Key Performance Indicator

In this section:

Introduction

The Key Performance Indicator (KPI) slice calculates a score based on weights assigned to code analysis rules in your development project. The weights are defined in a profile that can be customized to meet your development goals. When the slice is invoked, it counts the number of rule violations for each file in the project. For each rule, the slice multiplies the violation count by the specified rule weight in the profile. If a rule does not have a defined weight in the profile, the count for that rule is not gathered. The sum of the weighted counts is divided by the logical lines of code (METRIC.NOLLOCIF) to yield a KPI score. This artifact ships with the Security Compliance Pack for DTP 5.4.3, which includes configurations for calculating KPIs associated with security compliance standards.

KPI Slice Workflow

1. Install the Security Compliance Pack, which installs the KPI slice and weighted profiles.
2. Deploy the KPI slice to your DTP environment.
3. Run your code analysis tool using one of the Security Compliance Pack test configurations (see Security Compliance Pack for DTP 5.4.3 for additional information).
4. Run your code analysis tool using the Metrics test configuration. The Number of Logical Lines in Files metric (METRIC.NOLLOCIF) must be enabled.
5. Call the KPI slices' REST endpoint to invoke the calculation. You must include the filter ID, build ID, and KPI profile containing the weights. The calculation is a long-running process and typically best to execute as part of your automated nightly code analysis.
6. Add a Metrics widget configured to show data according to the KPI profile.

Requirements

- DTP and DTP Enterprise Pack 5.4.3.
- Parasoft C/C++test, dotTEST, or Jtest 10.4.3.
- The code analysis tool must be configured to report the Number of Logical Lines in Files metric (METRIC.NOLLOCIF) to the DTP. The tool must also be configured with the correct filter and build ID. See DTP Concepts for additional information about filter and builds, as well as the code analysis tool for information on how to configure these settings.
- DTP must contain a build with static analysis data. If you do not include a build ID in your KPI calculation request (see Invoking the Calculation), metrics will be reported to DTP in the latest build with static analysis data. Otherwise, the calculation will be reported to the specified build.

Installation

The KPI slice is installed with the Security Compliance Pack for DTP 5.4.3. After installing the Security Compliance Pack, you will need to deploy the slice using Extension Designer.

Upgrading from KPI 2.0 or Older

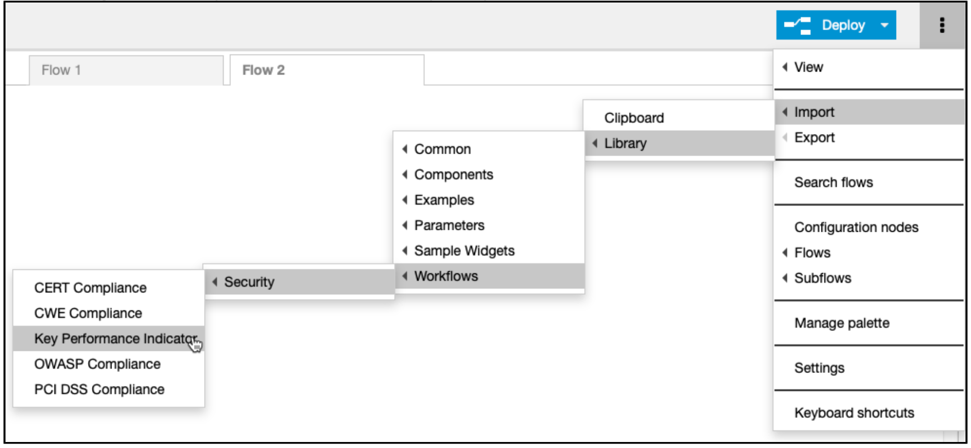
If you are upgrading from the standalone instance of the KPI slice (supported by DTP 5.3.3), uninstall the previous version of the artifact, including models and profiles, prior to installing the latest version. If you altered the default profile shipped with the previous version or created custom profiles, export the profile(s) and add the following attributes (also see Exporting and Importing Profiles):

- metricId: METRIC.KPIIF
- metircName: KPI in File

If you change the default values, we recommend setting the metric ID prefix to `METRIC.KPI.<profile name without spaces>`. Use a concise name for the metric (maximum 30 characters).

