checkmarx Path ID
- Coverity Merge Key
- Fortify Instance ID
- Tinfoil Api Issue ID
- Veracode App ID
- Veracode App Name
- Veracode Flaw ID
- Azure DevOps Work Item ID

Search results are displayed automatically.

For additional information on select search options, see the sections below.

Searching by the Finding's Location

The default "search by" option is Location, and search terms are case-sensitive. When searching by Location, the criteria can be any part of a file path. For example, to look for Findings in the *webapp/javascript* folder, enter `webapp/javascript`. To search Findings in files with the *.java* extension, enter `.java`. You can use `*` to indicate a wildcard: a search for `src/*.java` will match locations like `src/main/java/Example.java`. If you want to have the literal asterisk (`*`) as part of your search, use `*`. If you want to have a literal backslash (`\`) as part of your search, use `\`.

Searching by Finding ID

When searching by Finding ID, the same formatting rules apply as with the CWE search. To search for Finding 123, enter `123`. To search for Findings 123, 456, and 789, enter `123, 456, 789`. Note that the search will not look for Findings from other projects.

Searching by CWE ID

When searching by CWE, the criteria should be a number, or a comma-separated list of numbers. For example, to search for findings with a CWE of 91, simply enter `91`. To search for findings with a CWE of either 91 or 94, enter `91, 94`. Note that ranges (e.g., *100 - 200*) are currently not supported.

Searching by Type/Tool

When searching by Type / Tool, the criteria can be any text (case-insensitive) which may appear in the name or grouping of a Rule or Tool descriptor. For example, searching for "inject" by Type / Tool can match Rules like "SQL Injection," and Tool descriptors like *PMD / Security / Possible SQL Injection*. This search is case insensitive. Wildcards are not supported.

Working with Filters

Filters allow you to target which findings you want to display, along with providing specific information pertaining to the particular filter. For more information, see the following topics:

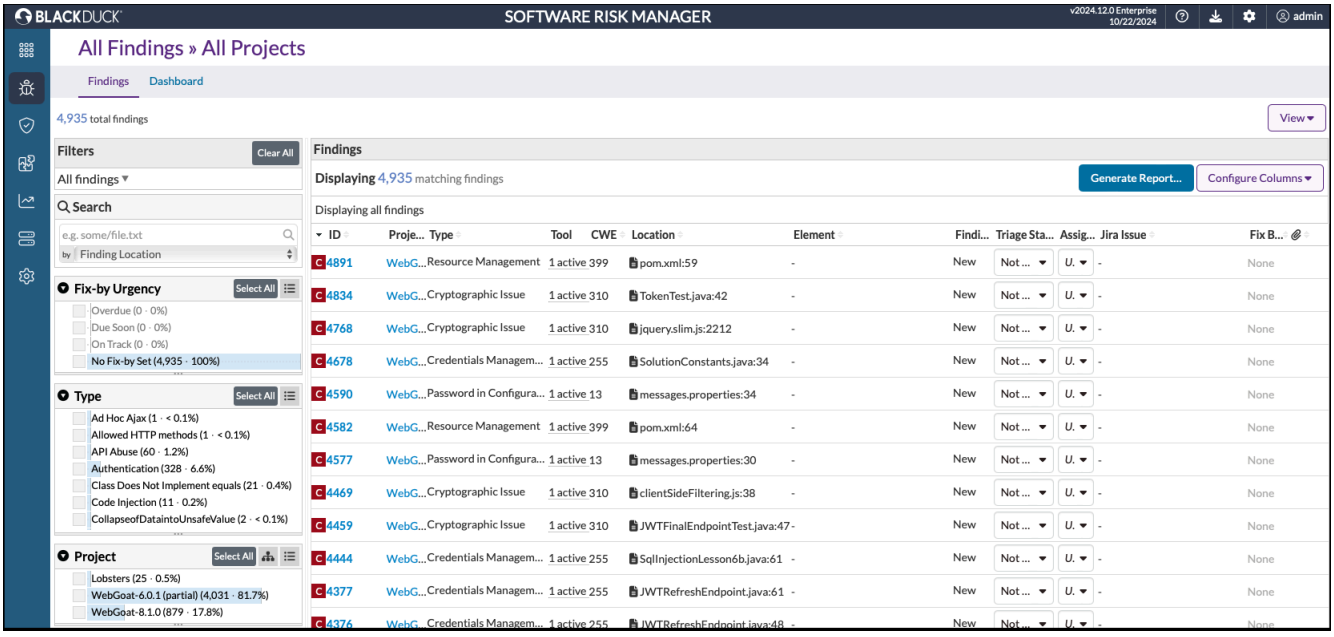
- Using Filters
- Configuring Filters

- Using Time Filters

Using Filters to Display Findings Data

Software Risk Manager provides a variety of filters and filter options that allow you to target specific information.

Click the Findings icon in the navigation bar to open the Findings page.



Filters are displayed to the left of the findings.

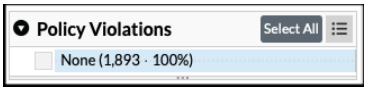
Clicking the arrow icon to the left of the filter name will expand the window and show the filter options and filter-specific information about the findings. Clicking any of the filter options will immediately apply those parameters to the list of findings.

The available filters include the following, and are detailed in the sections that follow:

- Policy Violations
- Fix-by Urgency
- Type
- Project
- Project Metadata
- Tool
- Detection Method
- Severity
- Location
- Container Image
- Age
- First Seen by SRM
- Date Modified
- Tool Overlaps
- Standards
- Tags
- Assignee
- Predicted Status
- Triage Status
- Pending Triage Status
- Finding Status
- Issue Tracker Association (if configured)
- Issue Tracker Resolution (if configured)

Policy Violations Filter

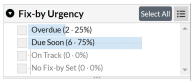
The Policy Violations filter allows you to filter findings based on existing policy violations. Expand the filter window and select which filter options you want to apply to the list of findings.



The filter window shows the number of findings and the percentage of violations compared to the total number of findings.

Fix-by Urgency Filter

The Fix-by Urgency filter allows you to filter findings based on urgency. Expand the filter window and select which filter options you want to apply to the list of findings.

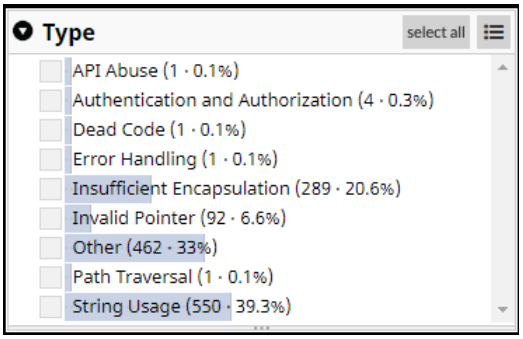


The filter window shows the number of findings for each of the following urgency levels:

- Overdue
- Due Soon
- On Track
- No Fix-by Set

Type Filter

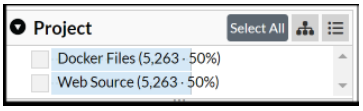
The Type filter allows you to filter findings based on finding type. Expand the filter window and select which filter options you want to apply to the list of findings.



The filter window shows the number of findings associated with a specific finding type.

Project Filter

The Project filter allows you to filter findings based on individual projects. Expand the filter window and select which filter options you want to apply to the list of findings.



Note: This filter only appears on "aggregated" versions of the Findings page, that is, for "All Projects," or for a project group with its members.

The filter window displays the projects associated with the findings in two ways:

- Projects displayed as a flat list.
- Projects displayed in a tree view.

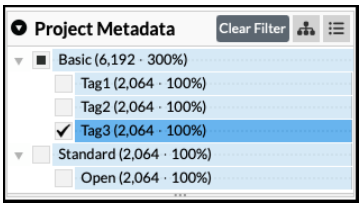
You can switch between display modes by selecting it from the first dropdown menu in the filter's header.

Project Metadata Filter

The Project Metadata filter allows you to filter findings based on defined metadata (see Adding and Configuring Project Metadata Fields).

Note: "Multiline" metadata is not supported.

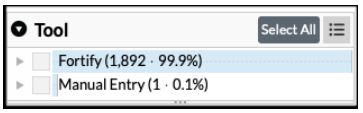
Expand the filter window and select which metadata options you want to apply to the list of findings.



Note: This filter only appears on "aggregated" versions of the Findings page, that is, for "All Projects."

Tool Filter

The Tool filter allows you to filter findings based on a specific tool's result types. Expand the filter window and select which filter options you want to apply to the list of findings.



The filter window displays a list of the tools used in the analysis and the number of findings associated with each tool. The tool result type hierarchy typically follows a hierarchy of "Tool" » "Category" » "Name," following the same hierarchy as in the Tool Config page.

Detection Method Filter

The Detection Method filter allows you to filter findings based on the detection method used to create the finding. Expand the filter window and select which filter options you want to apply to the list of findings.

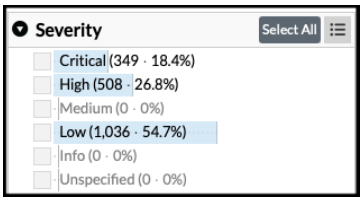
Note: Only the categories that apply to your project will be displayed.

The filter window lists the supported detection methods and displays the number of findings associated with each one:

- `Cloud Infrastructure Analysis`: Findings pertaining to cloud-hosted infrastructure
- `Component Analysis`: Third-party dependencies in your project that have known vulnerabilities
- `Container Analysis`: Findings within container runtimes or container images
- `Database Analysis`: Findings pertaining to databases
- `Dynamic Analysis`: Findings detected by Dynamic Application Security Testing (DAST) techniques
- `Hybrid Analysis`: Findings detected by multiple detection methods, for example, Static Analysis plus Dynamic Analysis
- `Interactive Analysis`: Findings detected by Interactive Application Security Testing (IAST) techniques
- `Network Analysis`: Findings pertaining to network infrastructure
- `Static Analysis`: Findings detected by Static Application Security Testing (SAST) techniques
- `Threat Modeling`: Findings detected by threat modeling techniques
- Plus any custom detection method

Severity Filter

The Severity filter allows you to filter findings based on the level of severity that is reported by a specific tool. Expand the filter window and select which filter options you want to apply to the list of findings.



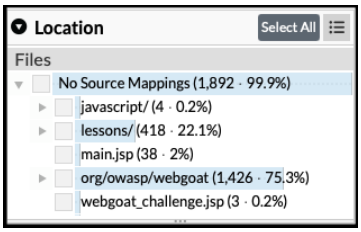
The filter window displays the number of findings belonging to each of the following risk categories:

- Info
- Low
- Medium
- High
- Critical
- Unspecified

For more information on risk mapping, see Tool Status and Severity Mapping.

Location Filter

The Location filter allows you to filter findings based on the finding's location. Expand the filter window and select which filter options you want to apply to the list of findings.



The filter window shows where each finding is located, reflecting the directory and file hierarchy of the codebase. Location categories that may apply to your project include files, URLs, and logical locations.

For .NET results, in some cases (especially if PDB files are not uploaded), source locations may not be available. Instead, a *Logical Locations*item will be shown, along with locations organized by namespace, class, and method.

Container Image Filter

The Container Image filter allows you to filter findings based on the names of container images that were discovered in Container Analysis results. Expand the filter window and select which filter options you want to apply to the list of findings.

Note: Images without an associated name are not shown in the filter.

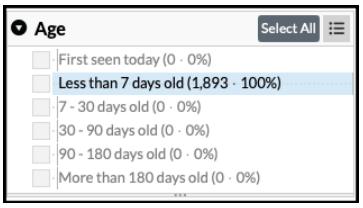
The filter window lists the supported container images and the findings associated with each.

Age Filter

The Age filter allows you to filter findings according to when that finding first appeared in an analysis. The Age filter calculates age based on tool-reported dates (when available). If the finding has no supported tools, the date reported is the first time the finding was seen by SRM. Note that the "first seen" date is fluid;

that is, if new data is ingested from a tool with an earlier date, the relevant findings will be updated to reflect this earlier date.

Expand the filter window and select which filter options you want to apply to the list of findings.

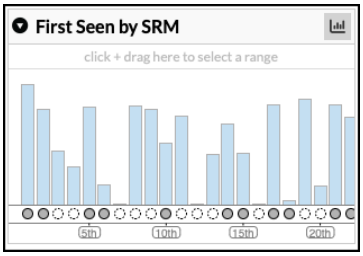


The filter window displays a set of pre-defined age ranges and the number of related findings.

First Seen by SRM Filter

The date the finding was first seen by SRM. Note: This date is not the same as the "first seen on" date for the finding, which is shown in the header of the Finding details page.

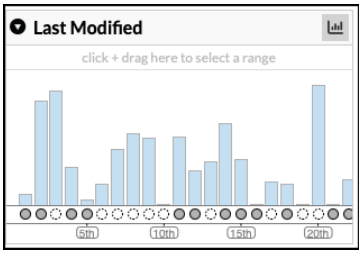
Note: This filter will not appear on "aggregated" versions of the Findings page, that is, for "All Projects," or for a project group with its members.



Last Modified Filter

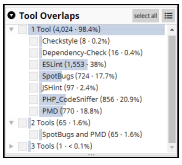
Displays when the finding was last modified.

Note: This filter will not appear on "aggregated" versions of the Findings page, that is, for "All Projects," or for a project group with its members.



Tool Overlaps Filter

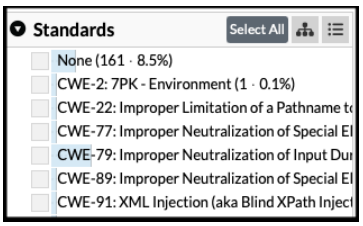
The Tool Overlaps filter allows you to filter findings based on correlation logic. Expand the filter window and select which filter options you want to apply to the list of findings.



The filter windows displays a breakdown of findings based on the degree of correlation of its associated tool results. For example, *Was a finding detected by 1 tool, 2 tools, or more?* Or *Were the 2 tools SpotBugs and PMD, or JSHint and PMD?* Actual correlation logic is determined by the project's Analysis Configuration.

Standards Filter

The Standards filter allows you to filter findings related to a specific industry standard. Expand the filter window and select one or more filter options to apply to the list of findings.



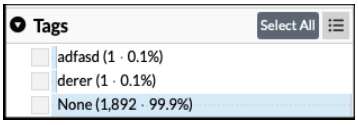
The filter window displays a list of the following standards and the number of findings related to each one:

- Architectural Concepts
- CERT C Secure Coding Standards
- CERT C++ Secure Coding Standards
- CERT Java Secure Coding Standards
- CISQ Quality Measures (2016)
- CISQ Quality Measures (2020)
- CLASP
- CWE All
- CWE Development Concepts View
- CWE Research Concepts View
- CWE Top 25 Most Dangerous Software Errors (2019)
- CWE Top 25 Most Dangerous Software Errors (2022)
- DISA STIG 3.10
- DISA STIG 4.10
- DISA STIG 5.1
- HIPAA
- Hardware Design
- MISRA C (2012)
- MISRA C++ (2008)
- NIST 800-53 Revision 4
- OWASP ASVS v4
- OWASP Mobile Top 10
- OWASP Top Ten (2013)
- OWASP Top Ten (2017)
- OWASP Top Ten (2019)

- OWASP Top Ten (2021)
- PCI DSS 3.1
- PCI DSS 4.0
- Seven Pernicious Kingdoms
- Software Fault Patterns
- WASC Threat Classification

Tags Filter

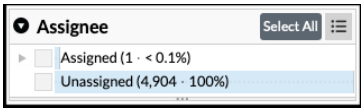
The Tags filter allows you to filter findings based on finding tags. Expand the filter window and select which filter options you want to apply to the list of findings.



The filter window shows the distribution of findings based on an assigned tag. Each number corresponds to the number of findings to which each tag has been assigned.

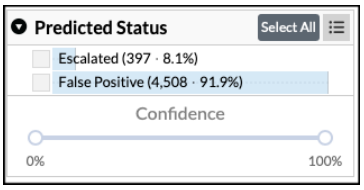
Assignee Filter

The Assignee filter allows you to filter findings based who was assigned to that finding. Expand the filter window and select which filter options you want to apply to the list of findings.



Predicted Status Filter (if configured)

The Predicted Status filter allows you to filter findings based on existing machine learning configuration settings. Expand the filter window and select which filter options you want to apply to the list of findings



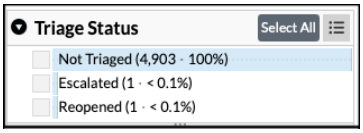
The Predicted Status filter is shown only if machine learning is enabled (see the Machine Learning Control Panel section).

Filtering options include filtering against findings with *Predicted Status* of To Be Fixed, False Positive, or Unknown, as well as filtering against *Prediction Confidence*, which ranges from 0 to 100 percent. Selecting multiple predicted statuses to filter on will include any finding that has any one of the selected predicted statuses. Selecting a sub range for prediction confidence will include any finding that has a predicted status matching one of the selected statuses as well as a prediction confidence that exists in the selected sub range (inclusively).

Note: This filter is only available in Software Risk Manager with the Machine Learning Triage Assistance add-on.

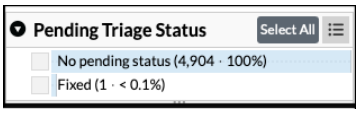
Triage Status Filter

The Triage Status filter allows you to filter findings based on the triage status of the finding (e.g., fixed, mitigated, etc.). Expand the filter window and select which filter options you want to apply to the list of findings.



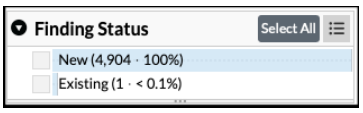
Pending Triage Status Filter

The Pending Triage Status filter shows triage status requests that are pending approval. Expand the filter window and select which filter options you want to apply to the list of findings.



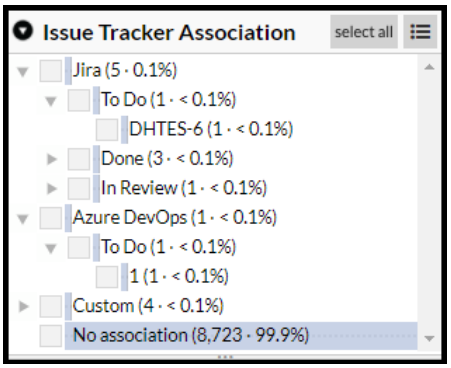
Finding Status Filter

The Finding Status filter allows you to filter findings based on the status of the finding (e.g., new, existing, or gone). Expand the filter window and select which filter options you want to apply to the list of findings.



Issue Tracker Association (if configured)

The Issue Tracker Association filter allows you to filter findings based on whether a finding has an associated issue. Expand the filter window and select which filter options you want to apply to the list of findings.

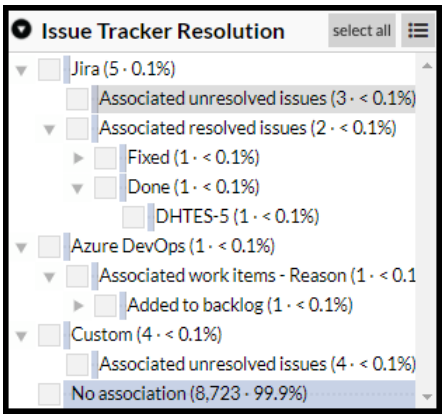


This filter option appears only if the project has been configured for issue tracking (see Issue Tracker Configuration). The filter window shows findings broken down by whether there is an associated issue, which issue tracker type (Jira, Azure DevOps, ServiceNow, GitLab, etc.) the issue is associated with, the issue's status, and the specific issue.

Note: Terminology can differ between different issue trackers (e.g., "issue" vs "work item," "status" vs "reason," etc.), but Software Risk Manager defaults to "issue" and "status" when a generic term is needed.

Issue Tracker Resolution (if configured)

The Issue Tracker Resolution filter allows you to filter findings based on resolution status. Expand the filter window and select which filter options you want to apply to the list of findings.

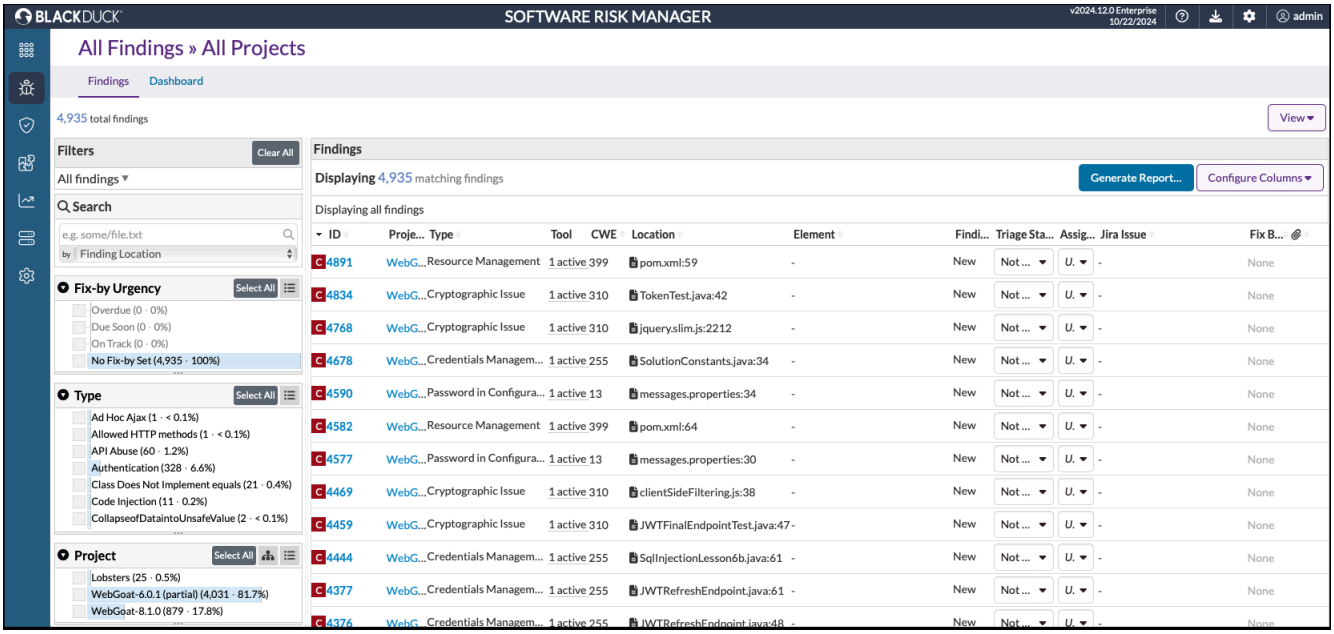


This filter option appears only if the project has been configured for issue tracking (see Issue Tracker Configuration). The filter window shows findings broken down by whether there is an associated issue, which issue tracker type (Jira, Azure DevOps, ServiceNow, GitLab, etc.) the issue is associated with, whether the issue is resolved, the resolution status, and the specific issue.

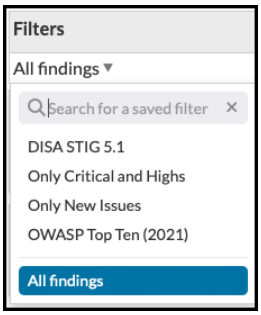
Configuring Filters

To make routine searches easier, you can configure and save filters that can be used by authorized users or user groups.

Click the Findings icon in the navigation bar to open the Findings page.



Filter options are located in the top left of the page.



Once a filter is selected, the list of findings will update to match the current filter state.

Configuring filters includes the following tasks:

- Creating or modifying a filter
- Viewing and setting filter permissions
- Renaming a filter
- Deleting a filter

Creating or Modifying a Filter

To create or modify a filter:

1. Click the Findings icon in the navigation bar to open the Findings page.
2. Select an existing filter from the Filters dropdown menu that you want to modify or select "All findings" to create a completely new filter.
3. Select the filter elements that will define this filter.
   When you make a change to the filter settings, an asterisk appears next to the filter name. From here you can click the "Save" icon to update the filter with the new settings or click the "Save as" icon to create a new filter with a new name.

   Clicking the "Reset" icon returns the filter to its previously saved state.
4. Click the "Save as" icon to create a new filter.
5. Enter a new name in the filter field.
6. Click anywhere on the page to save the new name.

Viewing and Setting Filter Permissions

To view and set permissions:

1. Click the Findings icon in the navigation bar to open the Findings page.
2. Select a filter from the Filter dropdown menu.
3. Click the "Permissions" icon to open the *Permissions* window (shown below).
   This window displays a list of users who have already been given permissions to view or edit this filter.
4. Select the *Shared* radio button, then click the *Users* or *Groups* button.
5. Use the "Add Users/Add User Groups" dropdown list to select the users or groups to add.
6. Click the "Add" icon.
7. Use the "Permissions" dropdown menu to set the desired permission.
8. Click Save to save your changes.

Renaming a Filter

To rename a filter:

1. Select the filter from the Filter dropdown menu.
2. Click the "Rename" icon.
3. Enter the new name in the filter field.
4. Click the "Rename" icon or click anywhere outside the filter field to save your changes.

Deleting a Filter

To delete a filter:

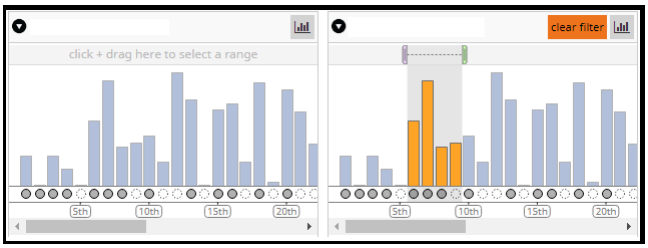
1. Click the Findings icon in the navigation bar to open the Findings page.
2. Select a filter from the Filter dropdown menu.
3. Click the "Delete" icon.
4. Click Delete.

Using Time Filters

A Time Filter is a special type of filter which groups findings by analysis, that is, the filter will decide which analysis each finding belongs to, and then display the groupings as a bar chart. The analysis number or analysis date will make up the X axis, while the finding count makes up the Y axis.

Unlike the other filters, a Time Filter may not be resized vertically. Instead, as more analyses are displayed it will grow horizontally, eventually adding a horizontal scrollbar.

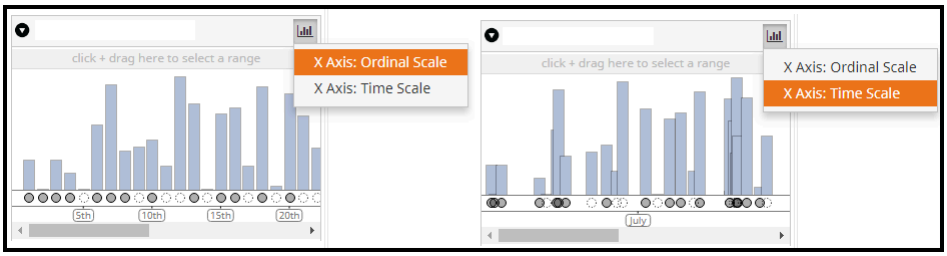
Making a filter selection with a Time Filter is done by drag-selecting a range in the selection area above the bar chart, as shown in the figure below.



The X axis of each Time Filter can be toggled between "Ordinal" and "Time."

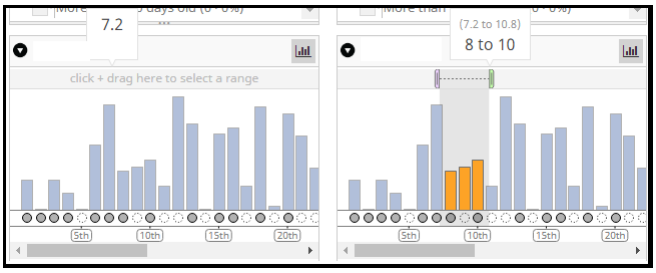
Ordinal scale uses the analysis number (e.g. 1st, 2nd, ...) to determine where each analysis is placed on the X axis. It allots the same amount of horizontal space for each analysis, making it a reliable way to visually separate one analysis from another. Ordinal scale is the default mode for each *Time Filter* when the page loads.

Time scale uses the start time of the analysis to determine where each analysis is placed on the X axis. Since the lifespan of a project may be very long, and several analyses may be clustered close together, bars may overlap when using time scale mode. This scale mode is most useful when you want to highlight a particular date range without separating individual analyses.

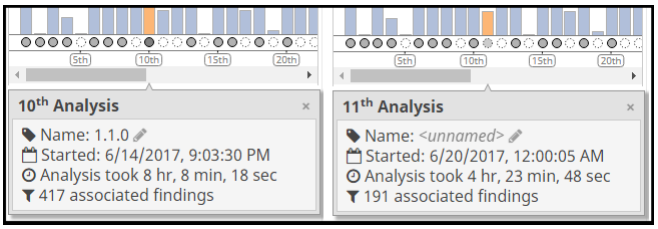


As you hover over the selection area of a Time Filter, you will see a tooltip indicating the X value that your cursor is hovering over. In time scale mode, the X value is a date and time. In ordinal scale mode, the smaller text indicates the "physical" selection range, while the larger text indicates the rounded range, which will be used as the actual filter selection. Once you click and drag to make a selection, the tooltip will expand to show you the "min" and "max" of your selection.

An existing selection can be altered by clicking and dragging either of the paddles (the purple and green paddles at either end of the selection) to resize, or the area between the paddles to pan. Double-clicking a paddle will move it all the way to the beginning or end of the chart. Double-clicking the area between the paddles will clear the selection.

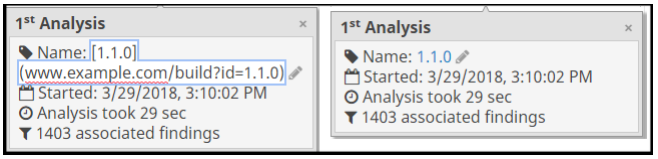


As you hover over bars in a *Time Filter*'s chart area, a tooltip will appear to display information about the currently-hovered analysis. The information includes the start time, duration, number of associated findings (the height of the bar), the analysis number (e.g. the 10th analysis in that project), and the Name.



