CWE Compliance

In this section:

Introduction

The Parasoft CWE Compliance artifact is a set of assets for your DTP infrastructure that enable you to track and visualize programming errors associated with CWE (Common Weakness Enumeration) guidelines. The artifact is shipped as part of the Security Compliance Pack for DTP 5.4.1. Contact your Parasoft representative for download and licensing information.

Supported Guidelines

The CWE Compliance artifact supports the following specific CWE implementations:

- 2011 CWE/SANS Top 25 Most Dangerous Software Errors.
- CWE List Version 3.1
- CWE List Version 2.11

Click on the following links to learn more about CWE guidelines:

- http://cwe.mitre.org/top25/
- https://cwe.mitre.org/data/

Prerequisites

One of the following Parasoft code analysis tools with appropriate Security Compliance licenses are required:

- Jtest 10.4.0 or later
- dotTEST 10.4.1 or later

Process Overview

1. Install the Security Compliance Pack to add the CWE assets to your system (see Installation instructions).
2. Deploy the DTP artifact using Extension Designer. Deploying the artifact will enable you to use the widgets, reports, and configurations to view Parasoft static analysis rule violations within the context of CWE.
3. Connect your code analysis tool to your project in DTP. Configure the settings that enable DTP to correlate analysis results, i.e., build ID, source control settings, etc. See the documentation for your analysis tool for details.
4. Analyze the project with your code analysis tool using one of the CWE test configurations.
5. Run the KPI workflow as part of your automated build process to generate the compliance data.
6. Use the DTP dashboard template, widgets, and reports to monitor compliance with security standards.

CWE Compliance Assets

The following artifacts are included in the package and added to your DTP environment when you install the Security Compliance Pack.

CWE Compliance.json

This is the core asset that extends DTP's data processing capabilities and produces CWE widgets and reports. DTP Workflows must be deployed using Extension Designer before they can be used (see Deploying the CWE Assets).

Dashboard Templates

Dashboard templates include preconfigured widgets to help you quickly view specific information about your projects. Review the Dashboards section if you are unfamiliar with dashboards in DTP. The following template files are included in the CWE Compliance artifact:

- CWE 3.1 - .NET (CWE-3_1-dotNET.json)
- CWE Top 25 2011 - .NET (CWE-Top-25-dotTEST.json)
- CWE Top 25 2011 - Java (CWE-Top-25-Java.json)

See Adding the CWE Dashboards for details.

Models and Profiles

Profiles provide a range of functions in a DTP infrastructure, such as providing inputs for custom calculations executed by an extension and providing data for compliance reports. Profiles take their structure from models, which define fields, headers, or other components used in the profile. See Working with Model Profiles for information about understanding profiles in DTP Enterprise Pack. The following profile files are included with the CWE artifact.

- CWE 3.1 - .NET profile (cwe-3_1-dotnet.json)
- CWE Security Impact - .NET profile (cwe-security-impact-dotnet.json)
- CWE Security Impact - Java profile (cwe-security-impact-java.json)
- CWE SANS Top 25 2011 - .NET profile (cwe-top25-2011-dotnet.json)
- CWE Compliance model (cwe-compliance.json)
- KPI model (KPI.json)

Compliance Categories

Individual code analysis rules belong to a category, such as Security, Exceptions, etc. The CWE Compliance artifact includes files that map code analysis rules to CWE-specific categories, i.e., weakness type or impact. You can configure widgets to report violations according to the categories defined in the following files to view them according to their CWE category:

- CWE 2.11 - Technical Impact - Java (CWE-2_11-Impact-java.xml)
- CWE 2.11 - Java (CWE-2_11-java.xml)
- CWE 3.1 - Development Concepts - .NET (CWE-3_1-Development-Concepts-dotNET.xml)
- CWE 3.1 - .NET (CWE-3_1-dotNET.xml)
- CWE 3.1 - Technical Impact - .NET (CWE-Impact-dotNET.xml)
- CWE SANS Top 25 2011 - Development Concepts - .NET (CWE-Top-25-2011-Development-Concepts-dotNET.xml)
- CWE SANS Top 25 2011 - .NET (CWE-Top-25-2011-dotNET.xml)
- CWE SANS Top 25 2011 - Java (CWE-Top-25-2011-java.xml)
- CWE SANS Top 25 2011 - Technical Impact - .NET (CWE-Top-25-Impact-dotNET.xml)
- CWE SANS Top 25 2011 - Technical Impact - Java (CWE-Top-25-Impact-java.xml)

See Custom Compliance Categories for additional information about rule categories in DTP.

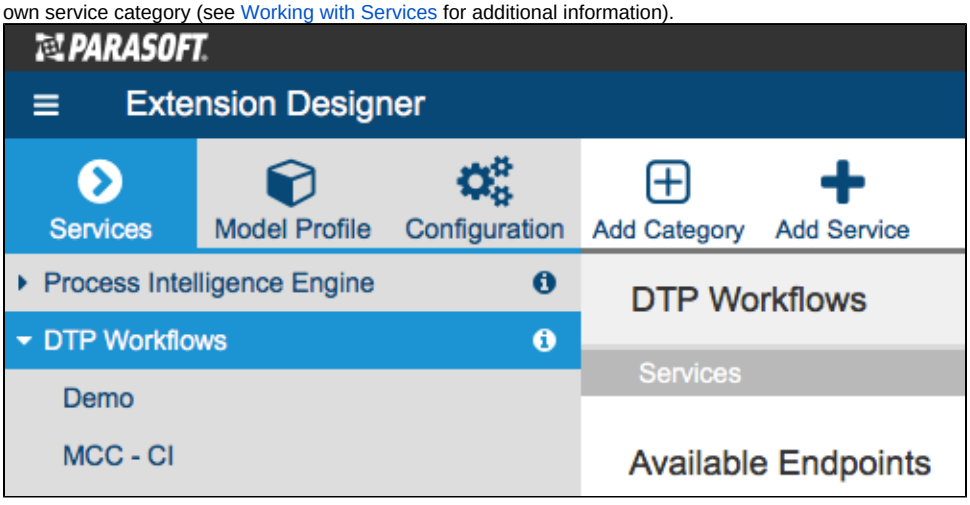
Key Performance Indicator.json

This DTP Workflow performs additional calculations to provide metrics data specific to CWE. The KPI Workflow is optional and is not specific to the CWE Compliance artifact. To use this workflow, deploy it to your DTP environment and manually add instances of the standard Metrics - Summary widget to your dashboard to view the data. See Calculating Security Impact for details.

Deploying the CWE Assets

The CWE Compliance assets are installed when you install the Security Compliance Pack (see Installation). After installing the artifact, you must deploy the assets to your DTP environment.

1. Choose **Extension Designer** from the DTP settings (gear icon) menu.
2. Click the **Services** tab and expand the **DTP Workflows** services category. You can deploy assets under any service category you wish, but we recommend using the DTP Workflows category to match how Parasoft categorizes the assets. You can also click **Add Category** to create your own service category (see Working with Services for additional information).



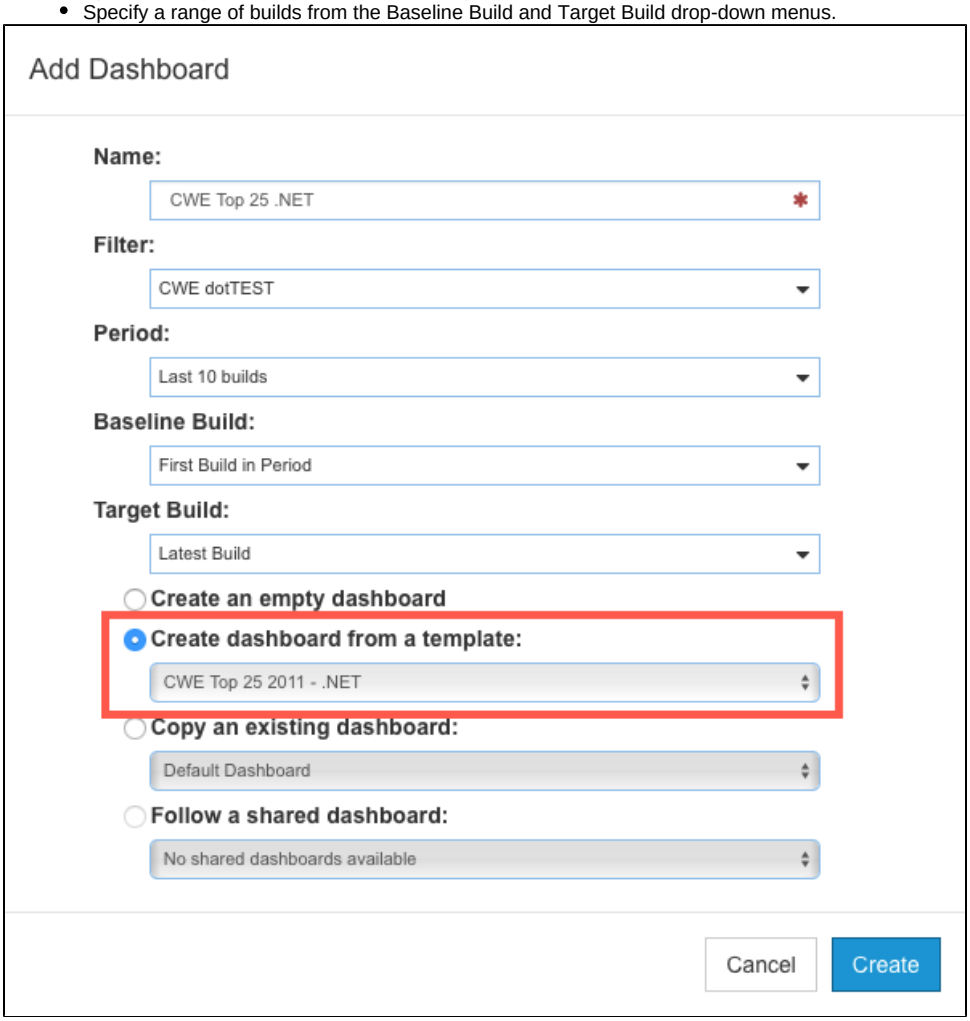
3. You can deploy the artifact to an existing service or add a new service. The number of artifacts deployed to a service affects the overall performance. See Extension Designer Best Practices for additional information. Choose an existing service and continue to step 5 or click **Add Service**.
4. Specify a name for the service and click **Confirm**.
5. The tabbed interface helps you keep artifacts organized within the service. Organizing your artifacts across one or more tabs does not affect the performance of the system. Click on a tab (or click the **+** button to add a new tab) and click the vertical ellipses menu.
6. Choose **Import> Library> Workflows> Security> CWE Compliance** and click anywhere in the open area to drop the artifact into the service.
7. Click **Deploy** and refresh your DTP browser window.

