



CAN UNCLASSIFIED



DRDC | RDDC  
technologysciencetechnologie

# **Security Posture Metrics (SPM) Command Line CLI (CLI) Software Design Description (SDD) Defence Research Development Canada (DRDC) Cyber Decision Making and Response (CDMR) Project**

Robert Le Van Mao  
Thales Systems Canada

Prepared by:  
Thales Systems Canada  
1 Chrysalis Way  
Ottawa, ON K2G 6P9  
Contractor Document Number: 2268C.005-SPM-CLI-SDD-01 Rev. 01  
PSPC Contract Number: W7714-155991  
Technical Authority: Amaya Arcelus, Defence Scientist  
Contractor's date of publication: November 2017

**Defence Research and Development Canada**  
**Contract Report**  
DRDC-RDDC-2018-C097  
May 2018

CAN UNCLASSIFIED

**IMPORTANT INFORMATIVE STATEMENTS**

This document was reviewed for Controlled Goods by Defence Research and Development Canada (DRDC) using the Schedule to the *Defence Production Act*.

Disclaimer: This document is not published by the Editorial Office of Defence Research and Development Canada, an agency of the Department of National Defence of Canada but is to be catalogued in the Canadian Defence Information System (CANDIS), the national repository for Defence S&T documents. Her Majesty the Queen in Right of Canada (Department of National Defence) makes no representations or warranties, expressed or implied, of any kind whatsoever, and assumes no liability for the accuracy, reliability, completeness, currency or usefulness of any information, product, process or material included in this document. Nothing in this document should be interpreted as an endorsement for the specific use of any tool, technique or process examined in it. Any reliance on, or use of, any information, product, process or material included in this document is at the sole risk of the person so using it or relying on it. Canada does not assume any liability in respect of any damages or losses arising out of or in connection with the use of, or reliance on, any information, product, process or material included in this document.

Unclassified

---

**THALES**

---

**Security Posture Metrics (SPM) Command Line CLI  
(CLI) Software Design Description (SDD)  
Defence Research Development Canada (DRDC)  
Cyber Decision Making and Response (CDMR) Project**

**Contract No.: W7714-155991 Testing of a Network Security Posture Metric**

**Document Control No.: 2268C.005-SPM-CLI-SDD-01 Rev. 01**

**Date: 19 November 2017**

**– RESTRICTIONS ON DISCLOSURE –**

The information contained in this document is proprietary to the Crown. The information disclosed herein, in whole or in part, shall not be reproduced, nor shall it be used by or disclosed to others for any purpose other than explicitly defined in Contract No. W7714-155991 Task 5 Testing of a Network Security Posture Metric. Due diligence shall be exercised in ensuring that the above conditions are strictly adhered to.

---

Unclassified

Security Posture Metric CLI SDD	Unclassified	Date: 19 November 2017
2268C.005-SPM-CLI-SDD-01 Rev. 01		Page 2

# THALES

## Security Posture Metrics (SPM) Command Line CLI (CLI) Software Design Description (SDD) Defence Research Development Canada (DRDC) Cyber Decision Making and Response (CDMR) Project

Contract No.: W7714-155991 Task 5 Testing of a Network Posture Metric

Prepared by:

Approved by:

---

Robert Le Van Mao  
SW Developer

---

Steeve Côté  
Project Manager

Proprietary Information. Use or disclosure of this data is subject to the Restriction of the title page of this document.	Unclassified	THALES
---	--------------	--------

**REVISION HISTORY**

Date	Rev.	Description	Author
19 November 2017	01	Initial release	Robert Le Van Mao

## TABLE OF CONTENTS

	<u>Page</u>
<b>1 INTRODUCTION .....</b>	<b>7</b>
1.1 Overview .....	7
1.2 Scope .....	7
<b>2 REFERENCE DOCUMENTS .....</b>	<b>8</b>
2.1 Government of Canada Documents .....	8
2.2 Thales Group Documents .....	8
2.3 Other Documents .....	8
<b>3 ARCHITECTURE BACKGROUND .....</b>	<b>9</b>
3.1 Overview .....	9
3.2 System Overview .....	9
3.3 Driving Requirements .....	10
3.4 Solution Background .....	15
3.5 Architectural Constraints and Assumptions .....	15
3.6 Architectural Decisions and Rationales .....	16
<b>4 SOFTWARE ARCHITECTURE DESCRIPTION THROUGH VIEWS .....</b>	<b>18</b>
4.1 Functional View .....	18
<b>5 SOFTWARE DESIGN FOR THE COMPONENT .....</b>	<b>19</b>
5.1 Software Components .....	19
<b>6 APPENDIX A: SPM CLI JAVA PROJECTS .....</b>	<b>50</b>
6.1 How to install, build and launch the tool .....	50
<b>7 APPENDIX B: OTHER UTILITIES .....</b>	<b>54</b>
7.1 Extract Data Python Script .....	54
7.2 SPME Launcher .....	55

## LIST OF FIGURES

	<u>Page</u>
Figure 3-1: SPM CLI use cases.....	10
Figure 4-1: Overview of software architecture.....	18
Figure 5-1: SPM CLI main class diagram.....	23
Figure 5-2: Sequence diagram showing the SPM CLI generating metrics. ....	24
Figure 5-3: SPM Engine API Class Diagram.....	26
Figure 5-4: Engine Cache API Class Diagram .....	27
Figure 5-5: Security Posture Metric API Class Diagram .....	28
Figure 5-6: Base Configuration API Class Diagram .....	29
Figure 5-7: Importer API Class Diagram .....	30
Figure 5-8: Metrics Exporter API Class Diagram .....	30
Figure 5-9: SPM CLI Package Diagram .....	31
Figure 5-10: Class Engine Class Diagram. ....	32
Figure 5-11: Class Diagram Showing Various SPM Methods.....	34
Figure 5-12: Attack Graph Class Diagram. ....	43
Figure 5-13: Edge and Vertex Class Diagram.....	44
Figure 5-14: Class Entity Class Diagram.....	45
Figure 5-15: Class AttributeSet Class Diagram.....	46
Figure 5-16: Media Importers Class Diagram. ....	47
Figure 5-17: Media Exporters Class Diagram. ....	49
Figure 6-1: SPM CLI GUI example.....	52
Figure 7-1: SPME Launcher example. ....	55

## LIST OF TABLES

	<u>Page</u>
Table 2-1: Government of Canada Documents.....	8
Table 2-2: Thales Group Documents .....	8
Table 2-3: Published Research Documents .....	8
Table 3-1: SPM CLI Driving Requirements .....	10
Table 3-2: Architectural Constraints and Assumptions .....	16
Table 3-3: Architectural Decisions and Rationale .....	17
Table 5-1: SPM CLI Command Line Tool options.....	19
Table 5-2: SPM CLI CSV output files. ....	20
Table 5-3: SPM CLI CSV file content formats. ....	21

Security Posture Metric CLI SDD	<b>Unclassified</b>	Date: 19 November 2017
2268C.005-SPM-CLI-SDD-01 Rev. 01		Page 6

## **LIST OF ACRONYMS AND ABBREVIATIONS**

### **A**

ARMOUR	Automated Network Defence
ARSM	Attack Resistance Security Metric

### **C**

CLI	Command Line Interface
COTS	Commercial Off The Shelf Software
CSV	Comma-separated Values
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System

### **D**

DRDC	Defence Research and Development Canada
------	---

### **J**

JRE	Java Runtime Environment
-----	--------------------------

### **N**

N/A	Not applicable
NVD	National Vulnerability Database

### **P**

PSM	Probabilistic Security Metric
-----	-------------------------------

### **S**

SDD	Software Design Description
SPM	Security Posture Metric

### **T**

TBD	To be determined
-----	------------------

### **X**

XML	Extensible Markup Language
-----	----------------------------



Security Posture Metric CLI SDD	<b>Unclassified</b>	Date: 19 November 2017
2268C.005-SPM-CLI-SDD-01 Rev. 01		Page 7

## 1 INTRODUCTION

### 1.1 Overview

1.1.1 The purpose of the Defence Research Development Canada (DRDC) Security Posture Metric (SPM) Command Line Interface (CLI) software design description (SDD) is to document the software architecture and software detailed design. The intended audience for this document includes software developers, software testers, configuration management, software integration and software support staff.

### 1.2 Scope

1.2.1 This 2268C.005-SPM-CLI-SDD-01 Rev. 01 is to describe the detailed design of SPM CLI component as a deliverable of DRDC W7714-155991 Task 5 Testing of a Network Security Posture Metric work item. This document includes the description for software structure and software components.

## 2 REFERENCE DOCUMENTS

### 2.1 Government of Canada Documents

**Table 2-1: Government of Canada Documents**

Document Code	Government of Canada Document Reference
[DRDC-RDDC-2015-C336]	Analysis of a Cauldron-based metrics suite and the development of an equivalent set in MulVAL, C. McKenzie, DRDC – Ottawa Research Centre, 2015.

### 2.2 Thales Group Documents

**Table 2-2: Thales Group Documents**

Document Code	Thales Group Document Reference
[83490057-DDQ-IFE-EN]	Java Development Guidelines, Jean-Loic Mauduy, Revision 002, April 3 2013.

### 2.3 Other Documents

**Table 2-3: Published Research Documents**

Document Code	Published Research Document Reference
N/A	Analysis of Attack Graph-based Metrics for Quantification of Network Security, A. Kundu, N. Ghosh, I. Chokski, and S. K. Ghosh, IEEE , 2012.
N/A	An attack graph based probabilistic security metric, L. Wang, T. Islam, T. Long, S. Singhal, and S. Jajodia, Data and Applications Security XXII, pp. 283–296, 2008.
N/A	An Approach for Security Assessment of Network Configurations using Attack Graph, N. Ghosh, and S. K. Ghosh, First International Conference on Networks & Communications, 2009.
N/A	Measuring the Overall Security of Network Configurations Using Attack Graphs, L. Wang, A. Singhal, and S. Jajodia, Data and Applications Security, 2007.

Security Posture Metric CLI SDD	<b>Unclassified</b>	Date: 19 November 2017
2268C.005-SPM-CLI-SDD-01 Rev. 01		Page 9

### **3 ARCHITECTURE BACKGROUND**

#### **3.1 Overview**

3.1.1 SPM CLI provides utilities for the user to generate security posture metrics based on a set of input including:

- (1) NVD CVE (1.2.1) CVSS base scores;
- (2) MulVAL input file (Input.P);
- (3) MulVal network mapping;
- (4) MulVAL generated attack graph; and
- (5) CVSS temporal metrics.

3.1.2 The results returned by SPM CLI will include:

- (1) CVSS-only metrics;
- (2) Current ARMOUR metrics;
- (3) Risk-based metrics; and
- (4) Attack-graph metrics.

#### **3.2 System Overview**

3.2.1 The results returned by SPM CLI are set of metrics that would be used as input to CMDR project MATLAB simulation.

### 3.3 Driving Requirements

3.3.1 The user cases for SPM CLI are shown in Figure 3-1.

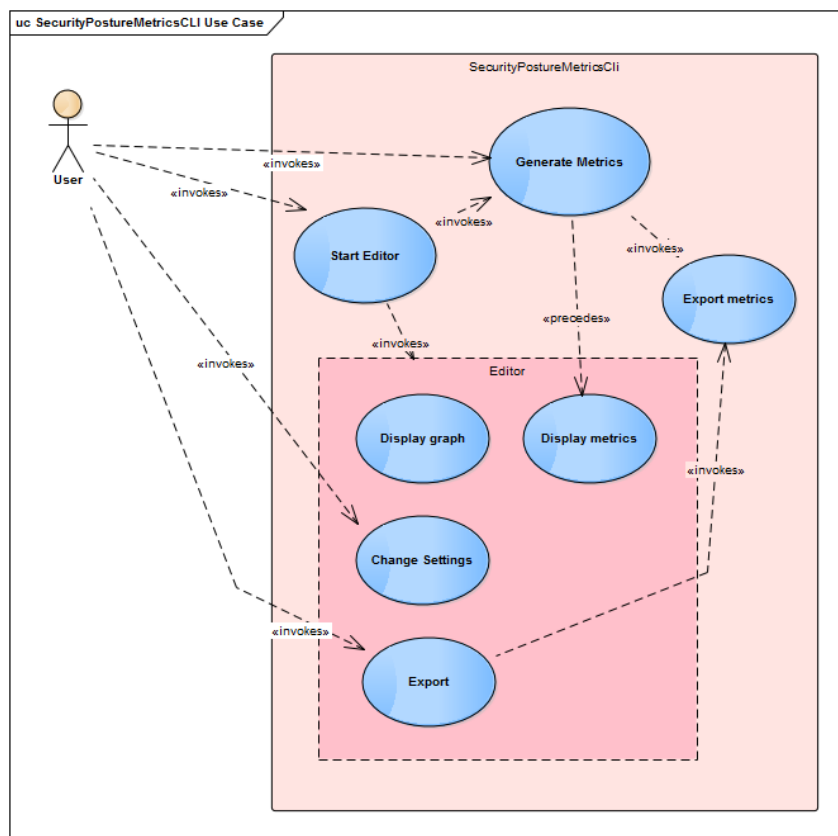


Figure 3-1: SPM CLI use cases

3.3.2 This section lists the functional requirements that drive the architectural design. The requirements are summarized in Table 3-1 below.

Table 3-1: SPM CLI Driving Requirements

Requirement	Description
TBD	SPM CLI shall allow the user to generate SPM using the following methods: <ul style="list-style-type: none"> <li>(1) CVSS-only metrics;</li> <li>(2) Current ARMOUR metrics;</li> <li>(3) Risk-based metrics; and</li> <li>(4) Attack-graph based metrics.</li> </ul>
TBD	SPM CLI shall compute CVSS-only metrics for: <ul style="list-style-type: none"> <li>(1) Host-level; and</li> <li>(2) Network-level.</li> </ul>

Security Posture Metric CLI SDD	Unclassified	Date: 19 November 2017
2268C.005-SPM-CLI-SDD-01 Rev. 01		Page 11

Requirement	Description
TBD	<p>For host-level CVSS-only metrics, SPM CLI shall use the following algorithm:</p> $SPM\_CVSS_{avg_{host}} = 1 - \frac{1}{10} average_{host}(CVSSBase)$ <p>Where <math>average_{host}(CVSSBase)</math> is the average CVSS base scores at host level.</p>
TBD	<p>For network-level CVSS-only metrics, SPM CLI shall use the following algorithm:</p> $SPM\_CVSS_{avg_{network}} = 1 - \frac{1}{10} average_{network}(CVSSBase)$ <p>Where <math>average_{network}(CVSSBase)</math> is the average CVSS base scores at network level.</p>
TBD	<p>SPM CLI shall compute current ARMOUR metrics for:</p> <ul style="list-style-type: none"> <li>(1) Host-level; and</li> <li>(2) Network-level.</li> </ul>
TBD	<p>For host-level ARMOUR metrics, SPM CLI shall use the following algorithm:</p> $SPM\_HC_{host} = \frac{1}{\sum_{vu \ln=1}^V \frac{1}{(1 - \frac{CVSSBase_{host,vu \ln}}{11})}}$ <p>Where CVSSBase are CVSS base scores of a vulnerability associated to the given host.</p>
TBD	<p>For network-level ARMOUR metrics, SPM CLI shall use the following algorithm:</p> $SPM\_HC_{network} = \min_{network}(SPM\_HC)$ <p>Where SPM_HC are the metrics of the hosts on the given network.</p>
TBD	<p>SPM CLI shall compute risk-based metrics for:</p> <ul style="list-style-type: none"> <li>(1) Host-level; and</li> <li>(2) Network-level.</li> </ul>