Deploying the Slice

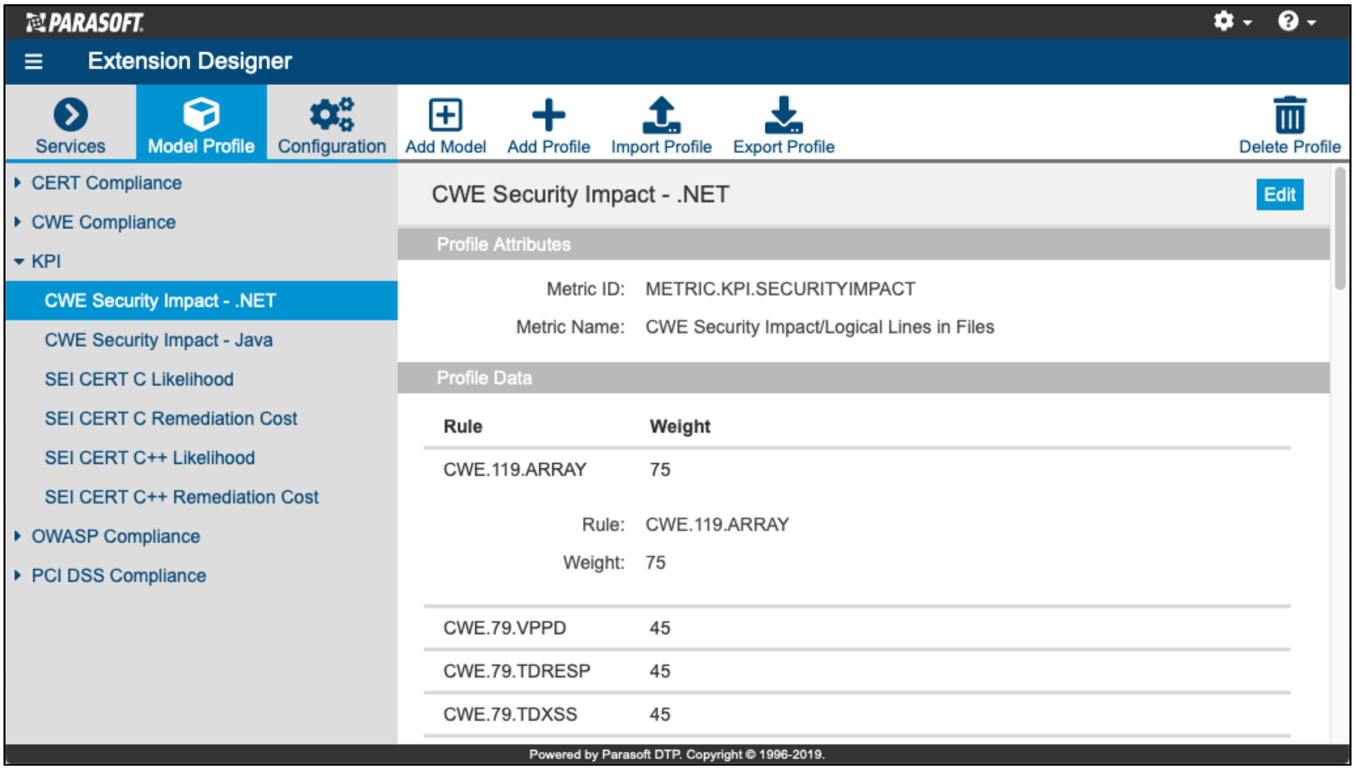
1. Open Extension Designer and click the **Services** tab.
2. Expand a service category and either click on an existing service or create new service in which to deploy the KPI slice.
3. If you are deploying the slice to an existing service, click the **+** button to add a new tab. Otherwise, go to the next step.
4. Choose **Import> Library> Workflows> Security> Key Performance Indicator** from the actions menu.