You can now add CWE widgets, use CWE compliance categories, and view CWE reports.

Adding the CWE Dashboards

The CWE Compliance dashboard templates will be available after installing the Security Compliance Pack. If you do not see the dashboard templates, restart DTP (see Stopping DTP Services and Starting DTP Services.

1. Click **Add Dashboard** in the DTP toolbar and specify a name when prompted.
2. (Optional) You can configure the default view for the dashboard by specifying the following information:
   - Choose the filter associated with your project from the Filter drop-down menu. A filter represents a set of run configurations that enabled custom views of the data stored in DTP. See Creating and Managing Filters for additional information.
   - Specify a range of time from the Period drop-down menu.
   - Specify a range of builds from the Baseline Build and Target Build drop-down menus.



3. Enable the **Create dashboard from a template** option and choose one of the CWE templates.
4. Click **Create** to finish adding the dashboard.

Repeat the process for any additional CWE dashboards you want to add to you DTP view.

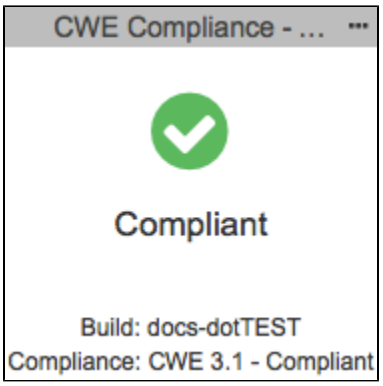
If you have already executed your code analysis tool using the correlated CWE test configuration, widgets will render data as soon as the dashboard is added. You can immediately begin using these widgets and working with the data to help you track your compliance goals.

About the .NET Dashboard Templates

The dashboard templates for .NET projects have the same widgets but are configured to show data related to either a CWE SANS Top 25 2011 - .NET compliance category or a CWE 3.1 - .NET compliance category. See Compliance Categories for a list of the available compliance categories. The following widgets are included:

CWE Compliance - Status

This widget shows the general compliance status of the project.



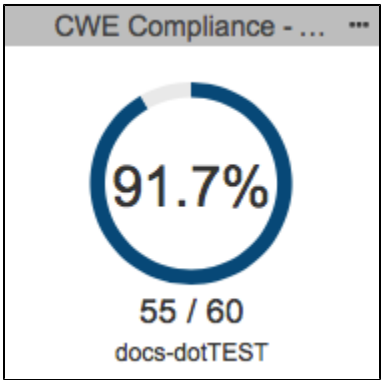
The widget can show one of several states:

- Compliant - Code meets all required guidelines.
- Compliant with Deviations - Code meets all guidelines, but deviations have been applied. Deviations are violations that you have determined to be acceptable (see Deviation Report for additional information about deviations).
- Not Compliant - Code does not meet all required guidelines.
- Missing rule(s) in analysis - Parasoft code analysis rules documented in your profile were not included in the specified build.

Click on the widget to open the CWE Compliance Report.

CWE Compliance - Percentage

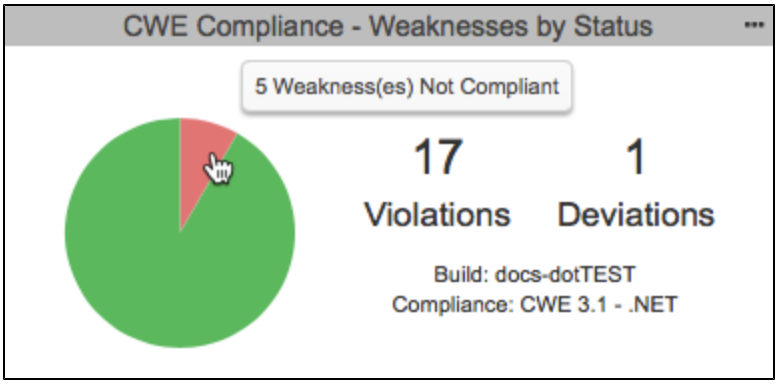
This widget shows how much of the project is in compliance with the CWE guidelines.



Click on the widget to open the CWE Compliance Report.

CWE Compliance - Weakness by Status

This widget shows the number of rules passed, violations, and deviations (suppressed code analysis violations). The green segment in the pie chart represents passing rules, while the red segment represents rules that have been violated.

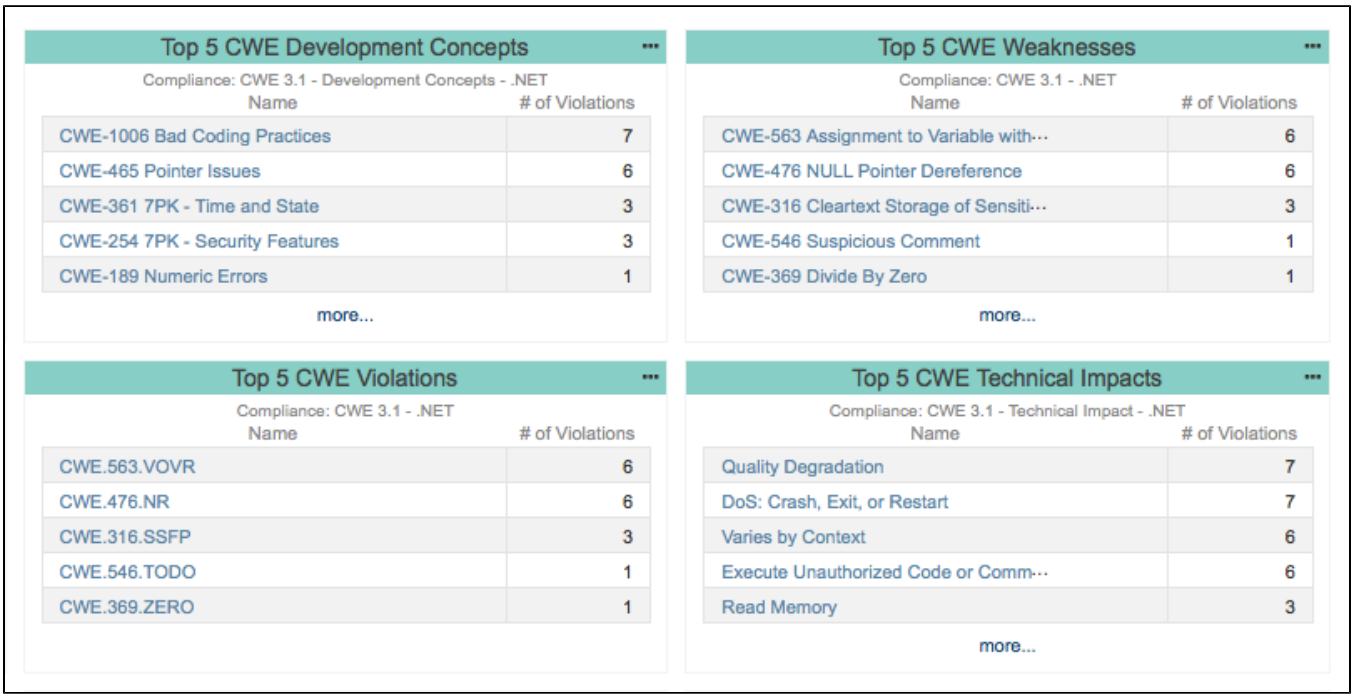


You can perform the following actions:

- Mouse over a segment of the pie chart to view details.
- Click on the passing segment of the pie chart to open the CWE Compliance Report filtered by passing guidelines.
- Click on the violations segment of the pie chart to open the CWE Compliance Report filtered by violations.
- Click on the Violations value to open an unfiltered instance of the CWE Compliance Report.
- Click on the Deviations value to open the Deviation Report.