Requirement	Description
TBD	<p>For host-level risk-based metrics, SPM CLI shall use the following algorithm:</p> $\text{SPM-RB}_h = 1 - \left[ \frac{1}{3}R_{\text{known}} + \frac{1}{3}R_{\text{unknown}} + \frac{1}{3}P_{\text{safeguards}} \right], \text{ where:}$ $R_{\text{known}} = \frac{\#Services_{\text{vulnerable}}}{\#Services_{\text{total}}} + \frac{1}{10} \text{average}_{\text{host}}(\text{CVSSExp}) + \frac{1}{10} \text{average}_{\text{host}}(\text{CVSSImp})$ $R_{\text{unknown}} = \frac{\#Vectors_{\text{possible,cross-domain}}}{\#Vectors_{\text{possible}}} + \frac{\#ServicePorts_{\text{open}}}{\#ServicePorts_{\text{total}}} + \frac{\#ServicePorts_{\text{internet-facing}}}{\#ServicePorts_{\text{total}}}$ $P_{\text{safeguards}} = \frac{\#Vectors_{\text{AttackGraph}}}{\#Vectors_{\text{possible}}}$ <p>Where CVSSImp are CVSS impact scores, CVSSExp are CVSS exploitability scores.</p>
TBD	<p>For network-level risk-based metrics, SPM CLI shall use the following algorithm:</p> $\text{SPM-RB}_N = 1 - \left[ \frac{1}{3}R_{\text{known}} + \frac{1}{3}R_{\text{unknown}} + \frac{1}{3}P_{\text{safeguards}} \right], \text{ where:}$ $R_{\text{known}} = \frac{\#Services_{\text{vulnerable}}}{\#Services_{\text{total}}} + \frac{1}{10} \text{average}_{\text{Network}}(\text{CVSSExp}) + \frac{1}{10} \text{average}_{\text{Network}}(\text{CVSSImp})$ $R_{\text{unknown}} = \frac{\#Vectors_{\text{possible,cross-domain}}}{\#Vectors_{\text{possible}}} + \frac{\#ServicePorts_{\text{open}}}{\#ServicePorts_{\text{total}}} + \frac{\#ServicePorts_{\text{internet-facing}}}{\#ServicePorts_{\text{total}}}$ $P_{\text{safeguards}} = \frac{\#Vectors_{\text{AttackGraph}}}{\#Vectors_{\text{possible}}}$ <p>Where CVSSImp are CVSS impact scores, CVSSExp are CVSS exploitability scores.</p>
TBD	<p>SPM CLI shall compute attack-graph metrics for:</p> <ol style="list-style-type: none"> <li>(1) Host-level; and</li> <li>(2) Network-level.</li> </ol>
TBD	<p>For host-level attack-graph metrics, SPM CLI shall compute the following metrics:</p> <ol style="list-style-type: none"> <li>(1) Number of paths;</li> <li>(2) Shortest path;</li> <li>(3) Mean path length;</li> <li>(4) Normalized mean path length;</li> <li>(5) Probabilistic security metric; and</li> <li>(6) Attack resistant security metric.</li> </ol>

Security Posture Metric CLI SDD	Unclassified	Date: 19 November 2017
2268C.005-SPM-CLI-SDD-01 Rev. 01		Page 13

Requirement	Description
TBD	For host-level number of paths, SPM CLI shall use the following algorithm: $NP_{host} = \#_{host} paths$
TBD	For host-level shortest path, SPM CLI shall use the following algorithm: $SP_{host} = \min_{host} \{length(path_1), length(path_2), \dots length(path_{NP_{host}})\}$
TBD	For host-level mean path length, SPM CLI shall use the following algorithm: $MPL_{host} = average_{host} \{length(path_1), length(path_2), \dots length(path_{NP_{host}})\}$
TBD	For host-level normalized mean path length, SPM CLI shall use the following algorithm: $NMPL_{host} = \frac{1}{NP_{host}} average_{host} \{length(path_1), length(path_2), \dots length(path_{NP_{host}})\}$
TBD	For host-level probabilistic security metric, SPM CLI shall use the following algorithm: $PSM_{host} = Probability_{host}(AttackGraph_{host})$
TBD	For host-level probabilistic security metric, SPM CLI shall use the approach described in [Other Documents].
TBD	For host-level attack resistant security metric, SPM CLI shall use the following algorithm: $AR_{host} = Resistance_{host}(AttackGraph_{host})$
TBD	For host-level attack resistant security metric, SPM CLI shall use the approach described in [Other Documents].
TBD	For network-level attack-graph metrics, SPM CLI shall compute the following metrics: <ul style="list-style-type: none"> <li>(1) Number of paths;</li> <li>(2) Shortest path;</li> <li>(3) Mean path length;</li> <li>(4) Normalized mean path length;</li> <li>(5) Probabilistic security metric; and</li> <li>(6) Attack resistant security metric.</li> </ul>
TBD	For network-level number of paths, SPM CLI shall use the following algorithm: $NP_{network} = \#_{network} paths$
TBD	For network-level shortest path, SPM CLI shall use the following algorithm: $SP_{network} = \min_{network} \{length(path_1), length(path_2), \dots length(path_{NP_{network}})\}$

Requirement	Description
TBD	For network-level mean path length, SPM CLI shall use the following algorithm: $MPL_{network} = average_{network}\{length(path_1), length(path_2), \dots length(path_{NP_{network}})\}$
TBD	For network-level normalized mean path length, SPM CLI shall use the following algorithm: $NMPL_{network} = \frac{1}{NP_{network}} average_{network}\{length(path_1), length(path_2), \dots length(path_{NP_{network}})\}$
TBD	For network-level probabilistic security metric, SPM CLI shall use the approach described in [Other Documents].
TBD	For network-level probabilistic security metric, SPM CLI shall use the following algorithm: $PSM_{network} = Probability_{network}(AttackGraph_{network})$
TBD	For network-level attack resistant security metric, SPM CLI shall use the following algorithm: $AR_{network} = Resistance_{network}(AttackGraph_{network})$
TBD	For network-level attack resistant security metric, SPM CLI shall use the approach described in [Other Documents].
TBD	On start-up, SPM CLI shall provide option to import CVSS base scores for use in SPM calculations from:  (1) NVD CVE (1.2.1) XML feed as pre-fetched XML files.
TBD	On start-up, SPM CLI shall provide option to import CVSS temporal metrics for use in SPM calculations from:  (1) A CSV file.
TBD	SPM CLI shall be able to generate SPMs for:  (1) Output from a single run of MulVAL; or (2) Output from multiple runs of MulVAL.
TBD	For output from a single run of MulVAL, SPM CLI shall import the following files:  (1) Input from file 'Input.P' in sub-folder 'source_data'; (2) Network configuration from XML file 'network_mapping.xml' in sub-folder 'network'; (3) Edges from CSV file 'ARCS.csv' in sub-folder 'attack_graph'; and (4) Vertices from CSV file 'VERTICES.csv' in sub-folder 'attack_graph'.
TBD	For output of multiple runs of MulVAL, SPM CLI shall import and compute SPMs for each set of data and repeat the process for all runs.



Requirement	Description
TBD	SPM CLI shall provide option(s) for the user to specify Internet-facing network(s) in the configuration file.
TBD	SPM CLI shall provide option(s) for the user to specify source vertices either: <ul style="list-style-type: none"> <li>(1) In the configuration file; or</li> <li>(2) As command line arguments.</li> </ul>
TBD	SPM CLI shall provide option(s) for the user to specify goal vertices either in: <ul style="list-style-type: none"> <li>(1) In the configuration file; or</li> <li>(2) As the command line arguments.</li> </ul>
TBD	If no option for source vertices are specified, SPM CLI shall derive the source vertices from the clause 'attackerLocated()' and its arguments from the Input.P file
TBD	If no option for goal vertices are specified, SPM CLI shall derive the goals vertices from the clause 'attackGoals()' and its arguments from the Input.P file
TBD	Although desirable, support for concurrent operations shall not constraint the design of SPM CLI.

### 3.4 Solution Background

3.4.1 SPM CLI software architecture described in this document should satisfy the following identifiable quality attributes:

- (1) Modifiability;
- (2) Capacity;
- (3) Portability;
- (4) Usability; and
- (5) Testability.

### 3.5 Architectural Constraints and Assumptions

3.5.1 This section lists the architecturally significant constraints and assumptions, as input to the architectural design. The issues are summarized in Table 3-2 below.

**Table 3-2: Architectural Constraints and Assumptions**

Constraint/Assumption	Description
Baseline	SPM CLI must run on all platforms supported by Oracle Java Runtime Environment (JRE) 1.8 or better.
Offline Use	SPM CLI must run on supported platforms without requiring an established connection to the internet.
Command Line Use	SPM CLI must provide an executable JAR that could be invoked from Windows command prompt.
Graphical User CLI	SPM CLI would provide graphical user interface for visualization.
Use of CVSS Base Scores from NVD CVE 1.2.1 XML files only	SPM CLI will use CVSS base scores from the NVD CVE 1.2.1 XML feed. The XML files will be downloaded from NIST site and included as data element into the build (sub folder ./data/nvd-cve-1.2.1).
CVSS Temporal Metrics file(s) will be created manually.	CVSS Temporal Metrics (nor CVSS Environmental Metrics) are not supported by NIST feeds and are not available in public domains. Thus, SPM CLI will import them from a CSV file on start up. If no option will be specified, a default CSV file (./data/cvss-extensions/temporalmetrics.csv), included as data element in the build will be used by default.
For attack graph based computations, If not explicitly specified by the user, the default source vertices will be automatically derived using built-in criteria.	<p>If not explicitly specified by the user, SPM CLI will deduce the default source vertices to be interactions that are associated to label 'attackerLocated()' and that applied to:</p> <ol style="list-style-type: none"> <li>1. Internet-Facing hosts if the flag "StartPointInternetFacingOnly" (config.xml) is set to "true"; or</li> <li>2. Any if the flag "StartPointInternetFacingOnly" (config.xml) is set to "false".</li> </ol>
For attack graph based computations, If not explicitly specified by the user, the default goal vertices will be automatically derived using built-in criteria.	If not explicitly specified by the user, SPM CLI will deduce the default goal vertices to be interactions that match the argument of 'attackGoals()' in the Input.P file and that are leaf.

### 3.6 Architectural Decisions and Rationales

3.6.1 This section provides a rationale for the major design decisions embodied by the software architecture. It describes any design approaches applied to the software architecture, including the use of design principles, architectural styles or design patterns, when the scope of those approaches transcends any single architectural view. The section also provides a rationale for the selection of those approaches.

**Table 3-3: Architectural Decisions and Rationale**

Decision	Description
Modularity and Extensibility	SPM CLI is designed in a modular and extensive manner. This design decision allows future capabilities to be added in a modular fashion with minimal impact on the existing implementation. SPM CLI design will maximize the decoupling between packages.
Maximize the use of encapsulation with specialization	The intent of this design decision is to have clear partition of subject matters, reduce class size and code complexity, and ease maintenance. SPM CLI design process will apply, whenever applicable, the single responsibility principle to the design of the Java classes.
Maximize the use of filter pattern to process graph	The intent of this design decision is to be able to re-use filters for similar searches.
Maximize the use of open-source COTS	The intent of this design decision is to speed up the development process and minimize risks associated to developing new software from the base up. The Java Graph Library JGraphT will be used to process graphs. The open-source package JGraphX will be used for visualization.
All Features Derived From Base Entity	The base entity will have a generic attribute set. This design decision was aimed to allow the applications to be able to programmatically add new attributes (criteria, categories, etc.) without having to modify and recompile the code.
Lightweight Application	SPM CLI is a lightweight application that requires minimal computing resources, uses open-source third-party software components and has minimal to no COTS pre-requisites. This design decision was made with the intent to minimize software development and deployment costs.
File System Repository	The initial design of SPM CLI uses the operating system file system for input and output instead of a database. This design decision reduces the complexity of the software application and re-uses the machine readable published plan file format as the data storage format. .
No Security and Validation	SPM CLI will not validate the integrity of the input data. The intent of this design decision is to simplify the implementation by shifting the responsibility of ensuring the validity and integrity (no corruption) of all input data to the operator.

4 SOFTWARE ARCHITECTURE DESCRIPTION THROUGH VIEWS

4.1 Functional View

4.1.1 The block diagram view of SPM CLI software architecture is shown in Figure 4-1.

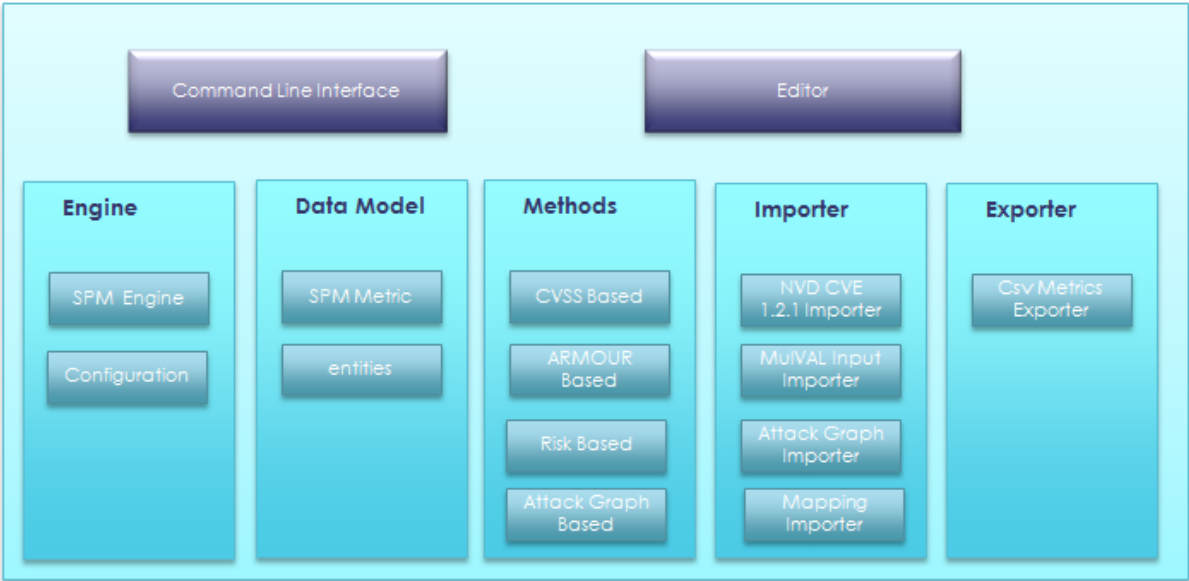


Figure 4-1: Overview of software architecture

## 5 SOFTWARE DESIGN FOR THE COMPONENT

### 5.1 Software Components

#### 5.1.1 SPM CLI Software Component

##### 5.1.1.1 Functionality Summary

5.1.1.1.1 The command line tool will be executed as an executable jar, and will include the options for the user to:

- (1) Specify the configuration XML file. The supported input file formats are included;
- (2) Specify the output files location and format. The following output formats are included:
  - a. CSV.
- (3) Select to display the results on standard output (verbose).

##### 5.1.1.2 SPM CLI Command Line Tool Options

5.1.1.2.1 The command line tool could be executed from Windows command tool as shown in the example below:

```
>java -jar SecurityPostureMetricsCLI-1.0.0.jar reference_network1
>java -jar SecurityPostureMetricsCLI-1.0.0.jar -o c:/temp reference_network2
>java -jar SecurityPostureMetricsCLI-1.0.0.jar -gs "attackerLocated('192.168.100.10')"-gg
"attackGoal(execCode('192.168.2.10',_))."-o c:/temp reduce_iter1
>java -jar SecurityPostureMetricsCLI-1.0.0.jar -g reference_network1
```

5.1.1.2.2 The standard invocation is:

SecurityPostureMetricsCLI-<version>.jar <options> <input folder>

Where the arguments are:

**Table 5-1: SPM CLI Command Line Tool options**

Argument	Description	Default value If option is not specified
<input folder>	The name of the folder that contains the input data (graphs, input P file, etc.) If the folder contains other subfolders, SPM CLI will verify each of those subfolders to find the ones that contain the valid input data and will process them accordingly.	N/A
<b>Options</b>		
-c <configuration file>	This option specifies the XML file that contains the custom configurations.	./config.xml
-gs <source specification>	This option specifies the criteria to select the source vertices.	Source(s) from input file.

-gg <goal specification>	This option specifies the criteria to select the goal vertices.	Attack goal(s) from input file.
-n <NVD CVE directory>	This option specifies the folder containing the NVD CVE XML files.	./data/nvd-cve-1.2.1
-o <output directory>	This option specifies the directory that contains the metric files.	./spm
-t <temporal metrics file>	This option specifies the name of the CSV file that contains temporal metrics.	./data/cvss-extensions/temporalmetrics.csv
-g	Start in editor mode.	Command line mode only
-v	This option specifies verbose is true.	False.
-h	This option print usage.	N/A

### 5.1.1.3 SPM CLI Output Format

#### 5.1.1.3.1 SPM CLI CSV Files

5.1.1.3.1.1 All output from SPM CLI are in a folder specified in the configuration file.

5.1.1.3.1.2 The output of the CSV file generated from the command line tool will contain the data listed in Table 5-2.

**Table 5-2: SPM CLI CSV output files.**

File	Data
CvssBased_HostLevel.csv	CVSS-only SPM at host-level.
CvssBased_NetworkLevel.csv	CVSS-only SPM at network-level.
ArmourBased_HostLevel.csv	Current ARMOUR SPMs at host-level.
ArmourBased_NetworkLevel.csv	Current ARMOUR SPMs at network-level.
RiskBased_HostLevel.csv	Risk-based SPMs at host-level.
RiskBased_NetworkLevel.csv	Risk-based SPMs at network-level.
AgBased_HostLevel.csv	Attack graph based SPMs (non PSMs and ARSMs) at host-level.
AgBased_NetworkLevel.csv	Attack graph based SPMs (non PSMs and ARSMs) at network-level.
AgBasedPsm_HostLevel.csv	Attack graph based PSM at host-level.
AgBasedPsm_NetworkLevel.csv	Attack graph based PSM at network-level.
AgBasedArsm_HostLevel.csv	Attack graph based ARSM at host-level.
AgBasedArsm_NetworkLevel.csv	Attack graph based ARSM at network-level.

