



Working Together to Build Confidence

USER'S GUIDE

BLADE TOIF OPEN SOURCE SOFTWARE

Document Version 1.4

Software Release 2.3.1



This document applies to Blade Tool Output Integration Framework (TOIF) Open Source Software (OSS) version 2.3.1 and may apply to subsequent releases. To check for newer versions of this document, please contact KDM Analytics Customer Support (e-mail: support@kdmanalytics.com).

Copyright © 2006-2016 KDM Analytics, Inc. All rights reserved.

KDM Analytics and KDM Analytics logo are trademarks of KDM Analytics, Inc. in the United States and/or other countries. All other names are trademarks or registered trademarks of their respective companies.

While every attempt has been made to ensure that the information in this document is accurate and complete, some typographical errors or technical inaccuracies may exist. KDM Analytics does not accept responsibility for any kind of loss resulting from the use of information contained in this document.

Due to continued product development the information contained in this document may change without notice. Any improvements or changes to either the product or the document will be documented in subsequent versions.

The software/documentation contains proprietary information of KDM Analytics, Inc. It is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. All rights are reserved. Reverse engineering of the software is prohibited. No part of this software/documentation may be copied, photocopied, reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or translated in another language without the prior written permission of KDM Analytics, Inc.

KDM Analytics™ is a trademark of KDM Analytics, Inc.

Microsoft®, Internet Explorer, Windows®, Windows NT®, Windows® 2000, Windows® 2000 Server, Windows® Server 2003, Windows® XP, MS-DOS™, Microsoft Visual Studio®, Microsoft .NET, and Microsoft Visual C++ are trademarks of Microsoft Corporation. Oracle®, the Oracle Logo, Java™ and all Java based trademarks and logos are trademarks or registered trademarks of Oracle America and/or its affiliates in the United States and other countries. Eclipse™ is a trademark of Eclipse Foundation, Inc. FindBugs and the FindBugs logo are trademarked by the University of Maryland. RATS is distributed by Secure Software, and owned by Fortify Software, Inc. WebGoat is a project maintained by Open Web Application Security Project (OWASP).

KDM Analytics, Inc.
1956 Robertson Rd
Suite 204
Ottawa, ON
Canada
(613) 627-1010

Internet E-Mail: support@kdmanalytics.com
Website: <http://www.kdmanalytics.com>

Contents

Introduction	1
Audience	1
Typographical Conventions.....	2
Getting Help	3
When to contact us	3
How does Blade TOIF OSS Work?	5
Blade TOIF OSS Components	6
Preparing to Install Blade TOIF OSS	7
Installation Overview	7
Supported Deployment Configurations	8
System Requirements	8
Client Hardware	8
Client Software.....	8
Installing Blade TOIF OSS Packages	11
Verify Blade TOIF OSS and TSV Output Installation	12
Blade TOIF OSS RCP	12
TSV Output	12
Installing Open Source Software (OSS) Static Code Analysis (SCA) Tools.....	13
Running the TOIF Adaptor	15
Running the TOIF Adaptor from the command line	15
Integrating with C project's build.....	16
Integrating with Java project's build	16
Running the TOIF Assimilator	17
TSV Output	19
Running the TSV Output.....	19

TOIF Findings View	21
<hr/>	
Upgrading the TOIF Findings View from 2.2.0 to 2.3.1 in Eclipse	21
New Install of TOIF Findings View in Eclipse	22
Opening the TOIF Findings View in Eclipse	23
Importing Vulnerability Findings from a KDM File to a General Project in Eclipse.....	24
Prioritizing Vulnerability Findings Reported in the TOIF Findings view	26
Re-order vulnerability type (SFP/CWE) to set order of importance	26
Setting the SFP/CWE Confidence levels per Vulnerability Detection Tool	27
Show or Hide Vulnerability Type (SFP/CWE) from being reported in TOIF Findings View	28
Exporting and Importing the TOIF Findings Configuration Preferences Table Settings.....	29
Scoping Vulnerability Findings in the TOIF Findings View	31
Search and term filter	31
Filter Options.....	31
Sorting by Column Heading.....	32
Filter on a Selected Project Source File(s).....	33
Analyzing and Citing Vulnerability Findings in the TOIF Findings View	34
Exporting Cited Vulnerability Findings to a *.tsv file	35
Importing Cited Vulnerability Findings from a *.tsv file	36
TOIF Findings View Interface	37
<hr/>	
TOIF Findings View toolbar	37
View Menu	37
Export Selection	39
Export Coverage	39
Defect Description.....	39
User Defined Sort Order.....	39
Sort by Multiple Findings	40
Term Filter Field	40
TOIF Findings View Context Sensitive Menu Options	40
Not a Weakness	40
Is a Weakness.....	40
Uncite Weakness.....	40
Trace.....	41
More Information.....	41
TOIF Findings View Column Sorting	41