Violations by Category

The dashboard includes several instances of the standard DTP Categories - Top 5 Table widget configured to show violations according to CWE guidelines.

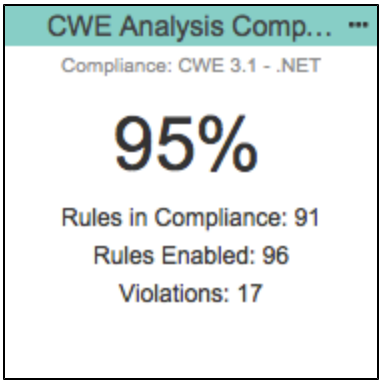


Each instance of the widget is driven by the compliance category configuration (see Compliance Categories).

Click on a category link in the Name column to open the Violations by Rule report. Click on the **more...** link (if more than five categories contain violations) to view the Violations by Compliance Category report.

Rules in Compliance

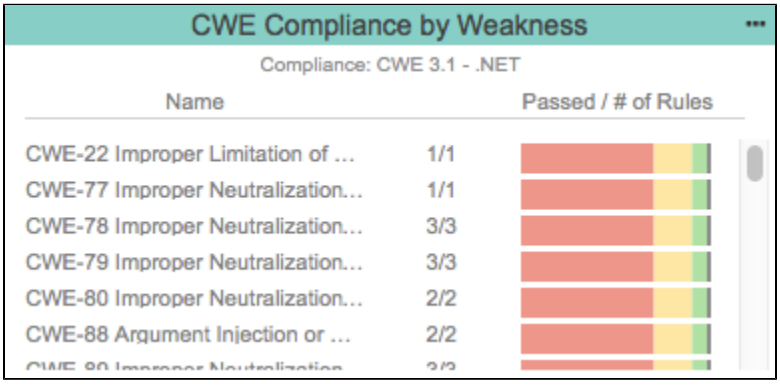
The dashboard includes an instance of the standard DTP Rules in Compliance - Summary widget configured for CWE. This widget shows what percentage of the rules are in compliance, number of rules in compliance, rules enabled, and number of violations. Click on the widget to view the Violations by Compliance Category report.



Click on the widget to view the Violations by Compliance Category report.

Compliance by Category

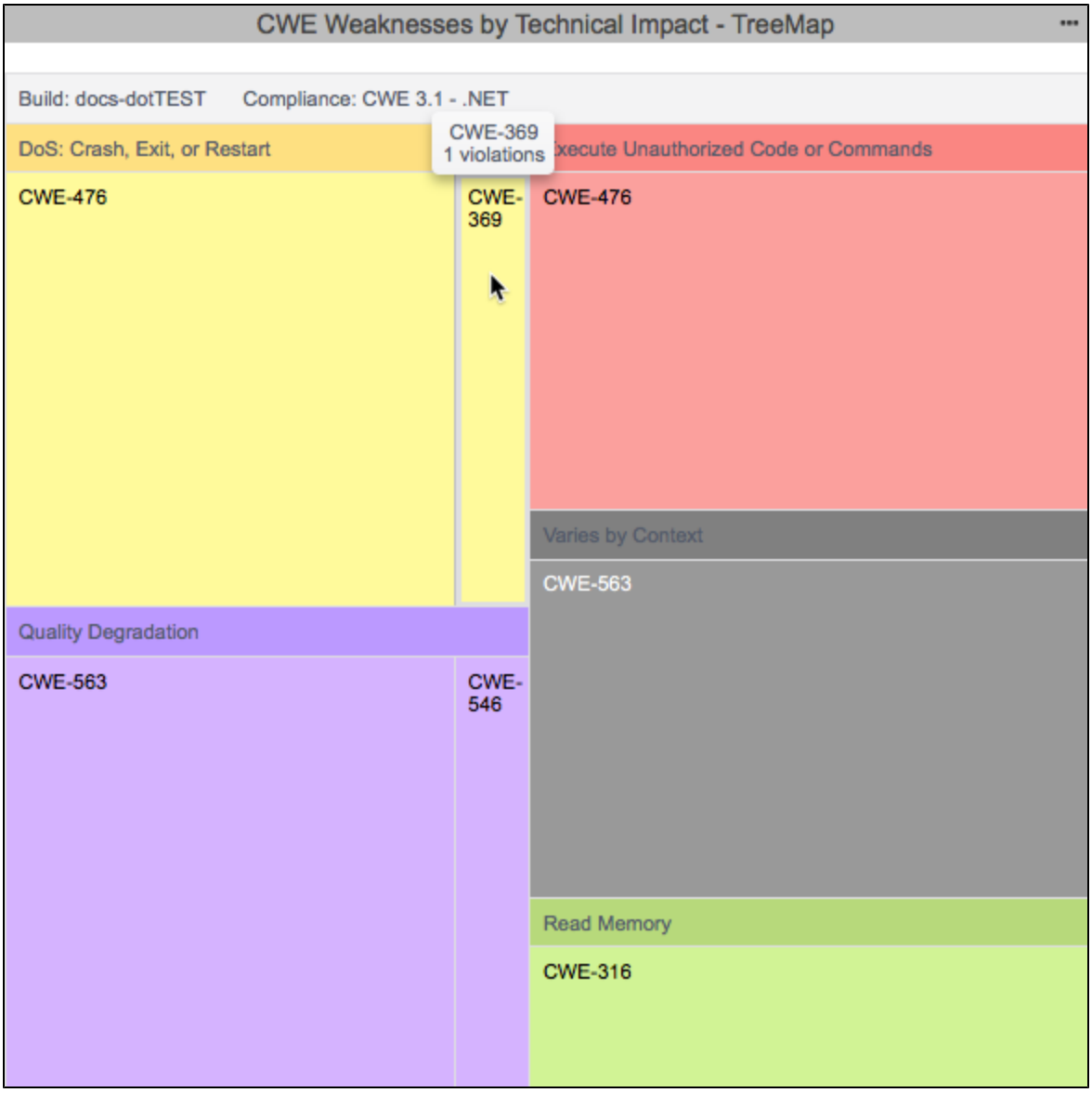
The dashboard includes an instance of the standard DTP Compliance By Category widget configured for CWE. This widget provides an overview of the compliance status for each category in the compliance configuration.



Click on the widget to open the Violations by Rule report.

CWE Weakness by Technical Impact - TreeMap

This widget shows how static analysis violations are concentrated according to their technical impact.



Mouse over a leaf in the widget to view details. Click on a leaf to open the Violations Explorer filtered by the compliance category.

About the Java Dashboard Template

Widgets in the CWE Top 25 - Java dashboard template includes standard DTP widgets preconfigured to show data according to a Java-oriented compliance category. See Compliance Categories for a list of the available compliance categories. The following widgets are included:

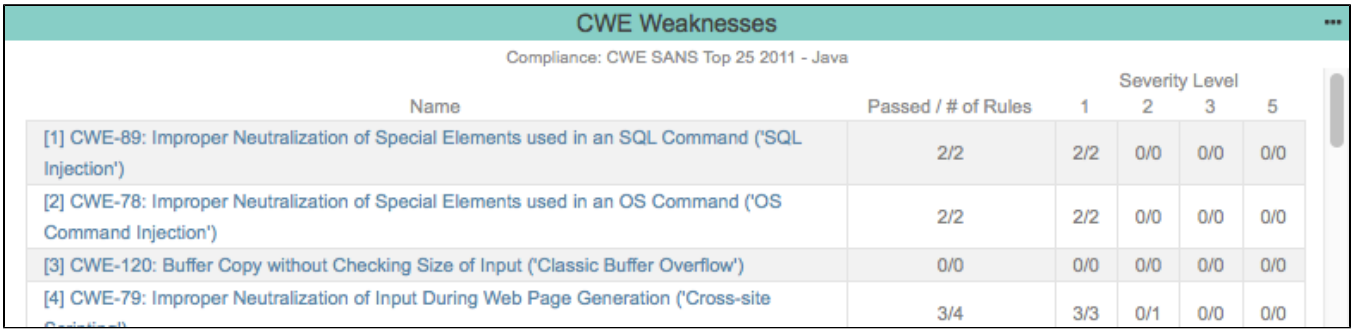
Rules in Compliance

The dashboard includes an instance of the standard DTP Rules in Compliance - Summary widget configured for CWE. This widget shows what percentage of the rules are in compliance, number of rules in compliance, rules enabled, and number of violations.

Click on the widget to view the Violations by Compliance Category report.