Note that when an analysis has a name, the circle below its bar in the bar chart will be filled in; when it has no name, the circle will only have a dotted outline. Users with the `update` role can edit analysis names by clicking the pencil icon next to the name. Naming an analysis can be useful if you want to associate it with a particular release version of your software, a Git commit, a Jenkins build, or anything similar.

In addition, analysis names allow you to write Markdown-style links, e.g., `[link text](link url)`.



First Seen by SRM Filter

The First Seen by SRM filter is a Time Filter that groups findings based on the analysis during which the finding was first seen in SRM; that is, the analysis that introduced the finding. This filter can be useful for

answering questions such as, "how many findings were introduced by release 2.1.0?" Note that the First Seen by SRM filter differs from the Age Filter in that the Age filter takes into account any first-seen dates provided by tools.

Last Modified Filter

The Last Modified filter is a Time Filter which groups findings based on the analysis during which the finding was last modified. (For findings modified by users or other non-analysis interactions, it picks the most recent analysis at the time of the modification.) An example usage of this filter could be in combination with the Status Filter to answer the question "How many findings have been fixed since release 2.1.0?" In this example, you would `unhide` the "completed" triage statuses by using the View button.

CWE Support

The Common Weakness Enumeration (CWE) is a community effort lead by MITRE to provide a common language to express software weaknesses.

Software Risk Manager leverages the CWE to provide correlation across the diverse set of testing tools it supports. Software Risk Manager also allows you to define your own correlation logic via the Rule Set page. This allows you to correlate based on a group of CWEs or tool specific rule codes.

Software Risk Manager uses the CWE identifier specified by the tool. In cases where the tool does not provide a CWE, that mapping is done automatically.

CWE information is readily available in Software Risk Manager. On the Findings page, you can search by CWE or filter by CWE. This includes grouping CWEs by various standards such as OWASP Top 10 or CWE/SANS Top 25. The CWE identifier is also shown in the Findings Table, and you can hover on that identifier to get the full CWE name.

CWE information is also provided on the Finding Details page. There you can see the full CWE name for the aggregated finding. For each individual tool result, the CWE used for each tool is also provided. In both cases, a link to MITRE's CWE website is provided.

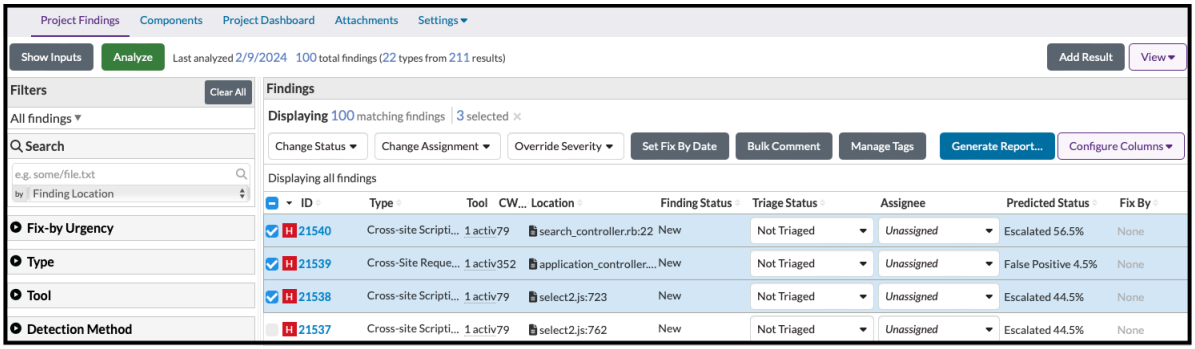
Finally, all reports (CSV, XML, PDF, Nessus, and AlienVault/NBE) contain CWE information.

Note: Software Risk Manager currently supports CWE Version 4.14.

Performing Bulk Operations

Bulk operations are actions that affect all findings that are currently selected. When the checkbox in the top left of the findings table is selected, these actions will apply to all findings currently displayed.

Note that bulk operations (aside from reporting) are not available on an aggregate version of the Findings page. In addition, the checkbox selection column in the findings table will be hidden.

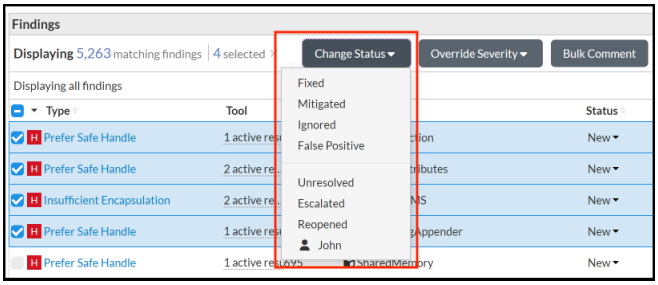


For more information on bulk operations, see the following:

- Change Status. This is used to change the triage status of multiple findings.
- Change Assignment. This is used to change the user assigned to multiple findings.
- Override Severity. This is used to change the severity of multiple findings.
- Bulk Comment. This is used to add comments to multiple findings.
- Generate Report. This is used to generate a variety of report types.
- Issue Tracker Integration. This is used to interact with a configured issue tracker.
- Manage Tags. This is used to assign tags to findings or unassign tags from findings in bulk.
- Set Fix By Date. This is used to set the fix-by date of multiple findings.

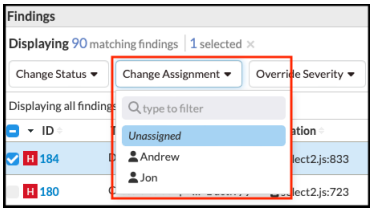
Change Status

The *Change Status* dropdown menu is available to users with the `update` role. It allows users to change the triage status of all findings currently selected.



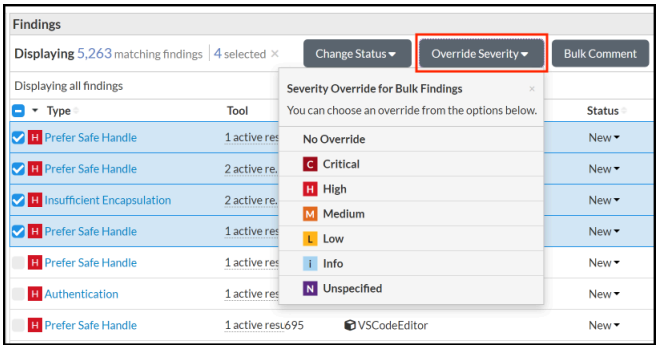
Change Assignment

The Change Assignment dropdown menu is used to change the user assigned to the selected findings.



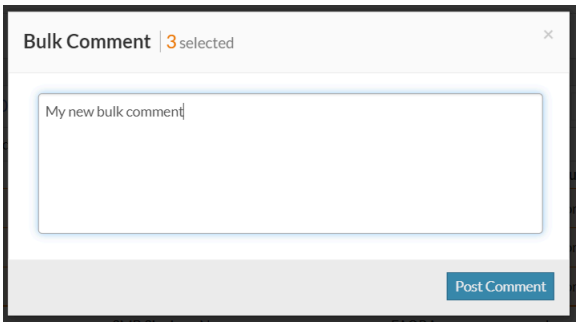
Override Severity

The *Override Severity* dropdown menu is available to users with `update` role. It allows users to change the severity of all findings currently selected.



Bulk Comment

The *Bulk Comment* button opens the *Bulk Comment* dialog. The dialog is used to comment on all findings that are currently selected.

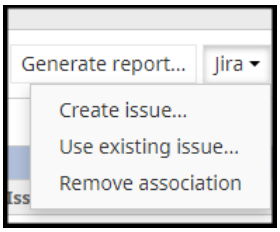


Issue Tracker Integration

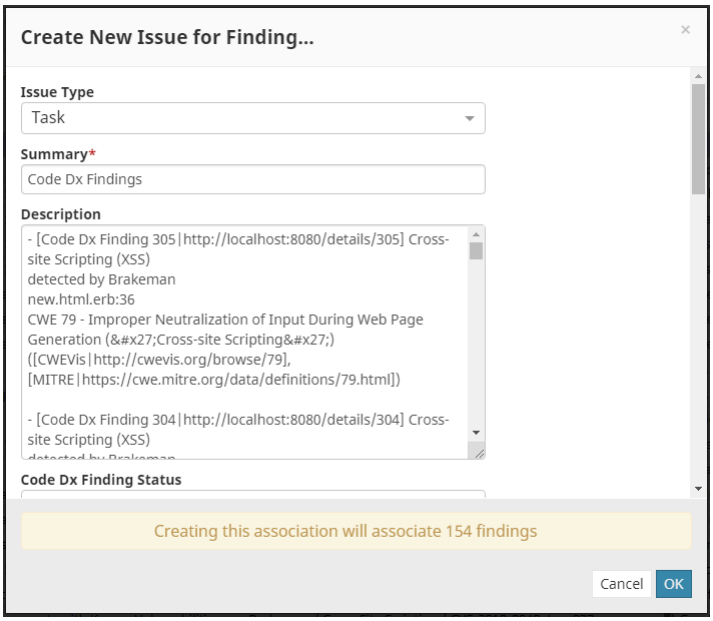
If a project has an Issue Tracker Configuration, the *Issue Tracker* dropdown menu will be available, allowing users with the `update` role to interact with the configured issue tracker. Software Risk Manager currently supports Jira, Azure DevOps, ServiceNow, and GitLab. For Jira and GitLab users, the options are create issue, associate with existing issue, and remove association. For Azure DevOps users, the options are create work item, associate with existing work item, and remove association. For ServiceNow users, the options are create incident, associate with existing incident, and remove association. The examples below assume Jira is the currently configured issue tracker.

Creating New Issues

To create a new issue for a Finding based on your current branch view, click the *Jira* dropdown menu and select the *Create issue...* option.



A dialog will open.



All of the fields are editable. Required fields will have a red asterisk by their name.

Use the template expressions that were defined when configuring the issue tracker to pre-populate the relevant fields with data from the active findings. Note that these pre-populated values will be based on the current branch view where the issue is created. Software Risk Manager provides default templates for the *Summary* and *Description* fields.

The *Description* field will be pre-populated with a brief description for each Finding. Jira descriptions can be set to allow for the use of WikiMarkup. Software Risk Manager takes advantage of that to make the descriptions more readable from within Jira.

Creating Associations with Existing Issues

For Jira users, associate a finding with an existing issue by clicking the *Jira* dropdown menu and selecting the *Use existing issue...* option. For Azure DevOps users, associate a finding with an existing work item by clicking the *Azure DevOps* dropdown menu and selecting the *Use existing work item ...* option. For ServiceNow users, associate a finding with an existing incident by clicking the *ServiceNow* dropdown menu and selecting the *Use existing incident ...* option. For GitLab users, associate a finding with an existing issue by clicking the *GitLab* dropdown menu and selecting the *Use existing issue...* option.

Enter the issue key, work item, or incident number that you want to associate with the finding(s). Clicking outside the textbox or pressing *Enter*will cause Software Risk Manager to look up the issue, work item, or incident in question. If SRM is able to find it, and if it's part of the same Jira, Azure DevOps, ServiceNow, or GitLab project you selected when you configured the issue tracker for this project, or if the same ServiceNow instance configured for this Software Risk Manager project, the issue or work item summary will be displayed, allowing you to confirm that you've entered the issue or work item you want. Click OK to associate the finding (or findings) with that issue or work item.

Refreshing Issue Status

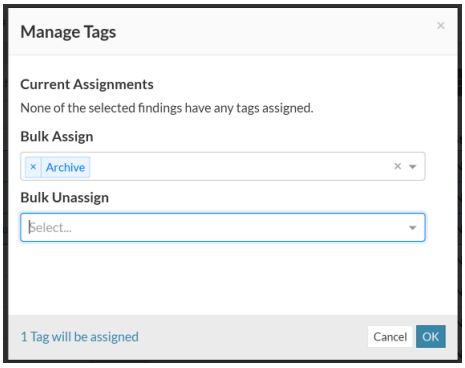
Software Risk Manager will regularly check the Issue Tracker server to refresh the status for all of the issues, work items, or incidents associated with findings in a given project. The interval at which the check is done is configurable in the Issue tracker configuration. However, you can also manually trigger a refresh of all the issues, work items, or incidents on the Findings page by clicking the *Refresh Issues*, *Refresh Work Items*, or *Refresh Incidents* button.

Removing Associations with Existing Issues

You can remove the issue, work item, or incident associations for all of the findings in the current filter by using the *Jira*, *Azure DevOps*, *ServiceNow*, or *GitLab* dropdown menu and selecting the *Remove association* option. Note this only removes the association in Software Risk Manager; it doesn't change the issue, work item, incident in the Issue Tracker.

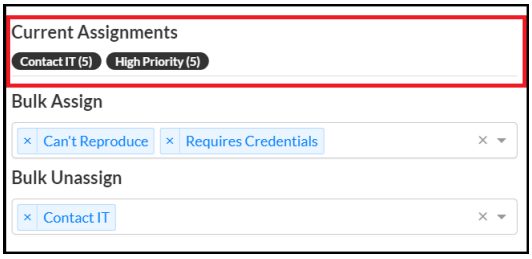
Managing Tags

The *Manage Tags* button opens the *Manage Tags* dialog. The *Manage Tags* dialog enables users to assign tags to findings or unassign tags from findings in bulk. The dialog is composed of three sections, each of which will be described in sequence. Note that no operations will be applied until the OK button in the footer of the dialog is clicked. (Clicking the Cancel will discard all dialog activity.)



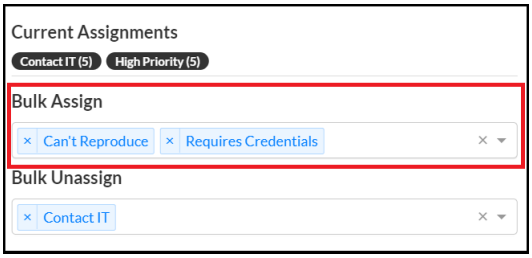
Current Assignments

The *Current Assignments* section presents a sequence of tags that have been assigned to at least one of the selected findings. Each tag in the sequence is paired with the number of selected findings to which that tag has been assigned (provided in parentheses next to the tag).



Bulk Assign

The *Bulk Assign* section allows users to select tags that should be assigned to all selected findings. The dropdown select menu (which will appear as soon as you begin typing the name of a tag) will be populated with tags that are available for assignment, including tags that have already been assigned to some of the selected findings. Admins can create tags inline if they attempt to assign a tag that does not exist.



The number of tags that will be attempted to be assigned once the OK button is clicked is shown in the footer of the dialog.

Bulk Unassign

The *Bulk Unassign* section allows users to select tags that should be unassigned from all selected findings. The dropdown select menu (which will appear as soon as you begin typing the name of a tag) will be populated with tags that appear in the *Current Assignments* section of the dialog.

The number of tags that will be attempted to be unassigned once the OK button is clicked is shown in the footer of the dialog.

Set Fix By Date

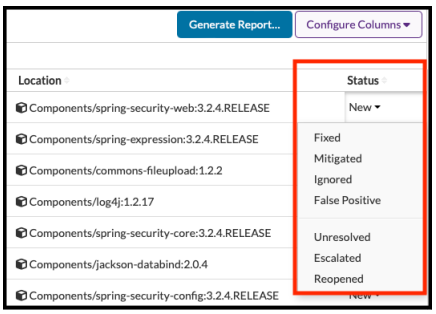
The *Set Fix By Date* button opens the *Set Fix By Date* dialog. The dialog is used to either set or remove the fix-by date on all findings that are currently selected.

If a project is configured to use policies, this bulk operation is used to override the fix-by date that was determined by the policy rules.

Findings Table

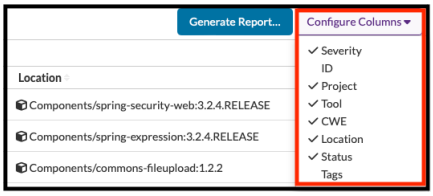
The *Findings Table* shows a concise representation of each individual finding. The number in the *ID* column is the unique identifier assigned to each finding and the text for the *ID* doubles as a link to the finding's details.

Users with the `update` role in a project can use the dropdown menu in the *Status* column to change the current status of a finding.



Projects often have more findings than can be displayed in the *Findings Table* all at once. Because of this, the table is split into pages. By default, each page shows 25 findings. Users can change the number of findings per page using the *Show* button, seen below.

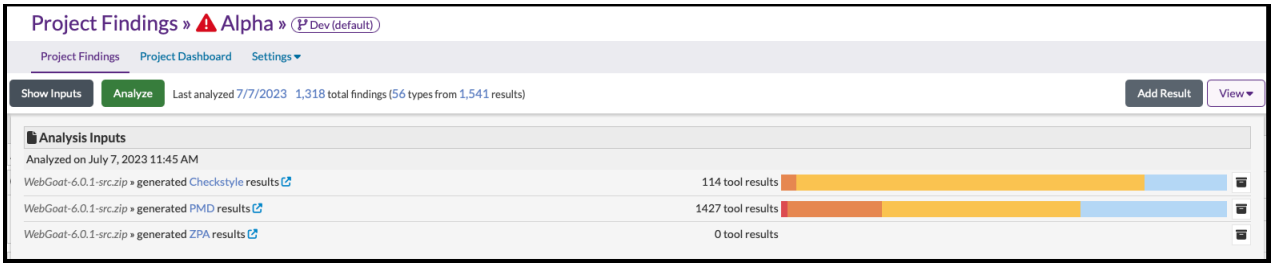
The *Findings Table* columns can be hidden or displayed using the dropdown menu in the upper right corner of the table. This is done by toggling the column name.



In the menu, visible columns have a checkmark to the left of the column name. Hidden columns can be made visible again by selecting them in the menu.

Analysis Input List

The *Analysis Inputs List* is a widget on the Findings page that shows the files that were provided to Software Risk Manager for analysis. It can be shown by clicking the *Show Inputs* button found in the Findings page header.



The *Analysis Inputs List* is broken down first by analysis, then by file. For example, when viewing a project in which two analyses had been performed, there would be a section for each analysis. Analyses are ordered by date, with the most recent analysis shown at the top of the list, and the oldest analysis at the bottom.

Within each section, individual entries represent files. For example, if a "spotbugs-results.xml" file had been uploaded to Software Risk Manager during one analysis, a corresponding entry would appear in the section for that analysis. Each entry has three main parts: input name, tool result summary, and archive button.

Input Name

The first part of an entry shows the file's name and the name of the tool it came from. For auto-generated tool outputs (i.e., files generated by any of the Software Risk Manager bundled tools), the name of the analyzed file will be shown instead of the name of the auto-generated temporary file. Next to the names, a download link allows users to download a copy of the file.

Tool Result Summary

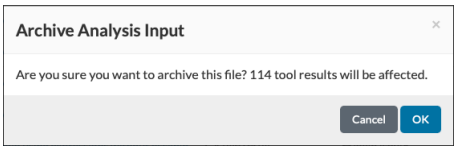
The second part of an entry shows a summary of the tool results originating from that file. Note that due to result correlation and other factors, the total tool result count will not necessarily match the total finding count. Next to the tool result count for each entry, a bar chart shows a breakdown of the tool results by

severity. The highest-severity results are shown in red, while the lowest-severity results are shown in gray. You can hover over each bar to see the severity it represents as well as the number of tool results belonging to that severity.

Archive Button

Users with the `create` role for a project have the ability to archive an analysis input using the Archive button located on the far right of each entry. Tool results from archived inputs will be removed. Any finding whose last tool result was removed in this manner will have its triage status automatically changed to `Gone`. Normally archival is done automatically.

When you click the Archive button for an analysis input, you will be prompted to confirm your choice.



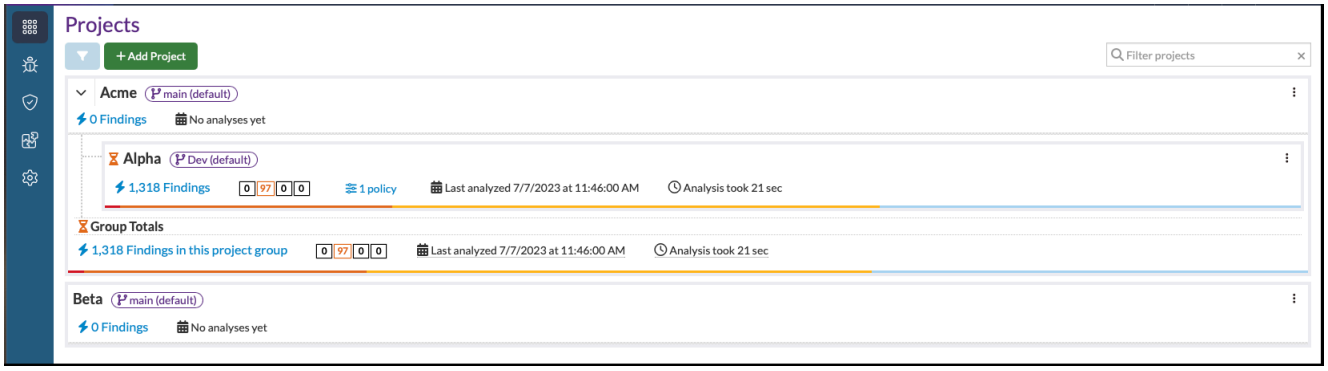
When you confirm, the archival will be performed. The page will update to reflect the updated tool result and finding counts.

Adding Manual Results

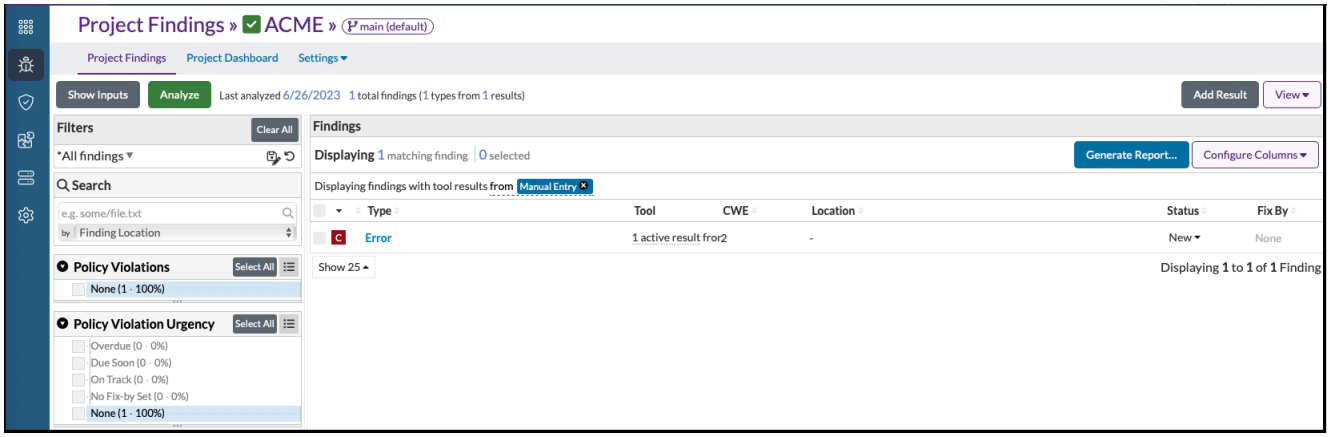
Software Risk Manager allows users with the *create* role to enter manual results to Findings data.

To add a manual result:

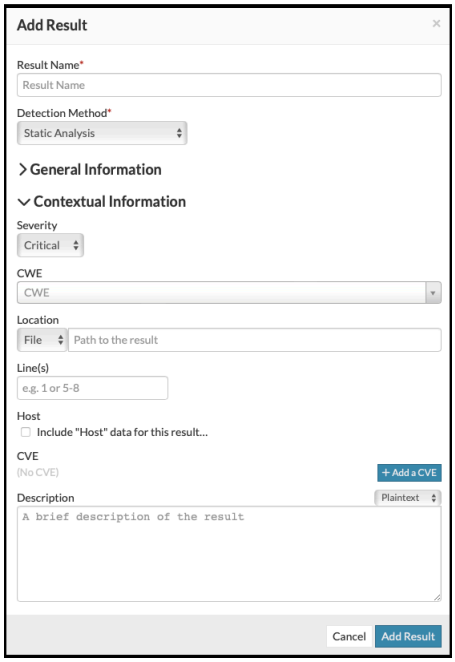
1. Click the Projects icon in the navigation bar to open the Projects page.



2. Locate the project and click the Project findings link to open the Project Findings page.



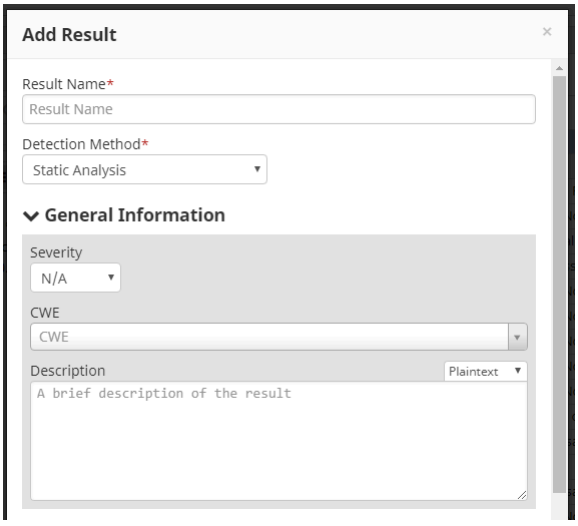
3. Click Add Result.



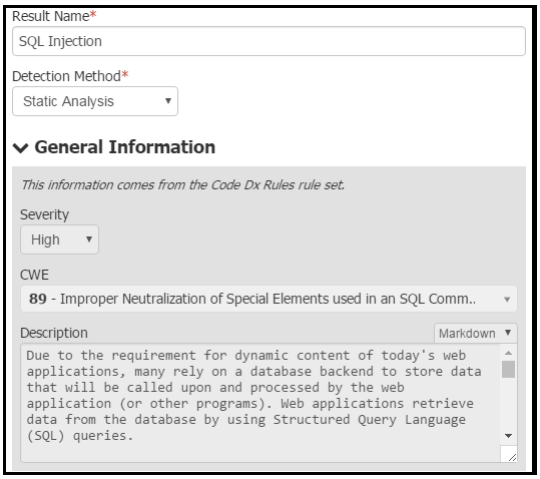
4. Enter a name for the result and select a detection method.
5. Add the required data to the "General Information" and "Contextual Information" fields. See the sections that follow for additional information on these sections.
6. Click Add Result.

General and Contextual Information

Information entered under the *Contextual Information* section describes the result itself. Expanding the *General Information* section of the form will allow values to be specified that will be shared among all manual results of the same name. Contextual information will override general information if specified. Note that this form creates *results*, which can be thought of as "evidence" for a finding. Multiple results may be correlated to a single finding. As with tool results, two manual results will typically be correlated if they have the same CWE, Location, and Detection Method, even if their names are different.

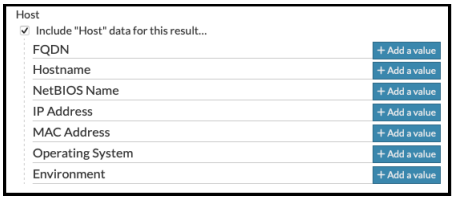


If the result name entered matches a rule in the current rule set, then the manual result will be associated with the general information for that rule. In this case, the general information can only be changed by revising the rule set. Both the general and contextual information will be included on the details page.



The *Tool* field allows the user to state that the manually-entered result actually came from a tool. The options available to this field are configured on the admin page, in the *Allowed Tools*section.

The *Host* field allows the user to describe the "host" on which the result was discovered. This normally will only pertain to results with the *Network Analysis* detection method, but could also relate to *Dynamic Analysis*. Host data entered on this field is considered "raw" data, (as opposed to the "normalized" data seen on the *Hosts* page). Raw host data may be joined with "normalized" host data through a process called "host normalization". By default, the "Include Host data for this result" checkbox is unchecked. Check it to expand the host data editor.



The *CVE* field allows the user to enter any number of CVEs that correspond to the result. By default, no CVEs are included. To start adding CVEs, click the *Add a CVE* button. When typing in a CVE text box, you can optionally start by only typing the numbers; the text box will fill in the rest for you. If your Software Risk Manager server is able to access the internet, it can check whether the CVEs entered by the user are real CVEs in the CVE database. This verification comes in the form of a checkmark or an "x" on the CVE textbox. Blank or invalid CVEs will be ignored when submitting the form.

Once you've completed the form, clicking the *Add Result* button at the bottom will dismiss the form and update the *Findings* page with the new finding. A notification will appear, indicating the ID of the finding to which the result was correlated. To delete or edit a manually added finding, click on the finding's ID in the *Findings Table* to access its details view. The result will appear in the *Evidence* section, where there will be buttons to edit and delete it.

Using Machine Learning with Project Findings

Note: This section is only applicable to Software Risk Manager users with the Machine Learning Triage Assistance add-on and requires that machine learning is enabled.

Users of Software Risk Manager may review findings and change their statuses. When a finding's status has been changed, we say that that finding has been *actively triaged*. The act of *actively triaging* a finding is considered a *past triaging decision*. Software Risk Manager is capable of learning from users' past triaging decisions in order to make predictions about findings that have yet to be *actively triaged*. More details will be described in the sections that follow.

Actionability of a Finding

We use the terms *Actionable* and *Non-Actionable* to denote findings that are “real” issues and “not-real” issues, respectively. A finding is said to be *Actionable* if it was actively triaged as Fixed, To Be Fixed, Mitigated, or Assigned, if it has a status of Gone, or if it has an issue tracker association. A finding is said to be *Non-Actionable* if it was actively triaged as False Positive or Ignored.

Training a Prediction Model

In order for Software Risk Manager to make predictions for findings, users will need to train a *prediction model*. Training a prediction model will collect all relevant data for findings that have been actively triaged and use that data to learn from users' past triaging decisions. See Machine Learning Control Panel for more information about how to train a prediction model.

