

Muting Defects

This guide shows you how to mute defects in Kiuwan.

Contents:

- [Introduction to defect muting](#)
- [The basics of defect muting](#)
 - [Scope of Mute](#)
 - [What to consider when muting at defect-specific level](#)
 - [Muting defects in Kiuwan Life Cycle \(baseline and deliveries\)](#)
 - [Review status of defects](#)
- [How to mute defects in Kiuwan](#)
 - [Muting at the defects/vulnerabilities tab of a baseline analysis](#)
 - [Muting at the mute defects sub-menu in the defects tab](#)
 - [Muting at the defects submenu of a delivery analysis](#)

Introduction to defect muting

While looking at analysis results, you could find that (for example) a Kiuwan rule is generally helpful and must be kept active, but, in some concrete cases, it is not applicable (or it is not properly working) and some defects should not be considered in the analysis.

If you want to keep the rule active but discard some specific defects. Kiuwan provides the Mute Defect functionality to do it.



Mute considerations

Muting defects is a feature that helps you to silence some defects.

The reasons to mute a defect can be of different natures, and one reason can be to silence **False Positives**, i.e. defects that are not really defects.

We strongly recommend you to report False Positives to Kiuwan Technical Support, and while we fix them, you can mute them.

Also, please consider the reasons and the functionalities that Kiuwan provides to manage defects that you don't consider as defects. Have a look at [How to manage Kiuwan defects when I do not completely agree with them](#)

The basics of defect muting

Scope of Mute

Defect muting can be applied to different **scopes**:

1. **Defect-specific**
 - a. By Line Number:
 - i. A specific defect (identified by the rule and the line number of a source file) is muted
 - ii. This kind of mute means that the defect will be kept muted in subsequent analyses if, and only if, the defect appears in the same line
 - b. By Source Code: (NEW)
 - i. A specific defect (identified by the rule, the file, and the source code line) is muted
 - ii. This kind of mute means that the mute is based in the content of the source line number and, therefore, it will be kept muted in subsequent analyses regardless of the line number where it appears
2. **File-scope**
 - a. To mute all the defects of a certain file, regardless of nature (rule) of the defect
3. **Rule-scope**
 - a. To mute all the defects coming from a specific rule, regardless of the file where the defects are appearing
4. **Rule-File or File-Rule** scope
 - a. Rule-File: To mute all the defects of a certain rule belonging to a specific file (or to a set of files)
 - b. File-Rule: To mute all the defects of a certain file coming from a specific rule

Kiuwan allows you to declare mute patterns for all the above situations, letting you tailor the Kiuwan muting mechanism to your specific needs.

What is important to remember is that **muted defects will not be considered when passing an Audit or calculating an Indicator**.

Muted defects are still there (you can inspect them) but will not be part of the calculations made by Kiuwan.

You are probably asking yourself:

1. Is muting a rule the same as deactivating that rule?
 - a. Yes, **muting a rule will mute all the current defects of that rule as well as future defects of that rule in further analyses**. This way, you don't need to deactivate the rule (that would imply to deactivate the rule for all the applications that use that model). Also, defects of that rule still exist (but muted), but will not be considered in Audits or in the Indicators. You can later un-mute again at a later stage and the rule will be considered as "live" again.
2. Is muting a file the same as excluding that file from the analysis?
 - a. Yes, the final effect is the same. **Muting a file will mute all the current defects of that file as well as future defects for that file**. As above, those defects will remain in the analysis but muted, not being considered in Audits and Indicators.

What to consider when muting at defect-specific level

When you select a defect to mute, you can decide whether to mute by line number or by source code.

EDIT EXISTING MUTE PATTERN

DEFINITION

Rule Do not allow external input to control resource identifiers

File WebGoat-6.0.0.RELEASE/webgoat-container/src/main/resources/static/js/jquery_form/jquery_form.js

Mute type ☒ By line number
Line 245
☐ By source code
Sink- \$.getOptions.closeKeepAlive, function() { jsahr = fileUploadFrame(a); }

EXPLANATION

Why

Comments

DELETE APPLY

If you mute a defect by line number, bear in mind that modifying the line number where that defect appears (by adding/removing lines before the defect line) will make the defect appear again.

Instead, if you mute that defect by source code, you can freely add/remove lines before that defect, the defect will be silenced as long as the source line text does not change.



When you mute a defect by source code, there's a condition that you must bear in mind:

- If, for example, you get 3 defects in different lines but the source code line is equal in all those defects, if you mute one of them by source code, the side-effect is that all three will be muted as well ⚠

It is important to know this side-effect because the mute-engine cannot distinguish between them (the source code line is the same for all of them, and the line number is not considered)

Finally, when the defect is an injection vulnerability (i.e. a defect coming from an injection security rule), the defect is uniquely identified by three factors: the sink, the source, and the propagation path.