CWE Weaknesses

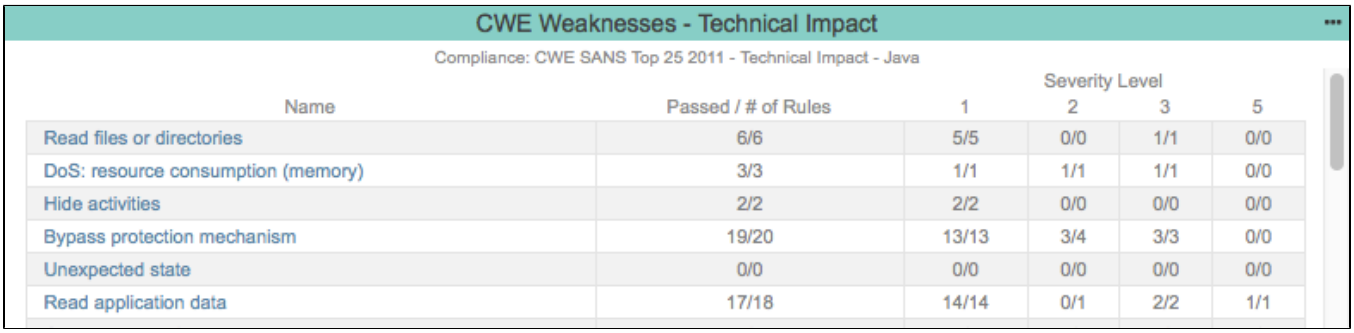
The dashboard includes an instance of the standard DTP Compliance by Category/Severity widget configured to show data according to the CWE SANS Top 25 2011 - Java compliance category. This widget lists the weakness categories, number of rules used to detect each weakness, number of passing rules for each weakness, and the breakdown of the security level(s) associated with the rules. Click on a weakness link in the Name column to open the Violations by Rule report.



Name	Passed / # of Rules	Severity Level 1	2	3	5
[1] CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	2/2	2/2	0/0	0/0	0/0
[2] CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	2/2	2/2	0/0	0/0	0/0
[3] CWE-120: Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')	0/0	0/0	0/0	0/0	0/0
[4] CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site	3/4	3/3	0/1	0/0	0/0

CWE Technical Impact

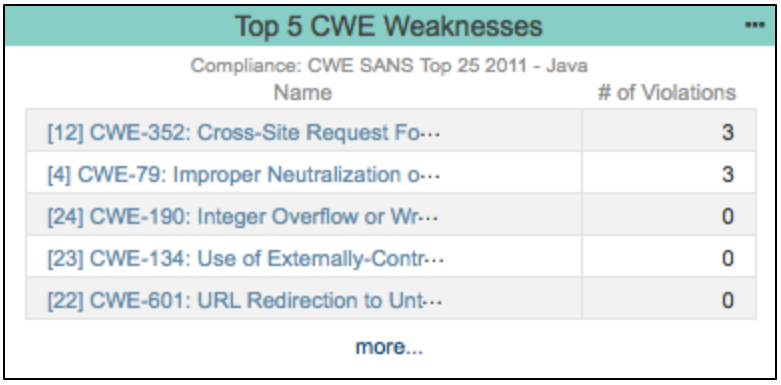
The dashboard includes an instance of the standard DTP Compliance by Category/Severity widget configured to show data according to the CWE SANS Top 25 2011 - Technical Impact - Java compliance category. This widget lists the technical impact categories, number of rules used to detect each impact category, number of passing rules for each category, and the breakdown of the security level(s) associated with the rules. Click on a technical impact category link in the Name column to open the Violations by Rule report.



Name	Passed / # of Rules	Severity Level 1	2	3	5
Read files or directories	6/6	5/5	0/0	1/1	0/0
DoS: resource consumption (memory)	3/3	1/1	1/1	1/1	0/0
Hide activities	2/2	2/2	0/0	0/0	0/0
Bypass protection mechanism	19/20	13/13	3/4	3/3	0/0
Unexpected state	0/0	0/0	0/0	0/0	0/0
Read application data	17/18	14/14	0/1	2/2	1/1

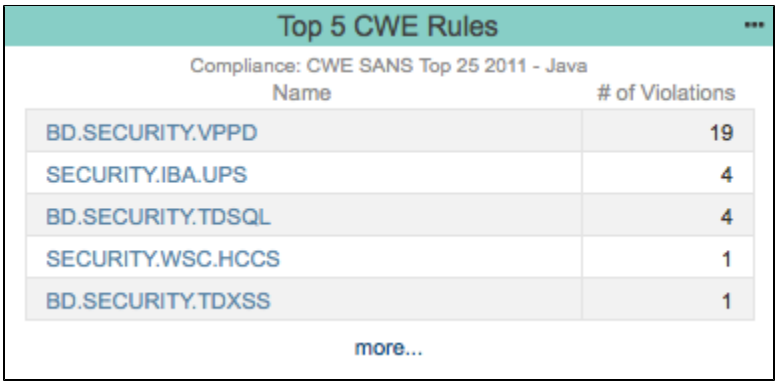
Top 5 CWE Weaknesses

The dashboard includes an instance of the standard DTP Categories - Top 5 Table widget configured to show data according to the CWE SANS Top 25 2011 - Java compliance category. This widget lists the five weaknesses with the highest number of violations. Click the **more...** link to open the Violations by Compliance Category report. Click on a weakness link in the Name column to open the Violations by Rule report.



Top 5 CWE Rules

The dashboard includes an instance of the standard DTP Rules - Top 5 Table widget configured to show data according to the CWE SANS Top 25 2011 - Java compliance category. This widget lists the five code analysis rules with the highest number of violations. Click the **more...** link to open the Violations by Compliance Category report. Click on a rule link in the Name column to open the Violations Explorer.

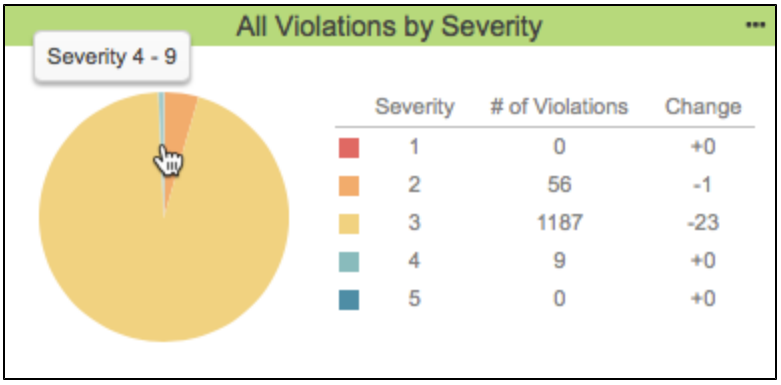


All Violations

The dashboard includes an instance of the standard DTP Violations - Summary Trend widget configured to show data according to the CWE SANS Top 25 2011 - Java compliance category. This widget shows the total number of failed or suppressed violations, the trend over a specified period, and the change from the first to last build in that period. Click on the widget to to open the Violations Explorer.

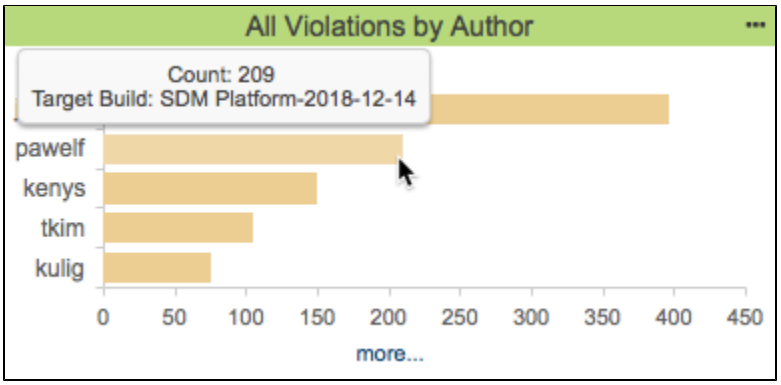
All Violations by Severity

The dashboard includes an instance of the standard DTP Severities - Pie widget configured to show data according to the CWE SANS Top 25 2011 - Java compliance category. This widget shows a pie chart of the violations in the project. Each segment represents a severity level. The legend includes the number of violations and changes from the baseline build to the target build. Click on the widget to to open the Violations Explorer.