Predicted Status and Prediction Confidence

When Software Risk Manager is making a prediction for a finding, we mean that Software Risk Manager is determining a *Predicted Status* for it. A *Predicted Status* for a finding is its *Actionability*. If Software Risk Manager predicts that a finding is *Actionable*, then we say that its *Predicted Status* is *To Be Fixed*, since Software Risk Manager thinks it's a real issue. If Software Risk Manager predicts that a finding is *Non-Actionable*, then we say that its *Predicted Status* is *False Positive*, since Software Risk Manager does not think it's a real issue. Every prediction that Software Risk Manager makes has a *Prediction Confidence*. A *Prediction Confidence* for a *Predicted Status* represents how certain Software Risk Manager is of its *Predicted Status* relative to the one it did not predict. Note that this is a prediction of a finding's *Actionability*. That being said, Software Risk Manager's prediction may not be correct.

Requirements for Making Predictions

Software Risk Manager will only attempt to make predictions for findings if a prediction model has been trained. See Machine Learning Control Panel for more information about how to train a prediction model.

When Will Software Risk Manager Make Predictions

Software Risk Manager will makes predictions for findings during the following situations:

- During an analysis
- After a manual result has been created
- After a prediction model has been (re)trained

In these situations, all predictions are being made automatically. During the first and third situations, predictions are automatically made for every finding in Software Risk Manager. During the second situation, a prediction is only made for the single manually created result. Since predictions are made automatically, a user may note that predictions for findings might differ between reviewing sessions.

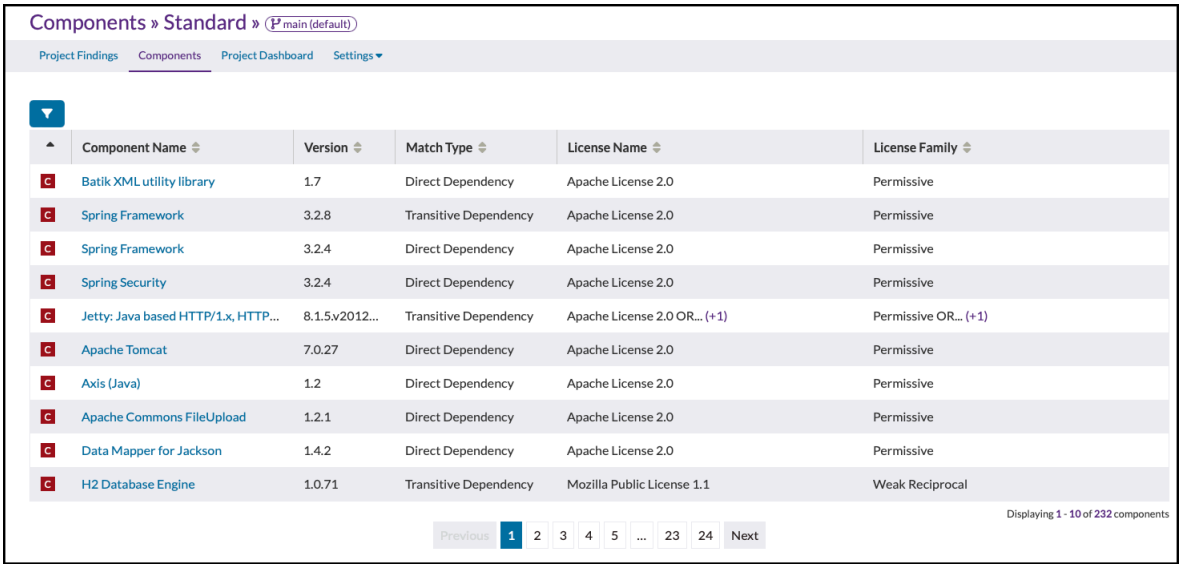
Predicted Status Column

Every value in this column consists of a *Predicted Status* and a *Prediction Confidence*.

Working with Components

Software Risk Manager provides a comprehensive list of a project's components that can be accessed through the Findings page.

Open a project's Findings page, then select the Components tab to display a list of that project's components.



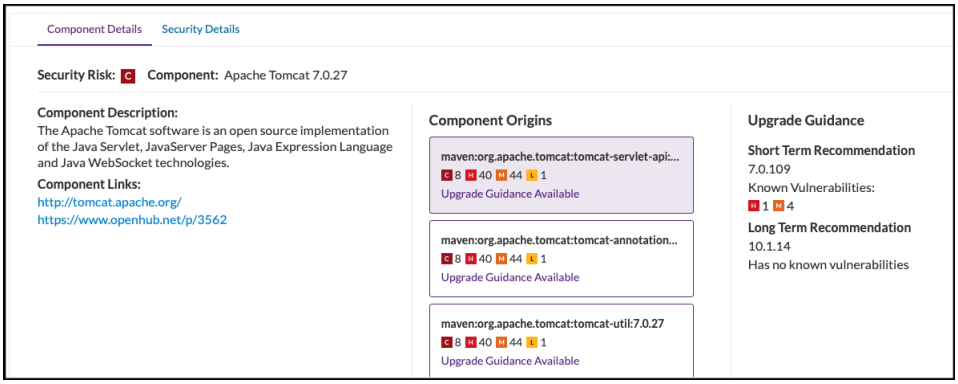
The component view shows all of the components that are part of the project, regardless of the status of any individual finding. (Use the filters to the left of the list to search; click the column headings to sort.)

The Component page provides the following information:

- Component Name
- Version
- Match Type (direct or transitive dependency)
- License Name
- License Family

Click the component name to open the Component Details and Security Details view for that specific component.

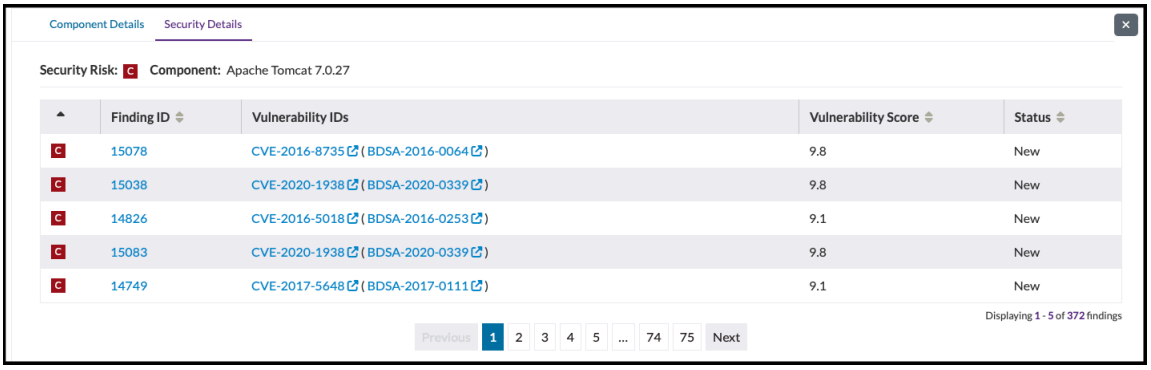
Select the Component Details tab to view detailed information about the component.



The Component Details tab provides the following information:

- Security Risk
- Component Name
- Component Description
- Component Links
- Component Origins
- Upgrade Guidance

Select the Security Details tab to view additional security information about the component.



The Security Details tab provides the following information:

- Security Risk
- Component Name
- Finding ID
- Vulnerability ID
- Vulnerability Score
- Status

Note: If the project is configured with a component correlation mode that doesn't include vulnerability, then the Vulnerability ID and Vulnerability Score columns won't appear in the table. (For more information, see Analysis Configuration Options.)

Working with Findings Reports

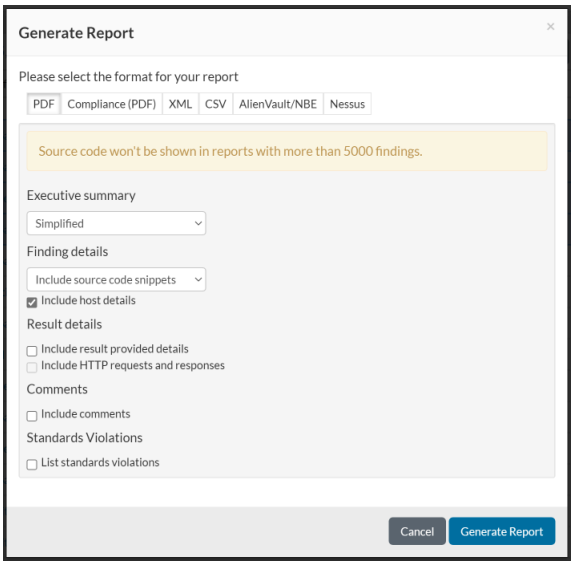
Software Risk Manager allows you to create custom reports, create report templates, and generate reports on a schedule. Reports provide information such the number of findings and status, triage status, and related findings details.

For more information on reports, see the following topics:

- Generating a Findings Report
- Creating a Findings Report Template

Generating a Findings Report

Click the Findings icon in the navigation bar and click Generate Report to open the *Generate Report* dialog. This dialog is used to select the type of report and allows you to customize the report.



SRM supports the following report types:

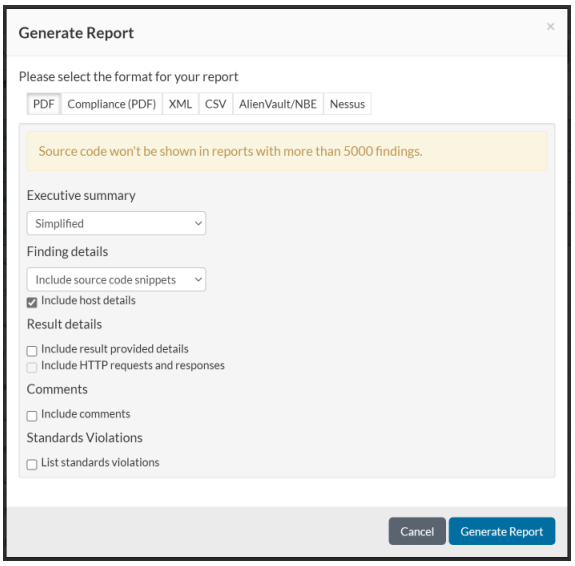
- PDF
- Compliance (PDF)
- XML
- CSV
- AlienVault/NBE
- Nessus

Instructions for generating each type of report are given below, according to report type.

PDF Report

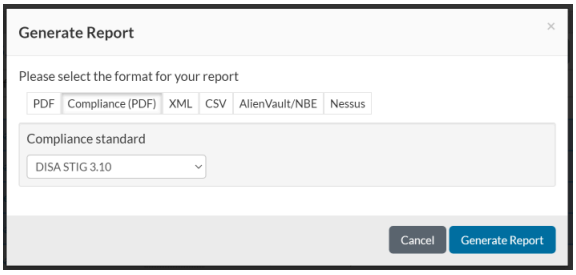
You can customize the PDF report in several ways. There are options to include or exclude a simplified or detailed executive summary section; finding details (with or without source code); tool details; and comments that appear in the Activity Stream (on the Finding Details page). The "Result details" section contains these options: "Include result provided details" and "Include HTTP requests and responses."

Note: The "Include result provided details" option *must* be selected if you want to include the HTTP requests and responses in the PDF report.



If you'd like your company logo to appear on the cover sheet, please contact your Software Risk Manager administrator to configure it for you.

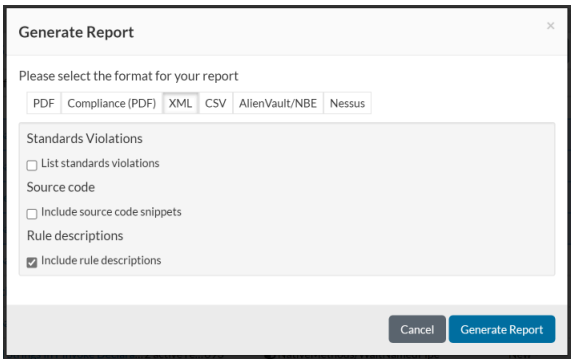
Compliance (PDF)



XML Report

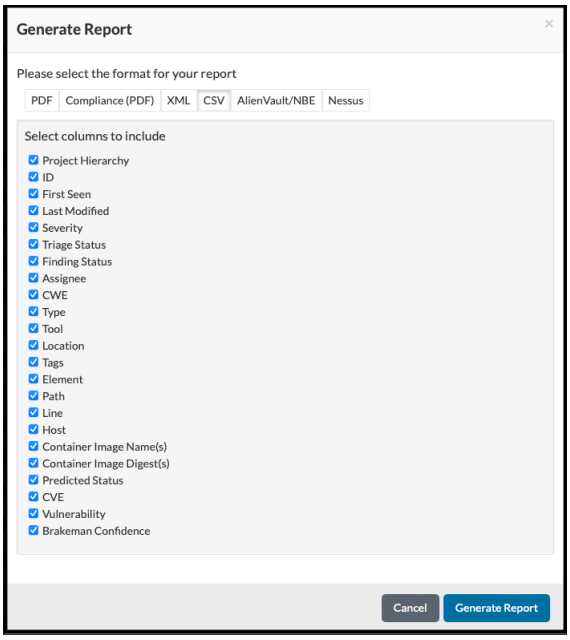
Customizations for the XML report include the option to enumerate standards violations for each finding, provide source code snippets, and whether to include copies of the rule descriptions for each finding.

Note: There is a limit of eight lines of code per source snippet for each finding. When the limit is exceeded, no source code is provided.



CSV Report

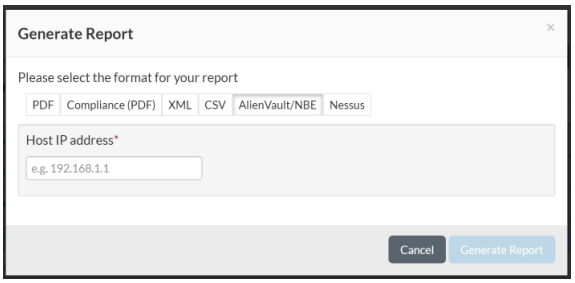
The CSV report provides options allowing you to select which columns will be included in the generated file.



AlienVault/NBE Report

Software Risk Manager users will be able to select the AlienVault/NBE report. This reporter generates an NBE report that is compatible with AlienVault.

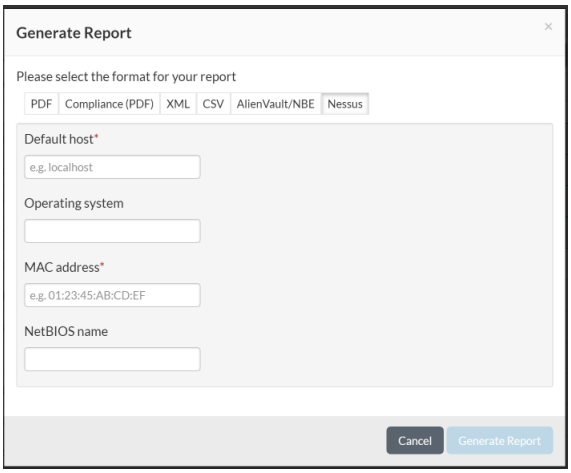
The report options require that a host address (IPv4) be specified for inclusion in the report.



Nessus Report

Software Risk Manager users will be able to select the Nessus report. This reporter generates a report in the *Nessus* format, which can be imported by many applications.

The default host and MAC address fields are required, while the operating system and NetBIOS name fields are optional. When exporting a finding that doesn't contain any request data, the default host value will be used.



Creating a Findings Report Template

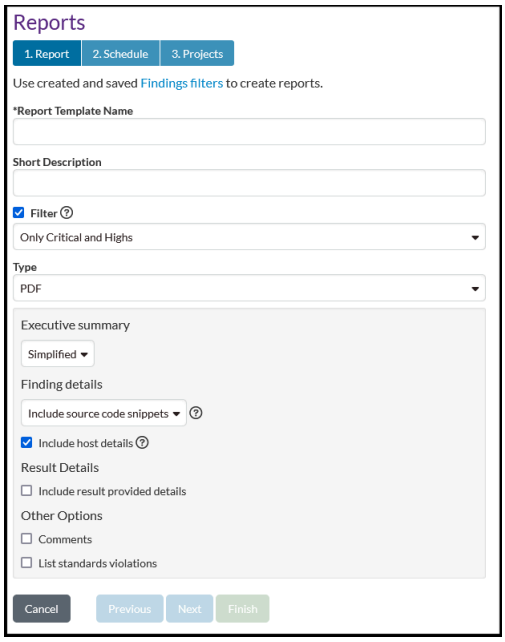
Click the Reports icon in the navigation bar to open the Reports page. This page lists all the existing custom report templates along with the following information:

- **Name.** The name of the report template.
- **Owner.** The owner of the report.
- **Type.** The type of report. Options are PDF, Compliance (PDF), XML, CSV, AlienVault/NBE, and Nessus.
- **Filter.** A dropdown list of available filters from the Findings page.
- **Schedule.** When a report should be generated.
- **Send To.** Email address to send the report.
- **In This Report.** The number of projects associated with the report. (Click the link to view the projects.)

Clicking the dropdown configuration icon allows you to view, edit, send now (report), or delete the report template.

To create a report template:

1. Click the Reports icon in the navigation bar.
2. Click Create Report Template.



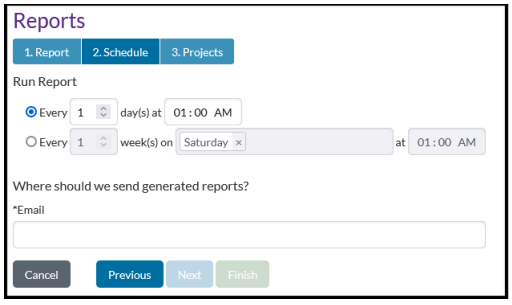
Creating a report template consists of three elements:

- Report type and definition
- Schedule
- Projects

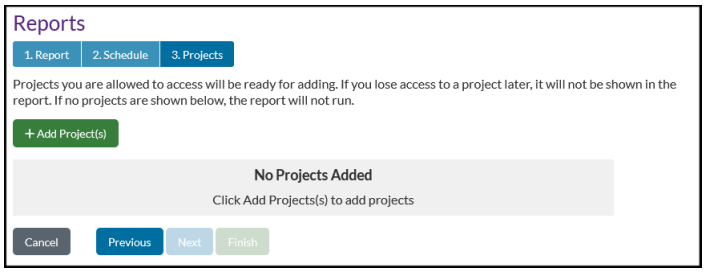
3. Enter a name for the template and a description (optional).
4. Add a filter by clicking the checkbox (optional) and selecting a filter from the dropdown list.
   Filter options will include your own saved filters, if any, along with any saved filters that are shared with you.

   Note: Saved filters that are shared with you must be copied before they can be used. A dialog will appear asking if you want to copy the shared filter.

5. Select and define a report type. (See Generating a Findings Report for detailed instructions.)
6. Click the Schedule tab.



7. Define how often you want to run the report.
   You can select a recurring day and time or day of the week and time.
8. Enter an email address where you want the report to be sent.
9. Click the Projects tab or click Next.



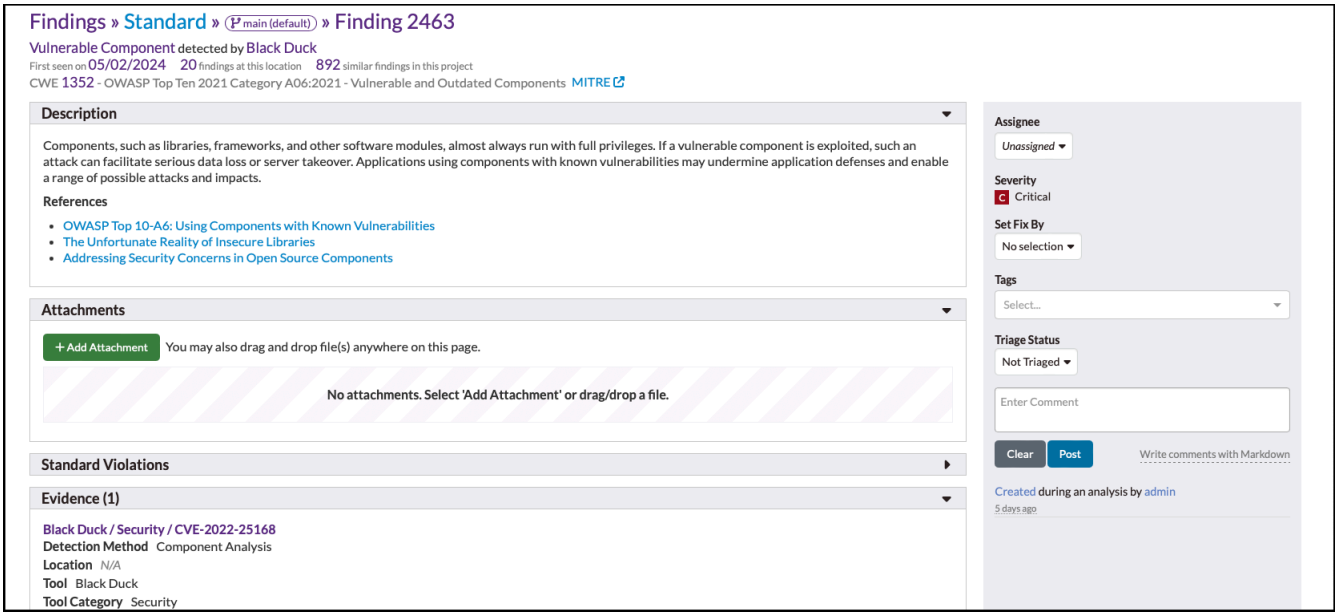
10. Click Add Project(s) to define which projects will be associated with this report.
11. Click Finish.

Working with Finding Details

The Findings page gives an overview of the findings in a project, focusing on a powerful filtering system, triage workflow, and issue tracking, with links to drill into more details via the Findings Details page.

To access the details for a single finding:

1. Click the Findings icon in the navigation bar to open the Findings page.
2. Locate the finding in the Findings list and click the link in the "Type" column, which is next to the Finding ID.



For more information on each element of the individual finding page, see the following:

- Branching. Allows you to view details for different branches.
- Details Summary. Provides a quick overview of the finding and the file where it is located.
- Severity Override. Provides the option to override the finding's severity level.
- Train Now Button. Provides a link to select training modules.
- Activity Stream. Allows you to change the status of the finding.
- Description. Provides a basic description of the finding.

- Attachments. Allows attachments to be linked to the finding.
- Standard Violations. Provides a list of standard violations.
- Training Video. Provides a link to select training videos.
- Evidence. Provides the raw data (results) that make up a finding.
- HTTP Activity. Shows the available data on the HTTP request.
- AI Insight (powered by Polaris Assist).* Generate remediation guidance for a SAST issue using a large language model (LLM). (The permission `Request and view finding assessments from Polaris Assist (Beta)` for the project is required, which is included in the default `Reader` role provided by SRM.)
- Source Code. Shows the contents of the file where the finding is located.
- Issue Tracker. Allows interaction with the configured issue tracker.
- Predicted Status. Provides prediction data for the finding based on the associated machine learning configuration.
- Fix By Override. Provides the option to either override or set a finding's fix-by date.

Branching

You can select which version of a finding's details to view by changing the actively selected branch in the dropdown. The list of branches available for selection will only include branches on which the finding appears. All details presented for a finding will be specific to the actively selected branch.

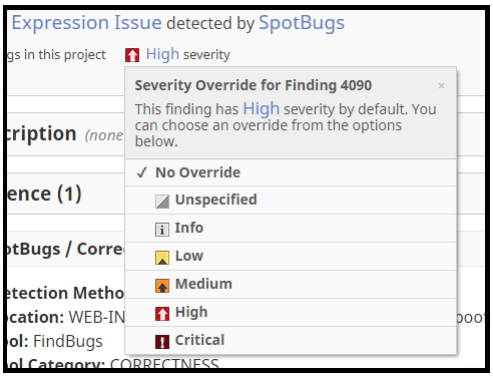
Details Summary

The *Details Summary* in the header gives a quick overview of the finding and the file where it is located. If the finding is associated with a CWE, the CWE is noted, with a link to the official CWE Mitre site. The "First seen on" date reflects the earliest date that the finding was observed, which will either be the date the finding was first observed in SRM or the earliest date reported by a supported tool.

The summary area also has "jump links." One link will scroll the source viewer to the location of the finding in the file. The other link (which appears once you scroll down the page) will bring you back to the top of the page.

Severity Override

Software Risk Manager allows users (with the `update` role) to change the finding's severity level. The severity override popup can be accessed by clicking the finding's severity icon from the Finding Details page or clicking the icon next to the finding ID on the Findings page.



Once the popup is opened, select the appropriate severity level. The popup will close and the new setting will be applied. When a finding has an overridden severity, the white border around its severity icon will change to green.

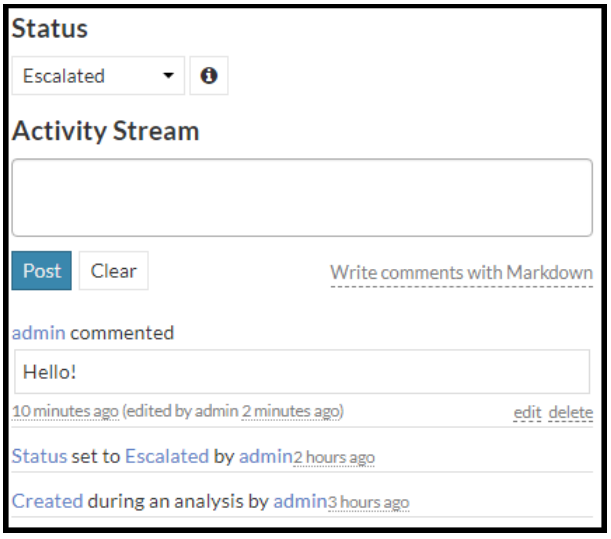
Accessing Additional Training Modules

Software Risk Manager integrates with Secure Code Warrior by linking developers to training modules that they can use to learn secure coding practices. If a training module is available that is related to the finding, a link will be provided to redirect the user to the training material.

Activity Stream

The *Activity Stream* area has widgets that let you change the status of the finding as well as comment on it. As users change the status and comment on a finding, messages appear in the activity stream, with newer messages at the top. Users can only edit or delete their own comments.

Note: Actions such as triage status updates and severity overrides apply only on the currently-selected branch. Comments are made independently of the current branch, effectively applying to all branches. The messages shown in the Activity Stream are filtered to those relevant to the current branch.

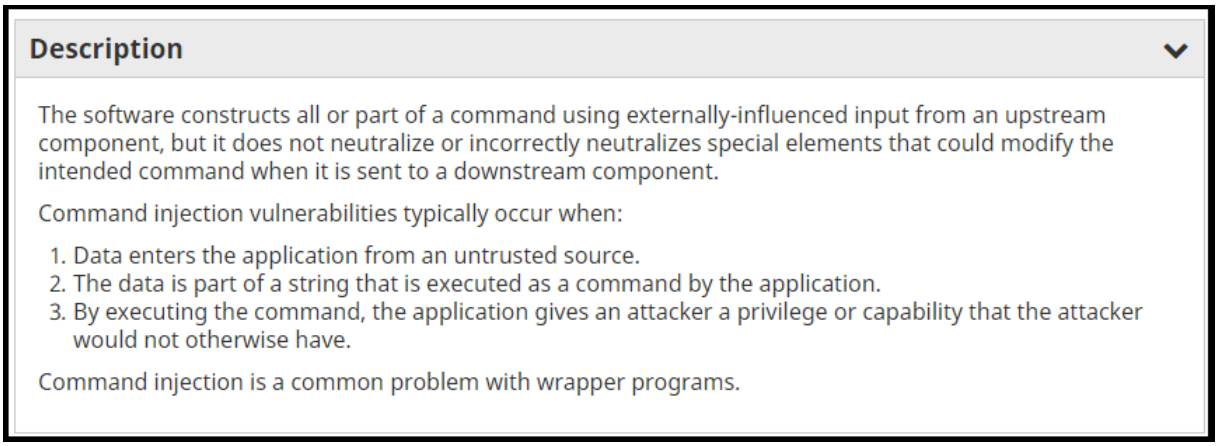


Descriptions

The description information shown by Software Risk Manager can come from a variety of sources, with varying levels of detail. At a high level, descriptions are divided into "general" and "contextual."

- "General" descriptions explain the *type* of finding, e.g. answering the question "What is SQL Injection?"
- "Contextual" descriptions explain a particular *instance* of the finding, e.g. answering the question "Why is this particular code a SQL Injection risk?"

The main "Description" section of the details page is a "general" description. Most of the time, the main description comes from a Rule Set. When a finding matches up to a rule, the main description section will use that rule's description. For findings created by *observed* tool results (i.e. types of findings that Software Risk Manager doesn't already know about - see the Tool Configurationsection), if the tool result does not match a rule, the general description may be created from that tool result, as long as the tool result provides one. This will often be the case with tools such as Fortify and Veracode.



The finding itself will not have a "contextual" description. This will instead be found on the individual results shown in the *Evidence* section. The "general" and "contextual" descriptions for results will be shown in the *Tool Rule Description* and *Contextual Description* sections of their display area, respectively (see below).

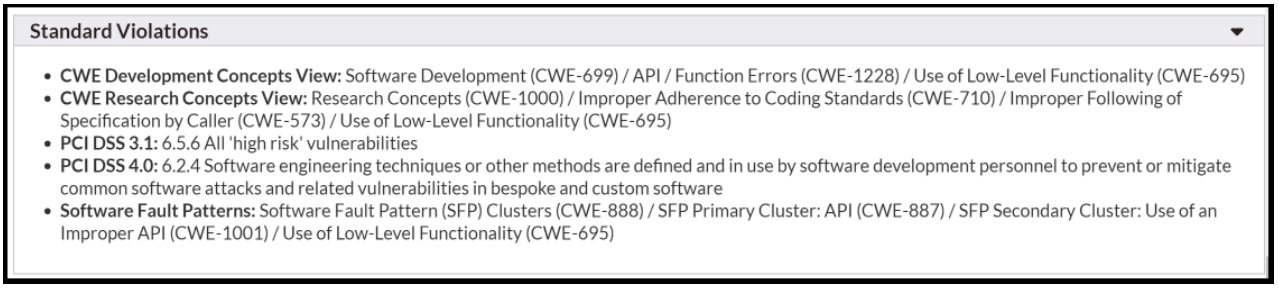
Training Video

Software Risk Manager integrates with Secure Code Warrior by providing training videos that developers can use to learn secure coding practices. Software Risk Manager will present these videos on the Details page when they are available.