5. Click on an area in the tab to drop the nodes and click **Deploy**.

Configuration

Several KPI profiles are added when the Security Compliance Pack for DTP 5.4.3 is installed. Each profile contains weights for the Parasoft rules that check a set of security compliance guidelines. You can use the default weights or review each rule and apply your own weights.



CWE Security Impact - .NET

Profile Attributes

Metric ID: METRIC.KPI.SECURITYIMPACT

Metric Name: CWE Security Impact/Logical Lines in Files

Profile Data

Rule	Weight
CWE.119.ARRAY	75

Rule: CWE.119.ARRAY

Weight: 75

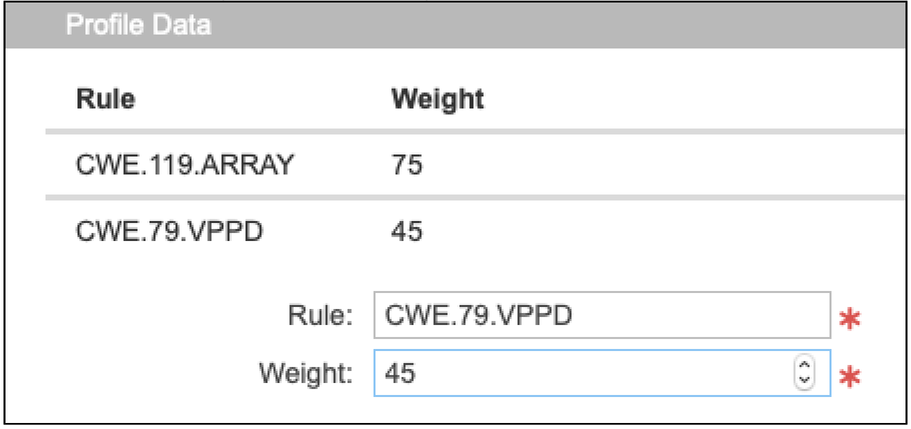
Rule	Weight
CWE.79.VPPD	45
CWE.79.TDRESP	45
CWE.79.TDXSS	45

Powered by Parasoft DTP. Copyright © 1996-2019.

To change the weight:

1. Click on a profile and click **Edit.**

2. Click on a rule and change the value in the Weight field.



Profile Data

Rule	Weight
CWE.119.ARRAY	75
CWE.79.VPPD	45

Rule: CWE.79.VPPD *

Weight: 45 *

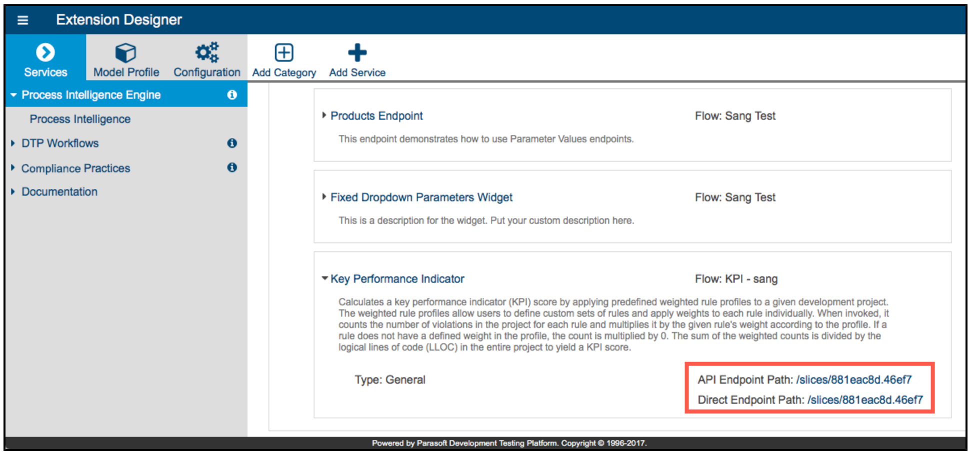
3. Click **Save.**

You can also add rules that are not in the profile to include in the calculation. Refer to Working with Model Profiles for additional information.