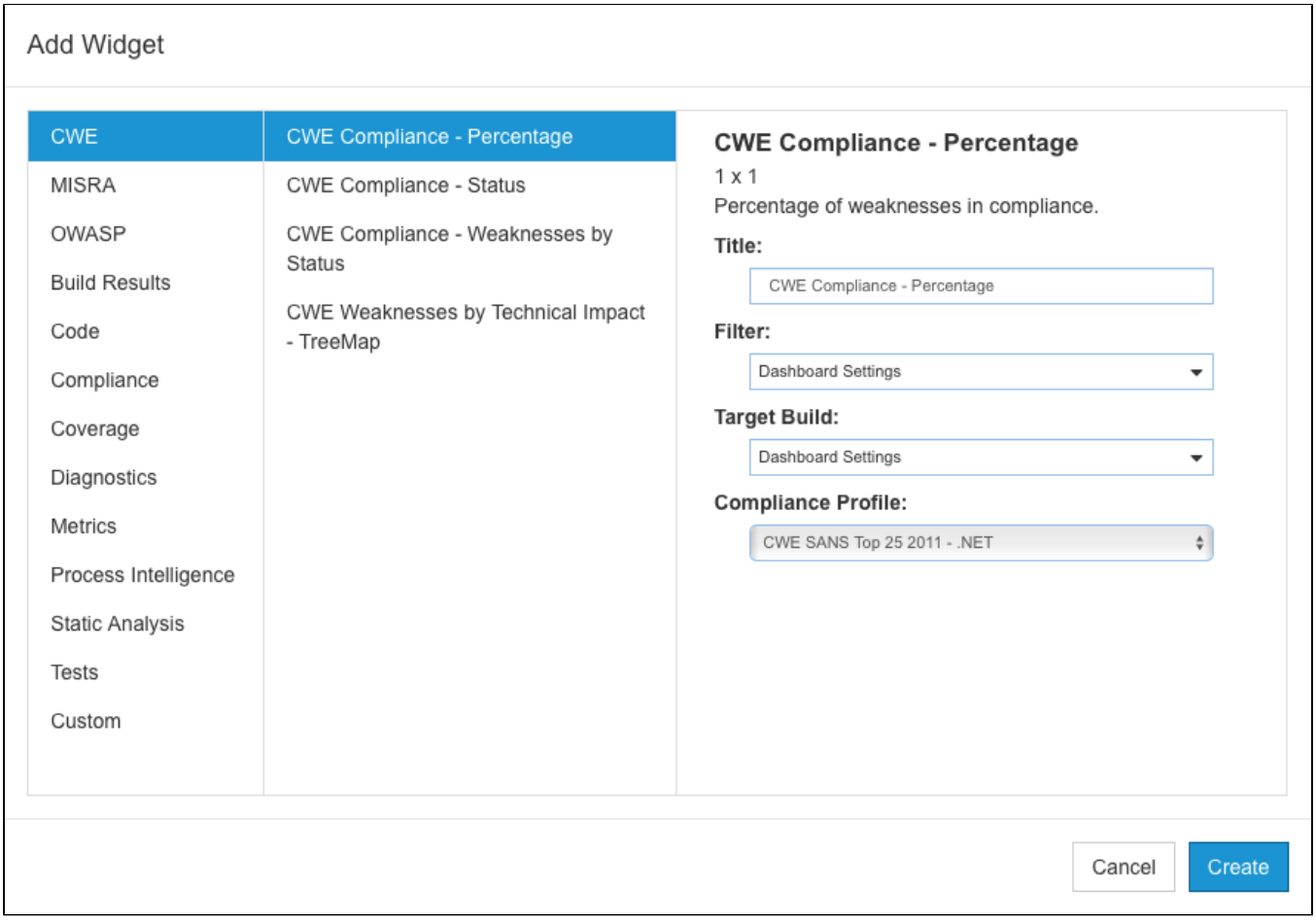
All Violations by Author

The dashboard includes an instance of the standard DTP Authors - Top 5 Bar widget configured to show data according to the CWE SANS Top 25 2011 - Java compliance category. This widget shows the five code authors with the highest number of violations. Click on the **more...** link to open the Authors Report. Click on a bar to view the author's violations in the Violations Explorer.



Manually Adding the CWE Widgets

You can manually add the CWE widgets to an existing dashboard. See Adding Widgets for general instructions on how to add widgets to a dashboard. After deploying the artifact, widgets will appear in the CWE category.



Title	You can rename the widget in the Title field. This setting is available for all widgets.
Filter	Choose a specific filter or Dashboard Settings from the drop-down menu. See Creating and Managing Filters for additional information. This setting is available for all widgets.
Target Build	Choose a specific build from the drop-down menu. The build selected for the entire dashboard is selected by default. See Using Build Administration for additional information about understanding builds. This setting is available for all widgets.
Compliance	Choose a compliance category group to view the data. See Compliance Categories for a list of the CWE-related compliance category groups. This configuration is not available for custom CWE widgets.
Compliance Profile	Profiles are assets shipped with an extension that enable DTP to perform additional calculations. See Models and Profiles for a list of the profiles shipped with the CWE Compliance artifact.

The CWE Compliance artifact includes profiles that you can use to calculate the security impact of detected weaknesses. Additional steps are required to leverage this functionality. See Calculating Security Impact for details.

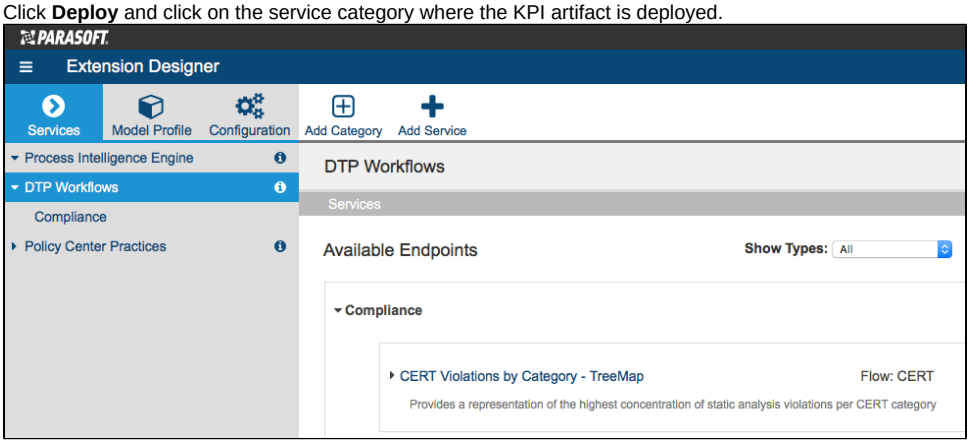
Calculating Security Impact

The Key Performance Indicator (KPI) DTP Workflow defines a KPI associated with static analysis rules so you can measure and quantify results. The build must have static analysis and metrics analysis data for the KPI extension to perform the calculation. Be sure that code analysis tool has been executed with the Metrics test configuration, as welll as the CWE 3.1 test configuration under the same build ID. The metrics analysis must also include data for the Logical Lines of Code metric (metricId METRIC.NOLLOCIF). Refer to the tool documentation for details about setting the build ID and executing the Metrics test configuration.

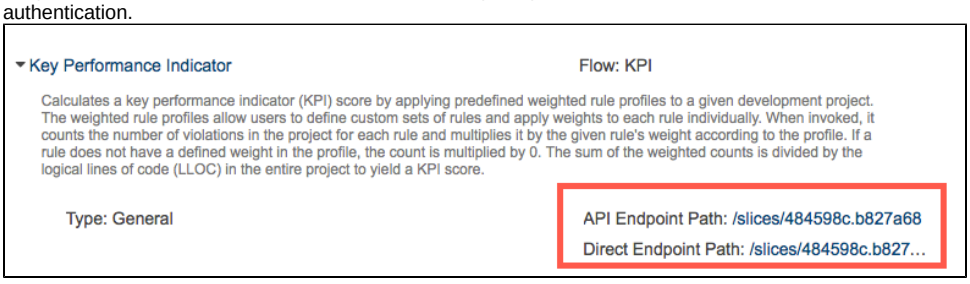
This artifact needs to be deployed manually before you can use it.

1. Open **Extension Designer** and click on the **Services** tab.
2. Choose a service under a service category for running the KPI artifact. We recommend using a service in the DTP Workflows category to match how Parasoft categorizes the assets. You can deploy the artifact to an existing service or add a new service. The number of artifacts deployed to a service affects the overall performance.
3. The tabbed interface helps you keep artifacts organized within the service. Organizing your artifacts across one or more tabs does not affect the performance of the system. Click on a tab (or click the **+** button to add a new tab) and click the vertical ellipses menu.

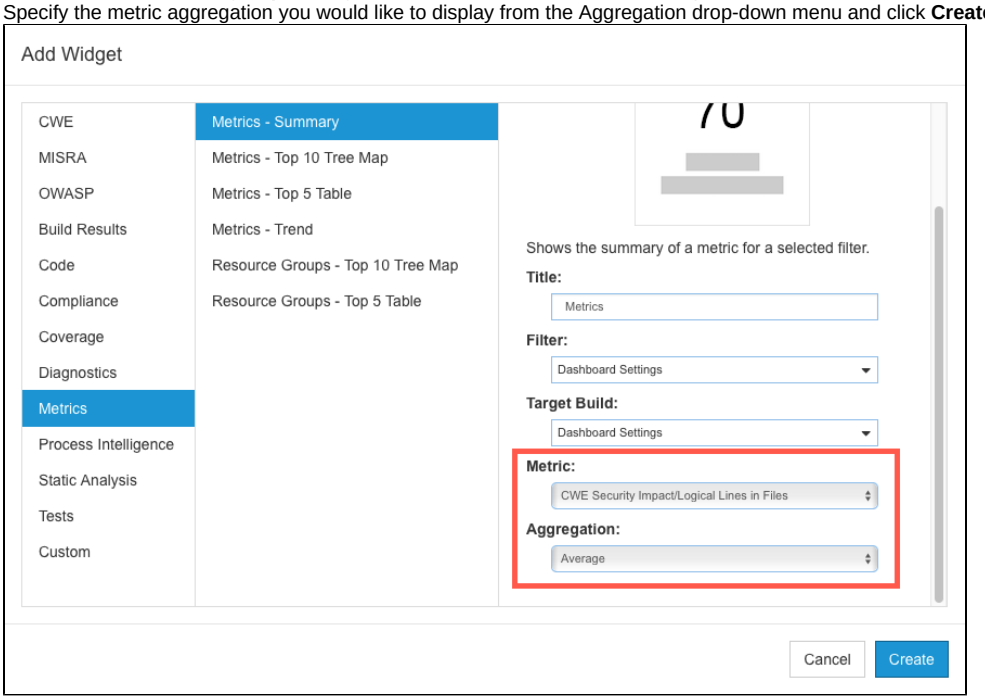
4. Choose **Import> Library> Workflows> Security> Key Performance Indicator** and click anywhere in the open area to drop the artifact into the service.
5. Click **Deploy** and click on the service category where the KPI artifact is deployed.