Then, if you select the source to mute, the mute window will show to you both the sink and source code lines.

In this case, if you mute **by line number**, the defect will be muted based on line numbers of sink and source code lines. As above, if line numbers of sink or source change, the mute will not be applied and the defect will rise again.

But, if you mute **by source code**, the mute applied to the source code of the sink, the source, and the propagation path. That means that although the sink and source code lines do not change, any change in the propagation path will be considered as a new defect and the mute will disappear.

Muting defects in Kiuwan Life Cycle (baseline and deliveries)

Kiuwan allows you to **mute defects at any moment of your application's life cycle**.

If you are using Kiuwan Life Cycle, most probably you will have application baselines (performed periodically at quite defined promotion to production stages) and deliveries (at nightly-build or quite often while continuous development).



In previous releases, Kiuwan only allowed you to mute defects in baseline analyses. Now, **you can also mute defects found during a delivery analysis**.

1. If you mute defects of a **baseline**, those defects will also be muted in further analyses (deliveries and baselines)
2. If you mute defects of **delivery**, all the further deliveries and baselines will also mute those defects.

IMPORTANT: you can only mute defects in a delivery executed over the last available baseline or in deliveries without related baseline analysis.

Once muted, those muted defects will be considered in further delivery and baseline analyses

Review status of defects

After an analysis, you will need to spend some time looking carefully at the defects found during the analysis, to fully understand them before considering submitting its correction to developers. During that review, some of them will be reviewed very fast but others may take a while.

Kiuwan can help you to mark the “Review Status” for any specific defect.

This way, as you review the defects you can mark them as **To review** or **Reviewed** (or leave them blank, of course) for review tracking purposes.

How to mute defects in Kiuwan

Kiuwan lets you manage to mute in several pages:

- At the Defects/Vulnerabilities tab of a baseline analysis
- At the Defects submenu of a Delivery analysis
- At the Mute Defects submenu in Defects tab

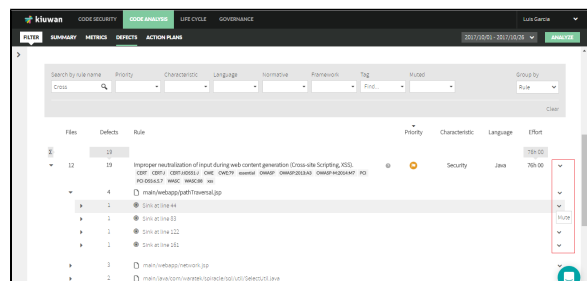
Let's go through them.

Muting at the defects/vulnerabilities tab of a baseline analysis

Once you select the last analysis of an application (either in Code Security or in code Analysis), go to Defects or Vulnerabilities tab.

For explanation purposes, in our explanation, we will refer to both as Defects tab. In case of any difference, we will note it.

We refer to the last analysis because you can only mute on the last analysis. The mute pattern applies to the current and further analyses .. (past analyses cannot be changed)



At the different scopes (rule, file, defect, etc), you can open the left menu and you can select the **Mute** option.

In our example, we will mute the 4 defects of the XSS rule on the selected JSP. So, clicking on the JSP will open the following dialog:

CREATE NEW MUTE PATTERN

DEFINITION

RuleImproper neutralization of input during web content generation (Cross-site Scripting, XSS).

Filemain/webapp/pathTraversal.jsp

EXPLANATION

WhyNone (default)

CommentsFalse positive
Too many defects
Generated code
Too complex code
Other

APPLY

When you mute something, you are creating a so-called **Mute Pattern**. Remember that a mute pattern can apply to a unique defect or to a set of defects, that's the reason for that nomenclature.

Besides descriptive data of the mute pattern (such as the involved rule, file, etc), you can add the reason (or **explanation**) that justifies the mute pattern.

You can select between common reasons to mute defects (it's a false positive, the defects are on generated code and cannot be changed, etc.), but you can also add your own.

Just in case you select to mute a rule with defects in more than one file, the dialog will be as in the figure

CREATE NEW MUTE PATTERN

DEFINITION

RuleImproper neutralization of input during web content generation (Cross-site Scripting, XSS).

Apply to files**myclass.java
**myjsp.jsp

Add file pattern

If no patterns are specified all defects reported by this rule will be muted

EXPLANATION

WhyNone (default)

Comments

APPLY

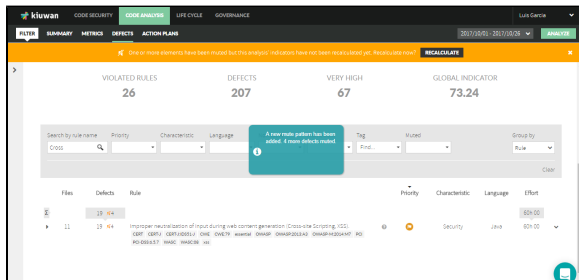
In this case, you will be able to specify as many **file patterns** as you want. In this case, the mute pattern will be applied to all the files that match any of the indicated patterns.