Contents	iii
Known Limitations	43
<hr/>	
Traceback displays numbers as reported by TOIF Adaptors.....	43
Jlint not supported on Fedora or RHE platform.....	43
Glossary of Terms	45
<hr/>	
Index	47
<hr/>	

Introduction

The *Blade Tool Output Integration Framework (TOIF) Open Source Software (OSS)* is a powerful vulnerability detection platform. It allows users to perform vulnerability sightings on a project utilizing multiple open source software (OSS) static code analysis (SCA) tools, and analyze the results in a common format using a single viewer. *Blade TOIF OSS* provides:

- ▶▶ Integration of multiple vulnerability detection tools and their findings as “data feeds” into a common repository
 - ▶▶ Addressing wider breadth and depth of vulnerability coverage
 - ▶▶ Common processing of results
- ▶▶ Normalization and correlation of “data feeds” based on discernable patterns described as Software Fault Patterns (SFPs) and Common Weakness Enumerations (CWEs)
- ▶▶ Collated SFP/CWE findings
- ▶▶ Prioritized report output with weighted results across tools/vendors
 - ▶▶ Exporting in XML and TSV format for further analysis in spreadsheet
- ▶▶ Utilization of open source development to advance the Software Assurance space
- ▶▶ A standard-based common protocol for exchanging vulnerability findings

Blade TOIF OSS is based on existing standard protocol for exchanging system facts, the OMG Knowledge Discovery Metamodel (KDM), now ISO/IEC 19506.

This document contains information about how to use *Blade TOIF OSS*. It includes procedures, notes, and other background information.

Audience

This document is intended to help system and software engineers, system analysts, security analysts, and system architects to use the *Blade TOIF OSS* in performing vulnerability sightings. *Blade TOIF OSS* users must have experience working with command line tools.

Typographical Conventions

Before you start using this guide, it is important to understand the terms and typographical conventions used in the documentation.

The table below identifies the formatting conventions used in KDM Analytics documents to represent different types of information.

Type of information	Formatting convention	Example
Slash characters in path names	Follow the UNIX convention (forward slash). Usually appears in <code>monospace font</code> as part of command-line input or output. Note: Substitute with a backward slash (\) for Windows platform.	<code>/workspace/Blade_TOIF</code>
File path and names	<code>monospace font</code>	Double click on the <code>eclipse.exe</code> file.
Information to be substituted with user-provided information	Description of information item to be supplied surrounded by angle brackets, <code>monospace font</code>	Extract the <code>eclipse-cpp-kepler-SR2-<platform version>.zip</code> file into a desired target folder.
Filename in descriptive text	<i>italics</i>	Locate the <i>com.kdmanalytics.toif.p2-2.3.1.zip</i> file
Step-by-step procedures.	1., 2., 3.,...	1) Click Start 2) Click Advanced tab
User interface fields and menu options	SMALL CAPITALS	Right click on MY COMPUTER and from
User interface command buttons	Bold	1) Click Start
Names of keys on the keyboard.	CAPITALS	SHIFT, CTRL, or ALT.
Key combinations for which the user must press and hold down one key and then press another	KEY+KEY	CTRL+P, or ALT+F4.

For more information on specialized terms used in the documentation, see the Glossary at the end of this document.

Getting Help

This PDF version of the User Guide is available as part of KDM Analytics product distribution.

The following sections provide details on when and how to contact KDM Analytics support if you encounter a problem and cannot resolve the issue after consulting the published information.