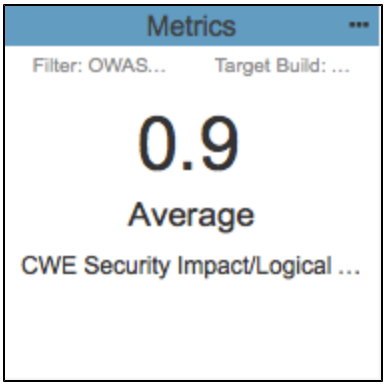
6. Expand the Key Performance Indicator section and copy the endpoint. Extension Designer presents two paths for the endpoint. The API Endpoint Path includes all API directories and can be used for exercising the endpoint in most cases. The Direct Endpoint Path is the direct path to the endpoint on the server and can be used if the API endpoint path is blocked or inaccessible, such as in some third-party integrations that require authentication.



7. Send a REST request to the endpoint along with the required parameters. See Execution Details.
8. Open your DTP dashboard and click **Add Widget.**
9. Choose the **Metrics> Metrics - Summary** widget in the overlay.
10. Choose the **CWE Security Impact/Logical Lines in Files** from the Metric drop-down menu
11. Specify the metric aggregation you would like to display from the Aggregation drop-down menu and click **Create**.



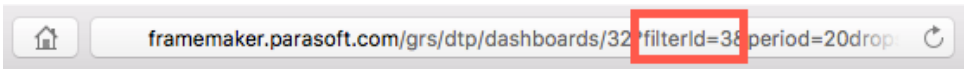
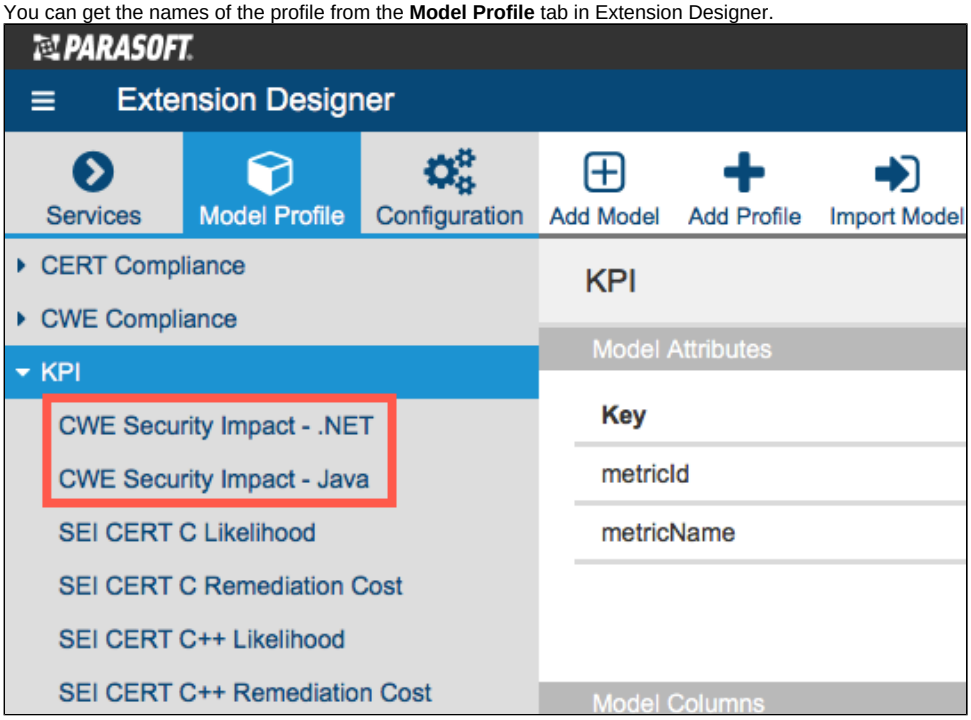
The widget will appear on your dashboard.



Clicking on the widget opens the Single Metric Overview Report.

Execution Details

You can execute the request in a browser, using a cURL command, or add it to a script. The following table describes the required parameters:

filterId	The filter ID for the project that the calculations will be performed on. You can quickly get the filter ID from URL of your dashboard.  You can also get the filter ID from the the Filters settings in DTP administration (see Creating and Managing Filters).
profile	The name of the profile that contains the rules and weights to for the calculation. Specify one of the following profiles to calculate security impact: • CWE Security Impact - .NET • CWE Security Impact - Java You can get the names of the profile from the **Model Profile** tab in Extension Designer. 
buildId	The build on which the calculation will be performed on. If you would prefer to use the latest unlocked build that has violations data and metrics calculated, you can use `latestBuild` as the value of this parameter. If no build id is provided, this parameter defaults to `latestBuild`.

Example API Call URL

```
http://framemaker.parasoft.com:8314/api/v1/services/5c0f0cae5d018e0630ae2789/slices/9acaecb1.7eb78?
filterId=9&buildId=docs-dotTEST&profile=CWE%20Security%20Impact%20-%20.NET
```

Example Successful Response

```
{
        "success":{
                "title":"KPI",
                "message":"Calculation has started for filter 'CWE dotTEST' using profile 'CWE Security Impact
- .NET'. Check debug output for any errors during calculation."
        }
}
```

Every rule can have a different weighting, and not every rule has to be in the profile, which enables you to run different KPIs for different purposes and different profiles for different subsets of rules. See Profiles for additional information.

Run KPI as part of your automated build process

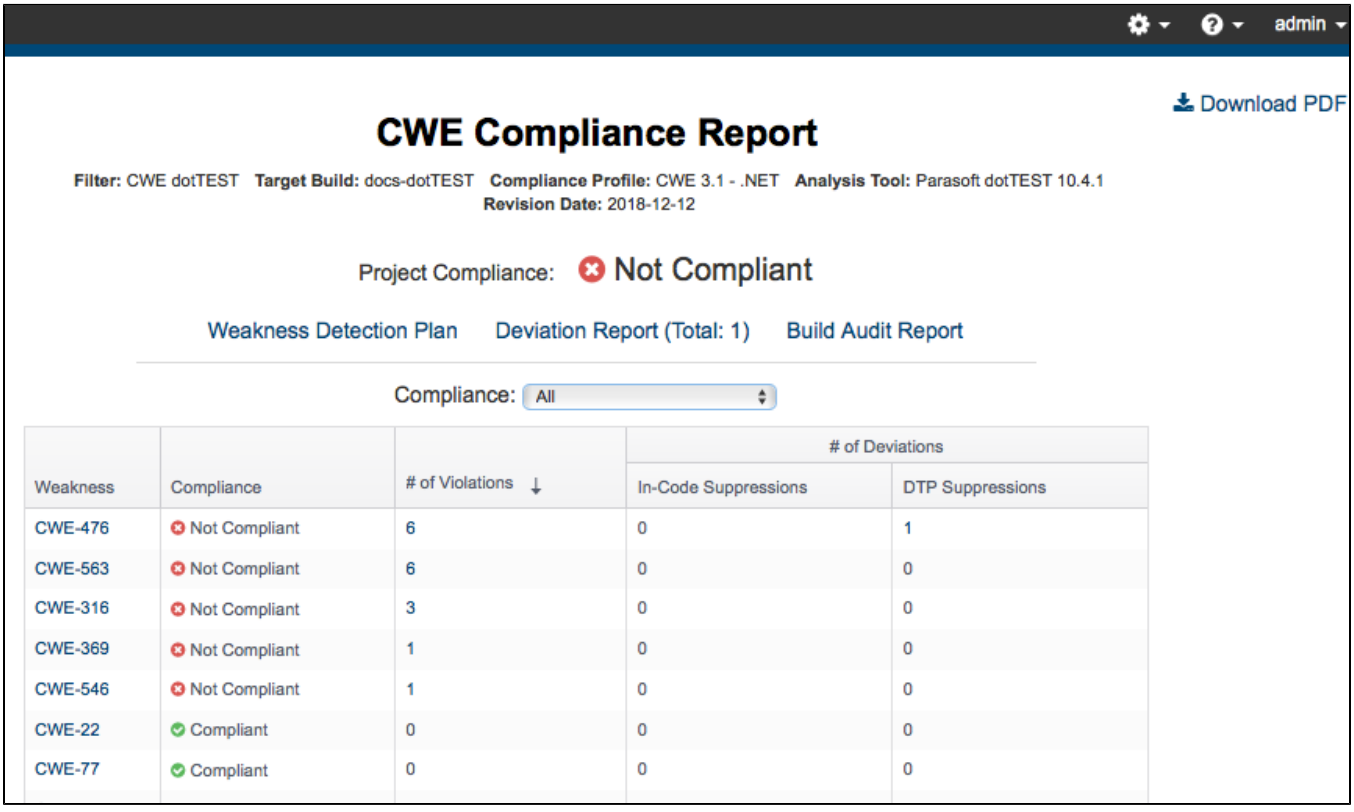
Depending on the volume of data being analyzed, KPI calculation may require multiple runs to acquire the core data and may take significant time, therefore triggering KPI calculation should be done as part of your build process or by manually using a trigger node in the KPI slice.