Remember that a file pattern must be indicated following **ANT pattern syntax**. For further help on ANT patterns syntax visit <https://ant.apache.org/manual/dirtasks.html>

After applied, muted defects will appear shadowed and with an icon.

You will also see a message (in yellow) indicating that there are muted defects but Indicators have not been recalculated yet.

If you need to add mode mute defect patterns ignore that message, otherwise click on **Recalculate** so the indicators are recalculated taking into account muted defects.

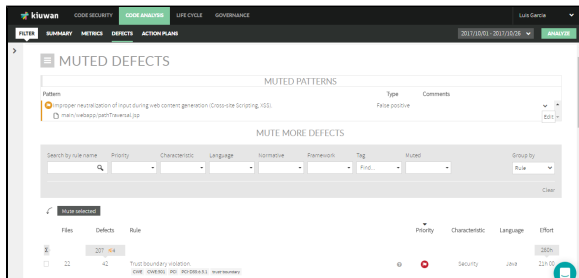


Muting at the mute defects sub-menu in the defects tab

As an alternative to the above page, you can also mute defects opening the Mute Defect sub-menu in the Defects tab.



This page can be used alone or together with the previous page.



In case you have already defined mute patterns, the **Muted Patterns** panel shows all defined so far. You can click on the menu of any one of them to **Edit** or **Delete** it.

But, you can also add new muted defects by selecting any set of defects, at any scope, just click on any row to open child nodes.



After you are done, just click on the **Mute selected** button to add those new ones to your list of Muted Patterns.

As before, a message indicating the need to **Recalculate** appears.

Muting at the defects submenu of a delivery analysis



If you run delivery analyses on an application, you could also mute defects in **any** delivery performed over the **last** baseline analysis or in deliveries without related baseline analysis.

Once muted, those muted defects will be considered in further delivery and baseline analyses

When you are at the Life Cycle module, you can see the list of deliveries as in the image below.

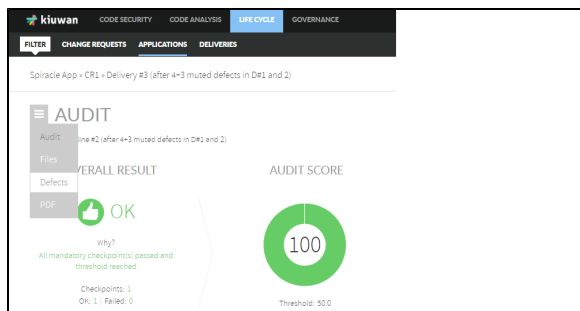
| Application | Date | Duration | # Deliveries | # Change requests | # OK deliveries | # Failed deliveries | # In progress deliveries | # Resolved deliveries | Status | Score |
|--|------------------|----------------|--------------|-------------------|-----------------|---------------------|--------------------------|-----------------------|--------|-------|
| Spiracle App | 2021/02/25 18:32 | 17m | 3 | 1 | 0 | 3 | 3 | 0 | 🟢 | 1 |
| Delivery label | Date | Change request | Branch | Files | Status | Score | | | | |
| Delivery #2 (after 4+3 muted defects in D#2 and 2) | 2021/02/25 18:32 | CR1 | - | 113 | 🟡 | 100 | | | | |
| Delivery #2 (after 4+3 muted defects in Delivery #2) | 2021/02/25 18:32 | CR1 | - | 113 | 🟡 | 100 | | | | |
| Delivery #2 | 2021/02/25 18:32 | CR1 | - | 113 | 🟡 | 100 | | | | |

In case you have muted defects, any delivery previously analyzed to the muting will have a warning icon indicating that the audit was done before the muting so results may not match shown defects.

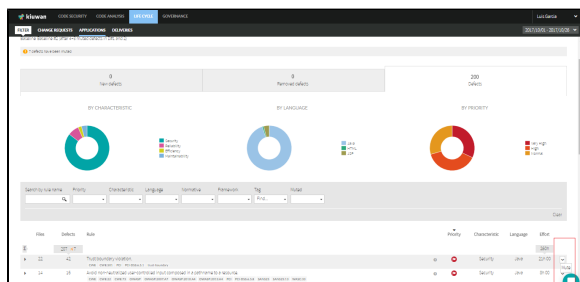
Mute patterns changed after this delivery was analyzed. Audit result may not agree with shown defects.

To mute defects in the delivery, just click on the Status icon of any delivery to open the Audit for it.

Afterward, you can select Defects sub-menu and mute defects over the delivery defects list as in a delivery analysis.



You can mute defects than either at the **New Defects** or the **Defects** tab.



Here you mute defects over the defects list as in a delivery analysis.