Note: The training videos use the `video/mp4` MIME type, which may not be supported by some browsers.

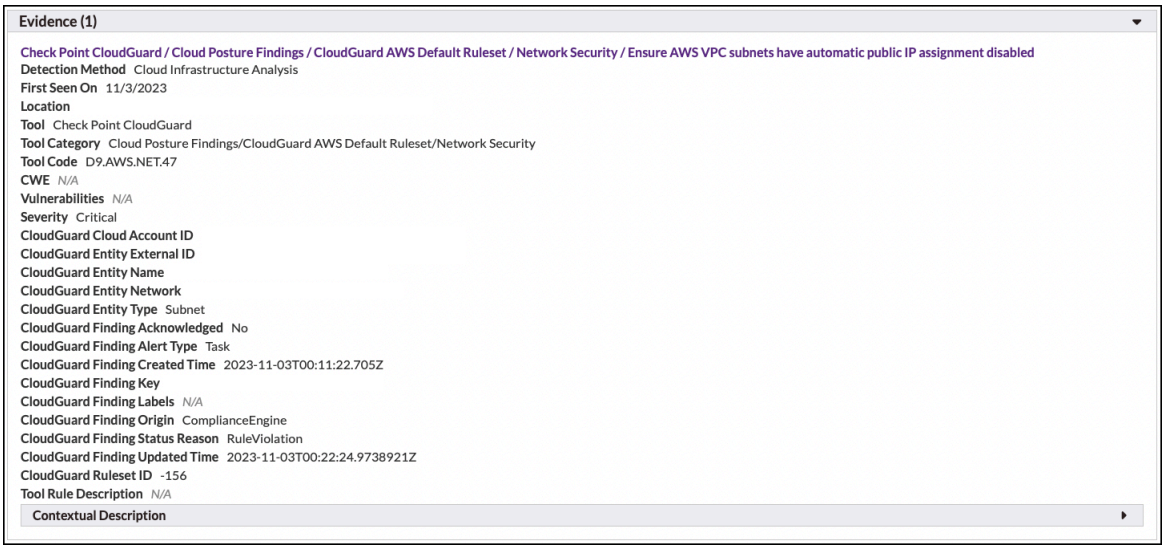
Standard Violations

The Standard Violations window lists which standards were violated in this finding.



Evidence

The *Evidence* section of the Finding Details page shows the raw results that make up a finding—regardless of whether the finding originated from an analysis tool or a manual entry. Each result in the Evidence section will be displayed in its own subsection, with the result's "type" as the header.



Each result in the evidence section includes the following fields:

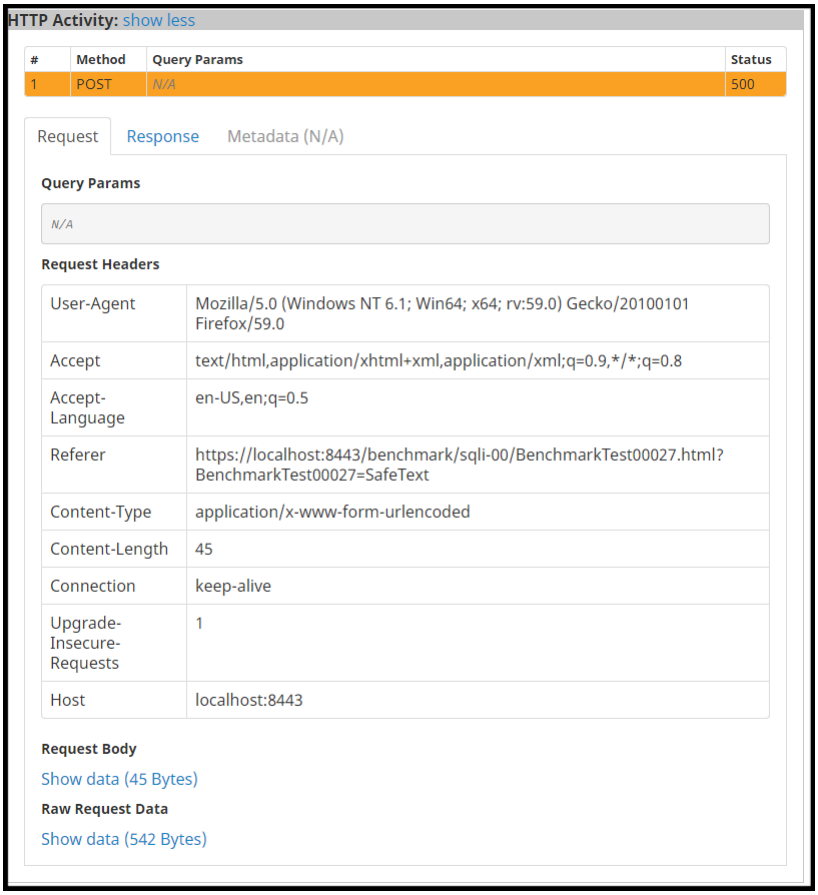
- **Detection Method.** Describes how the result was discovered (e.g., what type of analysis was performed), typically either "Static Analysis" or "Dynamic Analysis."
- **First Seen On.** For supported tools, reflects the date that the result was observed by that tool.
- **Location.** Describes exactly where the result was reported, i.e. a file, line number, and column number, or a URL. Note that because Software Risk Manager attempts to normalize locations, a result's location may differ slightly from the location of the finding it belongs to.
- **Tool.** The name of the tool that reported the result.
- **Tool Code.** The raw "code" that describes the type of the result (for example, "SQL Injection").
- **Tool Category.** The raw "category" (depending on the tool, the terminology may differ, e.g., "group") to which the tool code belongs.
- **CWE.** The Common Weakness Enumeration item reported by the tool, if available. Note that because Software Risk Manager uses *Rule Sets* to determine how results are combined into findings, a result's CWE may differ from its finding's CWE. To control this behavior, create or choose a different Rule Set to use in your project's *Analysis Configuration*.
- **CVE.** The Common Vulnerabilities and Exposures identifier reported by the tool, if available.
- **Severity.** The Software Risk Manager equivalent of the tool's reported severity. Note that some tools use different terminology for this, such as "priority," or different scales, like "1 through 10." Also note that when multiple results of different severities are combined, the finding will either take the severity of the rule that joined them or the highest severity from among the combined results.
- **Related Findings.** Lists any other findings that share the result, when applicable. This is more common when dealing with DAST and Hybrid results, as correlation between DAST results is less strict than with SAST.
- **Tool Rule Description.** The tool's unique description of the *type* of result being reported. For example, with a SQL Injection result, this is where you could find the tools' description of *what SQL Injection is*. Click the "show more" and "show less" links to expand and collapse the description.
- **Contextual Description.** The tool's description of the specific result. For example, with a SQL Injection result, this is where you could find out how this particular part of your codebase is vulnerable to SQL Injection. Click the "show more" and "show less" links to expand and collapse the description.

- **Host Info.** Shows the raw host data of the result being reported for Network Analysis results as well as certain DAST results. The host data is displayed as a table, with the left column indicating the host "field" (e.g., *Fully Qualified Domain Name (FQDN)*, *Hostname*, *IP Address*, etc.), and the right column containing the values for that field. The *Also include values from host normalization* checkbox will cause the page to include any values from the "normalized" version of this host to also be displayed. The "normalized" host is what is shown on the *Hosts* page. Values present on the normalized host and not present on the raw host will be rendered in green.
- **HTTP Activity.** This is shown for DAST results, and it will expand to its own subsection, which is described here.

Some tools report additional information that may appear in this section, such as Veracode's *Flaw ID*. When these additional fields are present in a project, some of them may become available as a finding search option on the Findings page.

HTTP Activity

The *Http Activity* section shows any detail Software Risk Manager knows about the HTTP request and response associated with a DAST result.



The table at the top of the *HTTP Activity* section enumerates the "variants" of request/response that were reported with the result. Some tools will attack the same URL with different variations of query parameters and form parameters to try and find vulnerabilities, then report each variant as part of the same result. Other tools will report each variation as its own result, but if Software Risk Manager sees that everything else is the same, it may join them together under a single result. Often times, there is only one variant reported, as

is the case in the screenshot above. In cases where there are multiple variants, click on the different rows of the variants table to show the details for that variant in the sections below.

For each variant, the details are described in the sections that follow.

Request Tab

The details of the HTTP request are broken down here:

- **Query Params** will show any applicable query parameters (i.e. parts of the URL after the ?, e.g. ? `foo=1&bar=2`)
- **Request Headers** shows each of the headers sent with the HTTP request, as a table.
- **Request Body** shows the body data sent with the request, if applicable. This is where form parameters go, or any other arbitrary content being sent to the server.
- **Raw Request Data** shows the raw, un-parsed version of the request. Note that some tools don't report this, instead reporting specific details of the request. In these cases, the raw request will be reconstructed automatically.

Response Tab

The details of the HTTP response are broken down here:

- **Response Headers** shows each of the headers sent with the HTTP response, as a table.
- **Response Body** shows the body data sent with the response, if applicable. Expanding it will show a text view of the raw response data. Note that response bodies are typically rather large, and are sometimes non-text data, which may make this view difficult to use. Software Risk Manager will not attempt to render whatever the data represents, since the assumption is that the response contained some vulnerability like cross-site scripting, or a malicious file.

Metadata Tab

Some tools will report extra "metadata" with their HTTP activity. When applicable, this data will be shown in the *Metadata Tab* as a table.

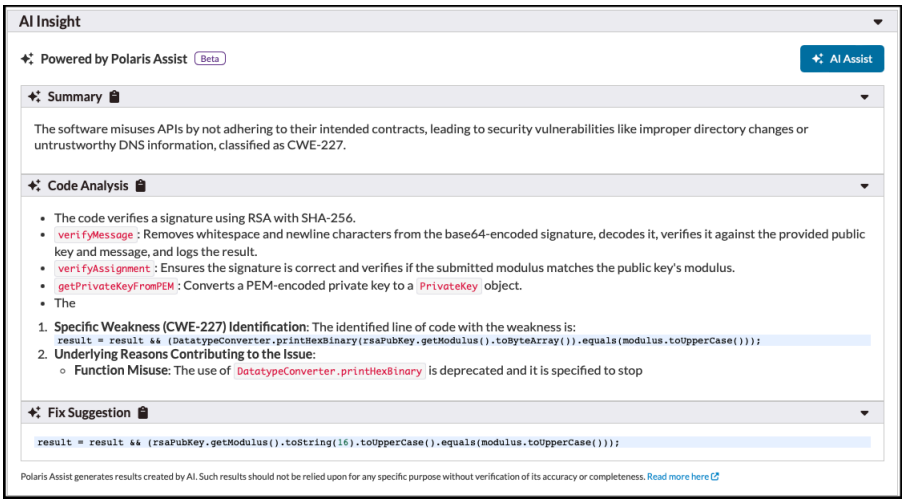
AI Insight Using Polaris Assist

When Polaris Assist is configured and enabled, an "AI Insight" section will be available for findings which have a *Static Analysis* detection method and have the required information for at least one of the available assessments listed below. (For information on configuration, see Polaris Assist.)

Note: Users must have the permission `Request and view finding assessments from Polaris Assist (Beta)` for the project, which is included in the default `Reader` role provided by SRM.

Warning: Polaris Assist generates results created by artificial intelligence (AI) or other automated technologies. Such results are provided for informational purposes only and should not be relied upon for any specific purpose without verification of its accuracy or completeness.

Click the AI Assist button to expand the fields.



When possible, SRM will offer up to three sub-assessments:

- **Summary.** Polaris Assist will provide a brief summary of the vulnerability and its impact. Requires finding description and CWE.
- **Code Analysis.** Polaris Assist will summarize an excerpt of the affected code and surrounding lines and a description of the vulnerability in the context of the affected code.
  - Code summary requires a file location with source code.
  - Vulnerability analysis requires a file location with source code, finding description, and CWE.
- **Fix Suggestion.** Polaris Assist will suggest an updated code snippet that attempts to address the vulnerability.
  - Requires a file location with source code, finding description, and CWE.

Source Code

The *Source Code* area shows the contents of the file where the finding is located. The "line" link will scroll the source display so that it shows the exact lines of the finding, which are highlighted in dark grey in the line number gutter. The presence of severity markers in the gutter denote other findings in the same file. When multiple findings are present in a single line, the severity marker will show the highest-level severity at that line. If you hover your mouse over any highlighted lines, a popup containing links to the *Finding Details* pages for the other findings will appear.

Source Search

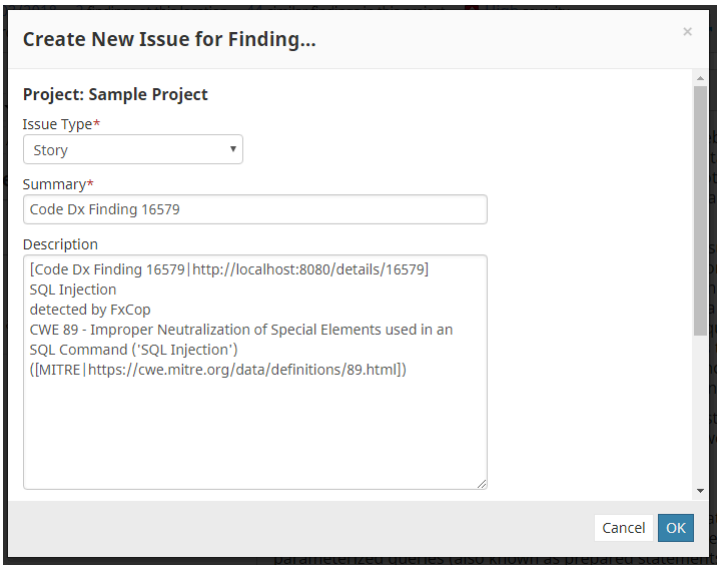
Searching within the *Source Code* area is separate from your browser's default search function. (For performance reasons, the *Source Code* view does not render the entire source file at once, so your browser might not be able to find lines that are not currently in view.) Click in the Source View first.

Issue Tracker

If a project has an Issue Tracker Configuration, the *Create issue* and *Use existing issue* buttons will be shown for Jira and GitLab users, *Create work item* and *Use existing work item* for Azure DevOps users, and *Create incident* and *Use existing incident* for ServiceNow users. Users with the `update` role are allowed to interact with the configured issue tracker.

Creating an Issue

You can click the *Create issue*, *Create work item*, or *Create incident* button, which will open a dialog.

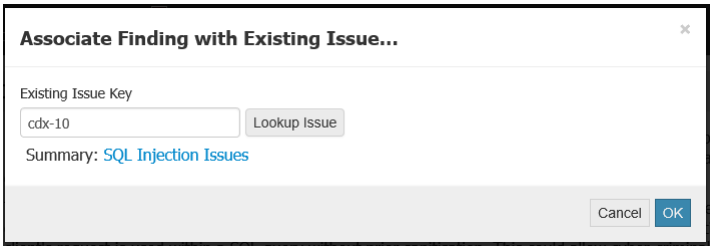


The dialog functions the same way as the dialog opened from the Bulk Operations area of the *Findings Table*, except the *Description* field will be pre-populated with information about this finding. When manually creating an issue in this way, the issue will be associated with the current branch view and will reference that finding's branch-specific data (e.g., *Severity* and *Status*).

Note: In contrast to manual issue creation, *Auto Create* currently only supports associating issues with a project's default branch.

Associating a Finding with an Existing Issue

Click the *Use existing Issue*, *Use existing work item*, or *Use exiting incident* button to associate this finding with an existing issue, work item, or incident. A dialog will open.



Refreshing Issue Status

Select the refresh icon to manually trigger a refresh of the issue or work item.

Removing Associations

Clicking the trash icon removes the association between the finding and its related issue or work item. Note: This only removes the association; it doesn't touch the issue or work item itself.

Predicted Status

Note: This section is only applicable to Software Risk Manager users with the Machine Learning Triage Assistance add-on.

A finding's prediction is included on the Finding Details page only if machine learning is enabled on the Machine Learning Control Panel. Each prediction is presented as a *Predicted Status*and a *Prediction Confidence*. Users can set a finding's Status to its *Predicted Status* by clicking on the Use Prediction button, which is next to the finding's prediction.

Fix By

Software Risk Manager allows users (with the `update` role) to change the finding's fix-by date. If the project is configured to use policies, a user can override a fix-by date that was determined by a policy rule. If the project is not configured to use policies, a user can simply set or remove a fix-by date.

The controls for setting the fix-by date are located under the finding severity information, above the tag controls.

The dropdown allows the user to either pick a specific date to set as the finding's fix-by date, or the user can choose to remove a finding's fix-by date.

Attachments

Allows files to be attached to the finding. Existing attachments are listed here.

To attach a file, click the Add Attachment button and select a file or use drag-and-drop. Click an attachment to download its contents. To delete an attachment, click the dropdown configuration icon and select Delete.

Triage Status Definitions

SRM Triage Status definitions are as follows:

- **Not Triaged.** (Not yet assigned a status.) The finding has not been assessed or categorized.
- **Fixed.** The finding has been directly fixed in the current version of the software and is awaiting confirmation by a later scan which would set the Finding Status to "Gone."
- **Mitigated.** The vulnerability has not been fixed, but steps have been taken to reduce its impact or likelihood.
- **Ignored.** The vulnerability has been deemed insignificant and does not currently warrant action.
- **False Positive.** The reported finding is not an actual vulnerability. After review, it is determined to be incorrect or misleading, and no action is needed.
- **To Be Fixed.** The finding has been assessed and flagged as important and therefore needs to be fixed.
- **Reopened.** The finding has been reopened per the analysis configuration settings. (See Analysis Configuration Options.)

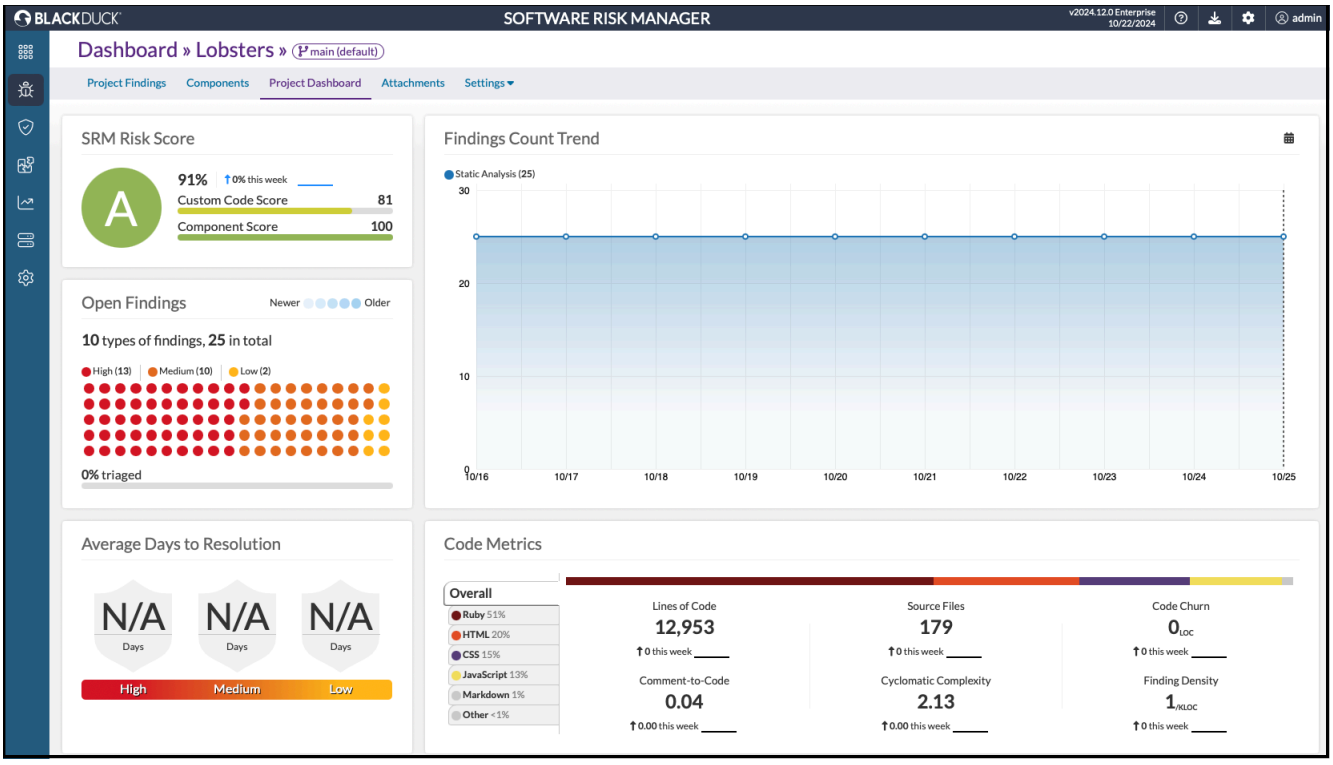
Finding Status Definitions

SRM Finding Status definitions are as follows:

- **New.** The finding was new in the last analysis that detected it.
- **Existing.** This finding has been detected in more than one analysis.
- **Gone.** This finding is no longer being reported by any tools.

Project Dashboard

The Project Dashboard provides a comprehensive overview of a project, displaying a set of analytic and trend data which are automatically updated as you use Software Risk Manager.



You can open the dashboard from the project list page or the project findings page:

- From the findings page, click the Project Dashboard tab at the top of the page.
- From the project list page, open the task dropdown list and select "Dashboard."

When viewing the Project Dashboard for the parent of grouped projects, you will have the option to include data from the child projects by using the roll-up feature. To do this, enable the *Include child projects* switch on the top right corner of the Project Dashboard page.

You can also access the Project Dashboard for "all projects" by clicking the "Dashboard" tab while on the aggregate "All Projects" Findings page.

While on the Project Dashboard page for a project with multiple branches, you can switch the view to a different branch by clicking the branch selector dropdown in the page header.

Dashboard Data

The *Project Dashboard* is divided into several sections:

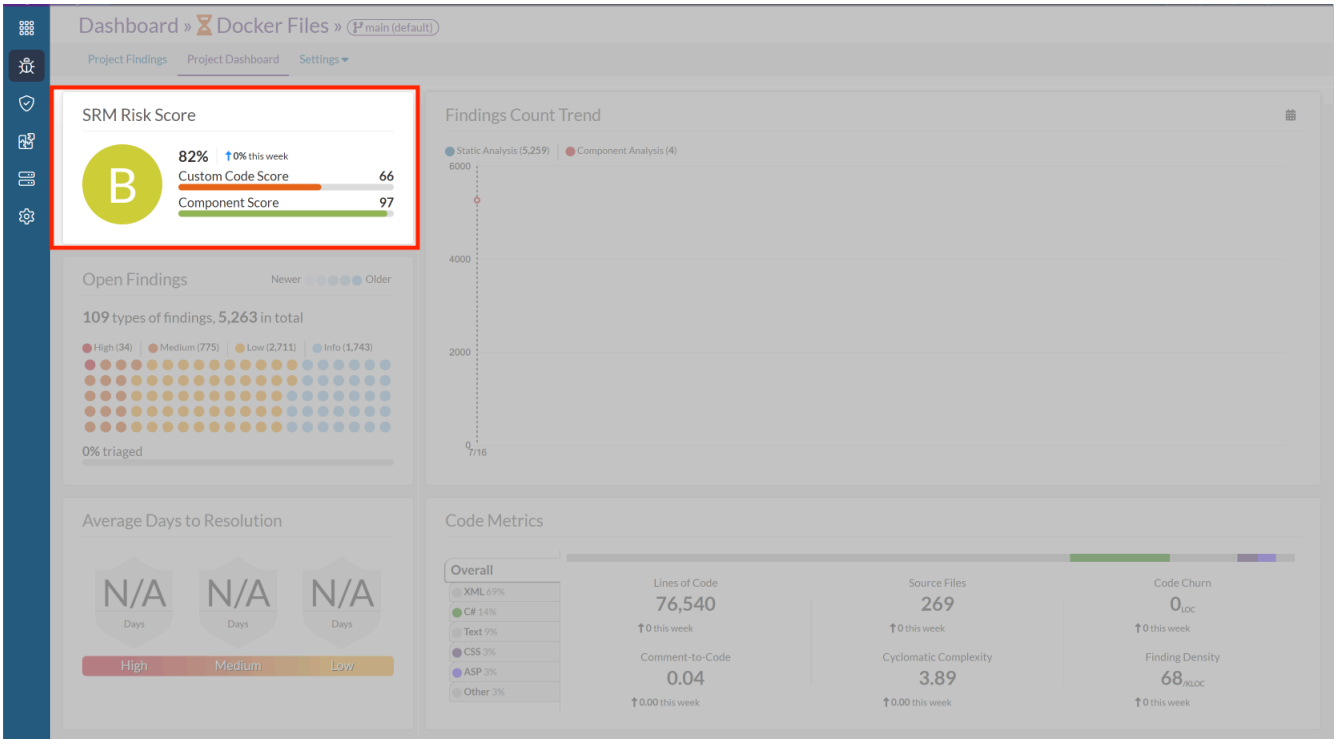
- Risk Score. Provides a "letter grade" to indicate the overall quality of the project.
- Open Findings. Displays the total number of findings, along with a breakdown of the number of findings in each category and the percentage of findings that have been triaged.
- Findings Count Trend. Shows the number of findings over time, broken out by detection method.
- Average Days to Resolution. Shows the average number of days it took to resolve an issue, broken out by severity.
- Code Metrics. Displays various metrics for the project's codebase, such as the number of lines of code, number of source files, etc., broken down by language.
- Analysis Frequency. Provides detailed information about when the analysis was run, how long it took, tools used, etc.

- Activity Monitor. Displays a "calendar heatmap," representing the analysis activity on the project over the past year.
- Created vs. Resolved. Provides a visual representation of the dueling trend of new findings that are added to the project and findings that are resolved by the team, along the difference between the two.
- Top Finding Types. Shows the top 10 types of findings in the project, by number of open findings.

Risk Score

The *Software Risk Manager Risk Score* section of the *Project Dashboard* provides a letter grade to indicate the overall "quality" of the project.

Figure 1: Risk Score



The letter grade is based on a percentage score, which is the average of the *Custom Code Score*, *Component Score*, and *Infrastructure Score*. Each score is weighted evenly, but note that an *Infrastructure Score* is not available for all projects. The score letter grades are = A, [80, 90) = B, [70, 80) = C, [60, 70) = D, [0, 60) = F.

- **Custom Code Score** starts at 100% and is reduced based on the "volume" and "variety" of non-"Component" and non-"Infrastructure" findings.
  - **volume penalty =**(3.0% × $\log_2$(*num_critical_findings* + 1)) **+**(1.5% × $\log_2$(*num_high_severity_findings* + 1)) **+**(0.75% × $\log_2$(*num_medium_severity_findings* + 1)).
  - **variety penalty =**(3.0% × *num_critical_finding_types*) **+**(1.5% × *num_high_severity_finding_types*) **+**(0.75% × *num_medium_severity_finding_types*).
- **Component Score** starts at 100% and is reduced based on the "volume" of "Component" findings. "Component" findings are any findings found when running Component tools.
  - **volume penalty =**(3.0% × *num_critical_component_findings*) **+**(1.5% × *num_high_severity_component_findings*) **+**(0.75% × *num_medium_severity_component_findings*).

- **Infrastructure Score** (only reported for projects that have findings from infrastructure analysis) starts at 100% and is reduced based on the "volume" of infrastructure findings. Infrastructure findings are any findings found when running Infrastructure or Cloud Infrastructure Tools.
  - **volume penalty =**(3.0% × *num_critical_infrastructure_findings*) **+**(1.5% × *num_high_severity_infrastructure_findings*) **+**(0.75% × *num_medium_severity_infrastructure_findings*).

Each of the "*num_*" values mentioned above refer to findings in the project which haven't been triaged (i.e., findings whose triage statuses haven't been marked as one of the "resolved" statuses like "Fixed" or "False Positive"). In the case of "volume," they refer to the *number* of findings. In the case of "variety," they refer to the number of distinct *types* of findings. Only critical, high, and medium severity findings are counted against the Software Risk Manager Risk Score.

Next to the letter grade, the specific percentage score is displayed alongside a spark-line that shows the general trend of the project's *Software Risk Manager Risk Score* over the past week.