### 5.1.1.3.2 SPM CLI CSV File Content Formats

5.1.1.3.2.1 The formats for each CSV file are described in Table 5-3.

**Table 5-3: SPM CLI CSV file content formats.**

File	Field Descriptions
CvssBased_HostLevel.csv	<b>Address:</b> host IP(v4) address <b>CVSS-only SPM (HL):</b> SPM value.
CvssBased_NetworkLevel.csv	<b>Address:</b> network IP(v4) address <b>CVSS-only SPM (HL):</b> SPM value.
ArmourBased_HostLevel.csv	<b>Address:</b> host IP(v4) address <b>ARMOUR-only SPM (HL):</b> SPM value.
ArmourBased_NetworkLevel.csv	<b>Address:</b> network IP(v4) address <b>ARMOUR-only SPM (HL):</b> SPM value.
RiskBased_Absolute_HostLevel.csv	<b>Address:</b> host IP(v4) address <b>CVSS Impact sub-score (HL):</b> CVSS impact sub-score from NVD CVE. <b>CVSS Exploit sub-score (HL):</b> CVSS exploitability sub-score from NVD CVE. <b>All services (HL):</b> number of services. <b>All vulnerable services (HL):</b> number of vulnerable services. <b>R known (HL):</b> computed R known value. <b>All service ports (HL):</b> number of service ports. <b>Open service ports (HL):</b> number of open service ports. <b>Internet-facing service ports (HL):</b> number of Internet facing service ports. <b>Possible vectors (HL):</b> number of possible attack vectors from attack graph. <b>Possible vectors cross-domain (HL):</b> number of possible cross-domain vectors. <b>R unknown (HL):</b> computed R unknown value. <b>All attack graph possible vectors (HL):</b> number of attack vectors targeting specific host. <b>P Safeguard (HL):</b> P safeguard value. <b>Risk-based SPM (HL):</b> SPM value.
RiskBased_Absolute_NetworkLevel.csv	<b>Address:</b> network IP(v4) address <b>CVSS Impact sub-score (NL):</b> CVSS impact sub-score from NVD CVE. <b>CVSS Exploit sub-score (NL):</b> CVSS exploitability sub-score from NVD CVE. <b>All services (NL):</b> number of services. <b>All vulnerable services (NL):</b> number of vulnerable services. <b>R known (NL):</b> computed R known value. <b>All service ports (NL):</b> number of service ports. <b>Open service ports (NL):</b> number of open service ports. <b>Internet-facing service ports (NL):</b> number of Internet facing service ports. <b>Possible vectors (NL):</b> number of possible attack vectors from attack graph. <b>Possible vectors cross-domain (NL):</b> number of possible cross-domain vectors. <b>R unknown (NL):</b> computed R unknown value. <b>All attack graph possible vectors (NL):</b> number of possible vectors from attack graph. <b>P Safeguard (NL):</b> P safeguard value.

Security Posture Metric CLI SDD	Unclassified	Date: 19 November 2017
2268C.005-SPM-CLI-SDD-01 Rev. 01		Page 22

	<b>Risk-based SPM (NL):</b> SPM value.
RiskBased_Relative_HostLevel.csv	<b>Address:</b> host IP(v4) address <b>CVSS Impact sub-score (HL):</b> CVSS impact sub-score from NVD CVE. <b>CVSS Exploit sub-score (HL):</b> CVSS exploitability sub-score from NVD CVE. <b>All services (HL):</b> number of services. <b>All vulnerable services (HL):</b> number of vulnerable services. <b>R known (HL):</b> computed R known value. <b>All service ports (HL):</b> number of service ports. <b>Open service ports (HL):</b> number of open service ports. <b>Internet-facing service ports (HL):</b> number of Internet facing service ports. <b>Possible vectors (HL):</b> number of possible attack vectors from attack graph relative to given host. <b>Possible vectors cross-domain (HL):</b> number of possible cross-domain vectors targeting specific host. <b>R unknown (HL):</b> computed R unknown value. <b>All attack graph possible vectors (HL):</b> number of attack vectors targeting specific host. <b>P Safeguard (HL):</b> P safeguard value. <b>Risk-based SPM (HL):</b> SPM value.
RiskBased_Relative_NetworkLevel.csv	<b>Address:</b> network IP(v4) address <b>CVSS Impact sub-score (NL):</b> CVSS impact sub-score from NVD CVE. <b>CVSS Exploit sub-score (NL):</b> CVSS exploitability sub-score from NVD CVE. <b>All services (NL):</b> number of services. <b>All vulnerable services (NL):</b> number of vulnerable services. <b>R known (NL):</b> computed R known value. <b>All service ports (NL):</b> number of service ports. <b>Open service ports (NL):</b> number of open service ports. <b>Internet-facing service ports (NL):</b> number of Internet facing service ports. <b>Possible vectors (NL):</b> number of possible attack vectors from attack graph relative to given network. <b>Possible vectors cross-domain (NL):</b> number of possible cross-domain vectors targeting specific network. <b>R unknown (NL):</b> computed R unknown value. <b>All attack graph possible vectors (NL):</b> number of possible vectors from attack graph targeting specific network. <b>P Safeguard (NL):</b> P safeguard value. <b>Risk-based SPM (NL):</b> SPM value.
AgBased_HostLevel.csv	<b>Address:</b> host IP(v4) address. <b>Number of paths (HL):</b> number of paths. <b>Shortest path (HL):</b> shortest path length. <b>Mean path length (HL):</b> mean path length. <b>Normalized mean path length (HL):</b> normalized path length.
AgBased_NetworkLevel.csv	<b>Address:</b> network IP(v4) address. <b>Number of paths (NL):</b> number of paths. <b>Shortest path (NL):</b> shortest path length. <b>Mean path length (NL):</b> mean path length. <b>Normalized mean path length (NL):</b> normalized path length.
AgBasedPsm_HostLevel.csv	<b>Address:</b> host IP(v4) address. <b>Service:</b> leaf vulnerable service name. <b>SPM Value:</b> computed PSM value.



AgBasedPsm_NetworkLevel.csv	<b>Address:</b> network IP(v4) address. <b>Service:</b> leaf vulnerable service name on the network. <b>SPM Value:</b> computed PSM value.
AgBasedArsm_HostLevel.csv	<b>Address:</b> host IP(v4) address. <b>Service:</b> leaf vulnerable service name. <b>SPM Value:</b> computed ARSM value.
AgBasedArsm_NetworkLevel.csv	<b>Address:</b> network IP(v4) address. <b>Service:</b> leaf vulnerable service name on the network. <b>SPM Value:</b> computed ARSM value.

5.1.1.3.2.2 An example of generated CSV file containing CVSS-only SPMs is shown below:

Address	CVSS-only SPM (HL)
192.168.1.20	0.2773
192.168.1.30	0.3748
192.168.100.10	0.3642
192.168.100.20	0.3696
192.168.2.10	0.064

#### 5.1.1.4 SPM CLI Structure Design Description

5.1.1.4.1 Figure 5-1 shows the class diagram of the tool. The tool class could be exported as executable jar or invoked from the IDE (eclipse).

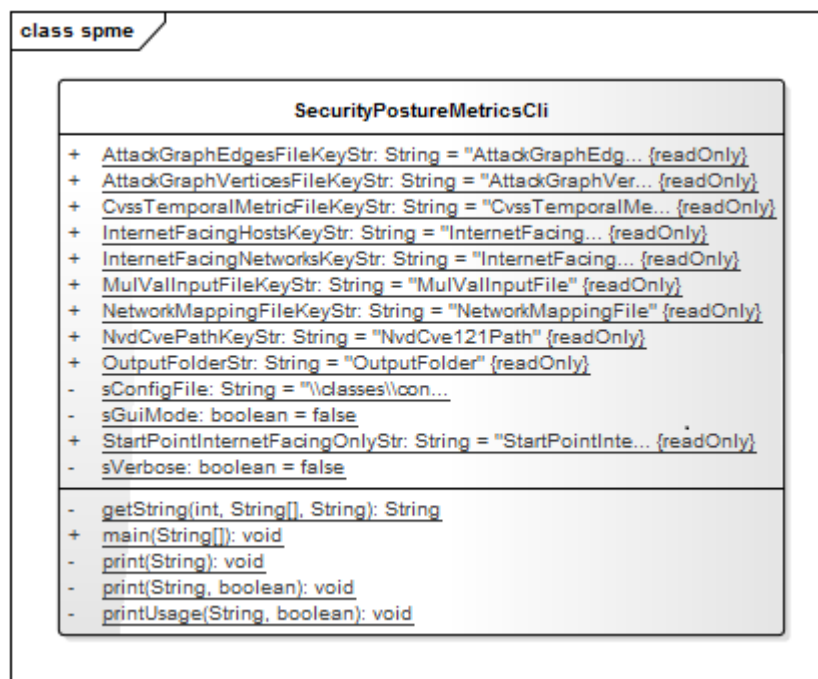


Figure 5-1: SPM CLI main class diagram

5.1.1.4.2 Figure 5-2 shows the sequence diagram of the tool while generating SPMs and exporting the results into CSV files.

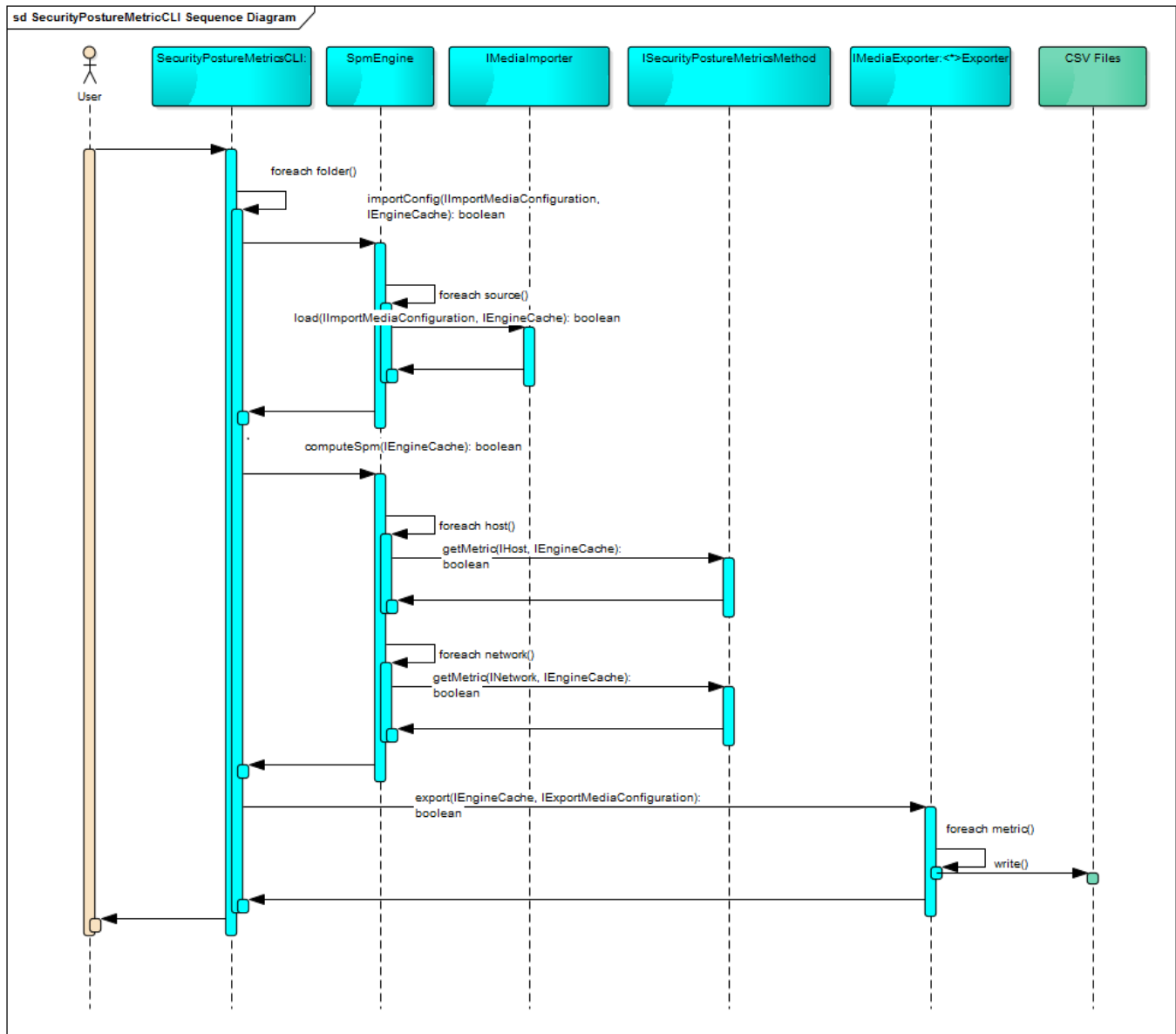


Figure 5-2: Sequence diagram showing the SPM CLI generating metrics.

Security Posture Metric CLI SDD	<b>Unclassified</b>	Date: 19 November 2017
2268C.005-SPM-CLI-SDD-01 Rev. 01		Page 25

## **5.1.2 SPM CLI Software Component**

### **5.1.2.1 Functionality Summary**

5.1.2.1.1 SPM CLI API functionalities include:

- (1) To provide the client application(s) with functions to generate SPMs from a given set of input data;
- (2) To provide the client application with a function to select one of the following methodologies:
  - a. CVSS-only method;
  - b. Current ARMOUR method;
  - c. Risk-based method; and
  - d. Attack graph based methods.
- (3) To provide the client application(s) with a function to import data from input files. The supported input file formats include:
  - a. CSV files as described in section 5.
- (4) To provide the client application(s) with a function to export the results to output file.

## **5.1.2.2 SPM CLI API**

### **5.1.2.2.1 Interface ISpmEngine**

5.1.2.2.1.1 The interface ISpmEngine defines the following functions:

- (1) Load data using a given import configuration;
- (2) Compute SPMs and save the results to the cache; and
- (3) Export the SPMs to media using a given export configuration.