Invoking the Calculation

The KPI slice is a long running slice. It should only be computed when invoked by a third-party, ideally as part of a nightly job.

1. Click on the service category on the Services tab and drill down to the endpoint. You can use the API or direct endpoint path (see Service Category Page for additional information).



Products Endpoint — Flow: Sang Test

This endpoint demonstrates how to use Parameter Values endpoints.

Fixed Dropdown Parameters Widget — Flow: Sang Test

This is a description for the widget. Put your custom description here.

Key Performance Indicator — Flow: KPI - sang

Calculates a key performance indicator (KPI) score by applying predefined weighted rule profiles to a given development project. The weighted rule profiles allow users to define custom sets of rules and apply weights to each rule individually. When invoked, it counts the number of violations in the project for each rule and multiplies it by the given rule's weight according to the profile. If a rule does not have a defined weight in the profile, the count is multiplied by 0. The sum of the weighted counts is divided by the logical lines of code (LLOC) in the entire project to yield a KPI score.

Type: General

API Endpoint Path: /slices/881eac8d.46ef7

Direct Endpoint Path: /slices/881eac8d.46ef7

Powered by Parasoft Development Testing Platform. Copyright © 1996-2017.

2. Copy the endpoint and send a REST request to it with the required parameters. Use the API endpoint path if available (see Working with Flows for additional information). The following table describes the required parameters:

filterId	The filter id for the project that the calculations will be performed on.
profile	Profile name with the rules and weights to use for the calculations.
buildId	The build id for which the calculations will be performed on. If no build id is provided, this parameter defaults to the latest build.

Example Invocation

You could run the following command to invoke the slice and run the calculation:

```
curl "http://localhost:8314/api/v1/services/5dcc38b803c7380f707268b9/slices/bd858e5f.965978?
filterId=2&buildId=docs-2019-11-13&profile=CWE%20Security%20Impact%20-%20Java"
```

If successful, you will receive a response such as the following:

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

Use HTTP(S)-compliance Parameters

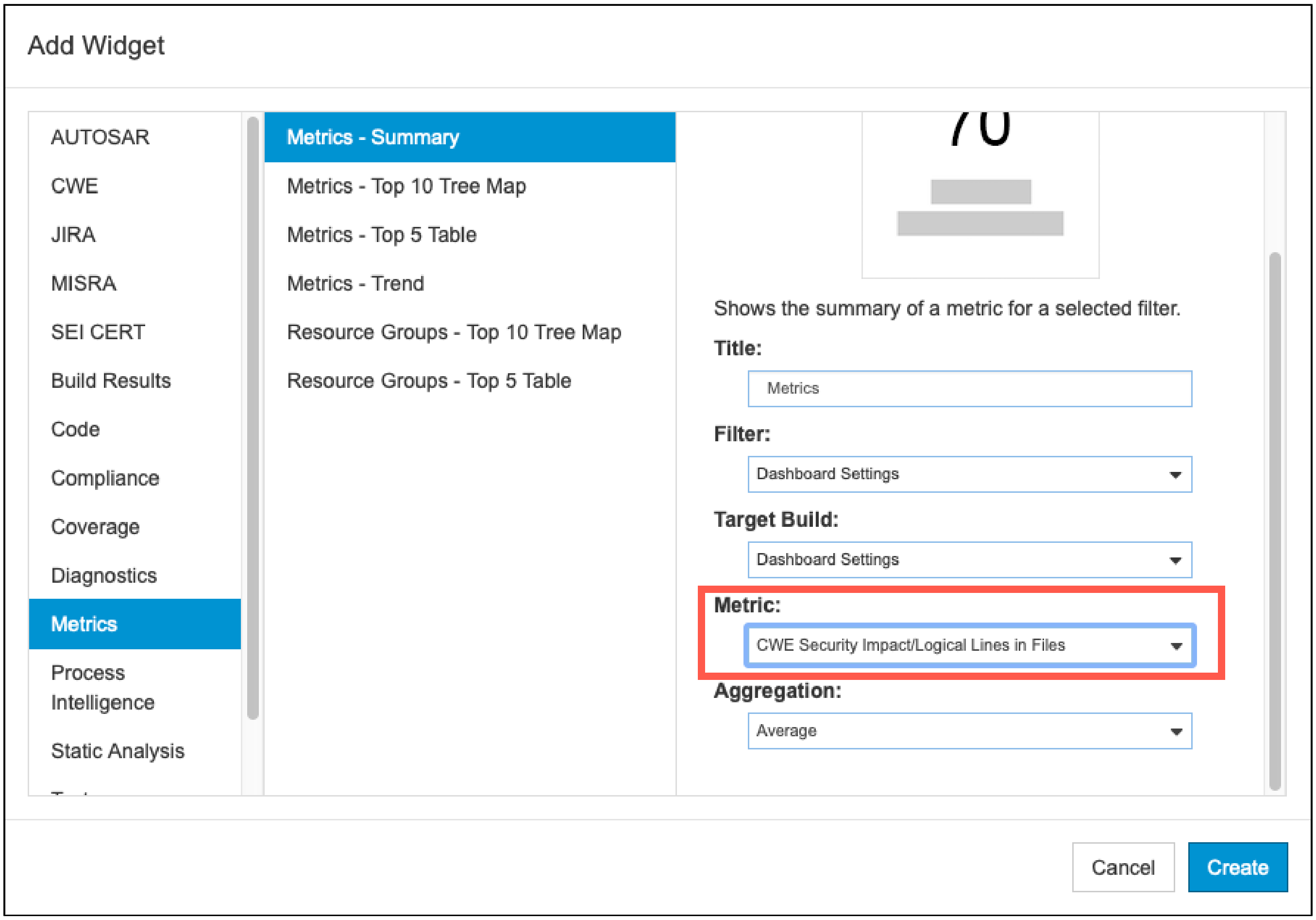
When adding your parameters, be sure to properly encode the string parameter values if they contain spaces, +, /, or any other characters that are not allowed in the HTTP(S) protocol.

The following parameter, for example, is not allowed and will prevent the artifact from functioning correctly: `buildId=c++test`. The encoded parameter would be `buildId=C%2D%2Dtest.`

You can use an encoding tool (e.g., http://www.url-encode-decode.com) to help you properly encode parameters to be compliant with the standard.

Viewing the Data in DTP

After computation has completed and the KPI metric has been reported back to DTP, you can add a Metrics widget to your DTP dashboard and choose a metric associated with your security compliance guidelines from the Metric drop-down menu. The value in the drop-down menu comes from the Metric Name field in the profile (see Configuration). Refer to Adding Widgets for additional information about adding widgets to DTP dashboards.



Shows the summary of a metric for a selected filter.