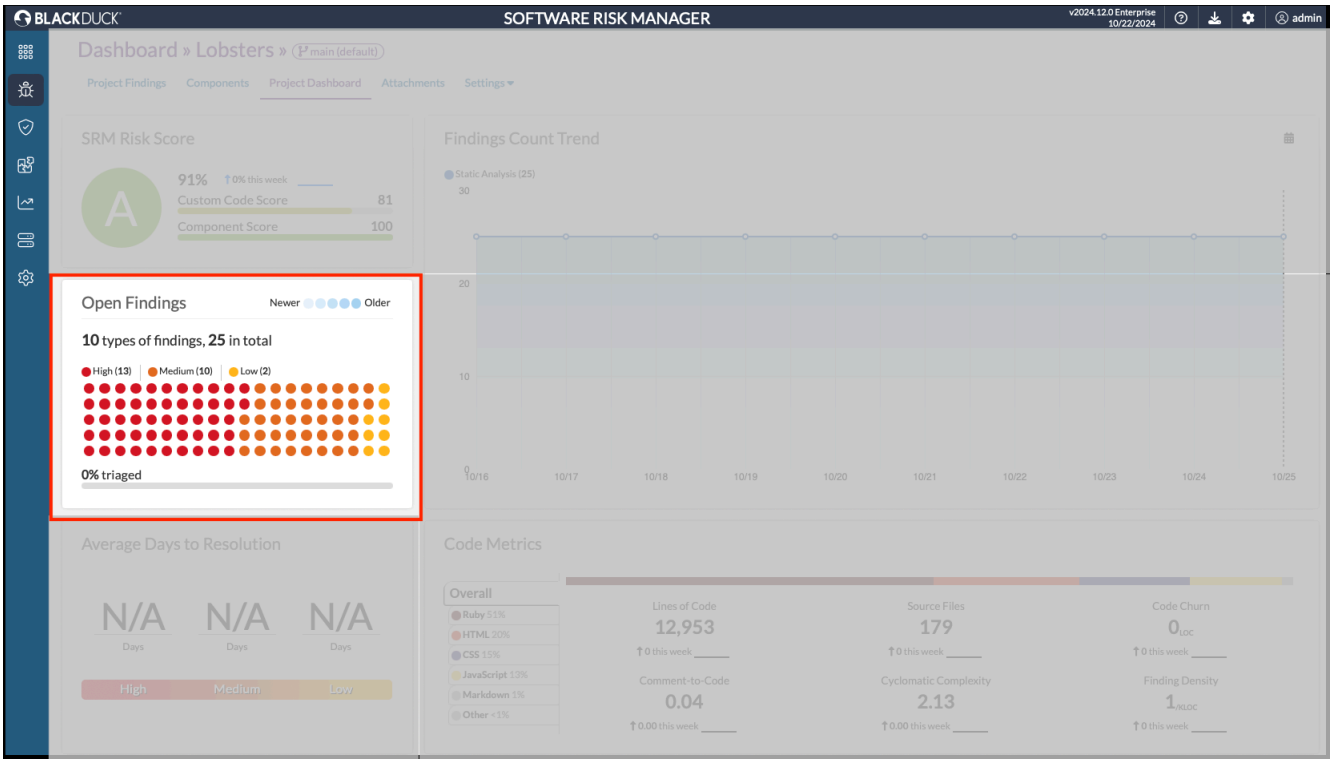
The individual scores for the *Custom Code Score* and *Component Score* are shown by a pair of "fill bars" next to the letter grade, below the overall score percentage.

For more information on how these scores are calculated and how to customize the formula, see the *Software Risk Manager Scoring Calculations* section in the *Software Risk Manager Install Guide.*

Open Findings

The Open Findings section shows the overall "triage status" of the project.

Figure 2: Open Findings



A waffle chart is used as a severity-age breakdown of the untriaged findings in the project. Different colors indicate different severities, as indicated by the legend. The number of dots of each color indicate the percentage (rounded) of findings in the project which have that specific severity. I.e. if there are 19 purple

dots, it means 19% of the untriaged findings have "critical" severity. Transparency is used to indicate the relative age of the findings, as indicated by the legend. A lighter (more transparent) version of the severity color indicates findings of that severity which are relatively new. A darker (more opaque) version of the severity color indicates findings of that severity which are relatively old.

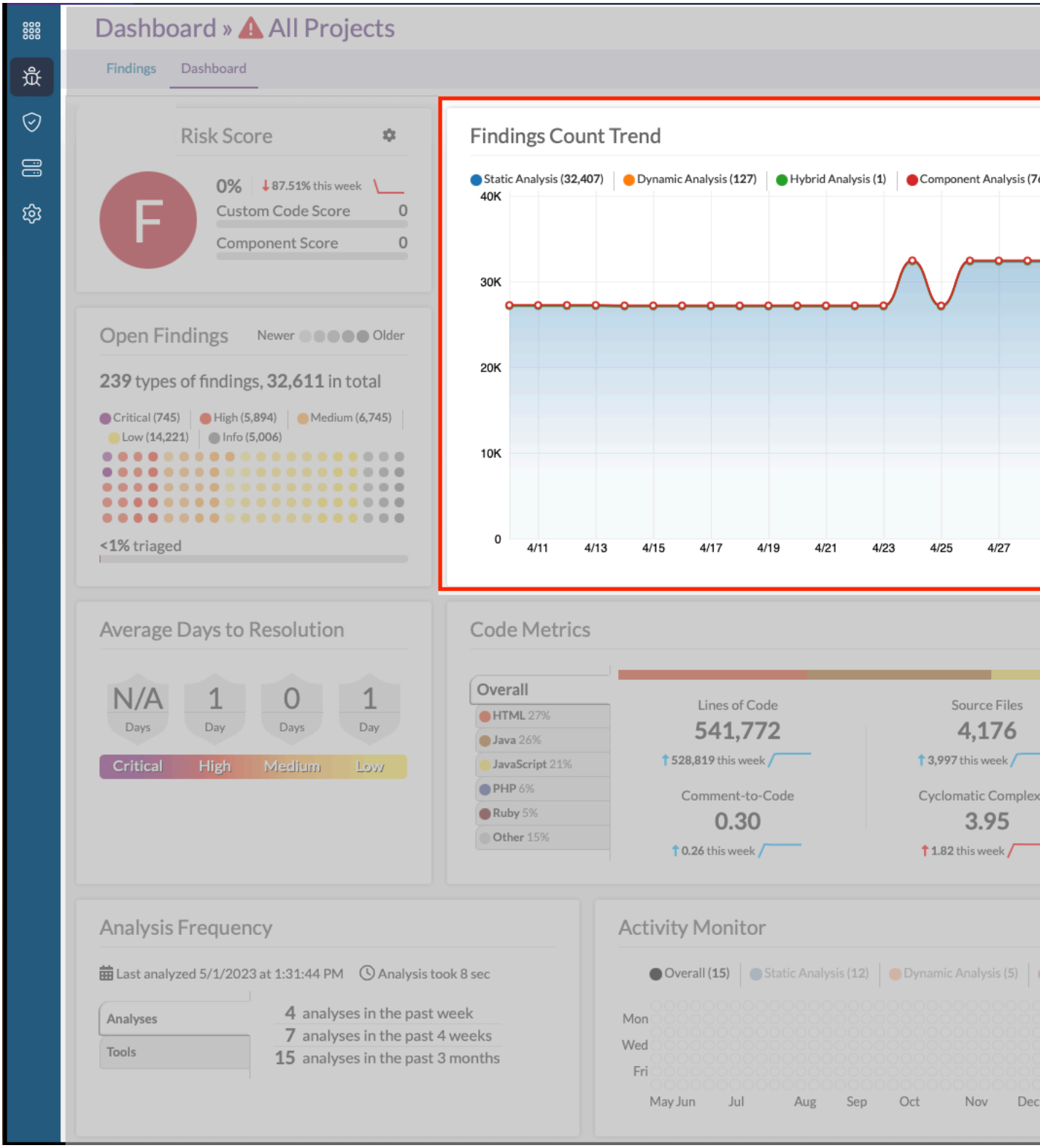
Clicking on the severity labels in the waffle chart's legend will cause the chart to focus on that severity, fading the other severities from view. Clicking again on the same label will reset that focus, returning the visualization to its normal state.

Hovering the mouse cursor over the severity labels in the waffle chart's legend, or over the colored dots in the waffle chart itself will cause the chart to temporarily focus on that severity. This effect is similar to the click effect described in the previous paragraph, but the effect does not persist if the mouse leaves the area that caused the focus. Hovering the mouse over the chart will also show a tooltip containing a summary of the respective hovered severity.

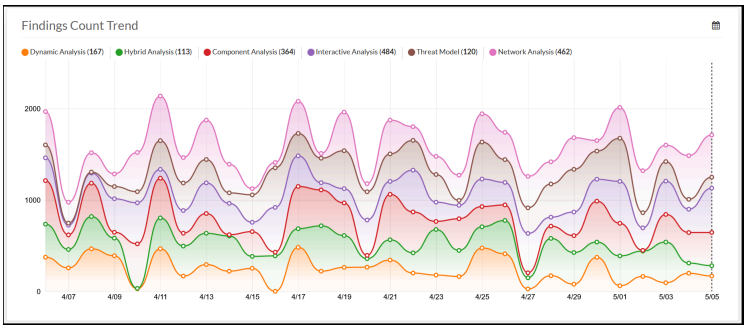
Below the waffle chart is a fill-bar indicating the percentage of findings which have been triaged (i.e. set to Fixed, Mitigated, False Positive), out of the total number of findings in the project, excluding findings that are marked "Gone".

Findings Count Trend

Figure 3: Findings Count Trend

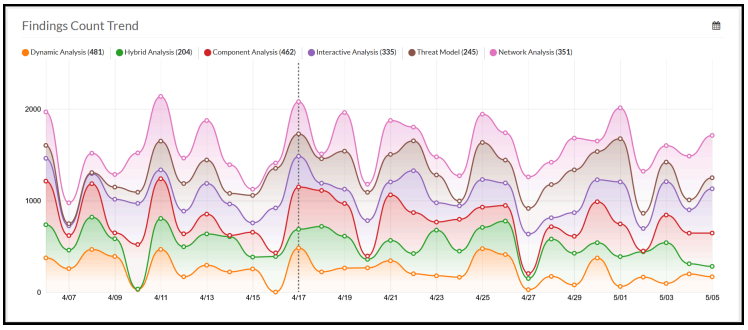


The *Findings Count Trend* section of the *Project Dashboard* shows a breakdown of findings by "detection method" over time.

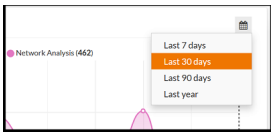


The *Findings Count Trend* visualization uses a stacked area chart, with "date" as the X axis, and total finding count as the Y axis. By default, an area for each detection method is shown, so that the stacked areas' total height indicates the total number of findings at a given date. Clicking one of the detection method labels in the legend will cause the visualization to focus on the respective detection method, hiding the other areas and moving the focused area to the bottom of the visualization. Clicking again on the same detection method label in the legend will remove the focus effect, returning the visualization back to its default state.

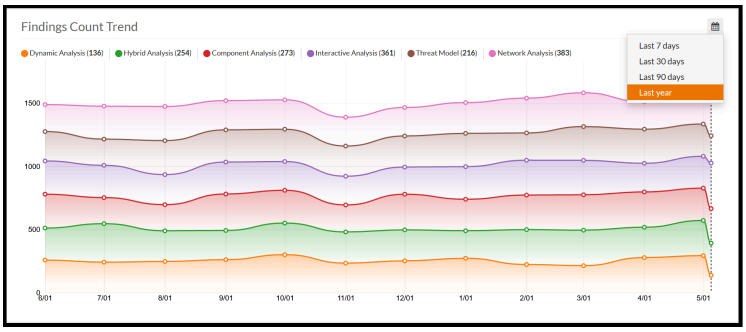
Hovering the mouse cursor over the visualization will cause a vertical line to snap to the nearest date, updating the legend to reflect the finding counts at that date. While the mouse cursor is not over the visualization, the vertical line will snap to the latest date, causing the legend to reflect the most recent finding counts



On the top-right of the trend graph is a calendar icon, which can be clicked to bring up a menu for selecting a date range.



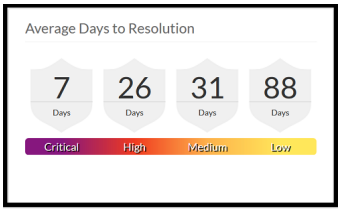
Selecting one of these range values will automatically refresh the graph to the selected range. For larger date ranges, each point in the graph can represent multiple dates by taking the *average* of data samples involved.



Average Days to Resolution

The *Average Days to Resolution* section of the *Project Dashboard* shows the average number of days it takes for a new finding in the project to be resolved.

In this context, resolution means the finding either becomes "Gone" (because developers fixed the issue, and a new analysis did not encounter the same finding), or its triage status was set to one of the "resolved" statuses: False Positive, Fixed, Ignored, or Mitigated.

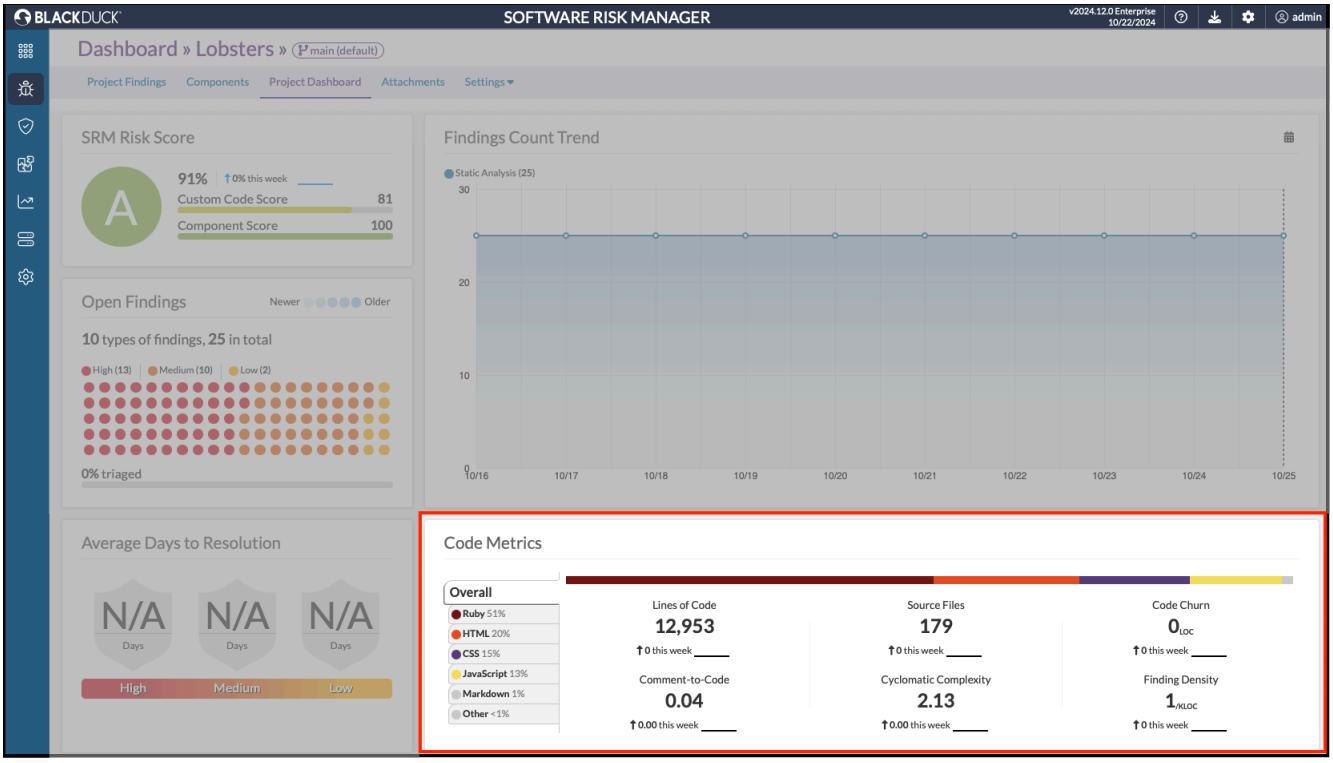


For each severity, the average number of days it takes to resolve a finding of that severity is displayed in a badge. Initially, each badge will display "N/A"; since no findings have been resolved, there is no "average" time. A colored bar below the badges acts as a legend, and hovering the mouse cursor over a badge causes it to become highlighted with that severity's respective color.

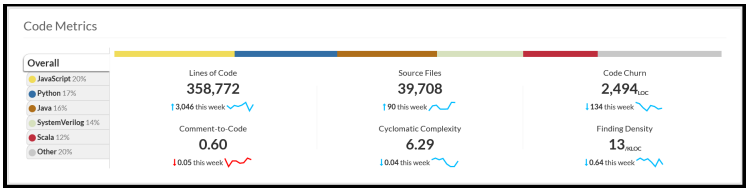
As a rule of thumb, teams may wish to prioritize addressing higher-severity findings, so team leads will want to see a lower number of days-to-resolution for higher-severity findings.

Code Metrics

Figure 4: Code Metrics



The *Code Metrics* section of the *Project Dashboard* displays a set of metrics for the project's codebase, broken down by language.

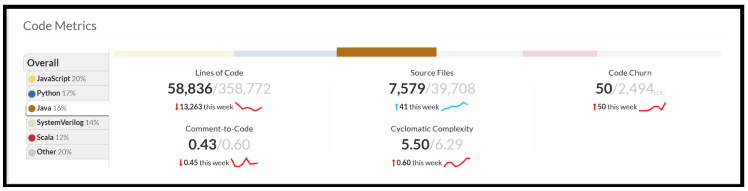


On the left of the section, a legend shows:

- An "Overall" group which represents the entire codebase; the sum of the metrics for each language.
- The top 5 languages (by percentage of lines of code in the respective language compared to the total number of lines of code).
- An "Other" group which contains the summation of any other languages after the top 5.

The colors assigned to each language are purely aesthetic, and are chosen using the same color scheme that Github uses.

By default, the "Overall" group is selected, so the metric areas to the right will show stats for the whole codebase. Clicking one of the languages, or the "Other" group in the legend will cause the metric areas to display language-specific stats. Clicking on the "Overall" group will return the display to its default state.



When focused on a particular language, each metric will show an "X / Y" value instead of the usual "Y". The "Y" indicates the metric's value for the entire codebase, and the "X" indicates the metric's value for the subset of the codebase which is written in the focused language.

Each metric area will also show a sparkline indicating that metric's trend over the past week. The sparklines will be colored blue for "good" changes, and red for "bad" changes.

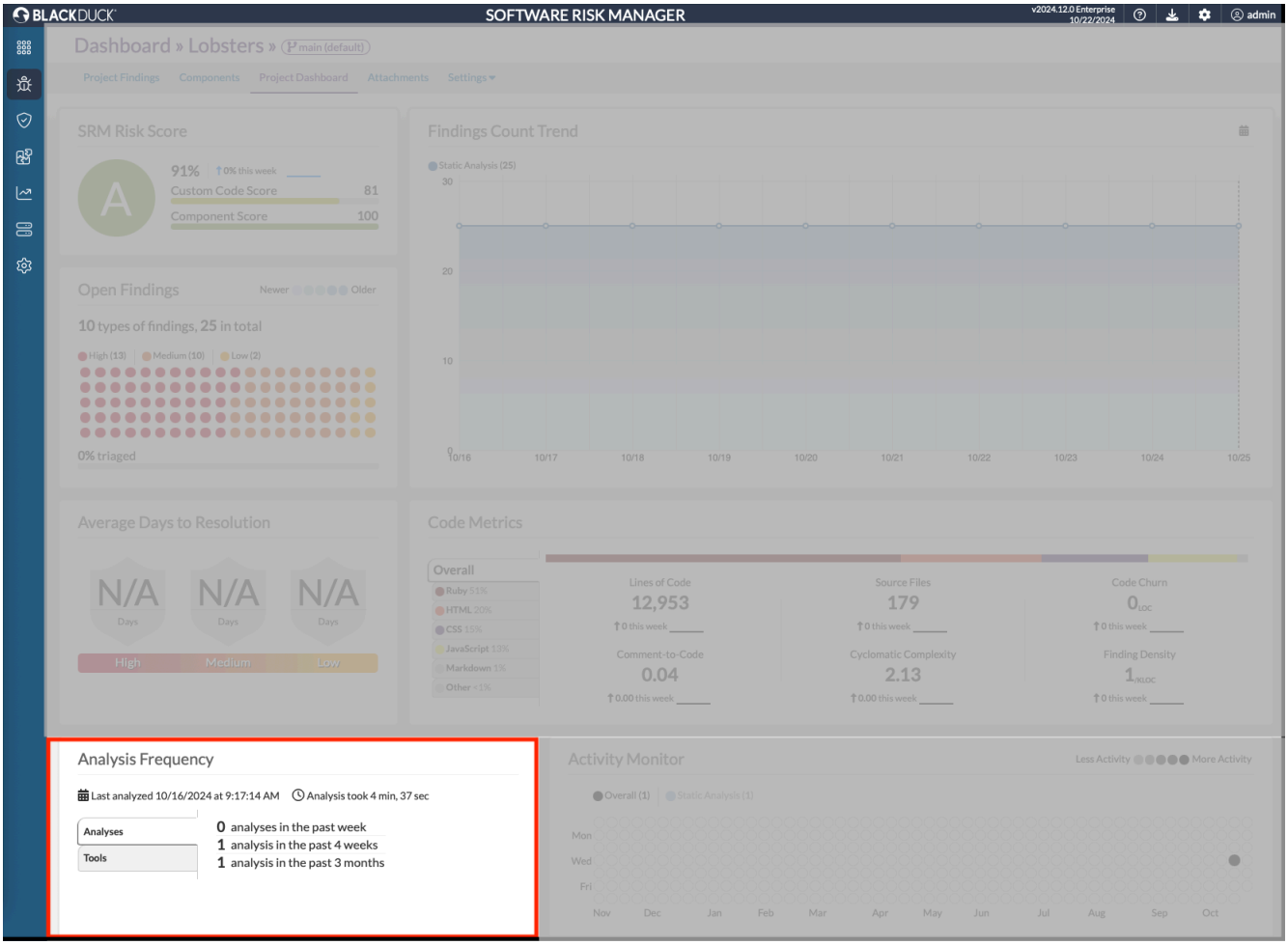
List of Code Metrics

- **Lines of Code** shows the number of lines of code (including blanks and comments) in the project's codebase. A rising number of lines of code is considered "good", as it implies the project is growing.
- **Comment-to-Code** shows the ratio of the number of comment lines to the number of total lines of code in the project's codebase. A rising comment-to-code ratio is considered "good", as writing comments is good development practice.
- **Source Files** shows the number of source files in the project's codebase. A rising number of source files is considered "good", as it implies the project is growing.
- **Cyclomatic Complexity** shows the average cyclomatic complexity number (CCN) of the project's codebase. A falling CCN is considered good, as less-complex code is easier to maintain.
- **Code Churn** shows the number of changed lines of code. Since all dashboard data is collected nightly, the value displayed today indicates the amount of code churn that occurred yesterday. A rising code churn is considered "bad", as high churn tends to increase the chance of introducing new vulnerabilities.
- **Finding Density** shows the number of findings per thousand lines of code in the project's codebase. A falling finding density is considered "good" since it means the defective percentage of the codebase is shrinking.

Analysis Frequency

The *Analysis Frequency* section of the *Project Dashboard* offers a summary of the project's most recent analyses.

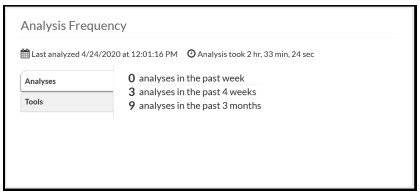
Figure 5: Analysis Frequency



At the top of the section, a text blurb describes when the latest analysis occurred, and how long it took. The rest of the section is broken down into three tabbed sections.

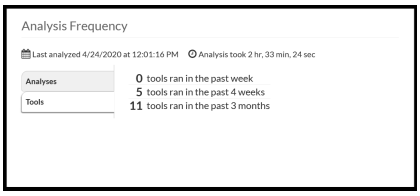
Analyses

This shows how many analyses were run on the project over the past week, 4 weeks, and 3 months.



Tools

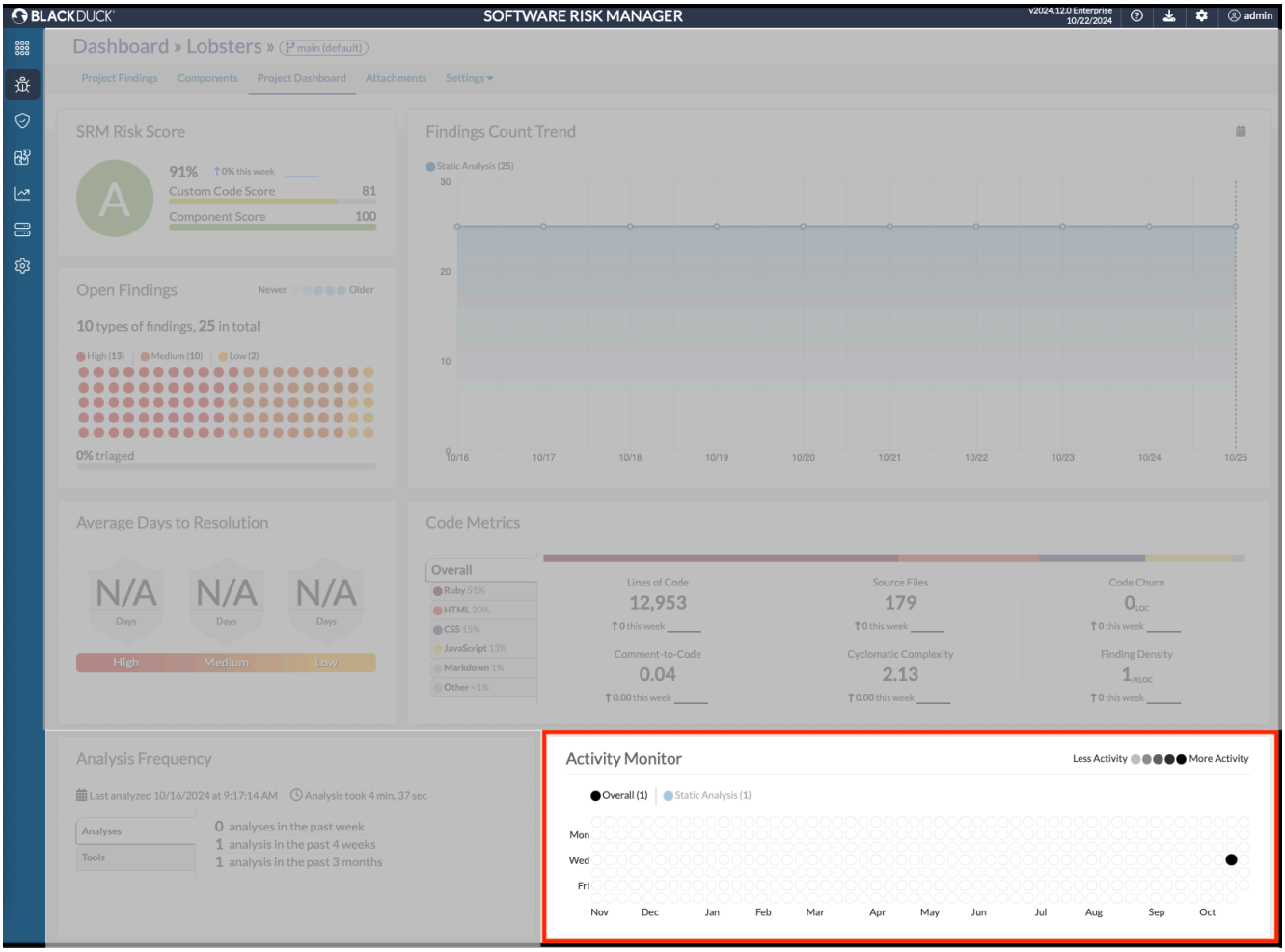
This shows how many unique tools were run in analyses on the project, over the same time periods. (Note that in this context, "tools that were run" means the set of Tools referenced by Findings in Software Risk Manager. It doesn't matter whether you ran the tool separately, or if Software Risk Manager orchestrated a run of the tool on its own.)



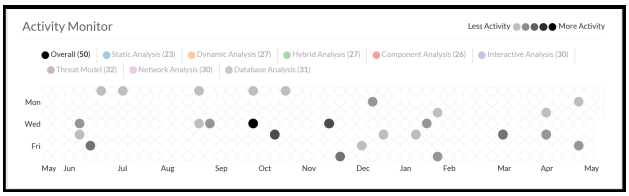
Activity Monitor

The Activity Monitor section of the Project Dashboard shows a "calendar heatmap" which represents the analysis activity on the project over the past year.

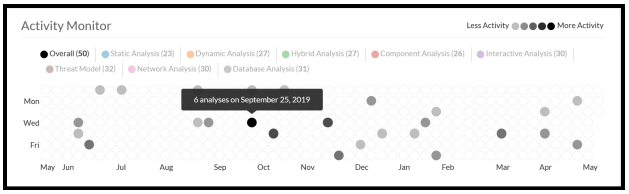
Figure 6: Activity Monitor

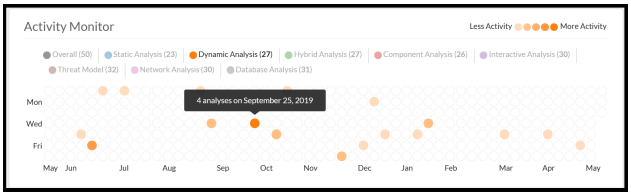


The far left represents dates from a year ago, and the far right represents recent dates. Stepping down a column of the chart, each bubble represents a day of the week, with Sunday at the top, and Saturday at the bottom. Hovering the mouse cursor over any of the bubbles in the chart will cause a tooltip to display the bubble's respective date, and the number of analyses that were run that day.



The analysis activity is broken down by different types of analyses, e.g. Static and Dynamic. The legend items below the visualization represent these different analysis types (i.e. "Detection Methods"). Note that any given analysis may result in findings of different detection methods, depending on what files were uploaded. Clicking the legend items below the visualization will cause the visualization to focus on the legend item's respective detection method. This can cause the number of analyses shown in the tooltip to change. For example, three analyses may have been run on a given day, but only two of those analyses resulted in data from Dynamic Analysis. In this case, if the "Overall" legend item was selected, the tooltip would show "3 analyses on {date}", but when the "Dynamic Analysis" legend item was selected, the tooltip for that same bubble would show "2 analyses on {date}."