5.1.2.2.2 The class diagram of ISpmEngine is shown in Figure 5-3.

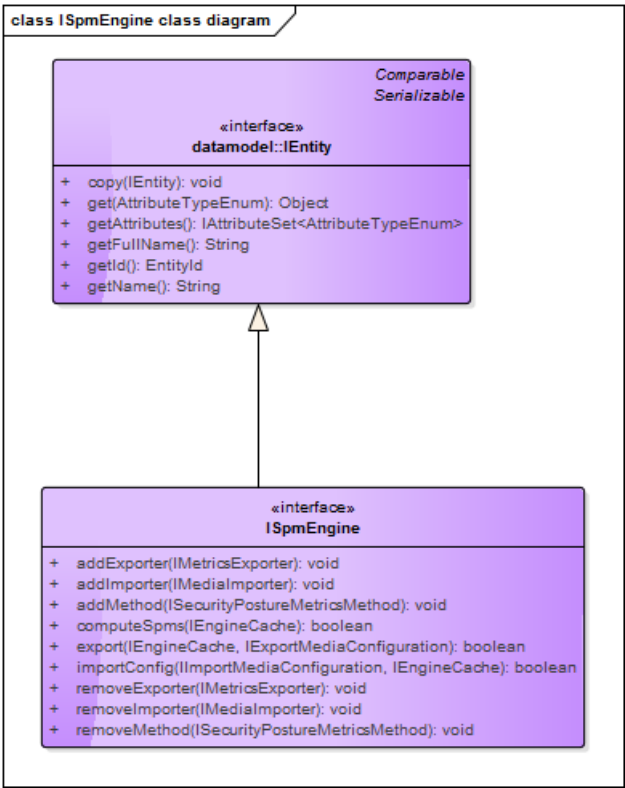


Figure 5-3: SPM Engine API Class Diagram

### 5.1.2.2.3 Interface IEngineCache

5.1.2.2.3.1 The interface IEngineCache defines a place holder for all configuration data and calculation results.

5.1.2.2.3.2 The class diagram of IEngineCache is shown in Figure 5-4.

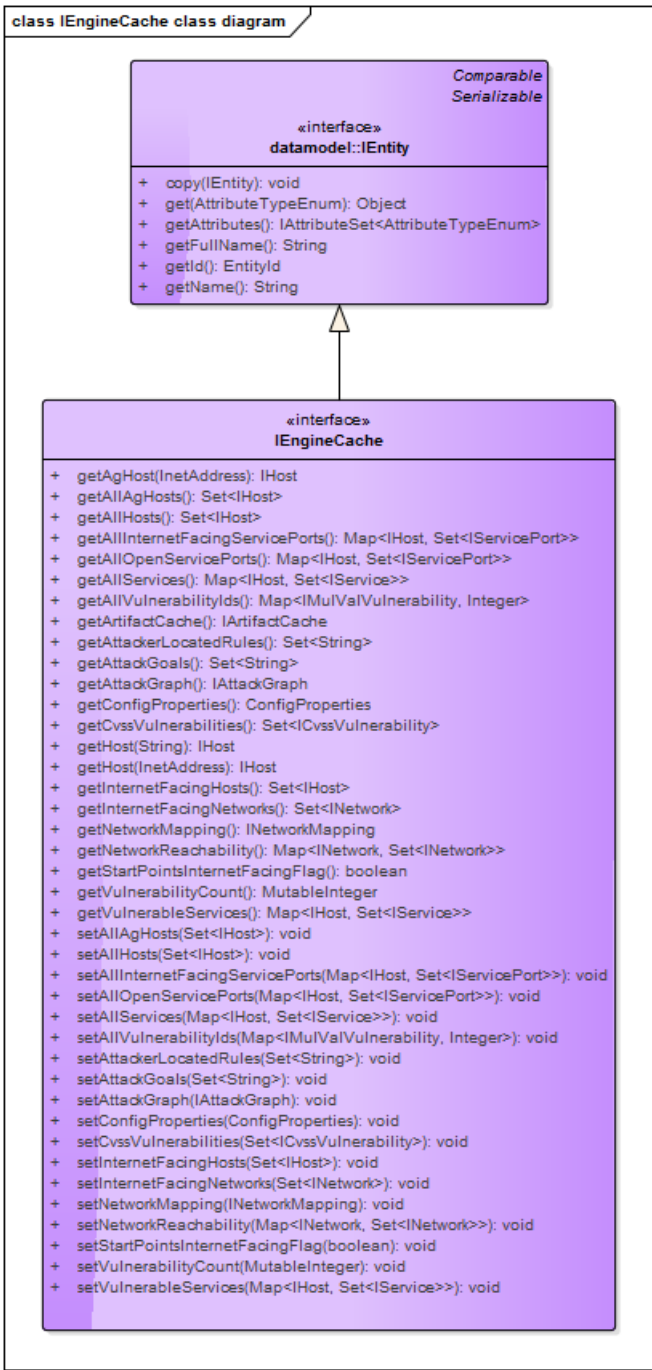


Figure 5-4: Engine Cache API Class Diagram

5.1.2.2.4 Interface ISecurityPostureMetric

5.1.2.2.4.1 The interface ISecurityPostureMetric defines the accessors to the computed SPM values.

5.1.2.2.4.2 The class diagram of ISecurityPostureMetric is shown in Figure 5-5.

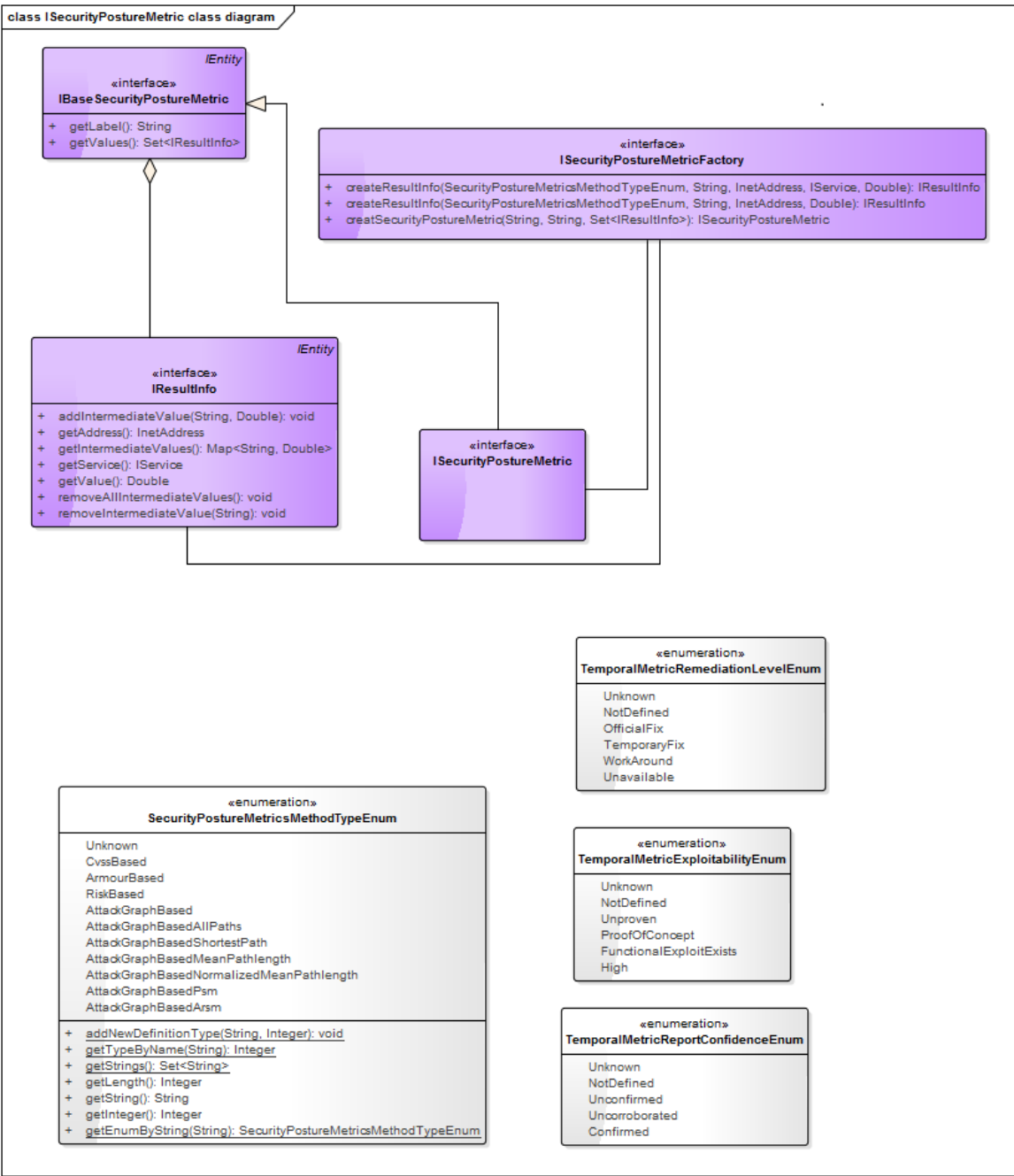


Figure 5-5: Security Posture Metric API Class Diagram

### 5.1.2.2.5 Interface IBaseConfiguration

5.1.2.2.5.1 The interface IBaseConfiguration defines the accessors for the following items:

- (1) Path to the NVD CVE XML files folder;
- (2) Path to the network mapping XML file; and
- (3) Path to the MulVAL input file.

5.1.2.2.6 The class diagram of IBaseConfiguration is shown in Figure 5-6.

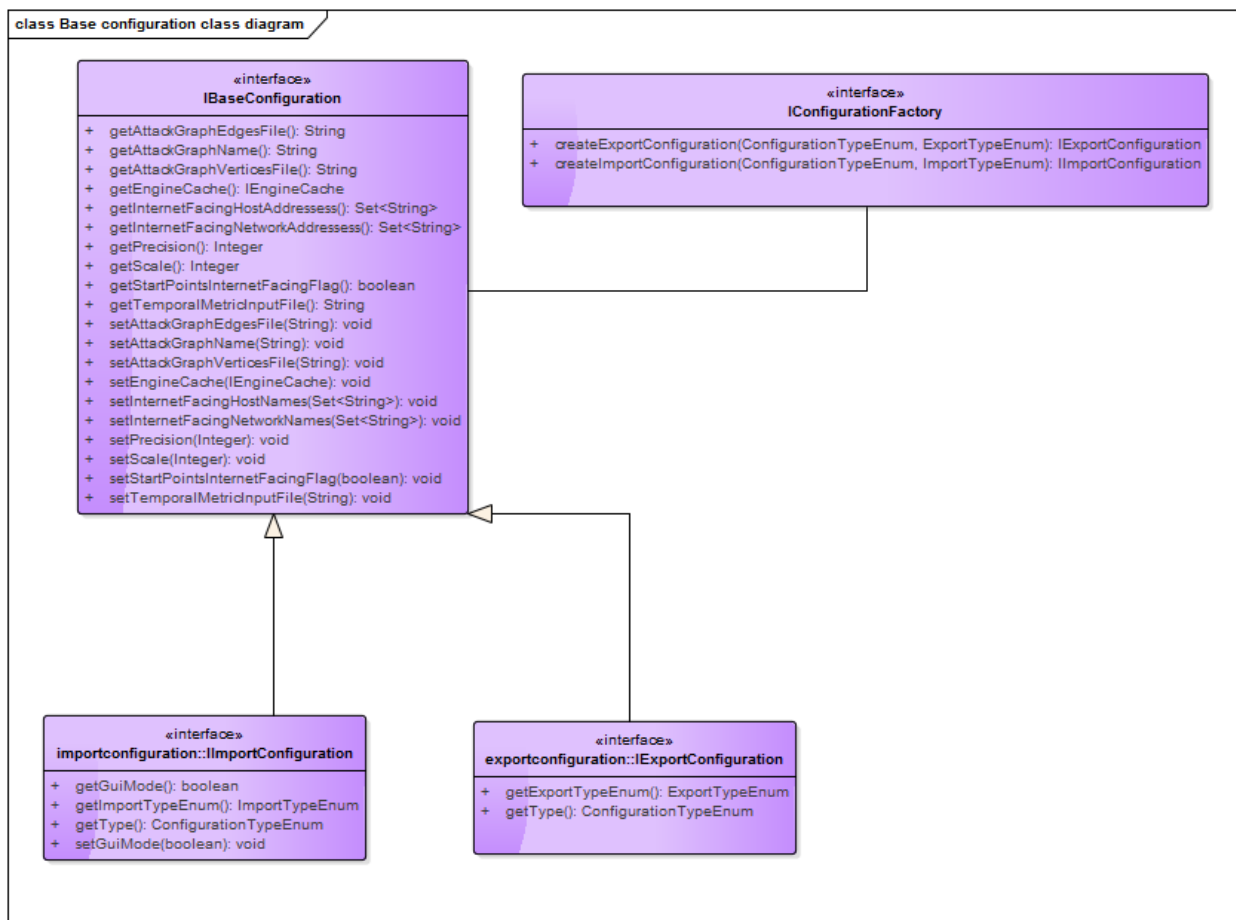


Figure 5-6: Base Configuration API Class Diagram

### 5.1.2.2.7 Interface IMediaImporter

5.1.2.2.7.1 The interface IMediaImporter defines the common methods to import given set of data using the given configuration, into a container of type IEngineCache.

5.1.2.2.7.2 The class diagram of IMediaImporter is shown in Figure 5-7.

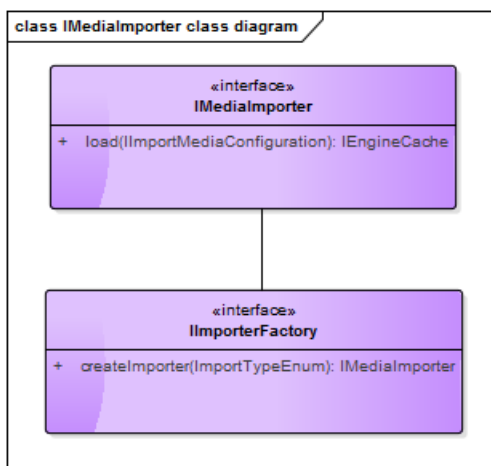


Figure 5-7: Importer API Class Diagram

### 5.1.2.2.8 Interface IMetricsExporter

5.1.2.2.8.1 The main purpose of the IMetricsExporter is to define the common methods to export given results to target media.