CWE Compliance Report

The following CWE widgets shipped with the CWE Compliance artifact link to the CWE Compliance Report:

- CWE Compliance - Status
- CWE Compliance - Percentage
- CWE Compliance - Weakness by Status

You can use the report to demonstrate compliance with CWE.



Weakness	Compliance	# of Violations	# of Deviations	
			In-Code Suppressions	DTP Suppressions
CWE-476	Not Compliant	6	0	1
CWE-563	Not Compliant	6	0	0
CWE-316	Not Compliant	3	0	0
CWE-369	Not Compliant	1	0	0
CWE-546	Not Compliant	1	0	0
CWE-22	Compliant	0	0	0
CWE-77	Compliant	0	0	0

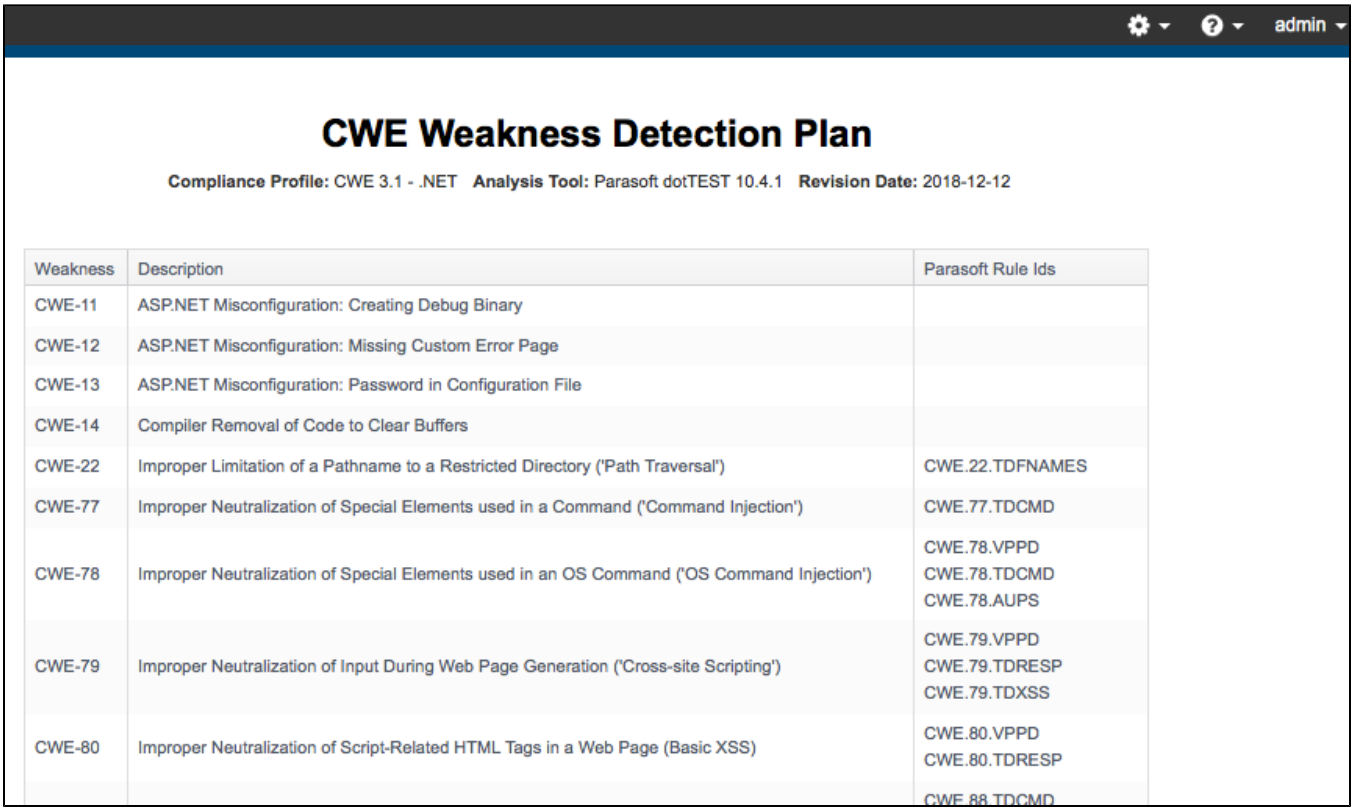
You can perform the following actions:

- Click on one of the following links to open a sub-report:
  - Weakness Detection Plan

    - Deviation Report
    - Build Audit Report
- Choose a state from the Compliance drop-down menu to filter weaknesses by their current state.
- Click on a column header to sort the report.
- Click on a link in the Weakness column to go directly to the weakness in the Weakness Detection Plan report.
- Click on a value in the # of Violations column to view the violations in the Violations Explorer.
- Click on a value in one of the # of Deviations columns to view the suppressed violation in the Violations Explorer.
- Click **Download PDF** to export a printer-friendly PDF version of the report data.

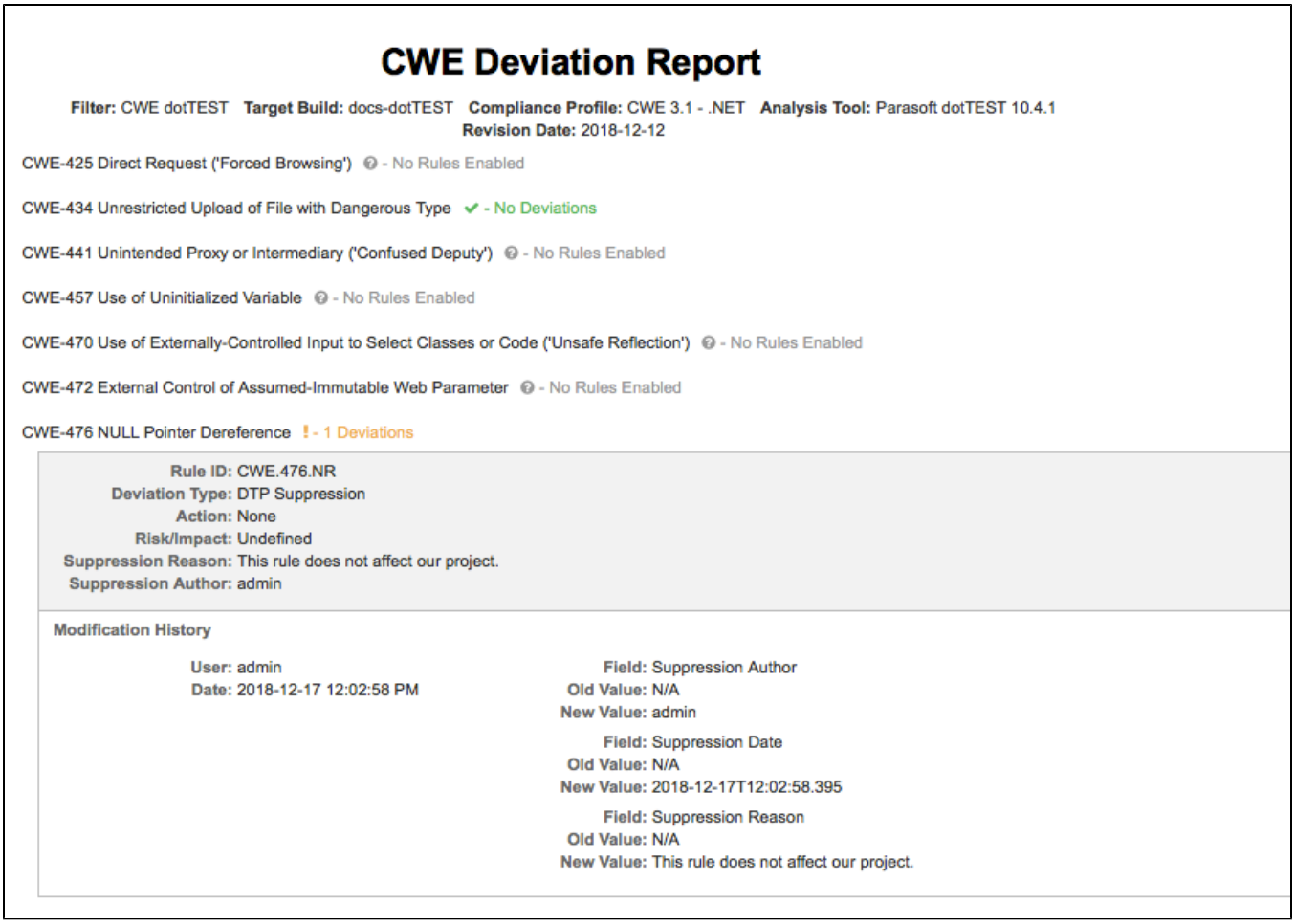
Weakness Detection Plan

The Weakness Detection Plan shows how Parasoft code analysis rules map to weaknesses. This report is populated with data from the selected compliance profile (see Models and Profiles).



Deviation Report

The Deviation Report shows information about which violations have been suppressed in the project. By default, the report shows all guidelines, but you can enable the **Only Deviations** option to filter out guidelines that have no suppressions associated with them. See Suppressing Violations for information about suppressions in DTP. Refer to the documentation for your analysis tool to learn about in-code suppressions.