When to contact us

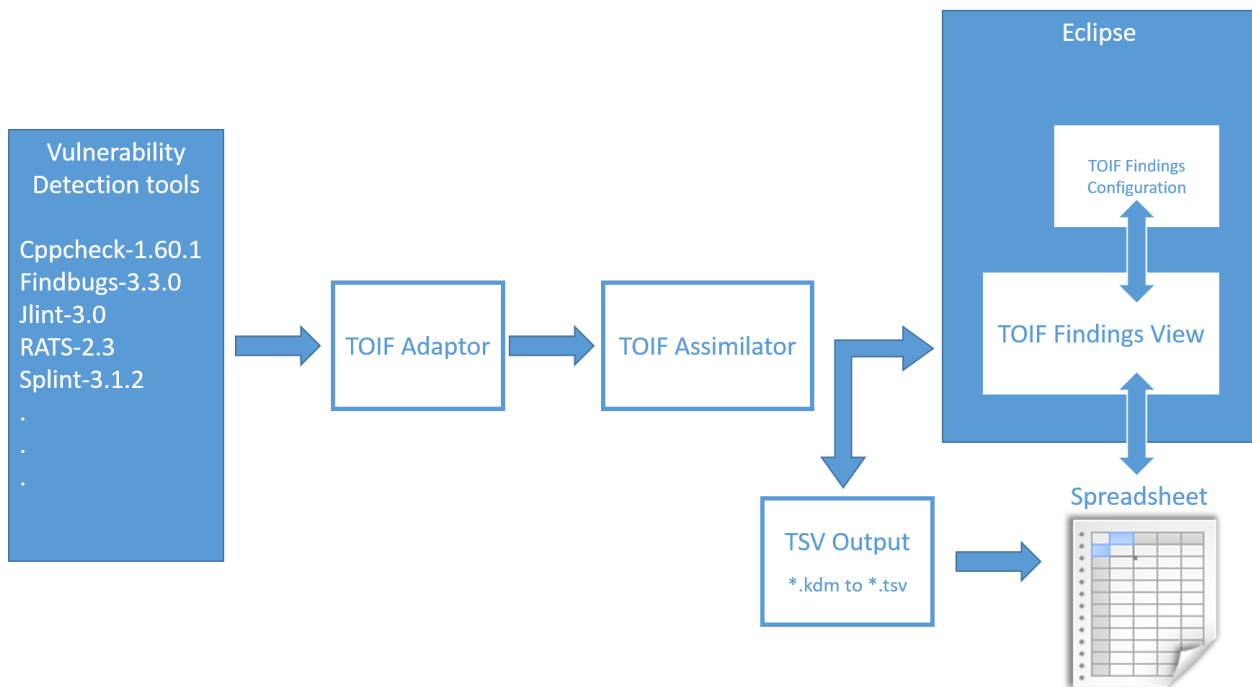
If you encounter a problem or deficiency working with our product and are unable to resolve it after consulting the published information, please contact KDM Analytics support by e-mail: support@kdmanalytics.com.

Chapter 1

How does Blade TOIF OSS Work?

Blade TOIF OSS takes the output of the supported Open Source Software (OSS) Static Code Analysis (SCA) tools (see page 13) and assimilates the vulnerability findings. The vulnerability findings can then be directly imported into Eclipse to view in the TOIF Findings View and exported to a tab-separated-values (tsv) format so that the results can be viewed by text editors or spreadsheets.

The following diagram illustrates the *Blade TOIF OSS* components and workflow.



Blade TOIF OSS Components

Blade TOIF OSS includes the following components:

- ▶▶ **TOIF Adaptor:** *TOIF Adaptor* is used to collect the output from various Open Source Software (OSS) Static Code Analysis (SCA) tools and convert their output into TOIF xml
- ▶▶ **TOIF Assimilator:** After running the *TOIF Adaptor* you need to run the Assimilator to merge TOIF findings and/or KDM data into a common fact-orientated repository or file
- ▶▶ **TOIF Findings View:** Once you have your TOIF findings assimilated you can use the *TOIF Findings View* to display the results in *Eclipse*. The TOIF Findings Configuration preferences table allows you to prioritize how vulnerability types and their associated confidence values are or are not reported in the TOIF Findings view. The report can also be exported to a *.tsv format to display the results in a text editor or spreadsheet
- ▶▶ **TSV Output:** Once you have your TOIF findings assimilated you can use the *TSV Output* to convert *.kdm to *.tsv to display the results in a text editor or spreadsheet

Chapter 2

Preparing to Install Blade TOIF OSS

The following sections provide the information required to install and get you working in the *Blade TOIF OSS*.