5.1.2.2.8.2 The class diagram of IMetricsExporter is shown in Figure 5-8.

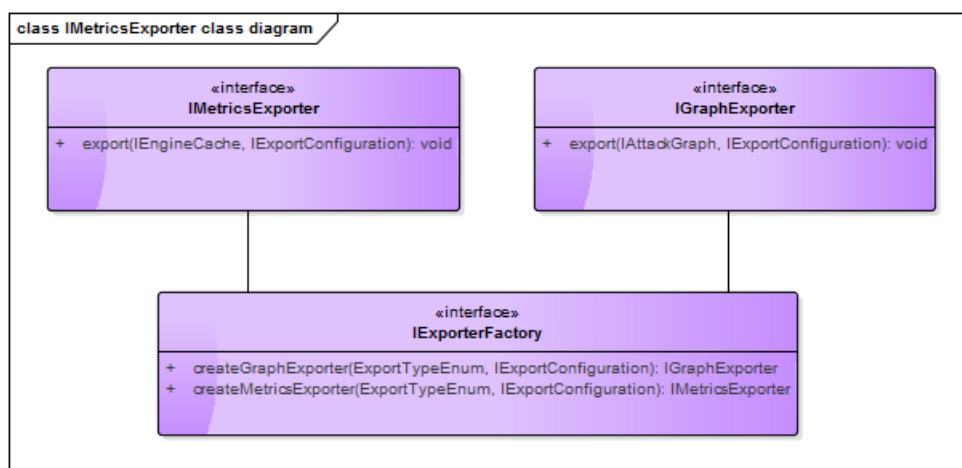


Figure 5-8: Metrics Exporter API Class Diagram



## 5.1.3 SPM CLI Structure Design Description

5.1.3.1 The various software components of SPM CLI are organized into packages as shown in Figure 5-10.

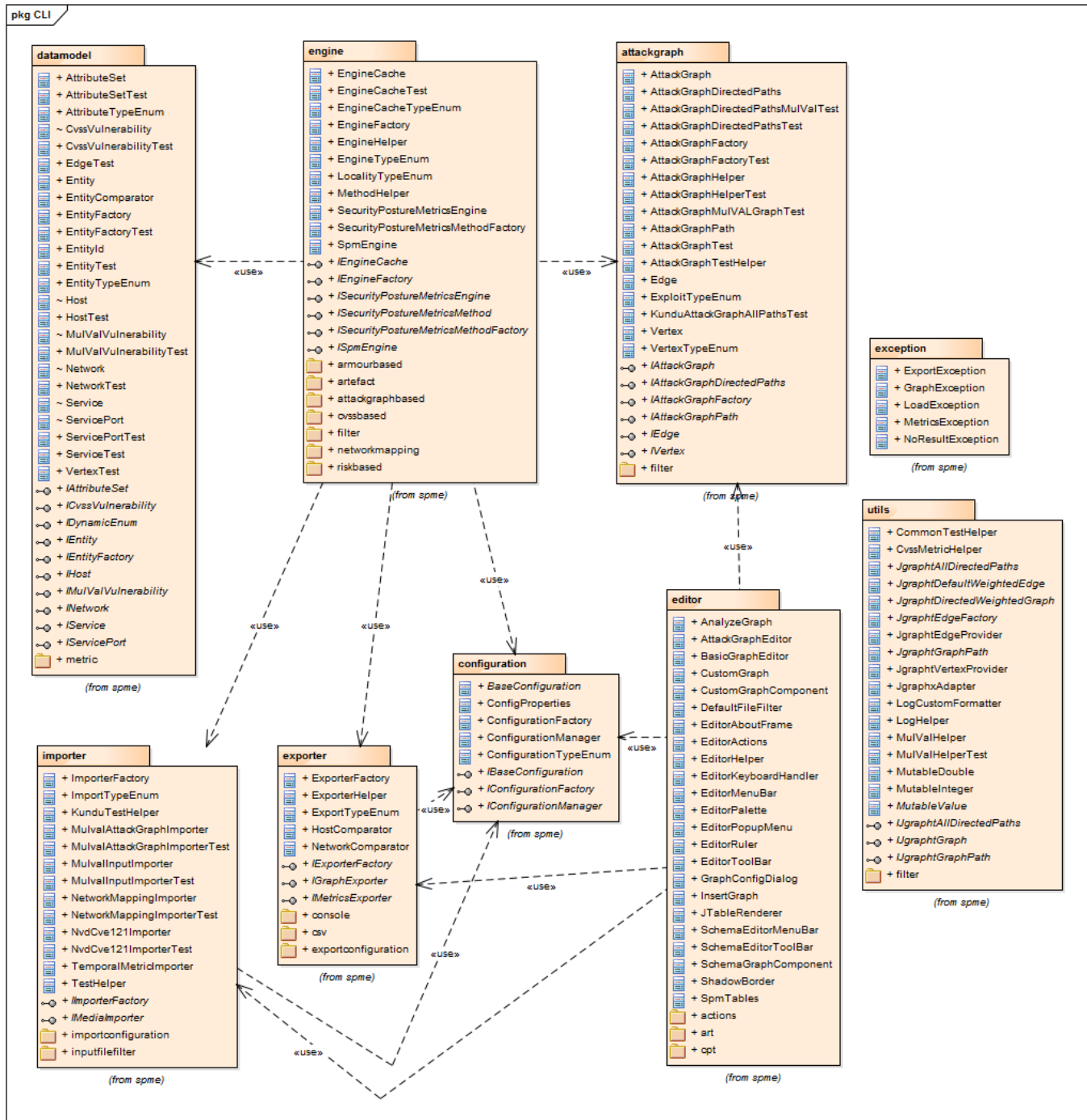


Figure 5-9: SPM CLI Package Diagram

### 5.1.3.2 Package Engine

#### 5.1.3.2.1 Class SpmEngine

5.1.3.2.1.1 The class SpmEngine implements the interface ISpmEngine.

5.1.3.2.1.2 The class diagram of SpmEngine is shown in Figure 5-10.

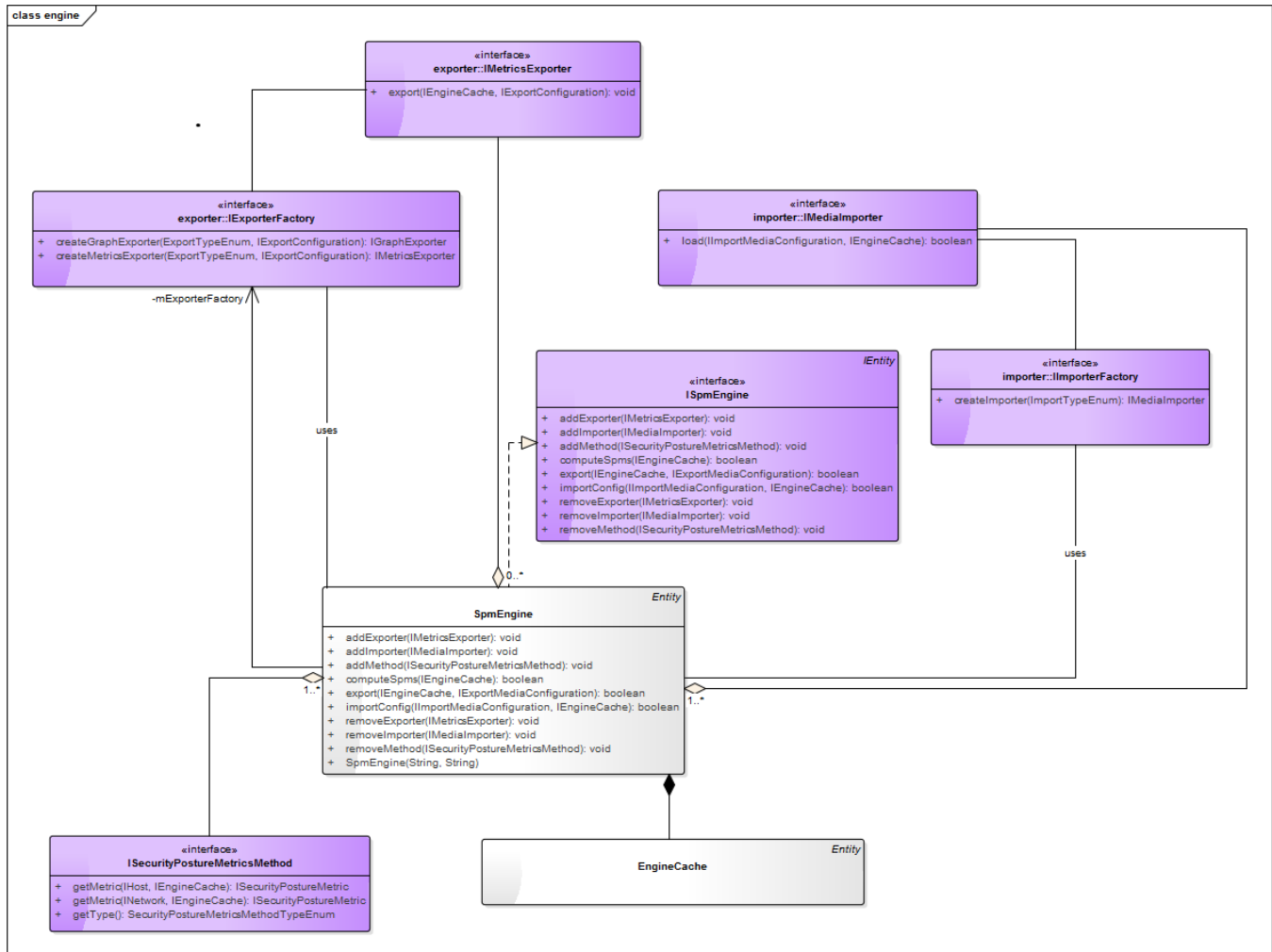


Figure 5-10: Class Engine Class Diagram.

Security Posture Metric CLI SDD	Unclassified	Date: 19 November 2017
2268C.005-SPM-CLI-SDD-01 Rev. 01		Page 33

### 5.1.3.2.1 Class of Type `ISecurityPostureMetricsMethod`

5.1.3.2.1.1 SPM CLI provides the following built-in methods that implement the interface of `ISecurityPostureMetricsMethod`:

- (1) **Class `CvssBasedMethod`**: this class performs CVSS-only SPM calculations at host-level and network-level;
- (2) **Class `ArmourBasedMethod`**: this class performs Current ARMOUR SPM calculations at host-level and network-level;
- (3) **Class `RiskBasedMethod`**: this class performs risk-based SPM calculations at host-level and network-level;
- (4) **Class `AgAllPathMethod`**: this class performs attack graph based SPM calculations based on paths at host-level and network-level;
- (5) **Class `AgPsmMethod`**: this class performs attack graph based probabilistic SPM calculations at host-level and network-level; and
- (6) **Class `AgArsmMethod`**: this class performs attack graph based attack-resistant SPM calculations at host-level and network-level.

5.1.3.2.1.2 The class diagram containing various methods is shown in Figure 5-11.

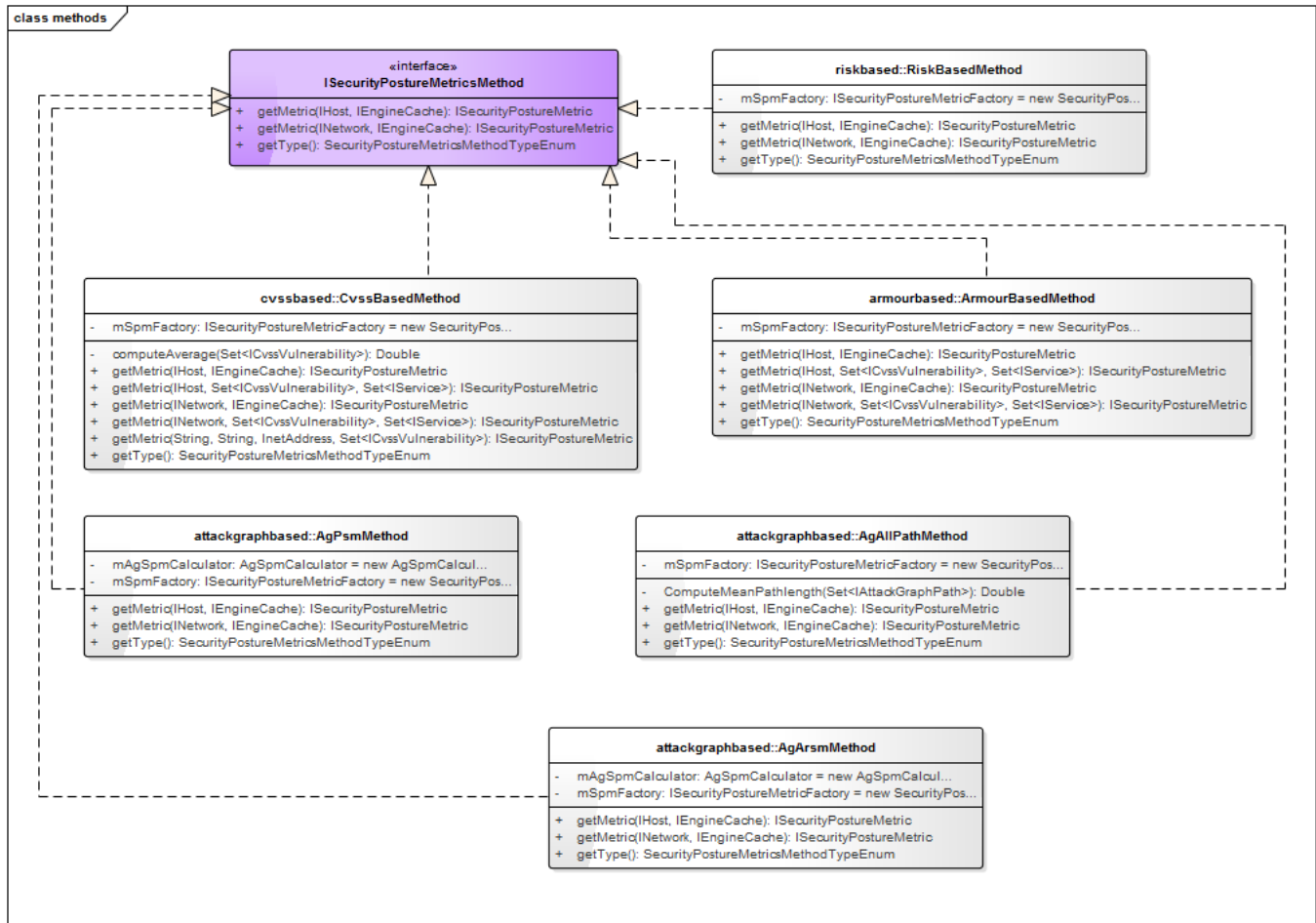


Figure 5-11: Class Diagram Showing Various SPM Methods.

### 5.1.3.2.1 Class CvssBasedMethod

5.1.3.2.1.1 The class CvssBasedMethods provides methods to compute CVSS-only metrics for the following levels:

- (1) **Host-level:** where the goal is a single host; and
- (2) **Network level:** where the goal is a target network.

Security Posture Metric CLI SDD	Unclassified	Date: 19 November 2017
2268C.005-SPM-CLI-SDD-01 Rev. 01		Page 35

5.1.3.2.1.2 For host-level metric, the input, algorithm and output of the class method getMetric() are :

(1) Input:

- (a) **Host:** as goal. The host object must contain a valid IPv4 address;
- (b) **EngineCache:** the cache with the following data defined:
  - (i) **CVSS scores:** the set of base scores imported from NVD CVE data source; and
  - (ii) **Vulnerable services:** the set of vulnerable services and their vulnerabilities imported from the OVAL input file.

(2) The metric is calculated as follows:

- (a) Create a set of vulnerabilities from all vulnerable services associated to the given host;
- (b) Compute the average of CVSS base scores of the vulnerabilities in the set; and
- (c) Compute  $SPM = 1 - \frac{1}{10} average(CVSSBase)$ .

(3) The output is an instance of ISecurityPostureMetric containing the value of the SPM.

5.1.3.2.1.3 For network-level metric, the input, algorithm and output of the class method getMetric() are :

(1) Input:

- (a) **Network:** as goal. The network object must contain a valid IPv4 address;
- (b) **EngineCache:** the cache with the following data defined:
  - (i) **CVSS scores:** the set of base scores imported from NVD CVE data source; and
  - (ii) **Vulnerable services:** the set of vulnerable services and their vulnerabilities imported from the OVAL input file.

(2) The metric SPM is calculated as follows:

- (a) Create a set of vulnerabilities from all vulnerable services associated to the given network;
- (b) Compute the average of CVSS base scores of the vulnerabilities in the set; and
- (c) Compute  $SPM = 1 - \frac{1}{10} average(CVSSBase)$ .

(3) Output an instance of ISecurityPostureMetric containing the value of the SPM.

Security Posture Metric CLI SDD	Unclassified	Date: 19 November 2017
2268C.005-SPM-CLI-SDD-01 Rev. 01		Page 36

### 5.1.3.2.2 Class ArmourBasedMethod

5.1.3.2.2.1 The class ArmourBasedMethods provides methods to compute Current ARMOUR metrics for the following levels:

- (1) **Host-level**: where the goal is a single host; and
- (2) **Network level**: where the goal is a target network.

5.1.3.2.2.2 For host-level metric, the input, the algorithm and the output of the method getMetric() of this class are:

- (1) Input:
  - (a) **Host**: as goal. The host object must contain a valid IPv4 address;
  - (b) **EngineCache**: the cache with the following data defined:
    - (i) **CVSS scores**: the set of base scores imported from NVD CVE data source; and
    - (ii) **Vulnerable services**: the set of vulnerable services and their vulnerabilities imported from the OVAL input file.

(2) The metric SPM is calculated as follows:

- (a) Create a set of vulnerabilities from all vulnerable services associated to the given host; and
- (b) Compute  $SPM = \frac{1}{\sum_{vuln=1}^V \frac{1}{(1 - \frac{CVSSBase_{host,vuln}}{11})}}$  where  $CVSSBase_{host,vuln}$  are the CVSS base scores of each vulnerability from the set.

(3) Output an instance of ISecurityPostureMetric containing the value of the SPM.

5.1.3.2.2.3 For network-level metric, the input, the algorithm and the output of the method getMetric() of this class are :

- (1) Input:
  - (a) **Network**: as goal. The network object must contain a valid IPv4 address;
  - (b) **EngineCache**: the cache with the following data defined:
    - (i) **CVSS scores**: the set of base scores imported from NVD CVE data source; and
    - (ii) **Vulnerable services**: the set of vulnerable services and their vulnerabilities imported from the OVAL input file.

(2) The metric SPM is calculated as follows:

- (a) Create a set of all hosts on the given network; and
- (b) Compute  $SPM = \min_{network}(SPM_{host})$  where  $SPM_{host}$  are the metrics for each host in the set.

(3) The output is an instance of ISecurityPostureMetric containing the value of the SPM.

### 5.1.3.2.3 Class RiskBasedMethod

5.1.3.2.3.1 The class RiskBasedMethods provides methods to compute risk-based metrics for the following levels:

- (1) **Host-level**: where the goal is a single host; and
- (2) **Network level**: where the goal is a target network.

5.1.3.2.3.2 For host-level metric, the input, the algorithm and the output of the method getMetric() of this class are:

(1) Input:

- (a) **Host**: as goal. The host object must contain a valid IPv4 address;
- (b) **EngineCache**: the cache with the following data defined:
  - (i) **CVSS scores**: the set of impact and exploit scores imported from NVD CVE data source;
  - (ii) **Vulnerable services**: the set of vulnerable services and their vulnerabilities imported from the OVAL input file;
  - (iii) **All services**: the set of all services imported from the OVAL input file;
  - (iv) **Open service ports**: the set of all open service ports imported from the OVAL input file;
  - (v) **Network mapping**: the network configuration;
  - (vi) **Internet facing network(s)**: the set of internet-facing network addresses; and
  - (vii) **Attack graph**: the generated attack graph.