CWE Deviation Report

Filter: CWE dotTEST **Target Build:** docs-dotTEST **Compliance Profile:** CWE 3.1 - .NET **Analysis Tool:** Parasoft dotTEST 10.4.1
Revision Date: 2018-12-12

CWE-425 Direct Request ('Forced Browsing') - No Rules Enabled

CWE-434 Unrestricted Upload of File with Dangerous Type - No Deviations

CWE-441 Unintended Proxy or Intermediary ('Confused Deputy') - No Rules Enabled

CWE-457 Use of Uninitialized Variable - No Rules Enabled

CWE-470 Use of Externally-Controlled Input to Select Classes or Code ('Unsafe Reflection') - No Rules Enabled

CWE-472 External Control of Assumed-Immutable Web Parameter - No Rules Enabled

CWE-476 NULL Pointer Dereference - 1 Deviations

Rule ID: CWE.476.NR
Deviation Type: DTP Suppression
Action: None
Risk/Impact: Undefined
Suppression Reason: This rule does not affect our project.
Suppression Author: admin

Modification History

User: admin
Date: 2018-12-17 12:02:58 PM

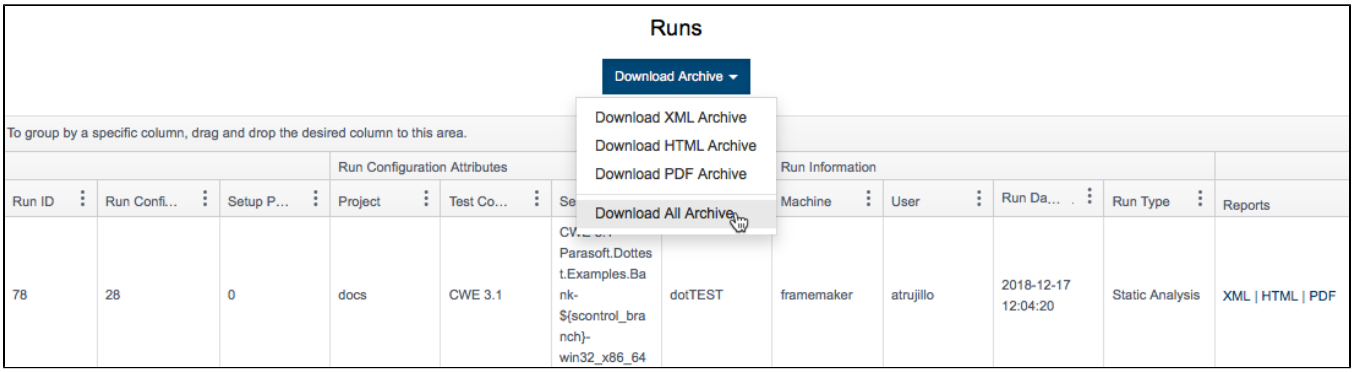
Field: Suppression Author
Old Value: N/A
New Value: admin

Field: Suppression Date
Old Value: N/A
New Value: 2018-12-17T12:02:58.395

Field: Suppression Reason
Old Value: N/A
New Value: This rule does not affect our project.

Build Audit Report

The Build Audit Report is native functionality in DTP. It shows an overview of code analysis violations, as well as test results and coverage information, associated with the build. This report also allows you to download an archive of the data, which is an artifact you can use to demonstrate compliance with CWE during a regulatory audit.



In order to download an archive, the build has to be locked. See Build Audit Report for additional details.

Profiles

The Security Compliance Pack for DTP 5.4.1 includes a set of profiles that perform custom calculations for the CWE 3.1 and CWE SANS Top 25 2011 standards and a set of profiles associated with calculating the CWE Security Impact KPI metrics for Java and .NET code.

CWE Compliance Profiles

The default profiles show the correlation between CWE guidelines and Parasoft code analysis rules and are suitable for most normal use cases.

Do not modify the CWE profiles

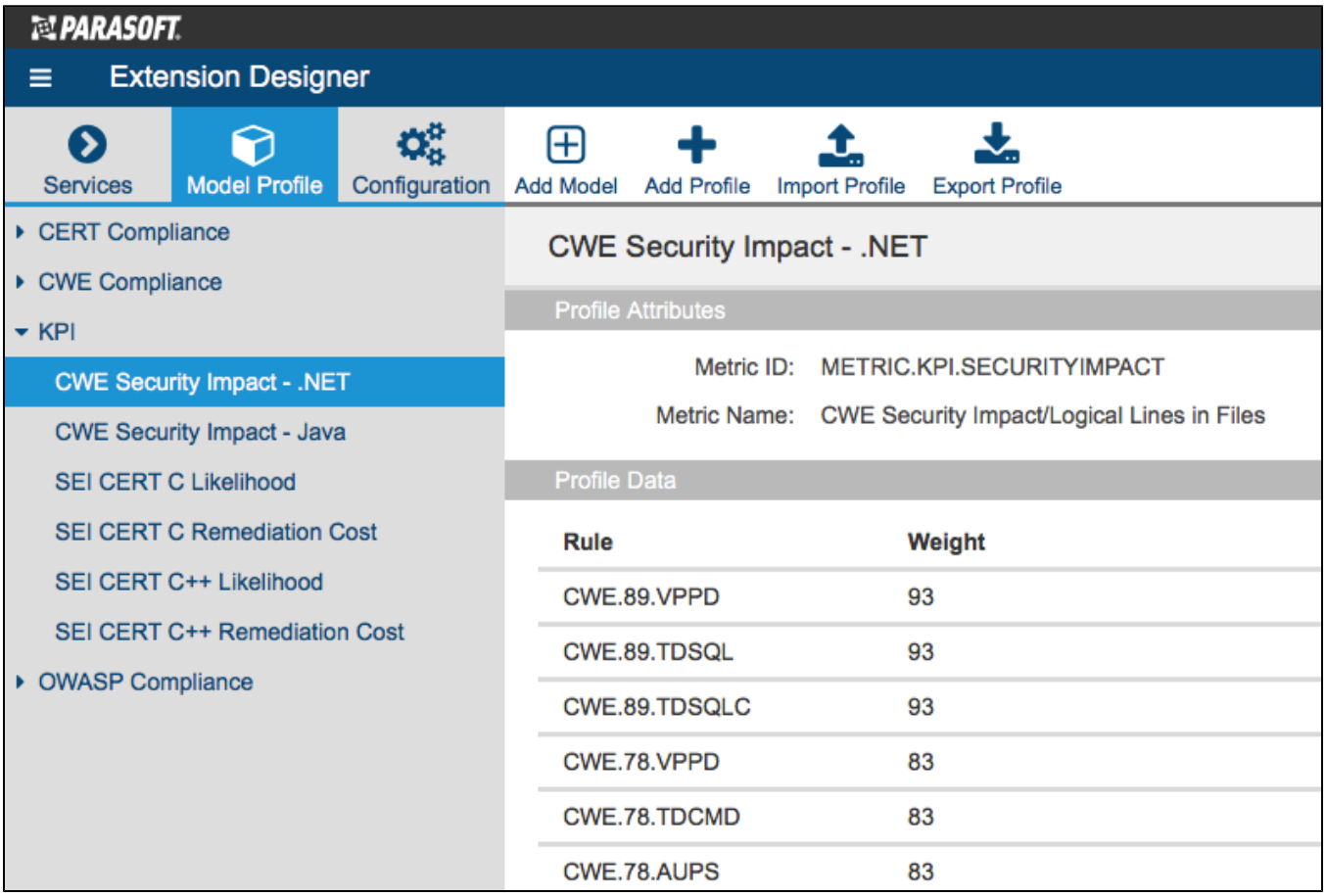
We strongly advise you to avoid changing the default CWE profiles because doing so will affect any reports you may need to generate for auditing purposes.

If necessary, you can make a copy of the default profile and adjust the correlation between Parasoft code analysis rules and CWE guidelines to achieve your software quality and compliance goals.

1. Open Extension Designer and click the **Model Profile** tab.
2. Expand the CWE Compliance model and choose either the **CWE 3.1 - .NET** or **CWE SANS Top 25 2011 - .NET** profile.
3. Click **Export Profile** to download a copy.
4. Click **Add Profile** and enter a name.
5. Click **Confirm** to create an empty profile.
6. Rename the copy of the default profile you exported and click **Import Profile**.
7. Browse for the copy and confirm to upload.
8. Click **Edit** and make your adjustments.
9. Click **Save.**

CWE KPI Profiles

The KPI artifact shipped with the Security Compliance Pack includes a SCWE Security Impact - .NET and CWE Security Impact - Java profiles. The profiles assign weights to the metrics analysis rules in order to calculate a KPI value for the build.



The default profile is suitable for most normal usage, but you can adjust the weights for each metrics rule if necessary.

1. Open Extension Designer and click the **Model Profile** tab.
2. Expand the KPI model and choose either the **CWE Security Impact - .NET** or **CWE Security Impact - Java** profile.
3. Click **Export Profile** to download a copy.
4. Click **Add Profile** and enter a name.
5. Click **Confirm** to create an empty profile.
6. Rename the copy of the default profile you exported and click **Import Profile**.
7. Browse for the copy and confirm to upload.
8. Click **Edit** and make your adjustments.
9. Click **Save.**