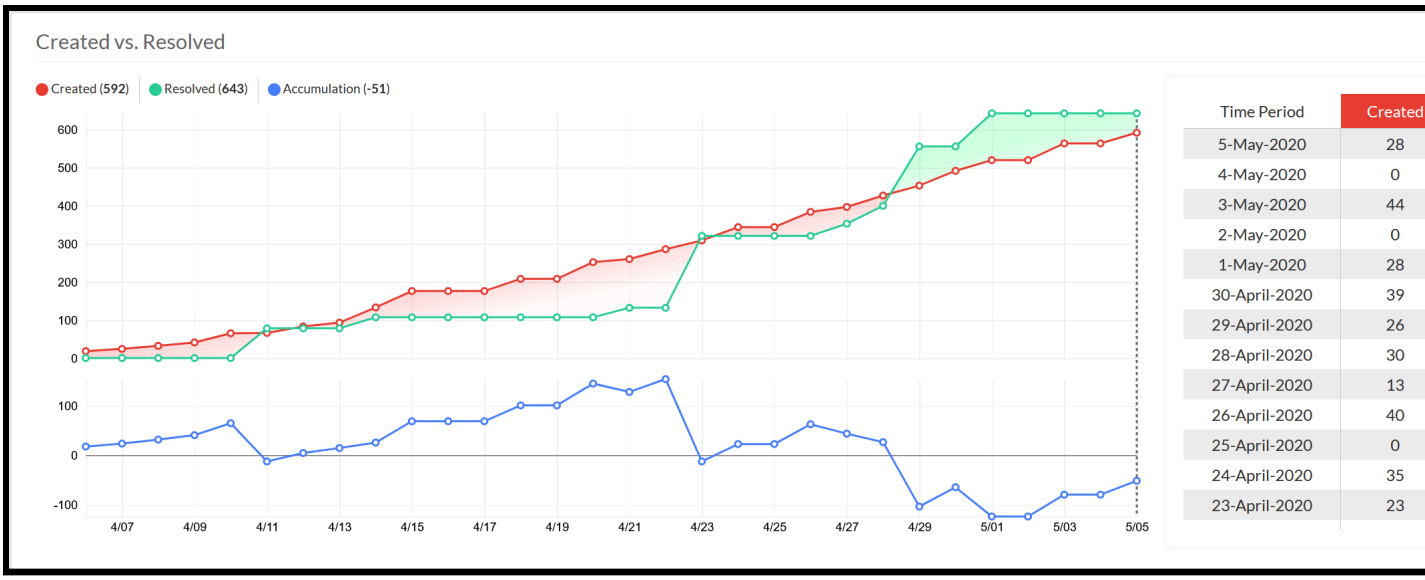




The visualization uses brightness to indicate more or less analysis activity for each given day, as indicated by the legend above the visualization. A darker shade of color indicates more analyses, and a lighter/whiter shade of color indicates fewer analyses.

Created vs. Resolved

The *Created vs. Resolved* section of the *Project Dashboard* shows the dueling trend of new findings that are added to the project, findings that are resolved by the team, and the difference between the two.



This section is broken into two pieces; the graph, and the table. Both represent the same data.

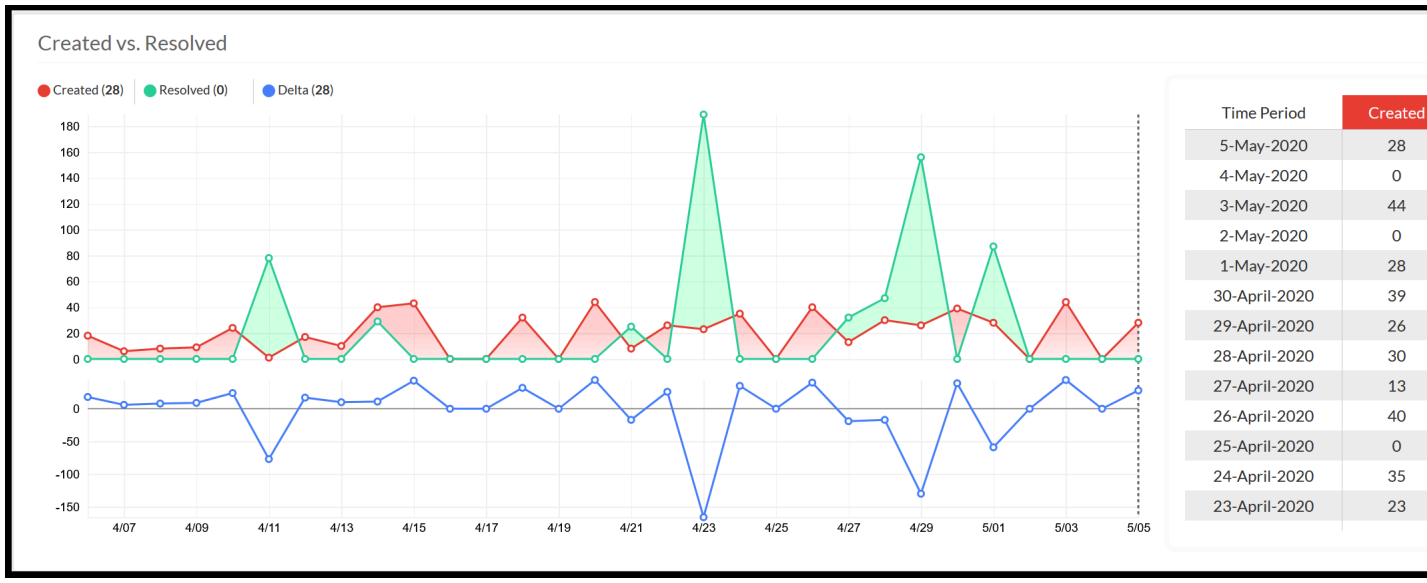
The graph is broken into two pieces; the "duel", and the "trend".

The "duel" section shows the number of created findings (in red) versus the number of resolved findings (in green). By default, the graph will show an accumulation of these numbers, starting from date at the far left of the graph. The icon in the upper-right corner of the *Created vs. Resolved* section opens a menu which allows you to toggle between "accumulated" and "daily" counts in the "duel" section. "Daily" counts show the exact number of created and resolved findings for any given day. The colored area between the lines in the "duel" section of the graph indicates which line is higher. A green fill means more findings were resolved as of that day (if using "accumulated" counts), or resolved *on* that day (if using "daily" counts).

The "trend" section of the graph shows the difference between the red and green lines of the "duel" (in blue). The "duel" and the "trend" graphs have their own separate Y axes representing cumulative finding counts, and count difference, respectively. The two graphs share the same X axis, which represents the date.

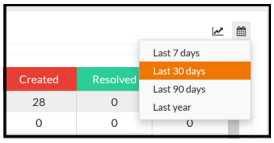
When hovering over the graph with the mouse cursor, a vertical line will snap to the nearest date to the mouse, causing the legend above the graph to update its numbers to reflect that date. The corresponding row in the table to the right of the visualization will be highlighted, and the table will auto-scroll to that row if necessary. Similarly, hovering over the table will cause the same changes, depending on which row in the table is hovered.

By default, the *Created vs. Resolved* section shows the *accumulated* number of findings since the beginning of the summary time window. Click the graph icon in the upper-right corner of the section, and select "Show daily counts" to switch the graph to Daily mode. Daily mode shows the change in values on a day-to-day basis. Accumulated mode can be considered the Integral of Daily mode, and Daily mode can be considered the Derivative of Accumulated mode.



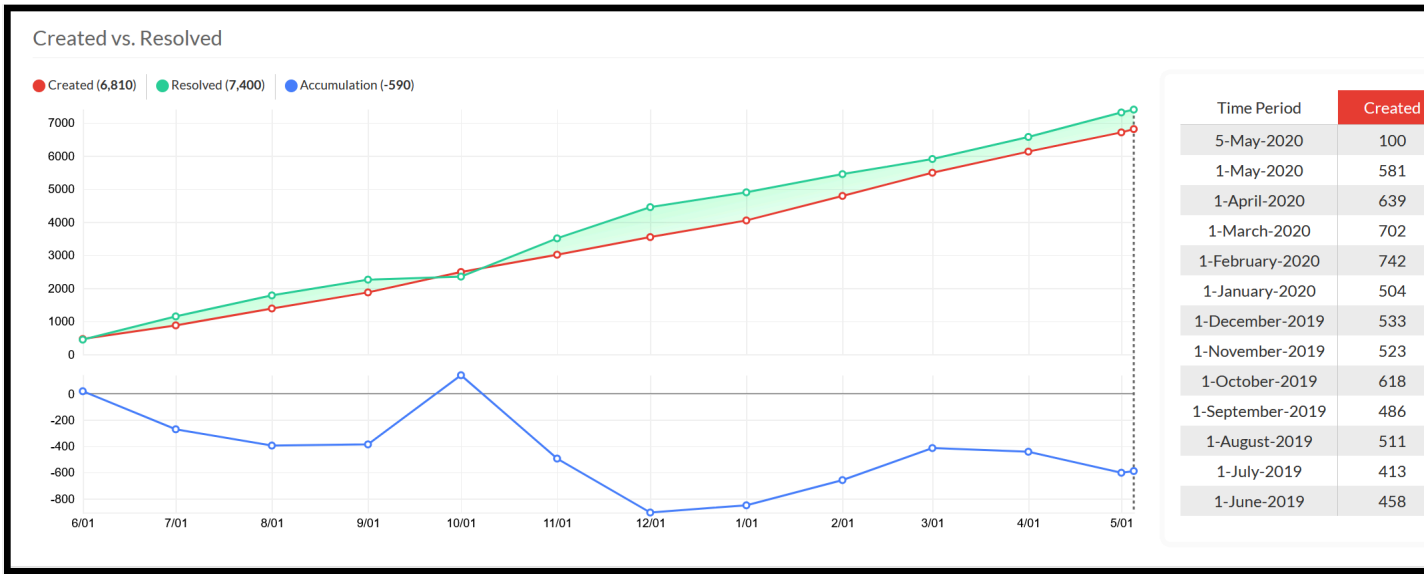
Time Period	Created
5-May-2020	28
4-May-2020	0
3-May-2020	44
2-May-2020	0
1-May-2020	28
30-April-2020	39
29-April-2020	26
28-April-2020	30
27-April-2020	13
26-April-2020	40
25-April-2020	0
24-April-2020	35
23-April-2020	23

On the top-right of the graph is a calendar icon, which can be clicked to bring up a menu for selecting a date range.

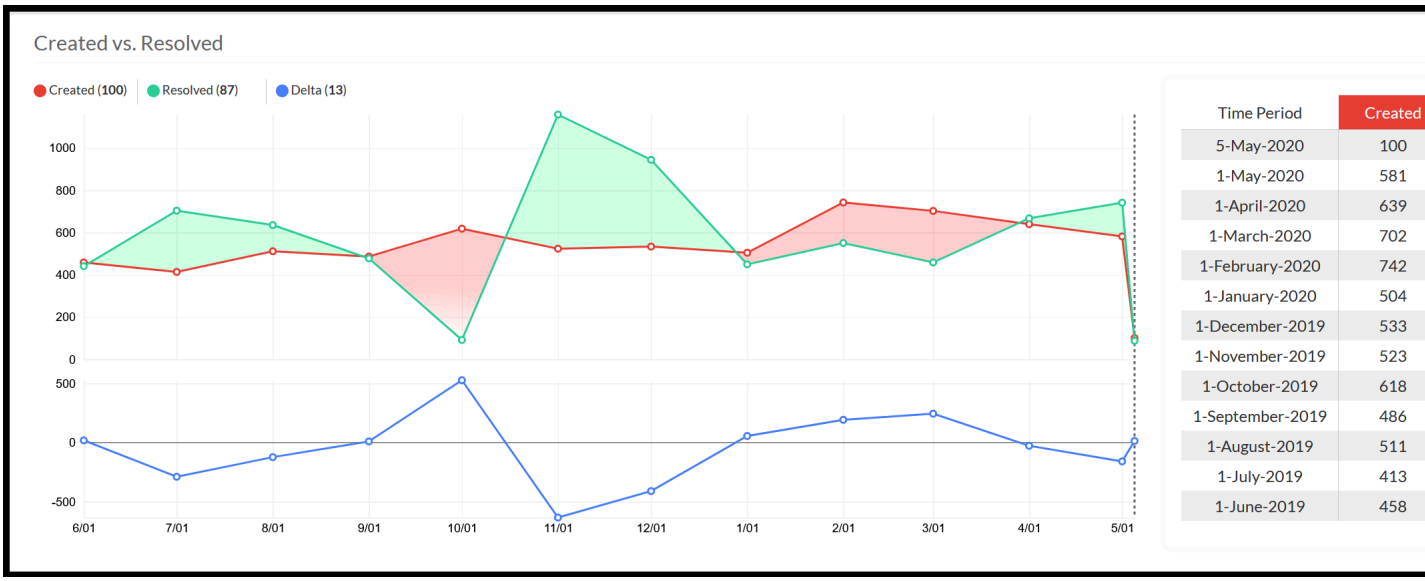


Selecting one of these range values will automatically refresh the graph to the selected range.

For larger date ranges, each point in the graph can represent multiple dates by taking the *sum* of data samples involved.



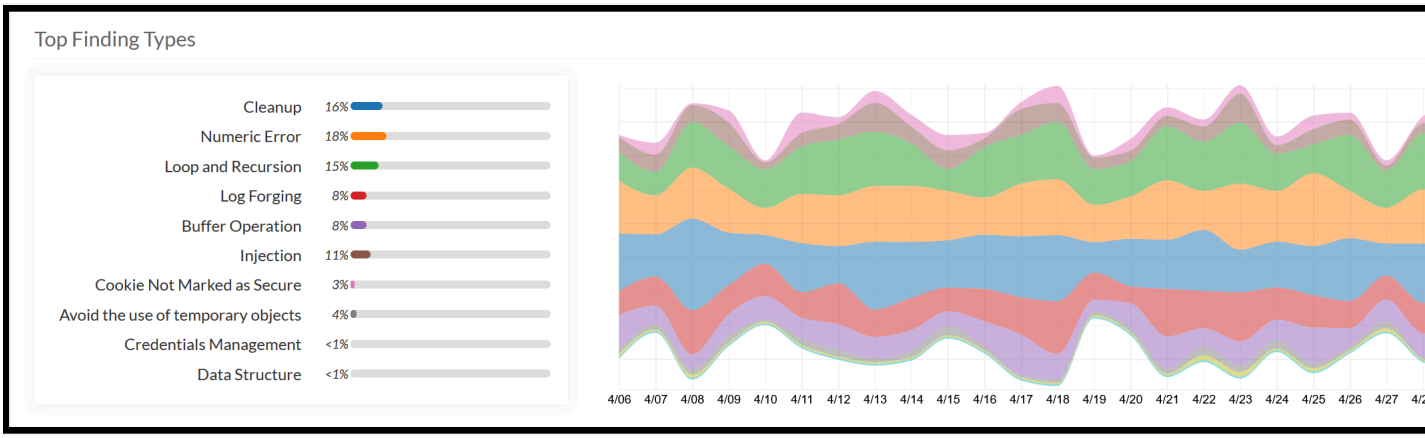
Time Period	Created
5-May-2020	100
1-May-2020	581
1-April-2020	639
1-March-2020	702
1-February-2020	742
1-January-2020	504
1-December-2019	533
1-November-2019	523
1-October-2019	618
1-September-2019	486
1-August-2019	511
1-July-2019	413
1-June-2019	458



Time Period	Created
5-May-2020	100
1-May-2020	581
1-April-2020	639
1-March-2020	702
1-February-2020	742
1-January-2020	504
1-December-2019	533
1-November-2019	523
1-October-2019	618
1-September-2019	486
1-August-2019	511
1-July-2019	413
1-June-2019	458

Top Findings Types

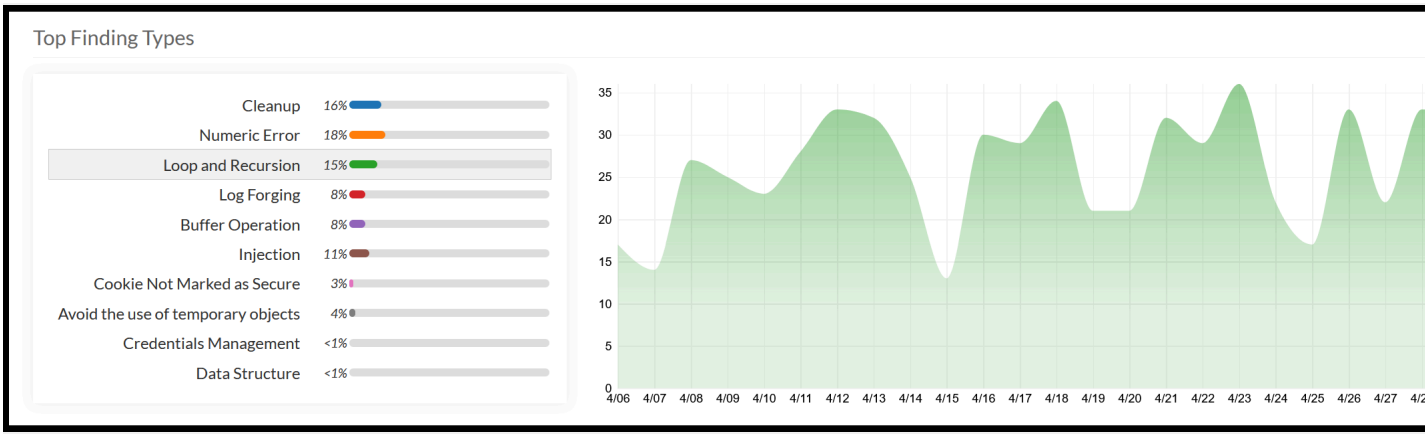
The *Top Finding Types* section of the *Project Dashboard* shows the top 10 types of findings in the project, by number of open findings.



The visualization uses a Stream Graph to represent the relative volume of the top finding types (Y axis) over time (X axis). Each stacked area of a given color represents a specific type of finding, e.g. "SQL Injection". The height of each area represents the number of findings of that type on a given day.

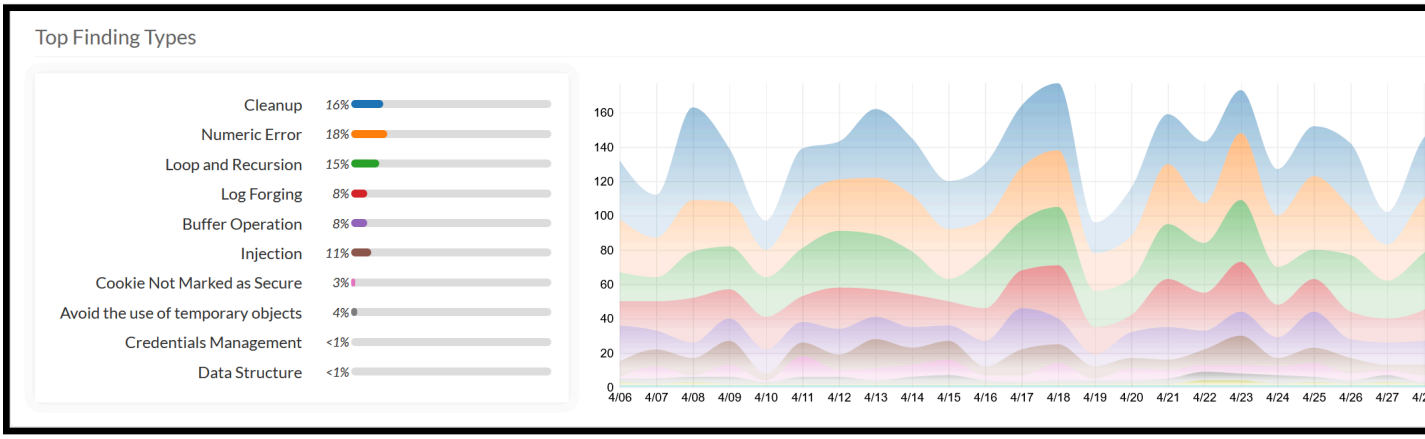
The table to the left of the visualization acts as a legend, where each of the finding types is labelled, and has a colored fill-bar indicating the respective finding type's percentage share of the project.

Hovering the mouse cursor over an item in the table to the left of the visualization will highlight the corresponding area in the visualization. Similarly, hovering the mouse cursor over an area in the visualization will highlight the corresponding item in the table. Clicking an item will cause that item to become "focused". Click the item again to undo the focused state, or click another item to change to another focused state.

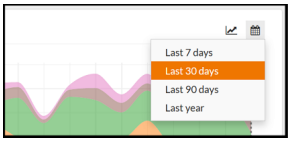


As with many of the other dashboard sections, hovering the mouse cursor over the visualization will cause a vertical line to snap to the date nearest to the mouse. When this happens, the table to the left of the visualization will update to reflect the percentages for that day.

Click the graph menu in the upper-right corner to access the "layout" options. By default, the graph uses "stream" layout. Switch to the "stack" layout to rearrange the items into a stack, such that the bottom of the stack aligns with the "0" on the Y axis. Note that with the "stream" layout, the Y axis's meaning differs from date to date, so no axis numbers will be displayed.

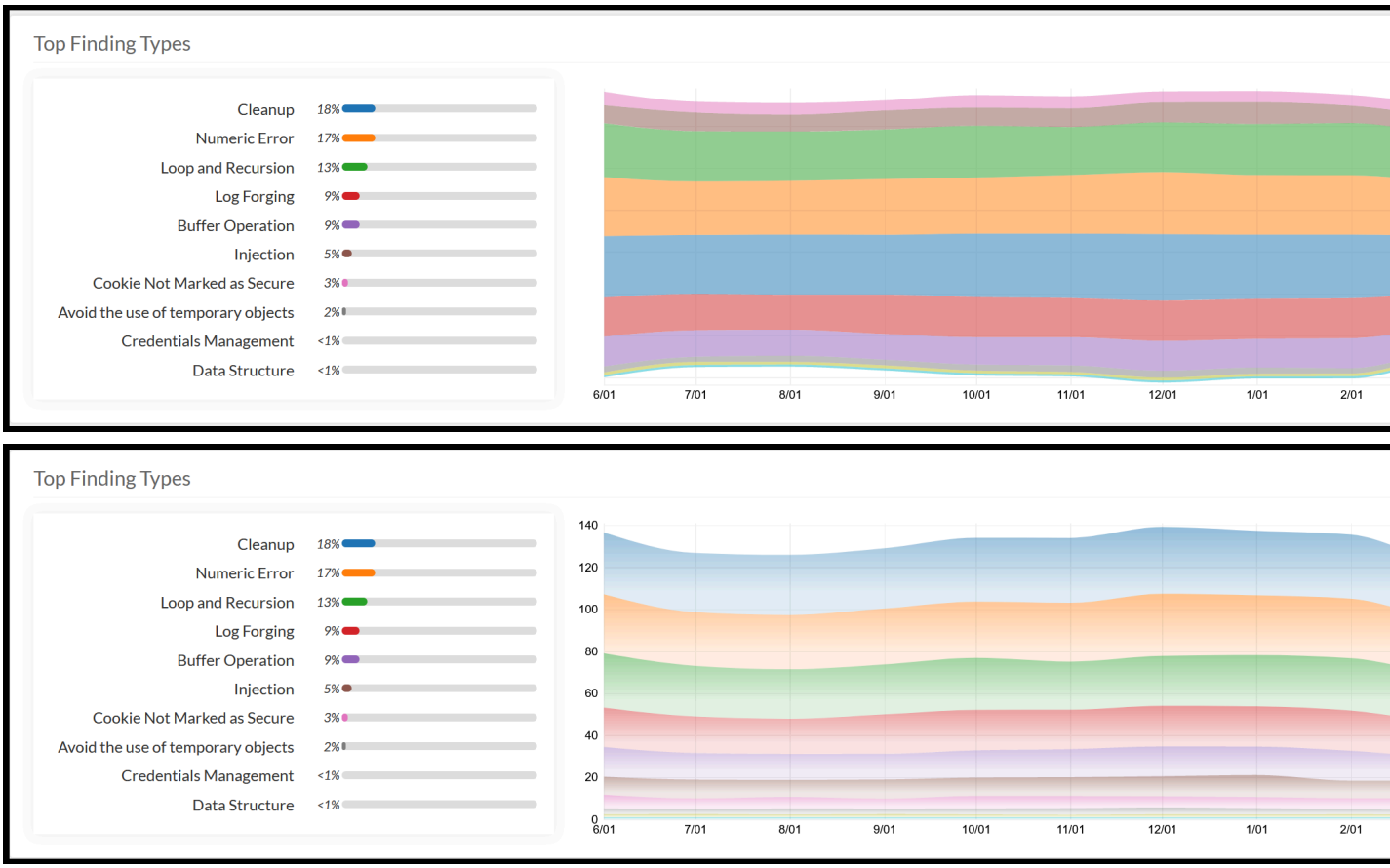


On the top-right of the graph is a calendar icon, which can be clicked to being up a menu for selecting a date range.



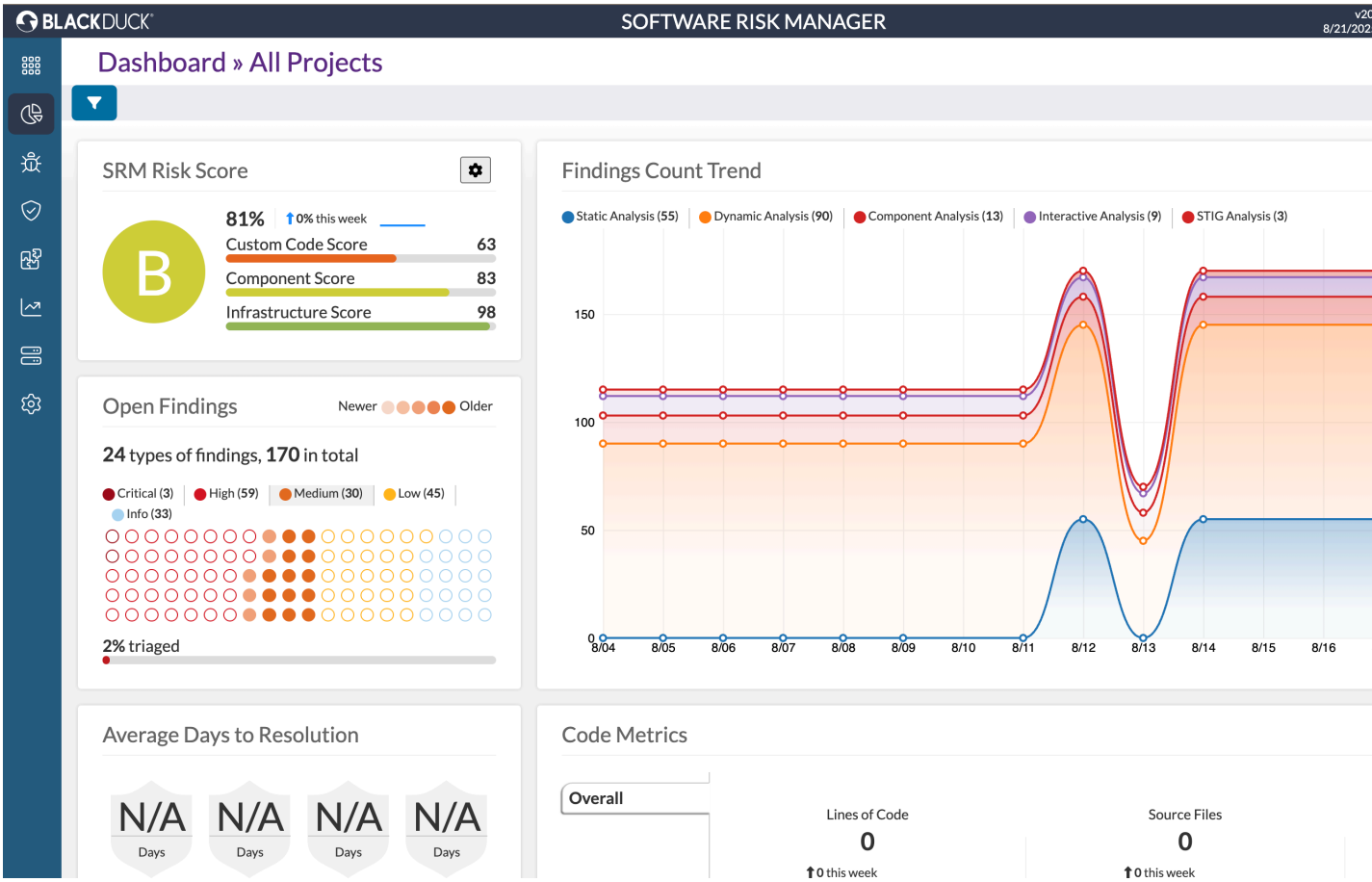
Selecting one of these range values will automatically refresh the graph to the selected range.

For larger date ranges, each point in the graph can represent multiple dates by taking the *average* finding counts of data samples involved.



Global Dashboard

The Global Dashboard provides a comprehensive overview of all projects in Software Risk Manager, displaying a set of aggregated analytic and trend data which are automatically updated as you use the system.



You can access the Global Dashboard from the main navigation menu. The dashboard offers a centralized view of your organization's software risk landscape, with powerful filtering capabilities to help you focus on specific subsets of projects.