(2) The metric SPM is calculated as follows:

- (a) Create a set of vulnerabilities from all vulnerable services associated to the given host;
- (b) Compute the average of exploit sub scores  $average(CVSSExp)$  from the set of vulnerabilities and CVSS scores;
- (c) Compute the average of impact sub scores  $average(CVSSImp)$  from the set of vulnerabilities and CVSS scores;

Security Posture Metric CLI SDD	Unclassified	Date: 19 November 2017
2268C.005-SPM-CLI-SDD-01 Rev. 01		Page 38

- (d) Compute the number of services  $\#Services_{total}$  as size of all services associated to the given host;
- (e) Compute the number of vulnerable services  $\#Services_{vulnerable}$  as size of all vulnerable services associated to the given host;
- (f) Compute  $R_{known} = \frac{\#Services_{vulnerable}}{\#Services_{total}} + \frac{1}{10} average_{host}(CVSSExp) + \frac{1}{10} average_{host}(CVSSImp)$ ;
- (g) Compute the number of open service ports  $\#ServicePorts_{open}$  as size of all service ports from the services that are referred by a host access statement 'haci' in the input file and that are associated to the given host;
- (h) Compute the number of service ports  $\#ServicePorts_{total}$  as size of all service ports from the services associated to the given host;
- (i) Compute the number of Internet facing service ports  $\#ServicePorts_{internet-facing}$  as size of all service ports from the services that are in the internet facing network and that are associated to the given host;
- (j) Compute the number of possible vectors  $\#Vectors_{possible}$  from the set of vertices in the attack graph of type 'execCode' or 'haci' with target being the given host;
- (k) Compute the number of possible cross-domain vectors  $\#Vectors_{possiblecross-domain}$  from the set of possible vectors and where the target and source hosts are not on the same network;
- (l) Compute  $R_{unknown} = \frac{\#Vectors_{possiblecross-domain}}{\#Vector_{possible}} + \frac{\#ServicePorts_{open}}{\#ServicePorts_{total}} + \frac{\#ServicePorts_{internet-facing}}{\#ServicePorts_{total}}$ ;
- (m) Compute the number of possible vectors in the attack graph  $\#Vectors_{attackgraph}$  from the set of vertices in the attack graph of type 'execCode' or 'haci' with target being any host;
- (n) Compute  $P_{safeguard} = \frac{\#Vectors_{attackgraph}}{\#Vectors_{possible}}$ ; and
- (o) Compute  $SPM = 1 - \left[ \frac{1}{3} R_{known} + \frac{1}{3} R_{unknown} + \frac{1}{3} P_{safeguard} \right]$ .

(3) Output an instance of ISecurityPostureMetric containing the value of the SPM.

5.1.3.2.3.3 For network-level metric, the calculation is same as in host-level metric with the input goal being the target network.

5.1.3.2.3.4 Risk-based metrics are computed for the following settings:

- (1) All vectors are relative to the host (at host-level) or the network (at network-level); and
- (2) All vectors are independent from the host (at host-level) or the network (at network-level).

Proprietary Information. Use or disclosure of this data is subject to the Restriction of the title page of this document.	Unclassified	THALES
--	--------------	--------



### 5.1.3.2.4 Class AgAllPathMethod

5.1.3.2.4.1 The class AgAllPathMethods computes SPMs based on attack graph for the following levels:

- (1) **Host-level:** where the goal is a single host; and
- (2) **Network level:** where the goal is a target network.

5.1.3.2.4.2 For host-level metric, the input, the algorithm and the output of the method getMetric() of this class are:

(1) Input:

- (a) **Host:** as goal. The host object must contain a valid IPv4 address;
- (b) **EngineCache:** the cache with the following data defined:
  - (i) **Attacker located rules:** the set of rules to filter vertices of type 'attackerLocated';
  - (ii) **Attack goals:** the set of rules to filter goal vertices; and
  - (iii) **Attack graph:** the generated attack graph.

(2) The metric SPM is calculated as follows:

- (a) Compute the set of starting vertices using attacker located rules;
- (b) Compute the set of goal vertices using attack goals that apply to the given host;
- (c) Compute the set of paths from starting vertices to goal vertices using the method AllDirectedGraph::getAllPaths() (Dijkstra) from the JGraphT library with the following parameters:
  - (i) If the graph has cycles, the maximum number of path length is the number from the XML configuration if defined or 12 by default. This is to prevent infinite loops. Otherwise the maximum number of path length is null; and
  - (ii) Simple path flag is set to true. All self-intersecting paths are not considered.
- (d) Compute the number of paths  $NP_{host}$  as the size of the set of paths;
- (e) Compute the shortest path length as  $SP = \min(\text{length}(\text{path}_1), \text{length}(\text{path}_2), \dots, \text{length}(\text{path}_{NP}))$ . Only interaction vertices are counted in the calculation of the length;
- (f) Compute the mean path length as  $MPL = \text{average}(\text{length}(\text{path}_1), \text{length}(\text{path}_2), \dots, \text{length}(\text{path}_{NP}))$ . Only interaction vertices are counted in the calculation of the length; and
- (g) Compute the normalized mean path length as  $NMPL = \frac{1}{NP} \text{average}(\text{length}(\text{path}_1), \text{length}(\text{path}_2), \dots, \text{length}(\text{path}_{NP}))$ . Only interaction vertices are counted in the calculation of the length.

(3) Output an instance of ISecurityPostureMetric containing the value of the SPM.

5.1.3.2.4.3 For network-level metric, the calculation is same as in host-level metric with the input goal being the target network.

### 5.1.3.2.5 Class AgPsmMethod

5.1.3.2.5.1 The class AgPsmMethods computes Probabilistic SPMs based on attack graph for the following levels:

- (1) **Host-level**: where the goal is a single host; and
- (2) **Network level**: where the goal is a target network.

5.1.3.2.5.2 For host-level metric, the input, the algorithm and the output of the method getMetric() of this class are:

- (1) Input:
  - (a) **Host**: as goal. The host object must contain a valid IPv4 address;
  - (b) **EngineCache**: the cache with the following data defined:
    - (i) **Attacker located rules**: the set of rules to filter vertices of type 'attackerLocated';
    - (ii) **Attack goals**: the set of rules to filter goal vertices;
    - (iii) **CVSS scores**: the set of base scores imported from NVD CVE data source;
    - (iv) **CVSS temporal metrics**: the set of temporal metrics from the temporal metrics CSV file; and
    - (v) **Attack graph**: the generated attack graph.
- (2) The metric SPM is calculated as follows:
  - (a) Compute the set of starting vertices using attacker located rules;
  - (b) Compute the set of goal vertices using attack goals that apply to the given host;
  - (c) Compute the individual scores for all vertices as  $p(e) = BS * E * RL * RC$  where BS is the CVSS base score, E is the exploitability metric (temporal), RL is the remediation level metric (temporal), and RC is the report confidence metric (temporal); and
  - (d) Compute SPM using the following algorithm:
    - (i) For all vertices V in the graph, mark V as processed if not of type 'execCode'. Mark V as unprocessed otherwise; and

(ii) While there exist an unprocessed vertex V:

<p><i>For all vertices <math>v</math> in the set <math>V</math> of vertices from the graph</i>  <i>If <math>v</math> is not of type 'execCode' then mark <math>v</math> as processed and let <math>SMP(v) = 1</math> else mark <math>v</math> as unprocessed and let <math>SMP(v) = 0</math>.</i></p> <p><i>While there exist unprocessed <math>v</math> in the set <math>V</math> of vertices</i>  <i>While there exists an unprocessed vertex <math>v'</math> whose predecessors are all processed</i>  <i>Calculate <math>SMP(v')</math> and mark <math>v'</math> as processed</i>  <i>For each vertex <math>v'</math> in a cycle that has more than one incoming edge</i>  <i>Calculate <math>SMP(v')</math> and mark <math>v'</math> as processed</i>  <i>For each unprocessed vertex <math>v'</math> in the cycles</i>  <i>Calculate <math>SMP(v')</math> and mark <math>v'</math> as processed</i></p>
---

(3) Output an instance of ISecurityPostureMetric containing the value of the SPM.

5.1.3.2.5.3 For network-level metric, the calculation is same as in host-level metric with the input goal being the target network.

### 5.1.3.2.6 Class AgArsmMethod

5.1.3.2.6.1 The class AgBasedMethods computes attack resistant SPMs based on attack graph for the following levels.

- (1) **Host-level:** where the goal is a single host; and
- (2) **Network level:** where the goal is a target network.

5.1.3.2.6.2 For host-level metric, the input, the algorithm and the output of the method getMetric() of this class are:

(4) Input:

- (a) **Host:** as goal. The host object must contain a valid IPv4 address;
- (b) **EngineCache:** the cache with the following data defined:
  - (i) **Attacker located rules:** the set of rules to filter vertices of type 'attackerLocated';
  - (ii) **Attack goals:** the set of rules to filter goal vertices;
  - (iii) **CVSS scores:** the set of base scores imported from NVD CVE data source;
  - (iv) **CVSS temporal metrics:** the set of temporal metrics from the temporal metrics CSV file; and
  - (v) **Attack graph:** the generated attack graph.

(5) The metric SPM is calculated as follows:

- (a) Compute the set of starting vertices using attacker located rules;
- (b) Compute the set of goal vertices using attack goals that apply to the given host;

- (c) Compute the individual scores for all vertices as  $r(e) = 1/p(e)$  where  $p(e) = BS * E * RL * RC$ , BS is the CVSS base score, E is the exploitability metric (temporal), RL is the remediation level metric (temporal), and RC is the report confidence metric (temporal); and
- (d) Compute SPM using the following algorithm.

```

For all vertices  $v$  in the set  $V$  of vertices from the graph

    If  $v$  is not of type 'execCode' then mark  $v$  as processed and let  $SMP(v) = 0$  else mark  $v$  as unprocessed and let  $SMP(v) = \infty$ .

While there exist unprocessed  $v$  in the set  $V$  of vertices
    While there exists an unprocessed vertex  $v'$  whose predecessors are all processed
        Calculate  $SMP(v')$  and mark  $v'$  as processed
    For each vertex  $v'$  in a cycle that has more than one incoming edge
        Calculate  $SMP(v')$  and mark  $v'$  as processed
    For each unprocessed vertex  $v'$  in the cycles
        Calculate  $SMP(v')$  and mark  $v'$  as processed

```

- (6) Output an instance of ISecurityPostureMetric containing the value of the SPM.

5.1.3.2.6.3 For network-level metric, the calculation is same as in host-level metric with the input goal being the target network.

### 5.1.3.3 Package Attack Graph

#### 5.1.3.3.1 Class AttackGraph

5.1.3.3.1.1 The class AttackGraph derives from JGraphT's class DirectedWeightedGraph.

5.1.3.3.1.2 The class AttackGraph provides basic methods to browse the underlying graph. The methods include:

- (1) Get all vertices matching a given attribute value;
- (2) Get an edge with given name;
- (3) Get a vertex with given name;
- (4) Return cycles detection flag; and
- (5) Get all cycles (as intersecting vertices).

5.1.3.3.1.3 The class diagrams for classes AttackGraph, Vertex and Edge are shown in Figure 5-12 and Figure 5-13.

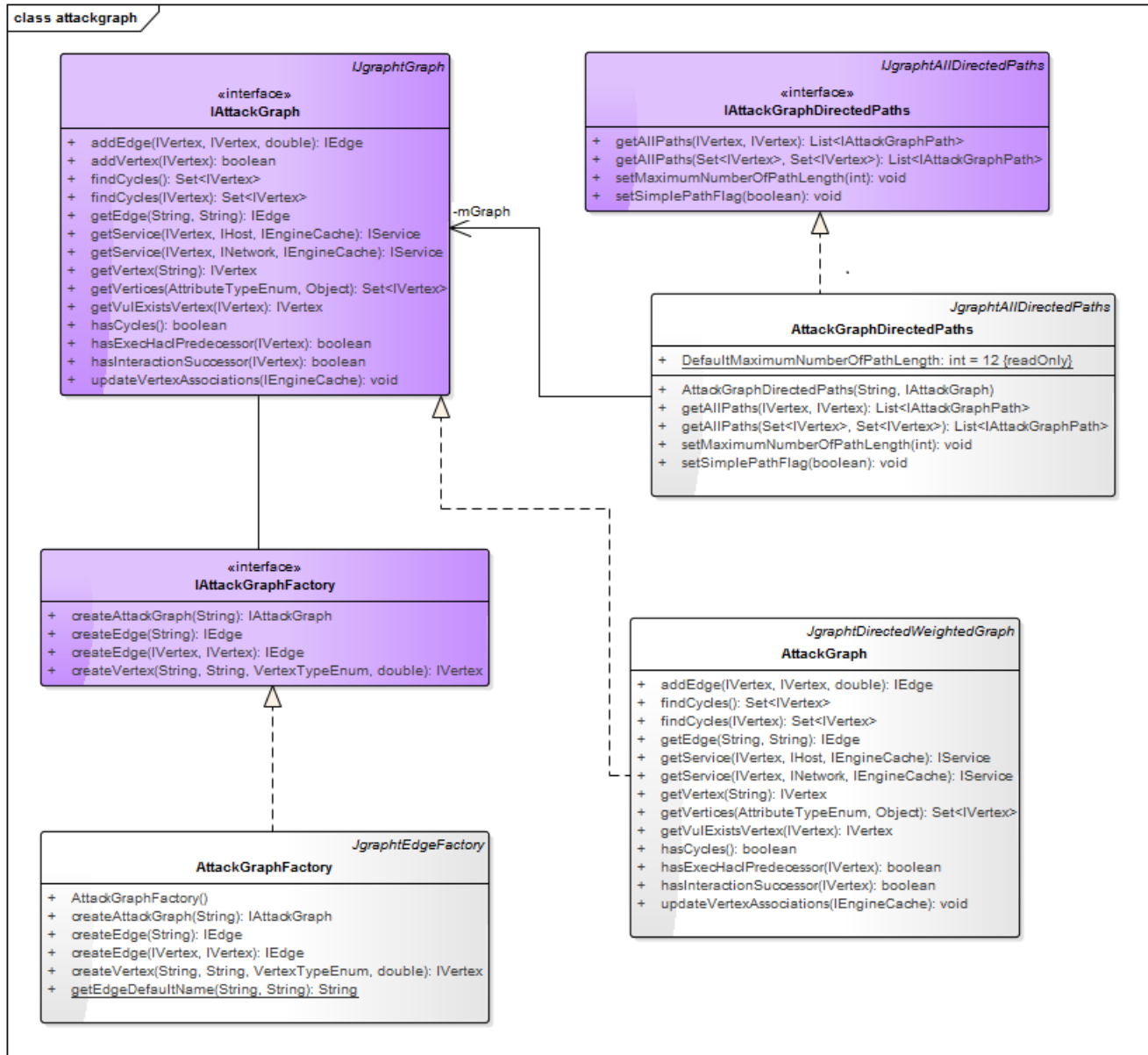


Figure 5-12: Attack Graph Class Diagram.