Installation Overview

This section provides the high-level steps for installing and running Blade TOIF OSS and viewing results in *Eclipse*.

- 1) Gather the installation packages for *Blade TOIF OSS* and *TOIF Findings View*.
 - Blade TOIF OSS package
 - *kdmanalytics-oss-toif-2.3.1.win32.win32.x86_64.zip* (windows)
 - *kdmanalytics-oss-toif-2.3.1.linux.gtk.x86_64.tar.gz* (linux)
 - TSV Output package
 - *kdmanalytics-oss-toif-tsvoutput-2.3.1.win32.win32.x86_64.zip* (windows)
 - *kdmanalytics-oss-toif-tsvoutput-2.3.1.linux.gtk.x86_64.tar.gz* (linux)
 - TOIF Findings view package: *com.kdmanalytics.toif.p2-2.3.1.zip*
- 2) Ensure that **Eclipse 4.4.2 (Luna)*** has been installed (<https://eclipse.org/downloads/packages/release/Luna/SR2>).
- 3) Read all of the information in this chapter before you install all the installation packages.
- 4) Unzip and install the *Blade TOIF OSS* package.
- 5) Ensure that the supported OSS SCA tools you want to run Blade TOIF OSS with are installed. See “**Installing Open Source Software (OSS) Static Code Analysis (SCA) Tools**” on page 13 for more information.
- 6) Run the *TOIF Adaptor* with the desired Open Source Software (OSS) Static Code Analysis (SCA) tool.
- 7) Run the *TOIF Assimilator* against the TOIF files generated by the *TOIF Adaptor*.
- 8) Decide if you are performing an upgrade or a new install of the *TOIF Findings* view in your Eclipse* instance or if you want to view vulnerability findings in a text file or spreadsheet.
- 9) Do one of the following to view the results:
 - a) Create a project and import the *.kdm file that was generated from running the TOIF Assimilator into an Eclipse* project and view it in the *TOIF Findings* view in **Eclipse**or

- b) convert the *.kdm to a tab-separated-values format so that the results can be viewed in a text editor or spreadsheet.

*Note, the *TOIF Findings View* is supported in Eclipse 4.4.2 (Luna) or Eclipse 4.3.2 (Kepler). Do not attempt to install the *TOIF Findings* view in another version of Eclipse.

Supported Deployment Configurations

The *Blade TOIF OSS* is a standalone program. The *TOIF Findings* view must be used within an instance of *Eclipse 4.4.2* (Luna).

System Requirements

To install Blade TOIF OSS your system must meet the minimum hardware and software requirements listed in the following sections.

Client Hardware

The following table lists the supported hardware platforms for Blade TOIF OSS.

Minimum	Recommended
500 MB free hard drive space	
2 GB RAM	8 GB RAM
Dual Core Processor	Quad core processor

Client Software

The following table lists the supported operating system platforms for Blade TOIF OSS.

Platform	Recommended
SUSE Enterprise Linux Desktop 12 SP1 (64 bit)	
Red Hat Enterprise Linux Desktop 7.1 (64 bit)	Red Hat Enterprise Linux Desktop 7.1 (64 bit)
Ubuntu 14.04.1 (64 bit)	
Fedora 17.x (64 bit)	
Microsoft Windows 7 (64 bit)	Microsoft Windows 10 (64 bit)

Microsoft Windows 10 (64 bit)

Additional Software

Recommended

Eclipse 4.4.2 (Luna) SR2 Eclipse IDE for C/C++ Developers or Eclipse IDE for Java Developers package

Eclipse 4.4.2 (Luna) IDE for C/C++ Developers, with JDT plugin

Eclipse 4.3.2 (Kepler) Eclipse IDE for C/C++ Developers or Eclipse IDE for Java Developers package

Chapter 3

Installing Blade TOIF OSS Packages

To install the *Blade TOIF OSS* for linux or windows, and the *TOIF Findings View* packages perform the following steps.