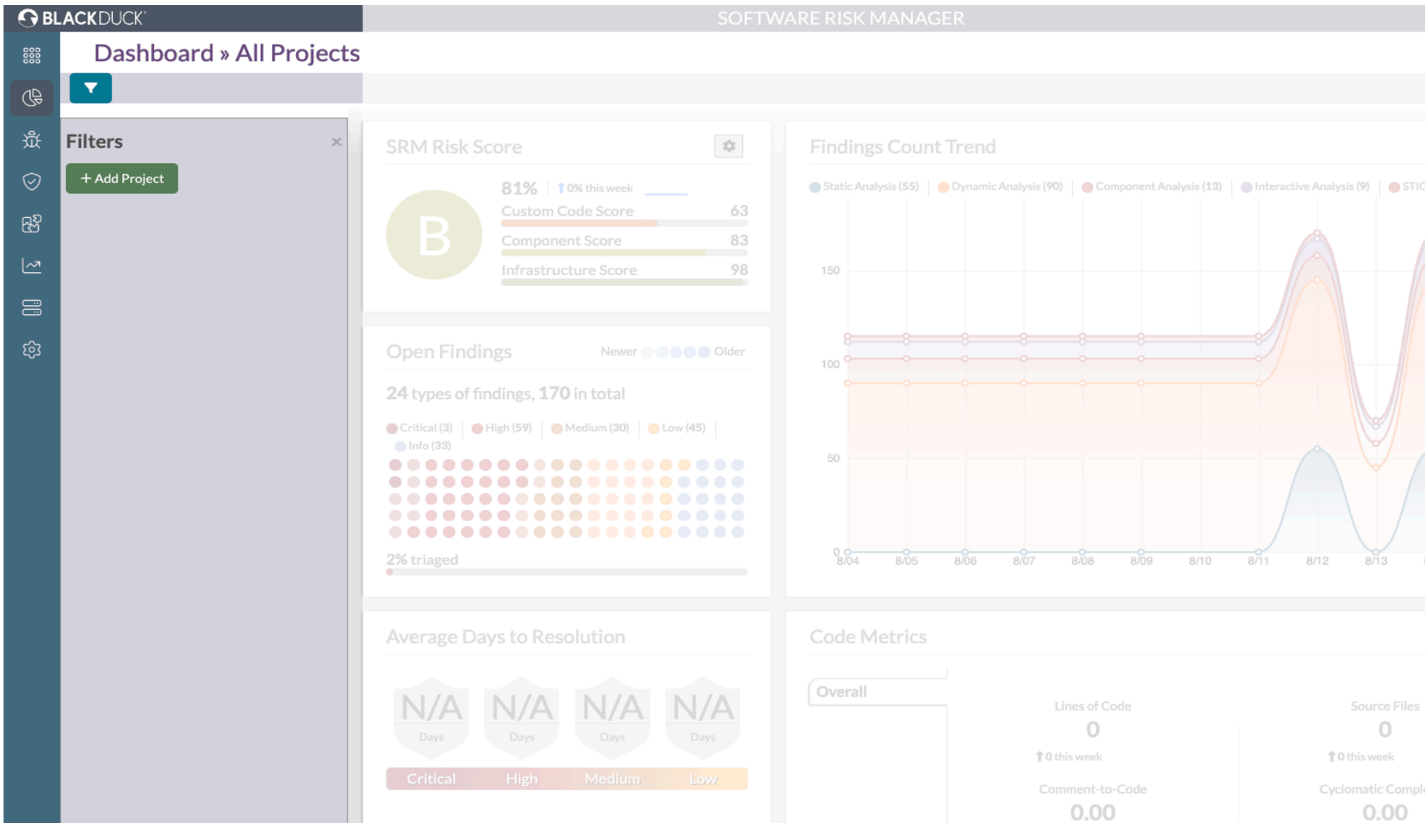
When viewing the Global Dashboard, you will have the option to include data from multiple projects and branches by using the filtering feature. To do this, use the filter panel on the side of the dashboard.

Dashboard Filter

The Global Dashboard includes a powerful filtering mechanism to help you focus on specific projects with their branches.

Filter Toggle

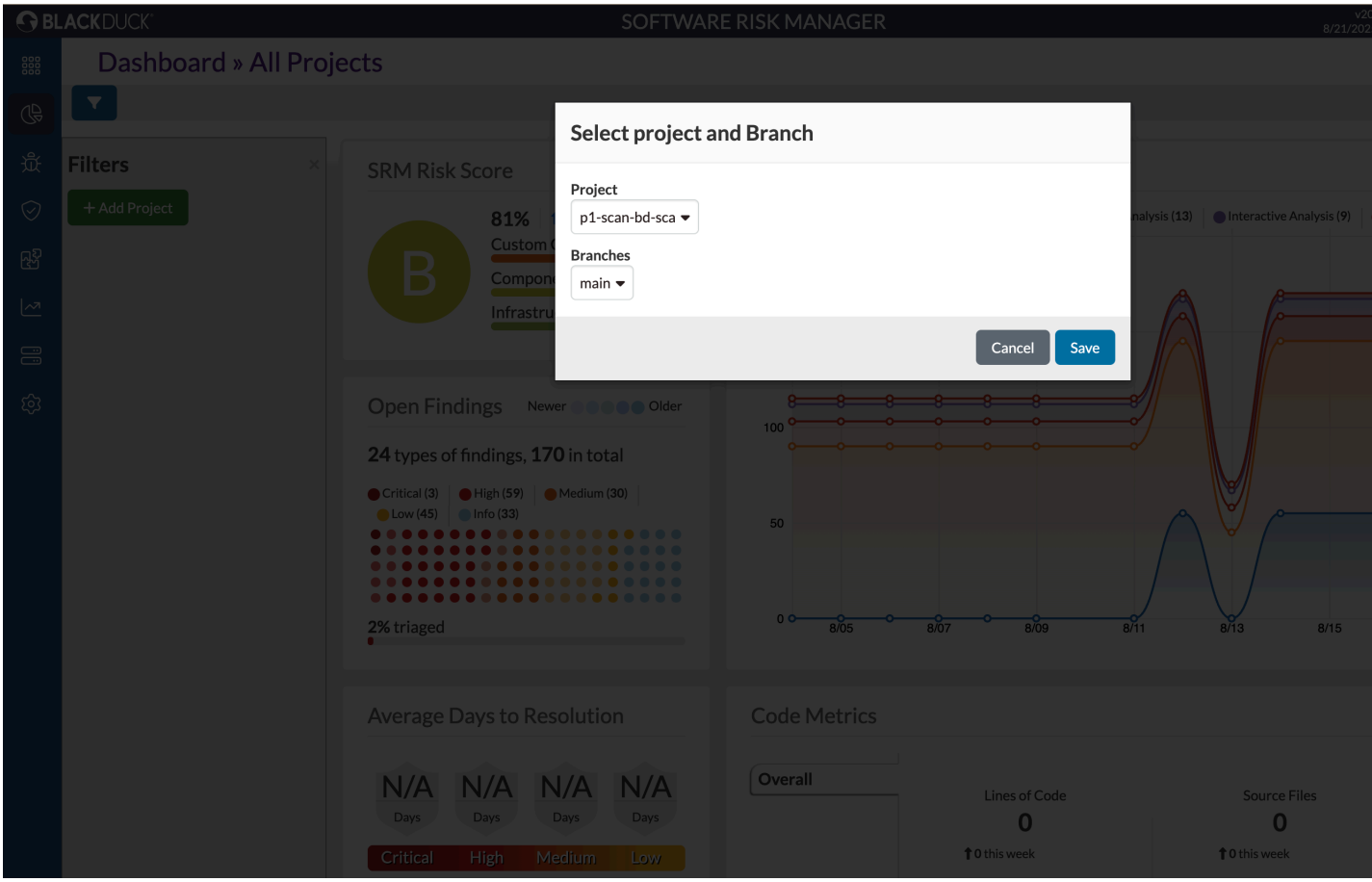
A toggle filter button is located on the left side of the dashboard. When clicked, it opens a filter panel that allows you to customize your dashboard view.



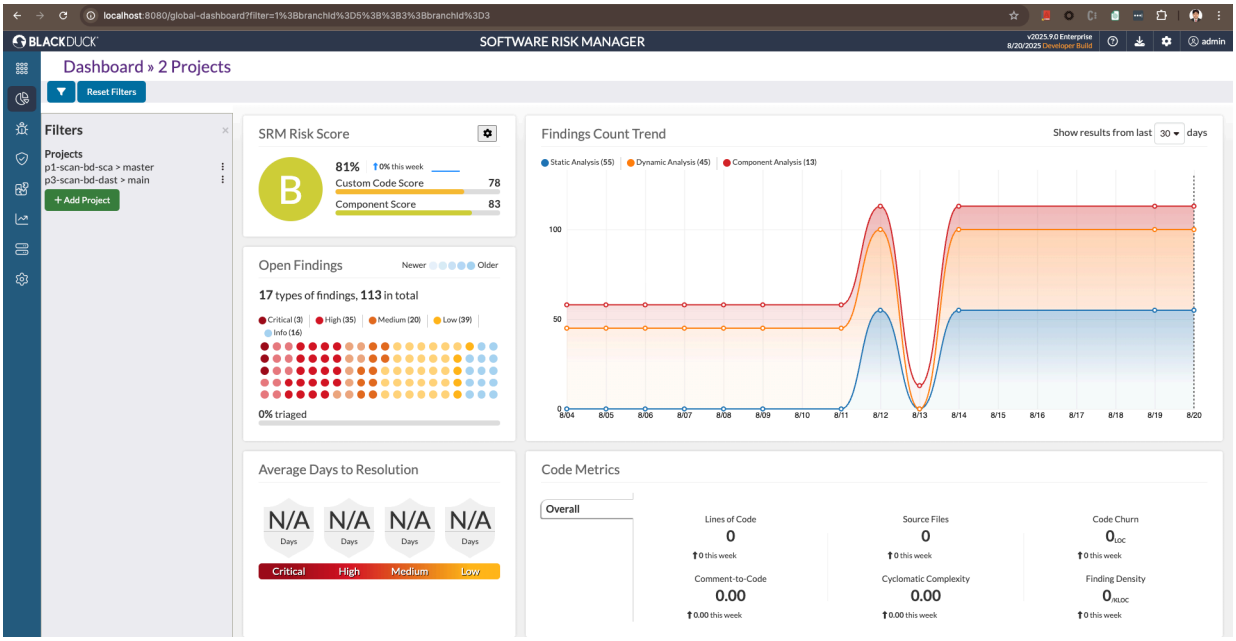
Adding and Removing Filters

In the filter panel, you can click on Add Project button. This will allow you to select projects and branches using dropdown options. The dashboard will be updated to reflect your selections based on

- Projects
- Project branches



You can select multiple projects and branches and save it to create a highly specific view of your software risk landscape. The selected filters will be shown in the filter panel.



- To close the filter sidebar, click the X icon next to the filter name in the filter panel.

- To add more filters, click the Add Project button again and select additional projects or branches.
- The filter panel allows you to easily manage your selected filters, including removing any that are no longer needed.
- The filter panel also renders the filters provided in the URL bar directly and dashboard gets filtered based on that (please look at the URL shown).
- A **"Reset Filters"** button is located on the right side of the Toggle Filter Button. When clicked, It will clear all the filter at a time
- As you add or remove filters, the dashboard visualizations will dynamically update to reflect the selected subset of projects.

The Global Dashboard contents are similar to the Project Dashboard and can be found here: Project Dashboard.

Hybrid Correlation

Hybrid Correlation, at its core, enables results from DAST tools to be correlated with results from SAST tools. This gives better visibility into how findings may actually be exploited in the wild as well as help identify test cases for those findings. Software Risk Manager currently has support for one form of Hybrid Correlation, which can infer possible execution patterns.

For more information, see the following section:

- Agentless Correlation

Agentless Correlation

Agentless correlation uses a static analysis approach on source code and binaries to correlate SAST and DAST results. No configuration steps are required to make use of agentless correlation. The only requirement is uploading source code at some point in an analysis.

Correlation Performance Impact

Agentless Correlation greatly expands the set of possibilities that must be considered to create a hybrid finding. Since exact code paths aren't provided, many inferred paths are created and evaluated during correlation. This can greatly impact the speed of correlation during analysis.

Requirements and Known Limitations

For information on requirements and known limitations, see the following topics:

- Requirements.
- Known Limitations.

Agentless Correlation Requirements

If Hybrid Correlation is enabled, Agentless Correlation is automatically applied for any project with correlation enabled and with uploaded source code.

Source Code

Agentless Correlation relies on the availability of source code to detect endpoints and their locations within a codebase. From this alone, DAST and SAST results that occur at an endpoint handling function can be correlated.

Only source code declaring and implementing endpoints are required. Source code for dependencies and utility libraries are not necessary, unless they declare and implement endpoints.

Endpoint detection is supported for a specific set of languages and web frameworks. These are as follows:

- **Java**: JSPs, Servlets, Struts, Spring MVC
- **C#**: ASP.NET MVC, Web Forms
- **Ruby**: Rails
- **Python**: Django

Effectiveness of endpoint detection can vary depending on the use of plugins and unconventional endpoint routing methods within the source code.

Binaries

Binaries for your application can also be uploaded to improve Agentless Correlation. If binaries are available, a call graph can be generated and explored to find code paths to SASTs from a detected endpoint. All relevant binaries for your application—the compiled application and its dependencies—should be uploaded with debug symbols for the best results.

Hybrid Correlation through call graph analysis is supported for binaries on the following runtime environments:

- JVM (Java, etc.)
- CLR (C#, etc.)

Agentless Correlation Known Limitations

Agentless Correlation explores a set of possible execution paths from an endpoint to find correlations with a code location. These explored paths may be inactive or incomplete due to undetected endpoints, inheritance and strategy patterns, anonymous functions, or other features for a given language and web framework.

Rule Sets

The Rule Set Page is accessed via the *Rule Set Associations* section of a project's *Analysis Configuration* dialog. When you access the Rule Set page, you will be able to view and sometimes edit a set of rules that can be used to determine how different types of findings will correlate with each other.

Each Rule Set has Rules, and each Rule has Criteria and identifying information.

Rule Sets are, as the name implies, a set of rules. Each rule acts as a strategy for combining results from different tools and providing standard information for the finding. Within a rule, a set of criteria can be defined, forming the underlying logic for the rule. The identifying information for a rule can optionally include a Severity, CWE, and Description, which will be shared by Findings created from that rule. For example, a general "SQL Injection" rule may be created to capture specific results from multiple tools and provide a shared description, making it easier to locate and recognize standard vulnerabilities.

When result data is uploaded to a Software Risk Manager project, as long as that project's *Prevent Correlation* setting is not enabled, its associated rule set will be responsible for determining which types of results represent the same types of problems. In this case, rules will be applied during ingestion, when findings are created from tool results. If there are multiple tool results belonging to the same rule and they occur at the same location, they will all be associated with the same finding. Whether a tool result "belongs" to a rule is determined by that rule's criteria.

After Changing Rule Sets

Since a project's configured rule set determines the manner in which results are correlated, changing that configuration necessitates an update of the correlation. This happens when the configured rule set for a project is modified in any way, or the *Analysis Configuration* is changed to use a different rule set. When this happens, the Findings page will display a notification prompting users to do so.

Additional Information

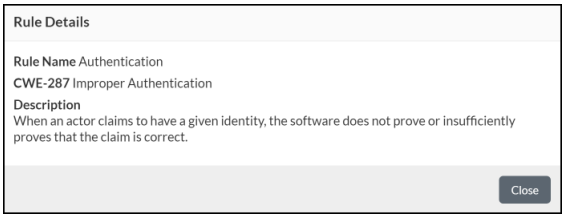
For more information on identifying information and rule criteria, see the following topics:

- Rule Identifying Information.
- Rule Criteria.

Rule Identifying Information

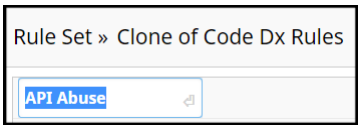
The identifying information for a rule includes severity, CWE, and a description. These fields are all optional. When provided, they will alter the corresponding values for findings associated with that rule.

Each rule's identifying information is collapsed by default. To expand it, click the dropdown configuration icon and select View Details.

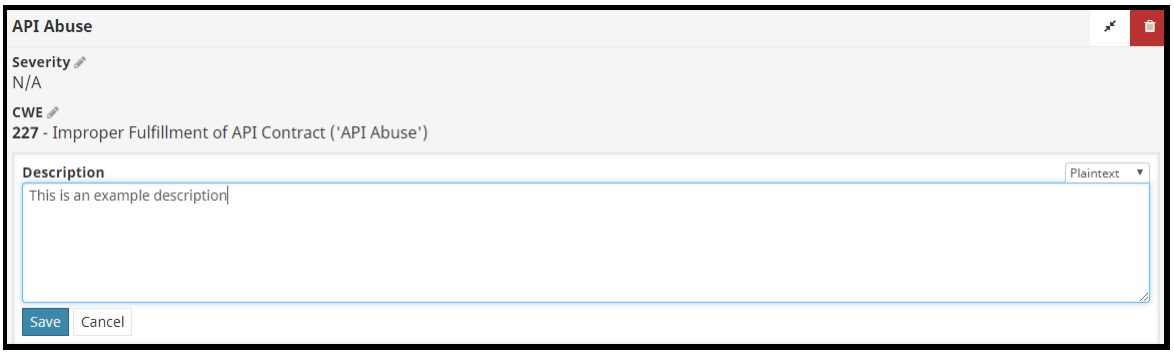


If you have the `admin` role, you can edit an existing rule's identifying information (aside from the read-only Software Risk Manager rule set).

To rename a rule, click on its name to open an edit window. Enter a new name then press Enter.



To change the severity, CWE, or description for a rule, expand the identifying information section, then click the pencil icon next to the corresponding header. This will activate an inline form allowing you to make changes to the value. Once you've set the desired value, click Save to apply the change. Click Cancel o discard your changes without saving.



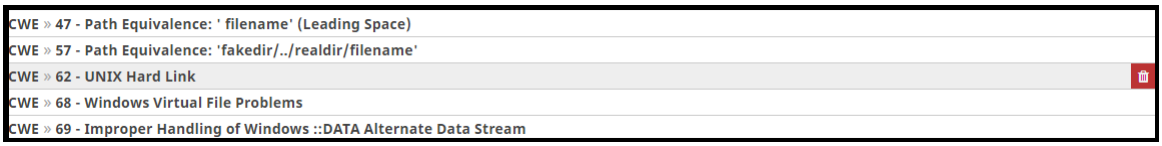
You can add criteria from editable rules via the forms at the bottom of each rule's criteria list.

Rule Criteria

A rule's criteria control which tool results will be matched with a rule. Note that each criterion can only appear once in a rule set. If you attempt to add a criterion that already exists in a different rule, you will be given the option to move the criterion out of that rule, or cancel. Users with the `admin` role can edit the criteria for each rule.

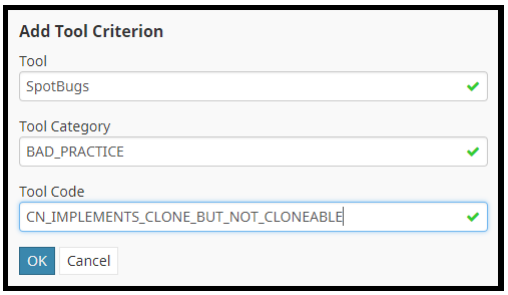
Criteria can be created for rules using the add criterion buttons for that rule. These buttons are located at the bottom of the criteria list.

Criteria can be deleted from rules using the delete button for that criterion. The button is hidden until you hover over the criterion in a rule's criteria list.

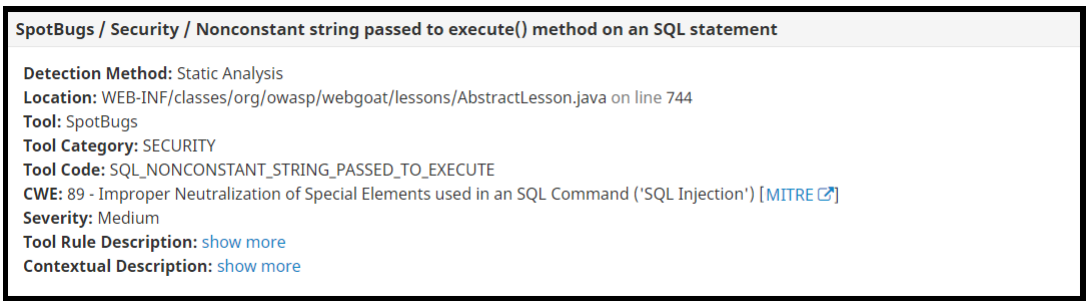


Tool Criteria

The *Add Tool Criterion* form allows you to create criteria that operate on a tool result's type. An individual tool criterion specifies a tool, category, and code. It will match tool results whose raw values match the values specified by the criterion.



The exact values for the tool criterion fields vary depending on what is reported by the tool. One way to discover these values is to look at the *Finding Details* page for existing findings in Software Risk Manager. The *Tool*, *Tool Category*, and *Tool Code* are displayed in the *Tool Details* for each associated tool result.



The category and code fields are both optional. Omitting both will create a criterion that matches all results from the specified tool. Omitting just the code will create a criterion that matches all results from the specified tool marked as part of the specified category. Some tools do not specify a tool category, in these cases the tool category field will need to be left blank.

Note: Leaving the tool category field blank does not act as a wildcard, so if the tool specifies categories, they must be included in all rule criteria.

CWE Criteria

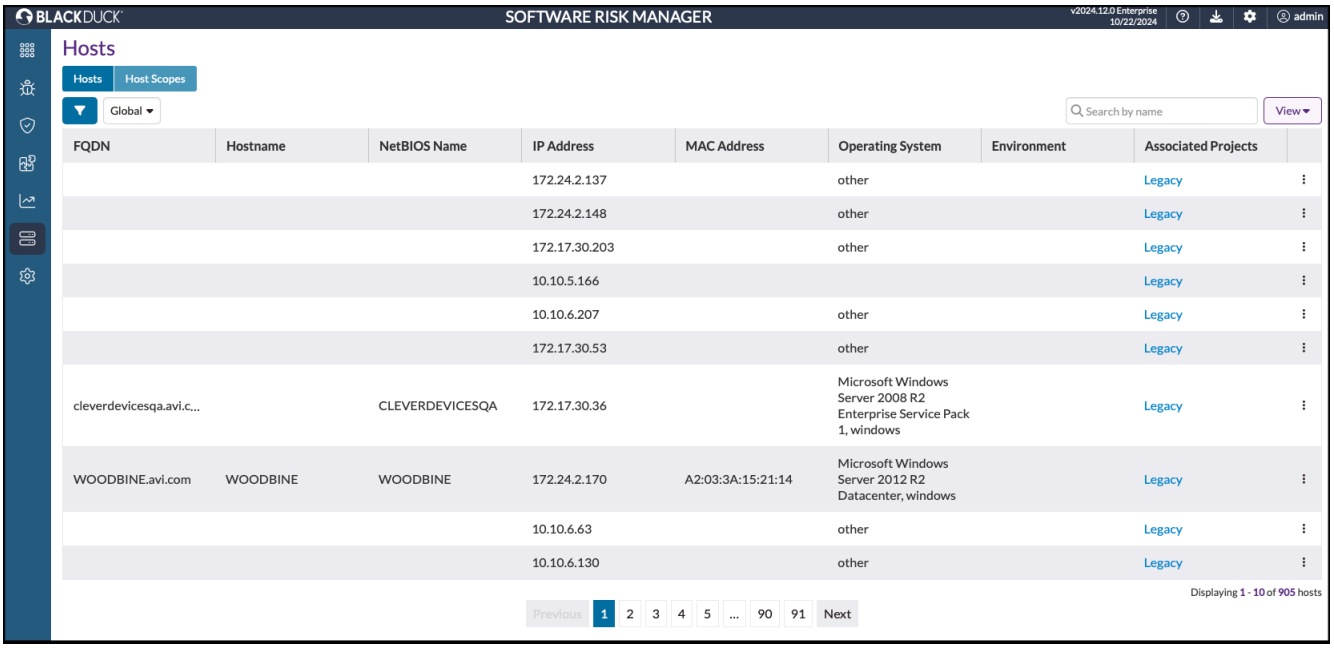
The *Add CWE Criterion* form allows you to create criteria that operate on a tool result's CWE. By specifying a CWE ID value, a CWE criterion will match tool results with that CWE value.

Hosts

Note: this section is only applicable to Software Risk Manager users with the InfraSec add-on.

When Software Risk Manager ingests *Network Security* results, the location of those results is typically expressed in terms of a "host," with the level of detail varying from tool to tool. The Hosts page is Software Risk Manager's location for interacting with host data directly, outside the context of Findings or Projects. Users will be able to access the Hosts page but the *Associated Projects* column will only populate for projects they belong to. Only the Software Risk Manager user with admin privileges will be able to create, edit, update, or delete host information.

Click the Hosts icon in the navigation bar to open the Host Scopes page.



This page shows a list of global host scopes with the following information:

- FQDN
- Hostname
- NetBIOS Name
- IP Address

- MAC Address
- Operating System
- Environment
- Associated Projects. Click the project name to open the project page.

Note: There's a basic filter field to sort hosts along with advanced filter options. Also, you can use the "View" button to select which columns to display (or hide).

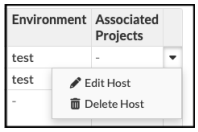
Tasks include the following:

- Creating a host
- Editing a host
- Deleting a host
- Managing host scopes. Click the Manage Host Scopes button to access the following options:
  - Importing host scopes
  - Exporting host scopes

Editing a Host Scope

To edit a host scope:

1. Click the Hosts icon in the navigation bar to open the Host Scopes page.
2. Click the dropdown list icon located in the last column of each row.



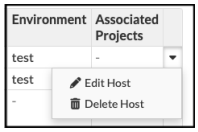
3. Select Edit Host.

4. Make changes as necessary.
   - Click the delete icon to delete an existing value.
   - Click the value button to add a field.
5. Click OK to save the changes.

Deleting a Host Scope

To delete a host scope:

1. Click the Hosts icon in the navigation bar to open the Host Scopes page.
2. Click the dropdown list icon located in the last column of each row.



3. Select Delete Host.
4. Click Delete to confirm.

Creating a Host Scope

To create a host:

1. Click the Hosts icon in the navigation bar to open the Host Scopes page.
2. Click Create Host.

3. Click the appropriate button to add values to the following fields.
   - FQDN
   - Hostname
   - NetBios Name
   - IP Address
   - MAC Address
   - Operating System
   - Environment
4. Click OK to save.

For more information

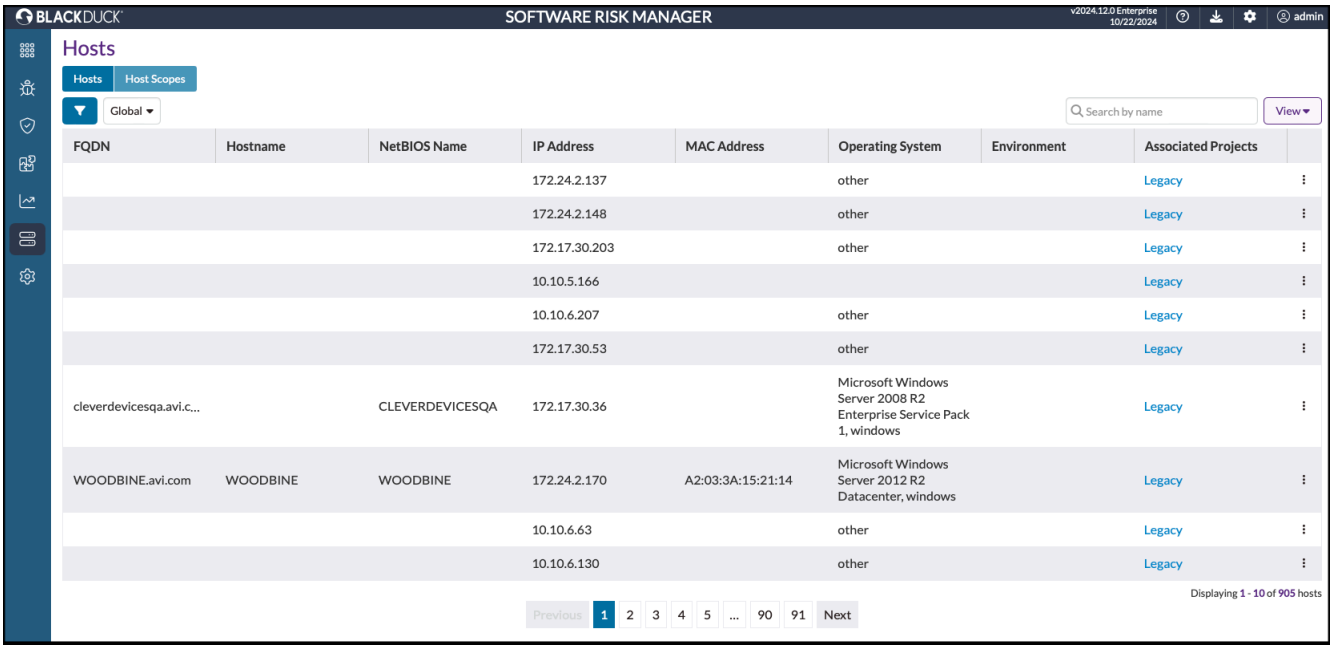
For information on host scope and the hosts table, see the following topics:

- Host Scopes
- Hosts Table

Host Scopes

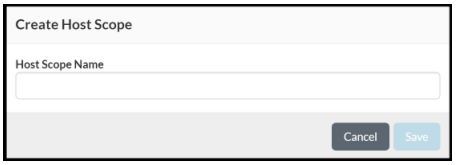
A Host Scope is effectively just a set of projects that share host information with each other. A Host Scope can be used to model a network where each project in a Host Scope contains vulnerability information for a vulnerable application that is housed on a potentially vulnerable host. Alternatively, one can simply use Host Scopes to isolate host information to particular sets of projects so that overlapping pieces of host information between Host Scopes don't interact with each other during Host Normalization and Finding Correlation. Each Host Scope can have multiple projects attached to them but each project can only be linked to one Host Scope. See Host Scope Associations for more information on how to set up projects with Host Scopes.

Click the Hosts icon in the navigation bar to open the Hosts page. Use the Hosts and Host Scopes buttons to display the associated data.



Managing Host Scopes

By default, the Global Host Scope will be the only one available. However, you can create new Host Scopes by clicking Add Host Scope.



Enter a name and click Save to create your Host Scope.

Once a host scope has been created, you can Import, Export, or Delete it.