Title: Metrics

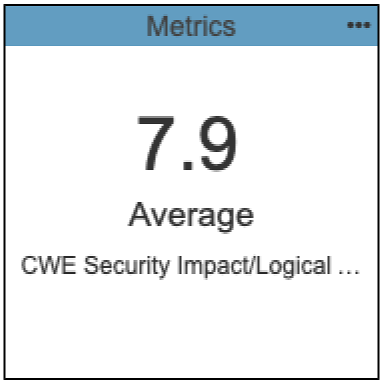
Filter: Dashboard Settings

Target Build: Dashboard Settings

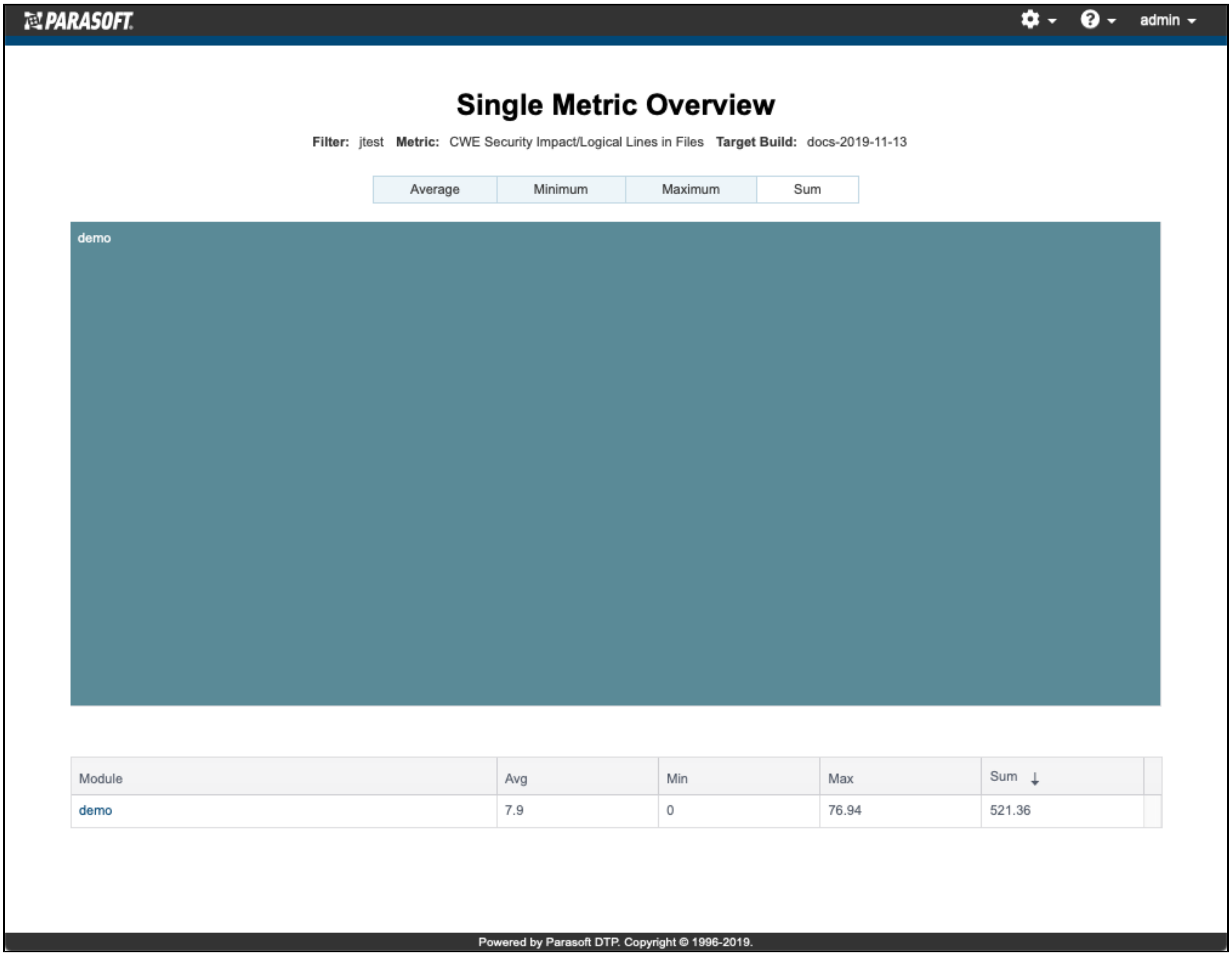
Metric: CWE Security Impact/Logical Lines in Files

Aggregation: Average

The widget will display the metric according to your specifications.



Click on the widget to open the Single Metric Overview Report.



Single Metric Overview

Filter: jtest **Metric:** CWE Security Impact/Logical Lines in Files **Target Build:** docs-2019-11-13

Module	Avg	Min	Max	Sum ↓
demo	7.9	0	76.94	521.36

Powered by Parasoft DTP. Copyright © 1996-2019.

Click on a link in the report to view the data in the Metrics Explorer.