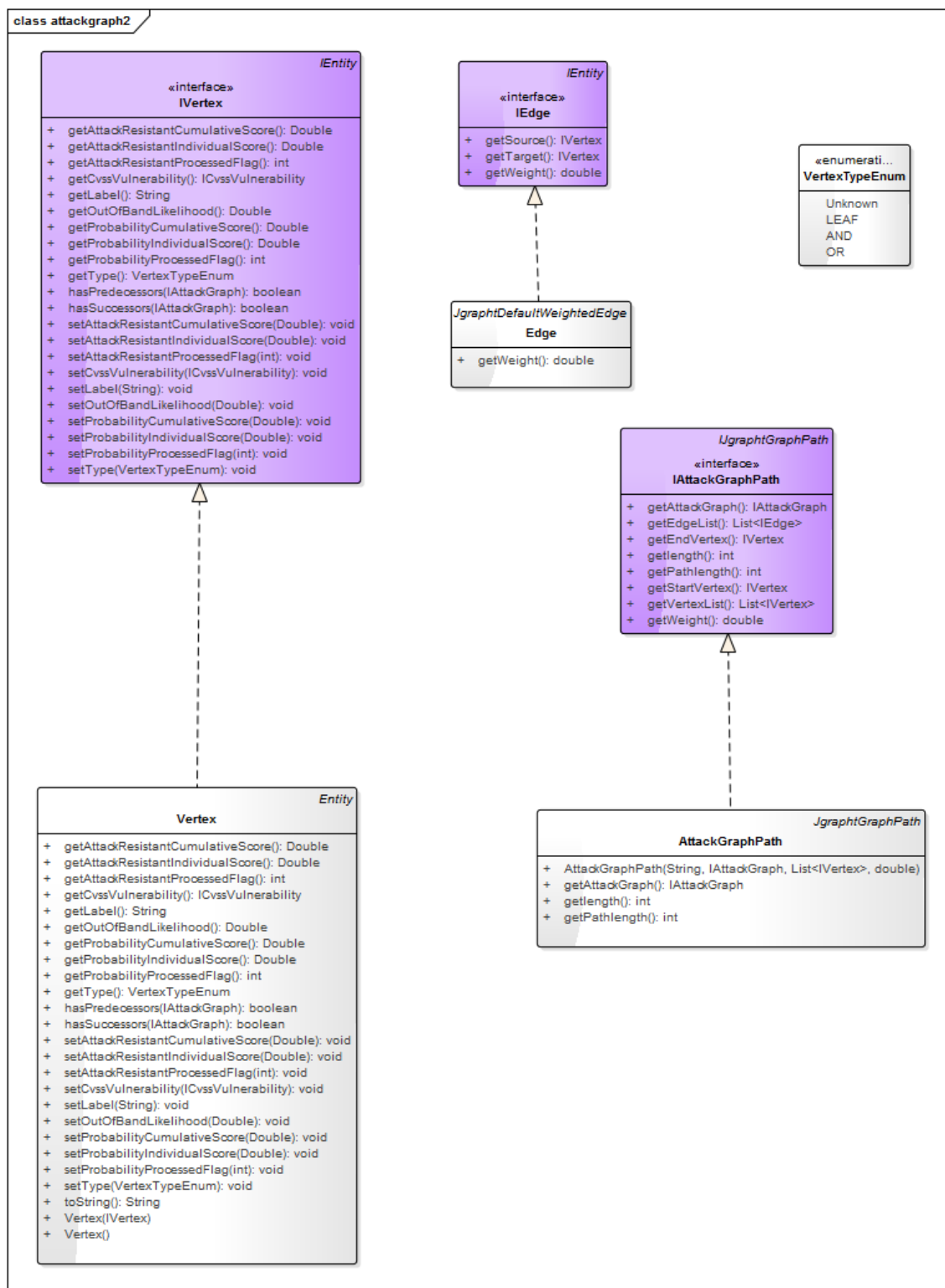


Figure 5-13: Edge and Vertex Class Diagram.

5.1.3.4 Package Datamodel

5.1.3.4.1 Class Entity

5.1.3.4.2 Class Entity defines the base class for all major features in the SPM CLI data model. Class Entity is mainly a set of attributes (of type IAttributeSet) and convenient accessors. The set of attributes of an Entity could be expanded as needed.

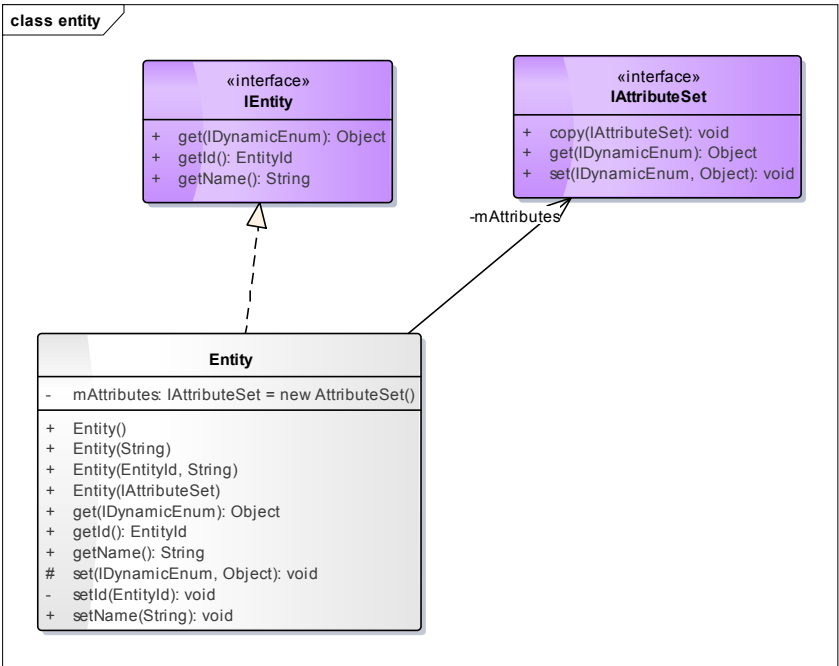


Figure 5-14: Class Entity Class Diagram.

### 5.1.3.4.3 Class AttributeSet

5.1.3.4.4 Class AttributeSet defines an expandable set of attributes for an Entity.

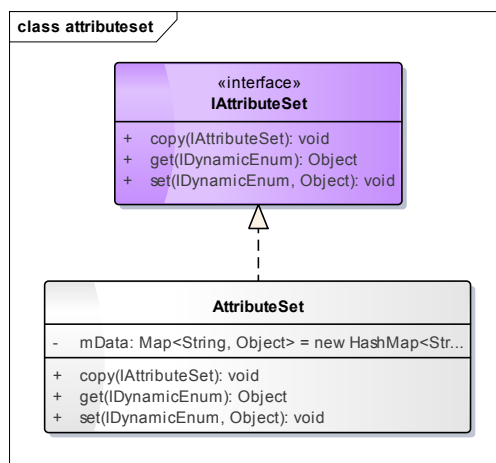


Figure 5-15: Class AttributeSet Class Diagram.

### 5.1.3.5 Package Importer

#### 5.1.3.5.1 Classes of Type IMediaImporter

5.1.3.5.1.1 SPM CLI provides the following built-in importers that implement the interface of IMediaImporter:

- (1) **Class NvdCve121Importer:** this class imports CVSS base scores from a set of XML files at a location specified in the import configuration. The XML files contains NVD vulnerabilities version 1.2.1 and are previously downloaded from NIST site (<https://nvd.nist.gov/vuln/data-feeds>);
- (2) **Class MulValInputImporter:** this class imports MulVAL input data from a .P file at a location specified in the import configuration;
- (3) **Class NetworkMappingImporter:** this class imports network configuration data from an XML file at a location specified in the import configuration;
- (4) **Class MulValAttackGraphImporter:** this class imports the graph from CSV files (default name are ARCS.csv and VERTICES.csv) at a location specified in the import configuration; and
- (5) **Class TemporalMetricImporter:** this class imports the temporal metrics from CSV files at a location specified in the import configuration.



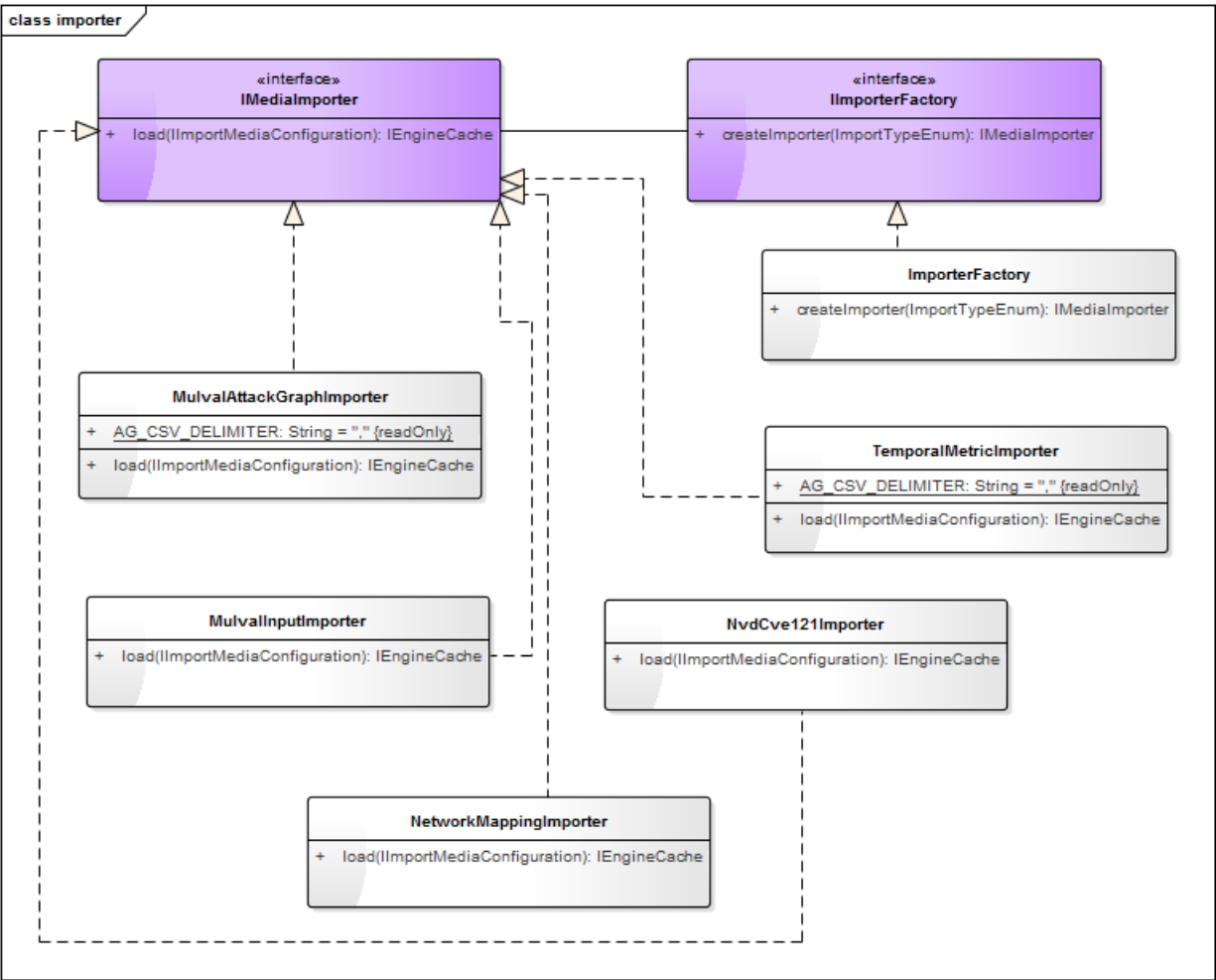


Figure 5-16: Media Importers Class Diagram.

### 5.1.3.6 Package Exporter

#### 5.1.3.6.1 Classes of Type IMetricsExporter

5.1.3.6.1.1 SPM CLI provides the following built-in exporters that implement the interface of IMetricsExporter:

- (1) **Class CsvCvssBasedMetricsExporter:** this class exports CVSS-only SPMs into 2 CSV files: one for host-level data and one for network-level. The default file names are:
  - (a) **CvssBased\_HostLevel.csv:** this file contains CVSS-only SPMs for host-level; and
  - (b) **CvssBased\_NetworkLevel.csv:** this file contains CVSS-only SPMs for network-level.

- (2) **Class CsvArmourBasedMetricsExporter:** this class exports Current ARMOUR SPMs into 2 CSV files: one for host-level data and one for network-level. The default file names are:
- (a) **ArmourBased\_HostLevel.csv:** this file contains Current ARMOUR SPMs for host-level; and
  - (b) **ArmourBased\_NetworkLevel.csv:** this file contains Current ARMOUR SPMs for network-level.
- (3) **Class CsvAgBasedMetricsExporter:** this class exports attack graph based SPMs into 2 CSV files: one for host-level data and one for network-level. The default file names are:
- (a) **AgBased\_HostLevel.csv:** this file contains attack graph based SPMs for host-level; and
  - (b) **AgBased\_NetworkLevel.csv:** this file contains attack graph based SPMs for network-level.
- (4) **Class CsvAgBasedPsmMetricsExporter:** this class exports attack graph based PSMs into 2 CSV files: one for host-level data and one for network-level. The default file names are:
- (a) **AgBasedPsm\_HostLevel.csv:** this file contains attack graph based PSMs for host-level; and
  - (b) **AgBasedPsm\_NetworkLevel.csv:** this file contains attack graph based PSMs for network-level.
- (5) **Class CsvAgBasedArsmMetricsExporter:** this class exports attack graph based ARSMs into 2 CSV files: one for host-level data and one for network-level. The default file names are:
- (a) **AgBasedArsm\_HostLevel.csv:** this file contains attack graph based ARSMs for host-level; and
  - (b) **AgBasedArsm\_NetworkLevel.csv:** this file contains attack graph based ARSMs for network-level.
- (6) **Class CsvTemporalMetricsExporter:** this class exports temporal metrics used to compute other SPMs into a CSV files (default: **CvssTemporalMetrics.csv**).

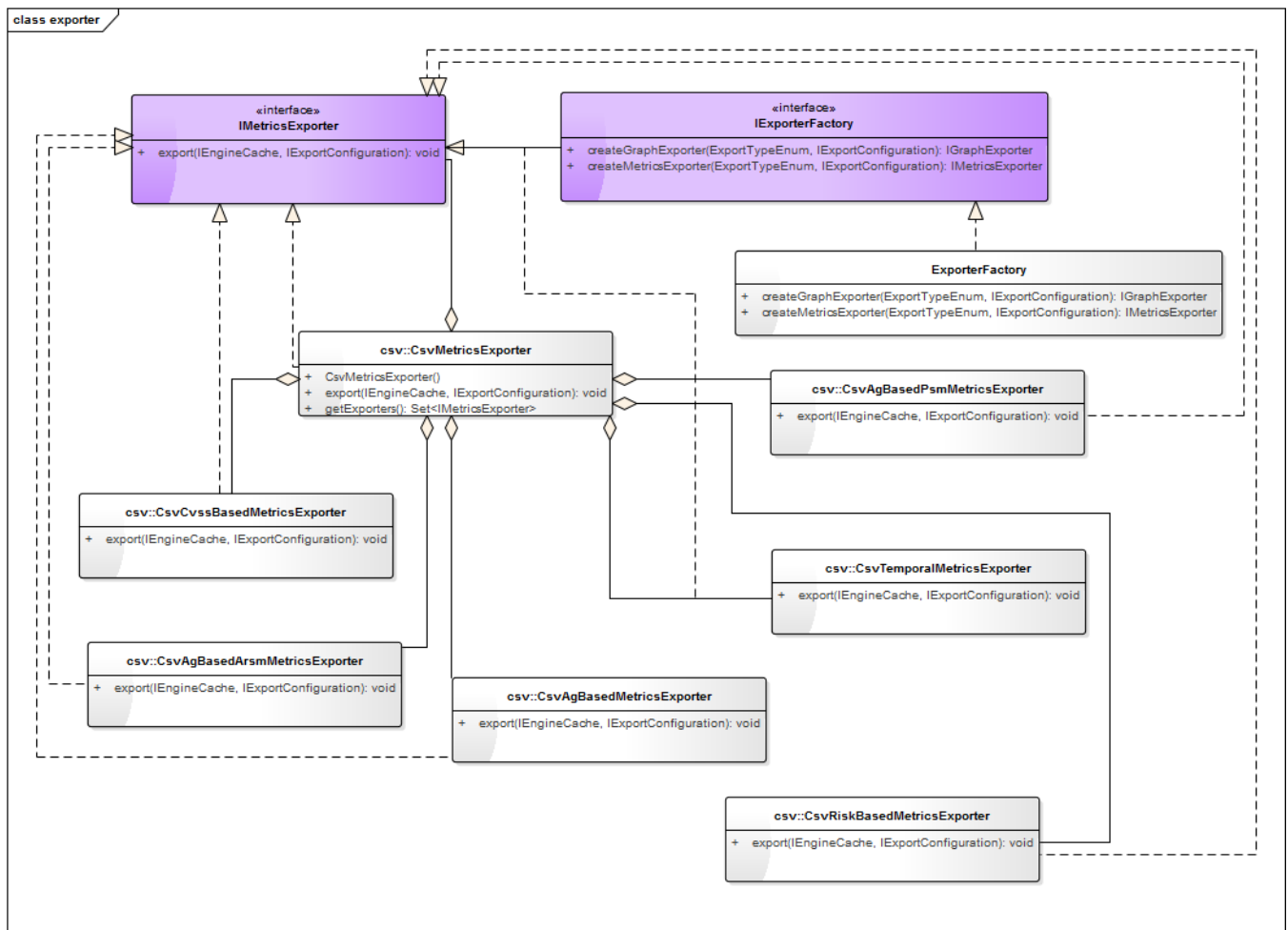


Figure 5-17: Media Exporters Class Diagram.