Clicking Import will allow you to import a custom set of host information into the Host Scope for which you clicked Import. Software Risk Manager currently only supports importing hosts defined in a `.json` file. Hosts are expected to be provided as JSON Objects of the form: `field-type: [values...]` SRM currently supports the following field-types: Hostname, FQDN, NetBIOS Name, IP Address, MAC Address, Operating System, Ports. Every value for a field type is simply a string, except for Ports, which is a special case expecting each value to be another JSON Object with the following structure:

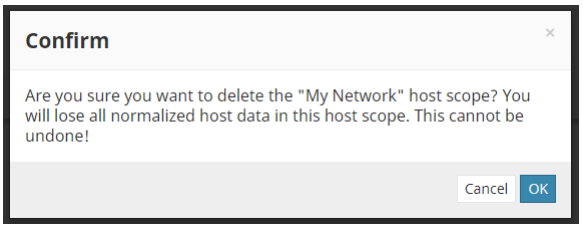
```
"Ports": {
 "Port": <port_number>
 "Protocol": <port_protocol>
 "State": <port_state>
```

```
}
```

Note: The Import button for non-selected Host Scopes will be disabled by default as you can only import hosts into the selected Host Scope.

Clicking *Export* will provide you a `.json` file containing all the normalized hosts in the Host Scope for which you clicked Export. The structure of the `.json` file matches that of the structure required for importing hosts into a Host Scope.

Clicking Delete will bring up a window allowing you to confirm that you would like to delete the relevant Host Scope. Deleting a Host Scope will delete all normalized host information belonging to that Host Scope. To delete a Host Scope, you will first need to delete any projects associated with that Host Scope.



You can confirm that you would like to delete the relevant Host Scope by clicking OK.

Hosts Table

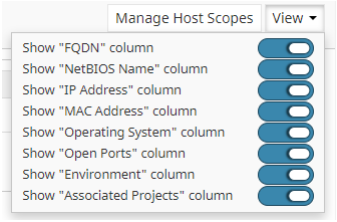
Selecting a Host Scope in the Host Scope management menu will populate the Hosts Table with normalized host information from the selected Host Scope. Normalized hosts are sets of hosts reported by different tools that are correlated to each other. Thus the information that the table is populated with is an aggregation of the host information from various tools that are referencing the same host.

Viewing Host Information

Each row in the table contains all of the host information that Software Risk Manager is aware of for a particular host, and each column in the table is a set of values appropriate for a particular field of interest. Currently, Software Risk Manager only displays FQDN, NetBIOS Name, IP Address, MAC Address, Operating System, Open Ports, Environment, and Associated Projects.

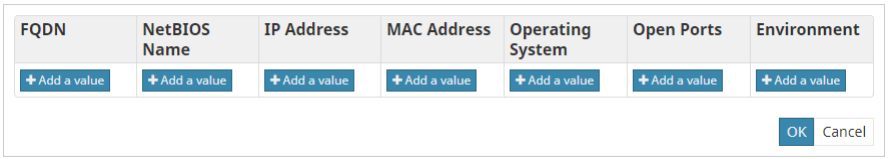
FQDN	NetBIOS Name	IP Address	MAC Address	Operating System	Open Ports	Environment	Associated Projects	
www.example5291.com	example-5091	51.15.1.164	7F:75:5B:68:64:66	Ubuntu	1127 (TCP)	Development	-	
www.example8024.com	example-1948	33.15.6.192	5B:A1:67:42:3E:53	Windows 7	4107 (TCP)	QA	-	
www.example6458.com	example-1982	165.25.0.124	9A:8A:E3:D8:6B:8A	Windows 8	1575 (IMAP)	Test	-	
www.example4930.com	example-5761	30.93.6.224	91:3D:C0:C4:AB:C6	macOS	6298 (IMAP)	QA	-	
www.example4391.com	example-9905	222.44.6.122	92:35:C6:6D:A3:AF	Ubuntu	411 (TCP)	Production	-	
www.example6922.com	example-6383	174.93.1.190	AD:94:E6:13:A8:F3	Windows 8	5636 (IMAP)	Test	-	
www.example3321.com	example-132	180.18.8.139	8E:F5:2A:AC:29:5D	Windows 7	5720 (HTTP)	Development	-	
www.example6626.com	example-4938	67.12.6.190	A9:A0:D1:7E:D8:E4	Ubuntu	7562 (IMAP)	Test	-	
www.example4959.com	example-1684	81.28.5.235	2E:25:B3:CF:77:71	Ubuntu	4396 (HTTPS)	Production	-	
www.example4830.com	example-6405	32.77.8.199	0F:2E:FF:07:98:92	Windows 10	3130 (HTTPS)	Production	-	
www.example8916.com	example-9954	187.16.9.194	D1:25:0F:9E:F9:33	Windows 7	163 (HTTP)	QA	-	
www.example388.com	example-7521	1.32.1.124	0A:D8:90:AD:35:B4	Windows 7	6937 (UDP)	Test	-	
www.example3181.com	example-470	27.23.6.164	33:07:34:70:58:EC	Windows 10	2109 (HTTP)	QA	-	
www.example7671.com	example-2585	144.91.8.173	5E:4B:14:33:49:03	Ubuntu	5646 (IMAP)	Development	-	
www.example8508.com	example-8751	197.86.3.239	5C:96:7D:6F:78:68	Windows 7	791 (HTTPS)	QA	-	

Click View in the top right corner and use the toggles to select what information to display.

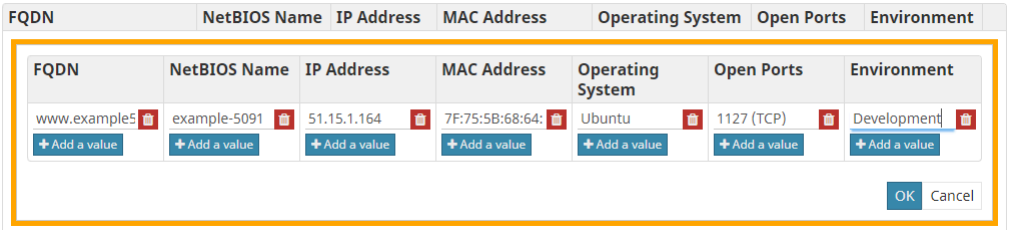


Manually Adding and Editing Hosts

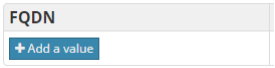
Software Risk Manager allows you to manually add new hosts to the selected Host Scope. The *Create Host* button is to the top right of the Hosts Table.



Clicking on it will bring up an interactive table where you can define the host that you are adding.

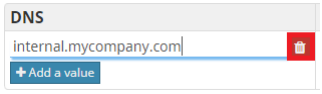


Clicking *Add Value* will produce a text field where you can enter a new value for the column you're editing. Note that a default value will be shown if the text field is empty and serves as an example of a valid value.



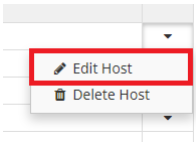
There is some validation applied to IP Address, MAC Address, and Open Ports. Typing in an invalid value for that field will cause the text field to be highlighted in red. Each invalid value is considered non-existent when clicking *OK* to create a Host with the specified values, and consequently will not appear in the Hosts Table when a host with invalid values is successfully created.

If you do not wish to include a value on a host, you may click the trash bin located at the far right of the cell to delete the value from consideration.

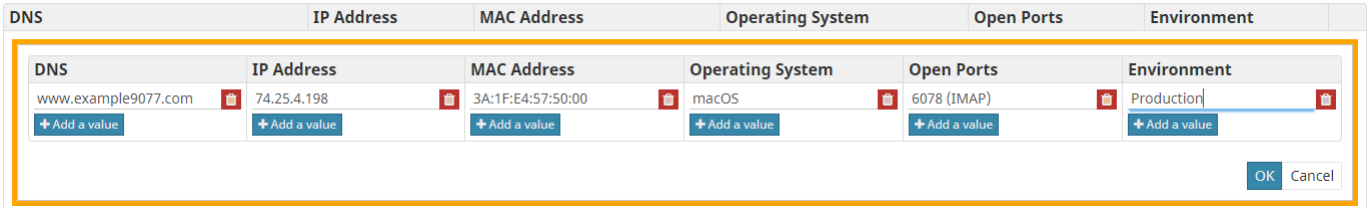


You can include any number of values for any particular column in the editor. Note that Associated Projects is not present in the editor. This is because Associated Projects is a derived field, found by determining if a host exists in a particular project. Also note that any columns that are missing in the Hosts Table as a consequence of disabling them in the "View" menu will still be shown while editing the hosts.

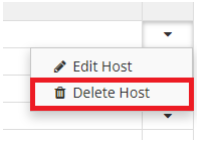
Software Risk Manager also allows you to manually edit existing normalized hosts in the Hosts Table. If you click on the button in the right most column of the Hosts Table, a drop down menu will appear. The first element in the drop down menu is "Edit Host".



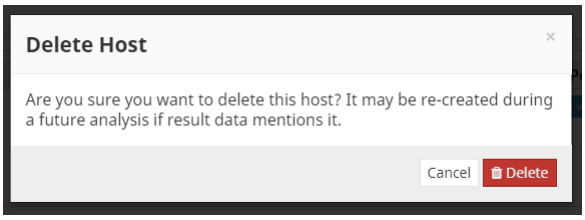
Clicking on "Edit Host" will bring up the same interactive table that appears when you're manually adding a new host, except now you will see it in the table, as opposed to above it.



Software Risk Manager also allows you to delete existing normalized hosts. In the same drop down menu that "Edit Host" appears, you will also see "Delete Host".

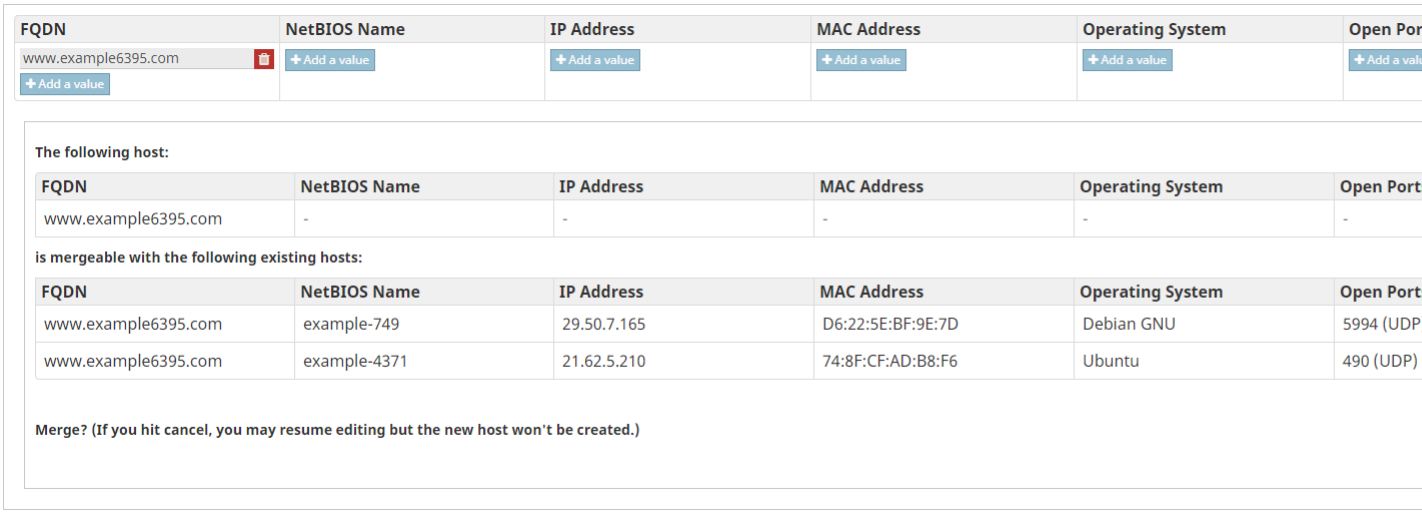


Clicking on it will prompt you with a message detailing the consequences of deleting a host.



You may confirm the delete by clicking the "Delete" button. Clicking *Cancel* will bring you back to the Hosts Table without deleting the host. Note that only the normalized host is deleted when clicking *Delete*. No host information acquired from results during an analysis will be lost.

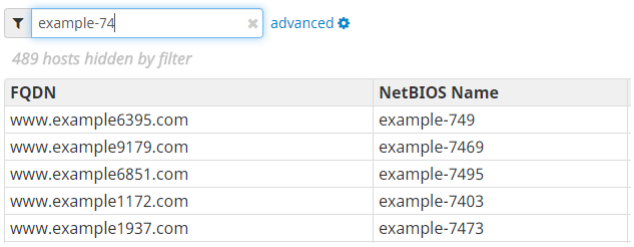
When creating or editing a host, you may end up introducing values for field-types that Software Risk Manager considers "identifying". Identifying fields for Hosts are the FQDN, Hostname, NetBIOS Name, IP Address, and MAC Address fields. If you introduce a value for an "identifying" field-type and it already exists on a host in the current Host Scope, clicking on "OK" will cause the editor to expand to include two new non-interactive tables.



The first new table will show you the host you tried to add, and the second new table will show you all hosts that already have some of the values for "identifying" field-types the host you tried to add has. Clicking *Merge* will cause the host you tried to add to be joined together with the other hosts that shared values for "identifying" field-types with that host. Clicking *Cancel* will bring you back to the editor and will not join the host with any other hosts. Note that you will be unable to edit the host you're trying to add until you either click *Merge* or *Cancel*.

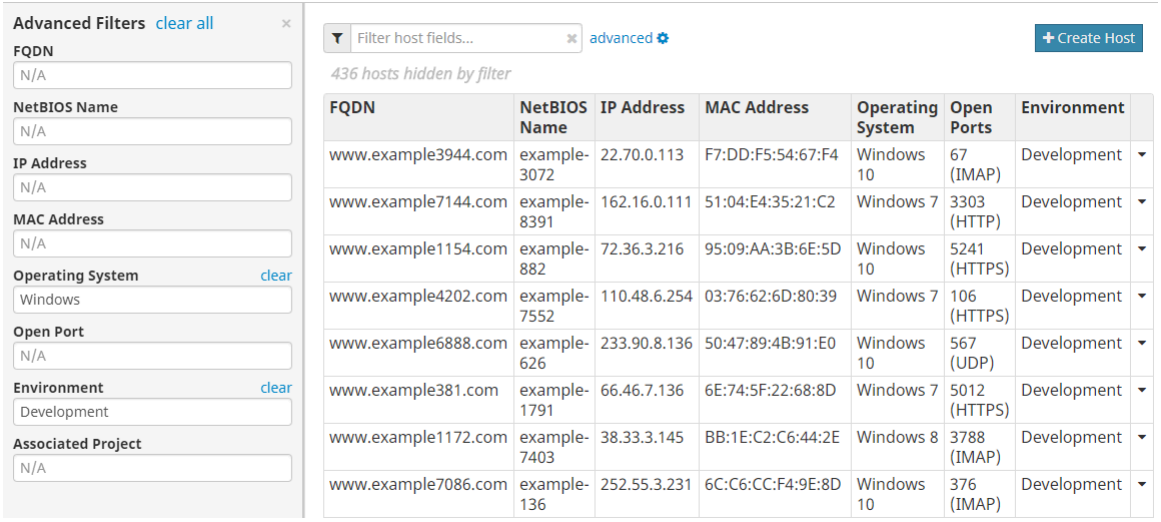
Filtering Hosts

Software Risk Manager also allows you to filter the hosts that appear in the Hosts Table. Above the table, you should see a text field.



This is the "Generic" filter. Any value provided here will be used to display only hosts for which at least one field value for any of the field-types satisfies the filter defined by the provided value.

Clicking *advanced*, which is next to the "Generic" Filter, will bring up the Advanced Filters sidebar.

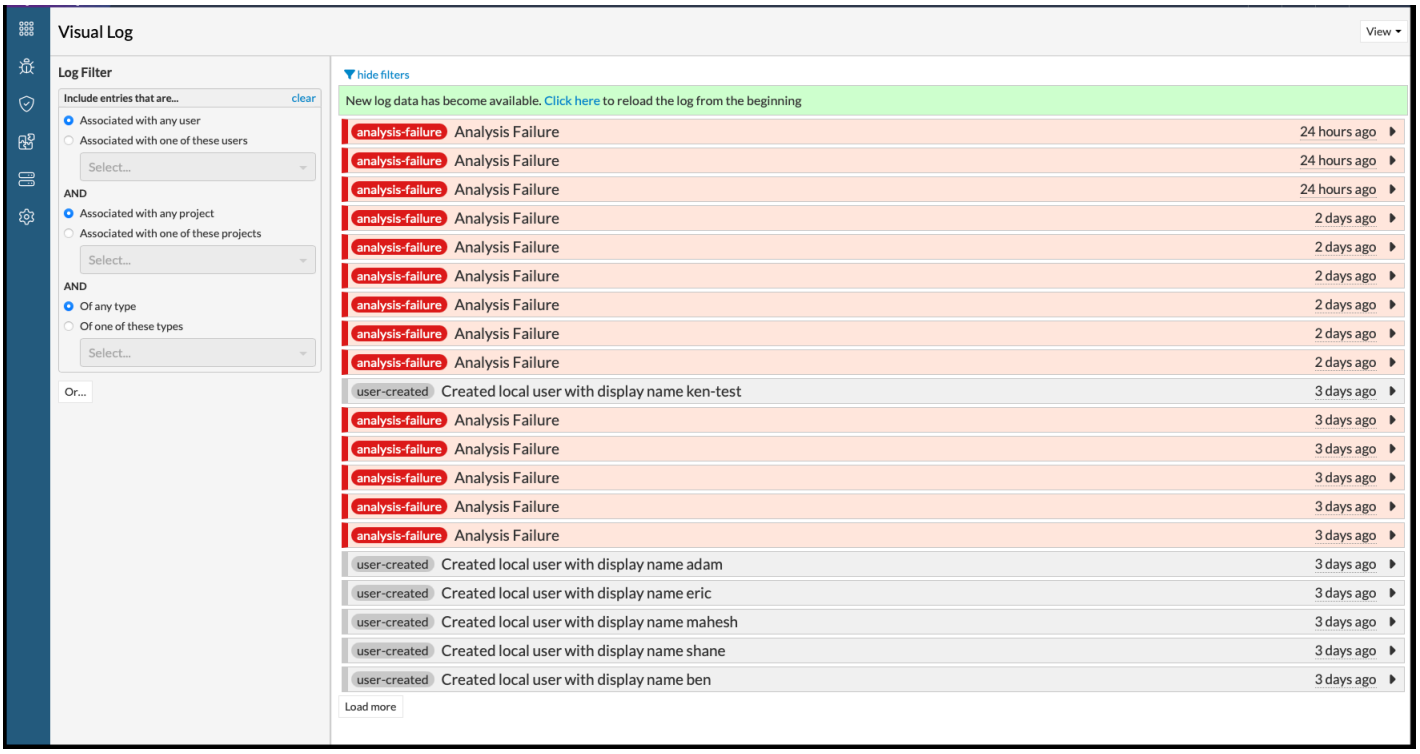


These filters are specific to a host field-type, and providing a value for any of these will display only hosts in the table for which the specified value exists in the specified column for that host. Note that each filter available will only filter by one value, so attempting to filter by multiple values for a particular field-type (or in the "Generic" filter) won't work.

Visual Log

The Visual Log page provides a helpful UI for certain events and errors that administrators might be interested in for auditing purposes. It is important to note that the log file generated by a running Software Risk Manager installation is not the same as the visual log. Most notably, arbitrary exceptions that appear in the log file will typically not appear on the Visual Log page.

Click the "gear" icon in the header bar and select Visual Log to open the Visual Log page.

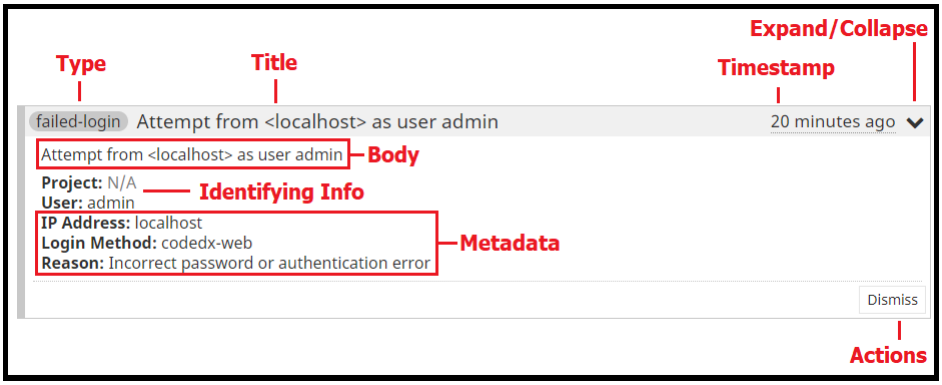


For information on log messages and filters, see the following topics:

- Visual Log Messages
- Visual Log Filter

Visual Log Messages

An entry in the visual log contains several useful parts:



- **Type** is an identifier for the type of event that the entry describes. The entry's type can be used as a filter criteria (explained below).
- **Title** is a brief description of the event.
- **Timestamp** indicates when the event happened. The default display mode for this is " ago", but hovering the mouse over the timestamp will reveal the precise date and time of the event.
- **Expand/Collapse** can be clicked to expand or collapse the log entry. By default, all log entries are collapsed. You can actually click anywhere in the colored title area to expand or collapse the log entry.

- **Body** a longer-form description of the event. In some cases, it may be the same as the title.
- **Identifying Info** points out the Project and User associated with the event, if applicable. For example, a login attempt for a Software Risk Manager user would be associated with that user. An analysis failure event would be associated with the Project for which the analysis failed.
- **Metadata** is unstructured information related to the event. In the example shown above, the *IP Address*, *Login Method*, and *Reason* are all metadata specific to the `failed-login`event type. Other event types can (and typically will) have different metadata fields.
- **Actions** refer to the bar at the bottom of an expanded event, containing one or more buttons.
  - The *Dismiss* button will cause the entry to become "dismissed," which will hide the event from view by default when visiting the page later.
  - The *Retry* button will be available for some error-type events, allowing users to retry whatever process whose failure generated the log entry.

As noted in reference to the *Timestamp*, the visual log is ordered in reverse-chronological order, so that the newest events will be at the top. As you scroll through the log, you'll eventually encounter a Load More button that will load the next portion of the log. You can keep scrolling and clicking Load More until you reach the end of the log (the earliest event). While you are on the Visual Log page, if new events happen, rather than interrupting your view of the log by immediately appearing and altering the UI, a notification will appear at the top of the page, prompting you to reload the log.

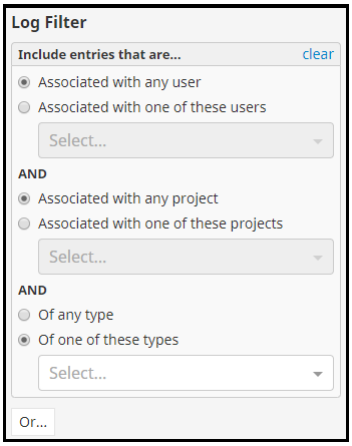
Dismissed Entries

Clicking Dismiss on a visual log entry will "dismiss" it, sending it into a semi-ignored state.

Once a reload is triggered (e.g., by refreshing the page, clicking the "click to reload" prompt, changing filter states, or changing view menu settings), dismissed entries will be hidden from view (assuming the *Include dismissed log entries* setting in the *View* menu is switched off). The *Include dismissed log entries* setting in the *View* menu can be toggled to show or hide dismissed log entries, causing them to appear with the checkered background style.

Visual Log Filters

The *Visual Log Page* offers a filtering capability that allows users to easily select a subset of the log.

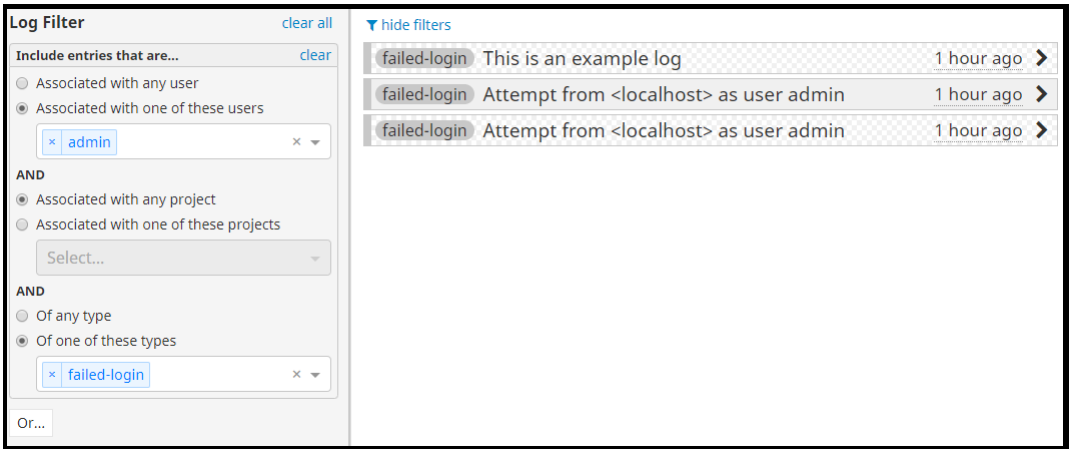


By default, the *Log Filter* section will contain a single blank filter block. Each filter block contains three optional criteria:

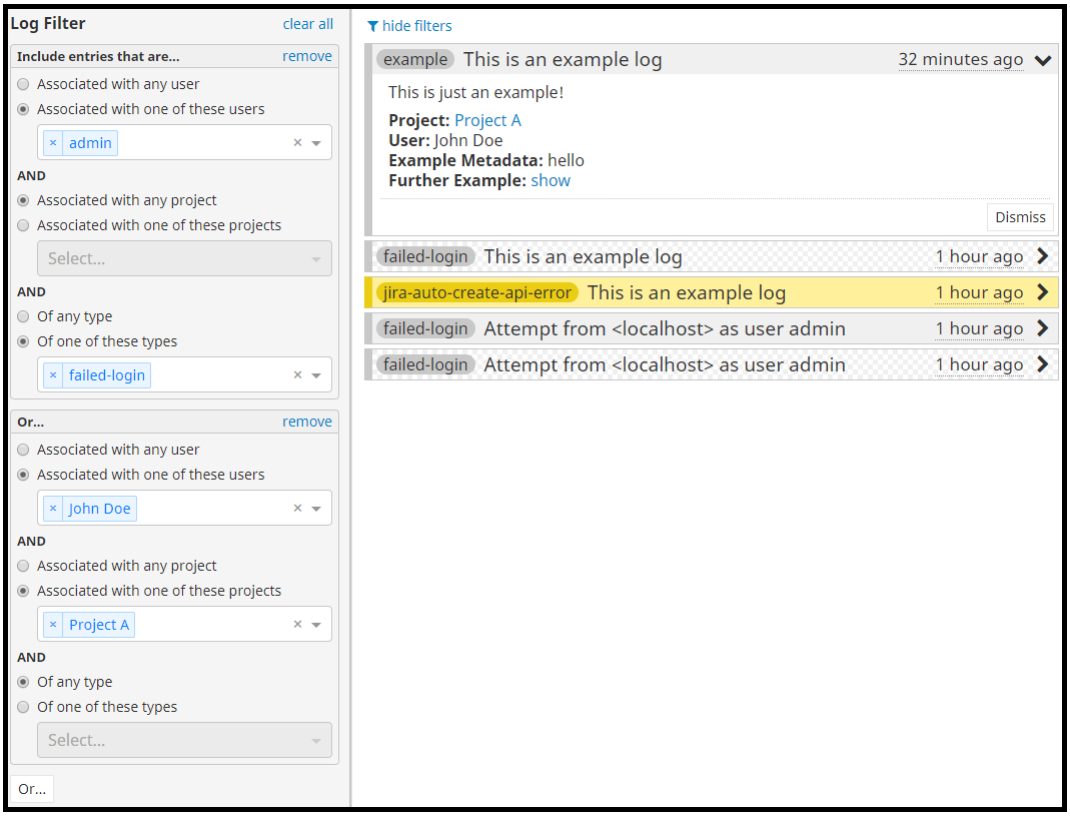
- View only those log entries whose *User* is one of the users selected in the filter block.

- View only those log entries whose *Project* is one of the projects selected in the filter block.
- View only those log entries whose *Type* is one of the types selected in the filter block.

For example, you could set a filter in order to only view `failed-login` events where someone attempted to log in as the "admin" user.



Clicking the *Or...* button below the filter will add an extra filter block, allowing you to set alternate criteria. In the example below, the filter will select `failed-login` events related to the "admin" user, **OR** any event related to "Project A" and the "John Doe" user.



Note: Although all log types will be available for selection in the filter, those types may not always be present in the log. For example, non-admin users will only be allowed to view log events that are directly related to a project they manage, so they inherently won't be able to see `failed-login` events, for example, because those events are never associated with projects. Also, the `successful-login` event is not recorded by default. See the Visual Log Configuration section in the install guide to enable recording of that event type.

Webhooks

Webhooks will allow SRM to issue POST requests to an external resource when finding or triage statuses are updated. Currently, webhooks are an API-only feature. Please see the Webhooks section of the API guide for more detailed information on how to configure webhooks.

Once a webhook has been configured, anytime the triage or finding status has been updated for one or more findings that belong to a project that matches a webhook configuration, a payload will be generated and a POST request will be made. The payload is a json object with the following properties:

- `trigger` – The event that triggered this payload to be sent.
- `reasons` – A list of reason objects detailing why the event was triggered. These objects may have the following properties:
  - `reason` – A short text describing the type of reason, can be one of the following: "analysis", "re-correlation", "archival", "system", "user-action", or "jira-sync".
  - `user` – An object detailing the user that triggered this event. Contains the ID and name of the user. Only available on "analysis", "re-correlation", "archival", and "user-action" reasons.
  - `analysisId` – The ID of an analysis. Only available on "analysis" reasons.
  - `input` – The name of an analysis input. Only available on "archival" reasons.
  - `action` – A short text description providing context to the reason. Only available on "jira-sync", "user-action", and "system" reasons.
- `findings` – A list of finding objects. See the Finding Table Data endpoint in the API guide to view the structure of the finding objects.

If a webhook is configured to use a secret, the requests made by SRM will contain the `X-Signature` header. The value of this header is generated by hashing the request json body using an HMAC hex digest.