|  |  |  |
| --- | --- | --- |
|  | |  |
|  | | Requirements Document |
|  | |  |
|  | Processor subsystem requirements    **Version: 1.0**  **Last Revised:** September 22, 2010  **Author: Anatoliy Lokshin** | |

Table of Contents

[Feature Overview 4](#_Toc272933344)

[1 Terminology 4](#_Toc272933345)

[2 Product Requirements 4](#_Toc272933346)

[2.1 Functional Area 5](#_Toc272933347)

[2.2 Detailed functional specification 5](#_Toc272933348)

Document Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Author** | **Revision No.** | **Date** | **Description of Change** |
| **Anatoliy Lokshin** | 1.0 | 09/22/2010 | First revision |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Approvers

|  |  |  |
| --- | --- | --- |
| **Name** | **Title** | **Date Approved** |
|  |  |  |
|  |  |  |
|  |  |  |

Reviewers

| **Name** | **Title** | **Date Reviewed** |
| --- | --- | --- |
| **Kyle Quest** |  |  |
|  |  |  |
|  |  |  |

References

|  |  |  |
| --- | --- | --- |
| **Document Name** | **Author** | **Location** |
| MetraTech Security Framework Specification | Kyle Quest | http://seceng.metratech.com/gf/download/docmanfileversion/20/78/MtSecurityFrameworkSpec.doc |
|  |  |  |

# Feature Overview

The Processor Subsystem provides pipeline input parameter data processing services where one or more engines are chained together to combine several processing steps. These processing steps can include different types of the Security Framework security services making complex composite processing possible.

# Terminology

Requirements that are labeled with an R (such as R-100) are Mandatory (or Required) for the targeted release.

Requirements that are labeled with an O (such as O-100) are Optional for the targeted release.

It is left to Engineering’s discretion on whether these can be met for the targeted release but it is imperative that Engineering take these into consideration when making any design or infrastructure decisions. This holds good for any requirement that might be listed under “Future Requirements” section as well.

**(R-1)** It is a Mandatory requirement that the infrastructure developed shall support the ‘feasibility’ of all feature requirements listed in the document – though the delivery of the features themselves could be scheduled for a later time.

# Product Requirements

Processor allows combine several engines those belong to one or several subsystems and then execute the chain as a single engine.

The chain is configured using the common configuration system that currently retrieves the data from XML files.

Processing engines are configurable using a set of engine properties and operation rules. Each operation has an engine to be executed at the processing step and a set of result interpreting rules.

It should be possible to have hybrid processing operation chains where one chain uses another chain for one of its operations.

Operation chains can be further subdivided into two categories:

• Ingress Data Object Filters.

• Output Data Object Filters.

Ingress Data Object Filters process specific data objects coming into a MetraTech application.

Egress Data Object Filters process specific data objects on the way out of a MetraTech application and to the application’s external user.

The rules used in the operation chains define which security services provided by the Security Framework need to be used to perform specific operation filter steps. These operation rules also define the initialization and runtime constrains that need to be used when other Security Framework services are invoked in the filter operations.

Processing engine (operations chain) takes an ApiInput object and returns ApiOutput. When an operation’s engine finds any security issue and throws an exception it can be collected in the processing engine’s result according to configured rules.

Processing engine provides a linear sequence of operations. Operation has a flag indicating whether a chain has to be continued when some issue is found at the step.

As an option another kind of the processing engine can be implemented. Each operation has to branches – for successful result and for unsuccessful one. Each branch is a link to another operation of the same kind. So, an operations tree can be defined.

An especial type of operation provides an endpoint for this kind of engine. It has no branches and only returns a value.

## Functional Area

| Requirement | Story | Constraints |
| --- | --- | --- |
| R-200 – Mandatory requirement | Processor subsystem provides an API for creating engine chains (processor engine) and using them as pipelines for processing input data. |  |
| R-201 – Mandatory requirement | The processor engine consists of a set of operations. Each operation includes an engine (of any subsystem) to be executed at the step and a set of rules, specific for the operation’s type. |  |
| R-202 – Mandatory requirement | Processor engine can be configured in the way to stop on any issue detection or to go through all steps and collect issues detected to process them all later. |  |
| R-203 – Mandatory requirement | Processor engine can use another processor engine as one of its step. |  |
| O-204 – Optional requirement | Processor engine can consist of a net of engine, i.e., each step has two branches for successful step’s execution result and for unsuccessful one. |  |

## Detailed functional specification