## 6 APPENDIX A: SPM CLI JAVA PROJECTS

### 6.1 How to install, build and launch the tool

6.1.1.1.1 The following sections provide instructions on how to install the tools required building and launching the SPM CLI tool.

#### 6.1.2 Pre-requisites

6.1.2.1.1 SPM CLI was tested on a Windows 10 platform. The following tools were used for the development and tests of SPM CLI:

- (1) Java JDK 1.8.0.144 or better;
- (2) Python 3.6.2 or better;
- (3) Eclipse Oxygen Release (4.7.0) or better; and
- (4) Standard GIT client or GIT bash.

#### 6.1.3 Install code base from DRDC GIT repository

6.1.4 If applicable, use GIT client to fetch the source code into the target machine from the DRDC repository.

6.1.5 The DRDC GIT repository is located here:

<https://<username>:<password>@toque.ottawa.drdc-rddc.gc.ca/git/metricsspmtal>

6.1.5.1.1 The project main folders are:

- (1) **Metricsspmtal**: this folder contains the SPM CLI project; and
- (2) **Third\_party**: this folder contains the SPM CLI project dependencies (pre-loaded).

6.1.5.1.2 The other folders are:


- (1) **Metricsspmtal-launcher**: this folder contains the SPME Launcher project; and
- (2) **Results**: this folder contains the generated data from input data set 'reduce\_iter1'.

#### 6.1.6 Build the tool within Eclipse

6.1.6.1.1 This step is optional since the GIT source contains a pre-built executable jar of the tool that could be used readily.

6.1.6.1.2 Start Eclipse and import the file system **Metricsspmtal**.

6.1.6.1.3 Create a (debug) configuration:

- (1) Click on debug menu icon  and select “Debug Configurations...”;
- (2) Select “Maven Build” and create new configuration. Give the new configuration a meaningful name; and
- (3) From the new configuration Main tab, enter the following settings:
  - (a) Base directory: `${project_loc:metricsspmtal}`;
  - (b) Goals: `clean install assembly:single`; and
  - (c) The rest should be left by default.
- (4) From the new configuration JRE tab, make sure that JRE is pointing to the JDK 1.8.0.144. This is important since the default non-JDK JRE doesn't include all the required dependencies;
- (5) Launch the new configuration by double clicking it; and
- (6) If the build is successful, a folder target will be created that contain the SPM CLI executable jar.

## 6.1.7 Run the tool from command prompt

### 6.1.7.1 Single MulVAL run as input

6.1.7.1.1 Prepare the input file. Example: `c:/reference_network2`. Make sure that `reference_network2` has the following contents (the content of the files will be explained in other sections):

- (1) A subfolder named 'source\_data': this subfolder should contain the .P file named 'Input.P';
- (2) A subfolder named 'network': this subfolder should contains the XML file name 'network\_mapping.xml'; and
- (3) A subfolder named 'attack\_graph': this subfolder should contains two (2) CSV files: 'ARCS.csv' and 'VERTICES.csv'.

6.1.7.1.2 Start a command prompt and go to the folder `~/GIT/metricsspmtal/target`.

6.1.7.1.3 Execute the following command:

```
>java -jar SecurityPostureMetricsCLI-1.0.0.jar -o c:/temp/reference_network2 c:/reference_network2
```

6.1.7.1.4 If successful, the metrics will be generated into the folder `c:/temp/ reference_network2`

6.1.7.1.5 The output folder will be `reference_network2_<timestamp>`.

### 6.1.7.2 Multiple MulVAL runs as input

6.1.7.2.1 The command will be the same as with single run.

6.1.7.2.2 Each input sub-folder should have the same structure, namely the three (3) folders 'source\_data', 'network' and 'attack\_graph'.

6.1.7.2.3 The command to execute is:

```
>java -jar SecurityPostureMetricsCLI-1.0.0.jar -o c:/temp/reference c:/reduce_iter1/reference
```

6.1.7.2.4 If successful, the metrics will be generated into the folder c:/temp/reference

6.1.7.2.5 The output folder will have several sub-folders each containing the SPMs of the corresponding MUIVAL run, example:

```
./parent/run_1_<timetsamp1>/...  
./parent/run_2_<timetsamp1>/...  
./parent/run_3_<timetsamp1>/...  
...
```

6.1.7.3 Visualization of input

6.1.7.3.1 The option -g provides a GUI for visualization of the MuIVAL graph and the computed results. Figure 6-1 shows an example of the GUI. The data in the example are from the from reference network. The layout mode is horizontal (Diagram→Layout).

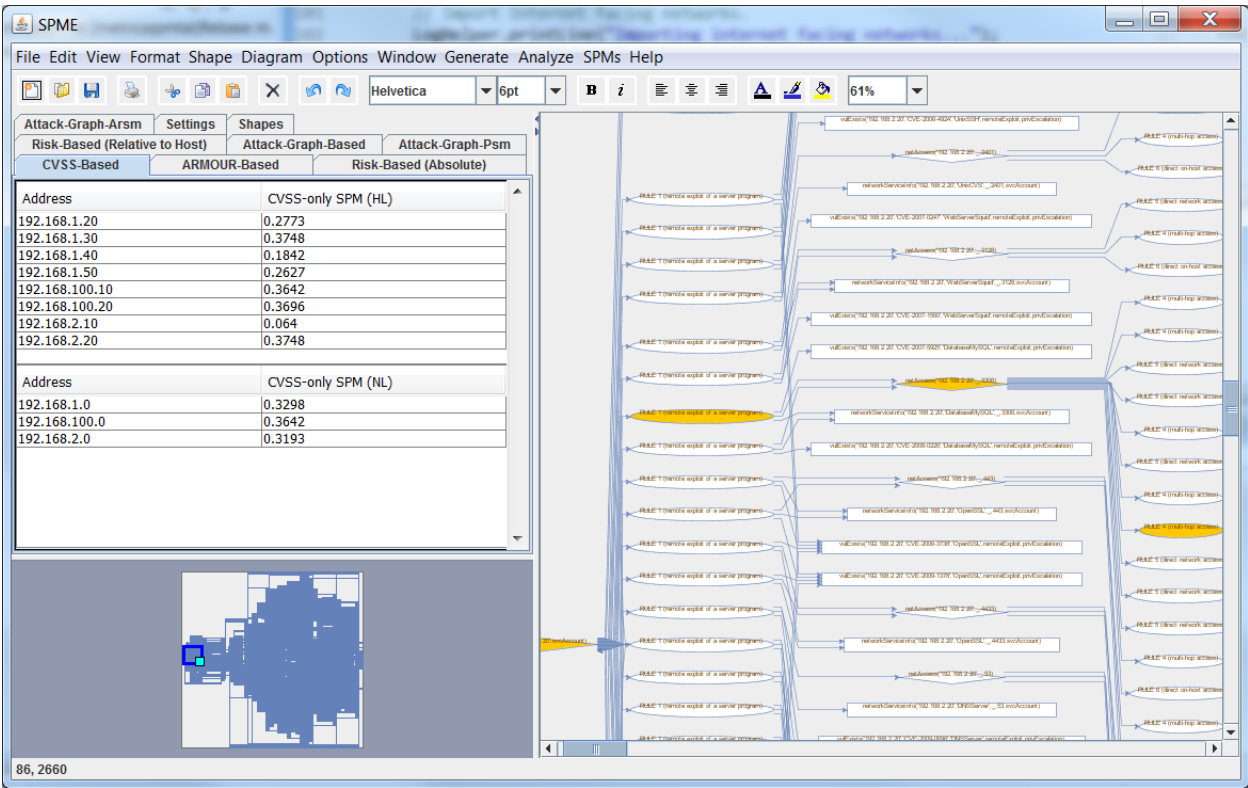


Figure 6-1: SPM CLI GUI example.

6.1.7.3.2 The SPM CLI in GUI mode could be launched either from:

- (1) Command line; or
- (2) SPME Launcher.

Security Posture Metric CLI SDD	<b>Unclassified</b>	Date: 19 November 2017
2268C.005-SPM-CLI-SDD-01 Rev. 01		Page 53

6.1.7.3.3 From the command line, the command to execute is shown in the example below:

```
>java -jar SecurityPostureMetricsCLI-1.0.0.jar c:/reduce_iter1/reference -g
```

Security Posture Metric CLI SDD	<b>Unclassified</b>	Date: 19 November 2017
2268C.005-SPM-CLI-SDD-01 Rev. 01		Page 54

## 7 APPENDIX B: OTHER UTILITIES

### 7.1 Extract Data Python Script

7.1.1 SPM CLI project comes with a python scrip 'extractData.py' to extract data generated from SPM CLI CSV files and save the results in tabular format to consolidated CSV files.

7.1.2 The python script could be launched either from:

- (1) Command line; or
- (2) SPME Launcher.

7.1.3 From the command line, the command to execute the script is shown in the example below (where input directory 'c:/temp/reference' is the directory containing the data generated by the SPM CLI):

```
>python extractData.py -i c:/temp/reference -o c:/temp/reference/extractedData
```

7.1.4 If successful, the consolidated metrics will be generated into the folder c:/temp/reference/extractedData



## 7.2 SPME Launcher

7.2.1 The SPME Launcher provides a GUI-based utility to run SPM CLI tools. Figure 7-1 shows an example of SPME Launcher main page.

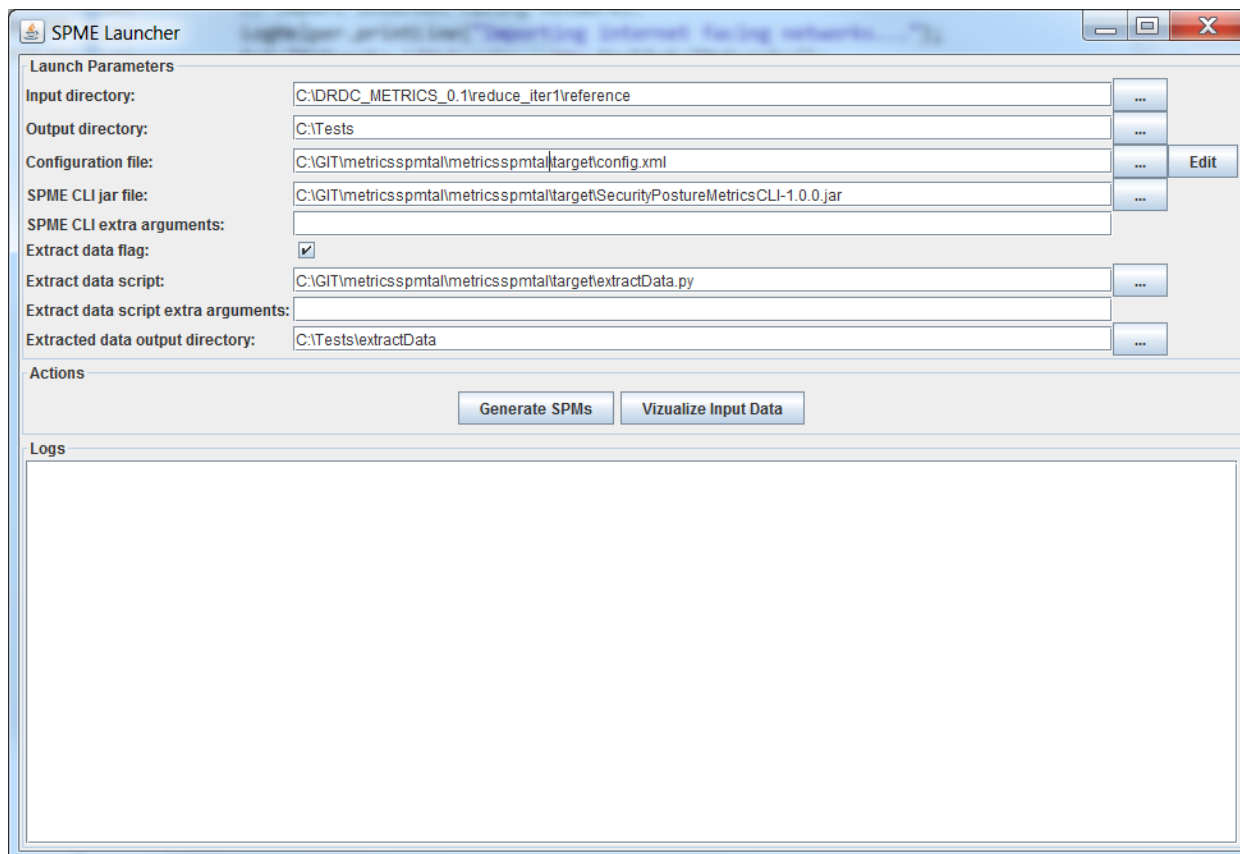


Figure 7-1: SPME Launcher example.

7.2.2 A batch Launcher.bat is provided to start the utility. From the command line, the command to execute is shown in the example below:

```
>cd metricsspmtal-launcher
>Launch.bat
```

7.2.2.1.1 The main fields in the GUI are described below:

- (1) **Input directory** (text field): directory containing the MulVAL data. The directory could be data generated from single run or from multiple runs;
- (2) **Output directory** (text field): directory containing the data generated by SPME CLI;
- (3) **Configuration file** (text field): configuration file used by SPME CLI;
- (4) **SPME CLI jar file** (text field): SPME CLI jar file. This field allows the user to select alternative SPM CLI versions;

Security Posture Metric CLI SDD	Unclassified	Date: 19 November 2017
2268C.005-SPM-CLI-SDD-01 Rev. 01		Page 56

- (5) **SPME CLI extra arguments** (text field): additional arguments to SPM CLI tool. Only options -gs, -gg, -n and -t are supported;
- (6) **Extract data flag**: true if run python script to extract data (post process) generated by SPME CLI tool;
- (7) **Extract data script** (text field): python script to extract data. This field allows the user to select alternative python script;
- (8) **Extract data script extra arguments** (text field): additional arguments to extract data script. This field is intended to provide arguments for customized scripts; and
- (9) **Extract data output directory** (text field): directory containing the data generated by extract data script.

DOCUMENT CONTROL DATA		
*Security markings for the title, authors, abstract and keywords must be entered when the document is sensitive		
1. ORIGINATOR (Name and address of the organization preparing the document. A DRDC Centre sponsoring a contractor's report, or tasking agency, is entered in Section 8.)  <b>Thales Systems Canada</b> <b>1 Chrysalis Way</b> <b>Ottawa, ON K2G 6P9</b>		2a. SECURITY MARKING (Overall security marking of the document including special supplemental markings if applicable.)  <b>CAN UNCLASSIFIED</b>
		2b. CONTROLLED GOODS  <b>NON-CONTROLLED GOODS</b> <b>DMC A</b>
3. TITLE (The document title and sub-title as indicated on the title page.)  <b>Security Posture Metrics (SPM) Command Line CLI (CLI) Software Design Description (SDD)</b> <b>Defence Research Development Canada (DRDC) Cyber Decision Making and Response (CDMR)</b> <b>Project</b>		
4. AUTHORS (Last name, followed by initials – ranks, titles, etc., not to be used)  <b>Le Van Mao, R</b>		
5. DATE OF PUBLICATION (Month and year of publication of document.)  <b>November 2017</b>	6a. NO. OF PAGES (Total pages, including Annexes, excluding DCD, covering and verso pages.)  <b>56</b>	6b. NO. OF REFS (Total references cited.)  <b>6</b>
7. DOCUMENT CATEGORY (e.g., Scientific Report, Contract Report, Scientific Letter.)  <b>Contract Report</b>		
8. SPONSORING CENTRE (The name and address of the department project office or laboratory sponsoring the research and development.)  <b>DRDC - Ottawa Research Centre</b> <b>Defence Research and Development Canada</b> <b>3701 Carling Avenue</b> <b>Ottawa, Ontario K1A 0Z4</b> <b>Canada</b>		
9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)	9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)  <b>W7714-155991</b>	
10a. DRDC PUBLICATION NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)  <b>DRDC-RDDC-2018-C097</b>	10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)  <b>2268C.005-SPM-CLI-SDD-01 Rev. 01</b>	
11a. FUTURE DISTRIBUTION WITHIN CANADA (Approval for further dissemination of the document. Security classification must also be considered.)  <b>Public release</b>		
11b. FUTURE DISTRIBUTION OUTSIDE CANADA (Approval for further dissemination of the document. Security classification must also be considered.)		

12. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Use semi-colon as a delimiter.)

Cyber Security Metrics; Metrics; Computer Network Defence (CND); Software Design and Architecture

13. ABSTRACT/RÉSUMÉ (When available in the document, the French version of the abstract must be included here.)