1) Download:

- ▶ Blade TOIF OSS
 - a) `kdmanalytics-oss-toif-2.3.1.win32.win32.x86_64.zip` (windows)
 - b) `kdmanalytics-oss-toif-2.3.1.linux.gtk.x86_64.tar.gz` (linux)
- ▶ TSV Output
 - a) `kdmanalytics-oss-toif-tsvoutput-2.3.1.win32.win32.x86_64.zip` (windows)
 - b) `kdmanalytics-oss-toif-tsvoutput-2.3.1.linux.gtk.x86_64.tar.gz` (linux)
- ▶ TOIF Findings view
 - a) `com.kdmanalytics.toif.p2-2.3.1.zip`.

Note: These files are located on the release page of GitHub and the KDM Analytics, Inc. website (www.kdmanalytics.com/toif/download.html).

2) Run an unzip utility to extract the:

- ▶ Blade TOIF OSS
 - a) `kdmanalytics-oss-toif-2.3.1.win32.win32.x86_64.zip` (windows) or
 - b) `kdmanalytics-oss-toif-2.3.1.linux.gtk.x86_64.tar.gz` (linux)into a desired target folder.
- ▶ TSV Output
 - a) `kdmanalytics-oss-toif-tsvoutput-2.3.1.win32.win32.x86_64.zip` (windows) or
 - b) `kdmanalytics-oss-toif-tsvoutput-2.3.1.linux.gtk.x86_64.tar.gz` (linux)into a desired target folder.

Verify Blade TOIF OSS and TSV Output Installation

To verify that the *Blade TOIF OSS* package have installed correctly, perform the following tasks.

Blade TOIF OSS RCP

- 1) Open a terminal or command prompt window and navigate into the Blade TOIF OSS installation directory.

For example, C:\toif-2.3.1

- 2) Do one of the following:

- a) Windows: type `toif --version`

- b) Linux: type `./toif --version`

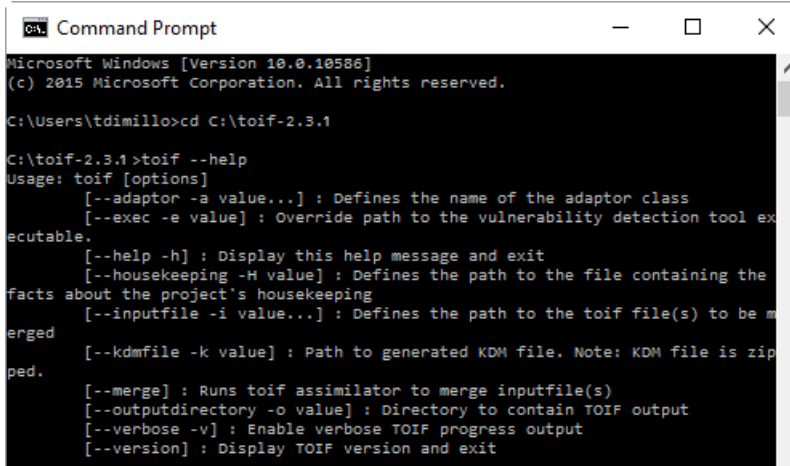
The terminal or command prompt displays `Version=2.3.1`.

- 3) Do one of the following:

- a) Windows: type `toif --help`

- b) Linux: type `./toif --help`

The terminal or command prompt displays the following:



```
Command Prompt
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\tdimillo>cd C:\toif-2.3.1

C:\toif-2.3.1>toif --help
Usage: toif [options]
    [--adaptor -a value...] : Defines the name of the adaptor class
    [--exec -e value] : Override path to the vulnerability detection tool executable.
    [--help -h] : Display this help message and exit
    [--housekeeping -H value] : Defines the path to the file containing the facts about the project's housekeeping
    [--inputfile -i value...] : Defines the path to the toif file(s) to be merged
    [--kdmfile -k value] : Path to generated KDM file. Note: KDM file is ziped.
    [--merge] : Runs toif assimilator to merge inputfile(s)
    [--outputdirectory -o value] : Directory to contain TOIF output
    [--verbose -v] : Enable verbose TOIF progress output
    [--version] : Display TOIF version and exit
```

TSV Output

- 1) Open a terminal or command prompt window and navigate to the *TSV Output* installation directory.

For example, C:\tsvoutput-2.3.1

- 2) Do one of the following:

- a) Windows: type `tsvoutput --version`

b) Linux: type `./tsvoutput --version`

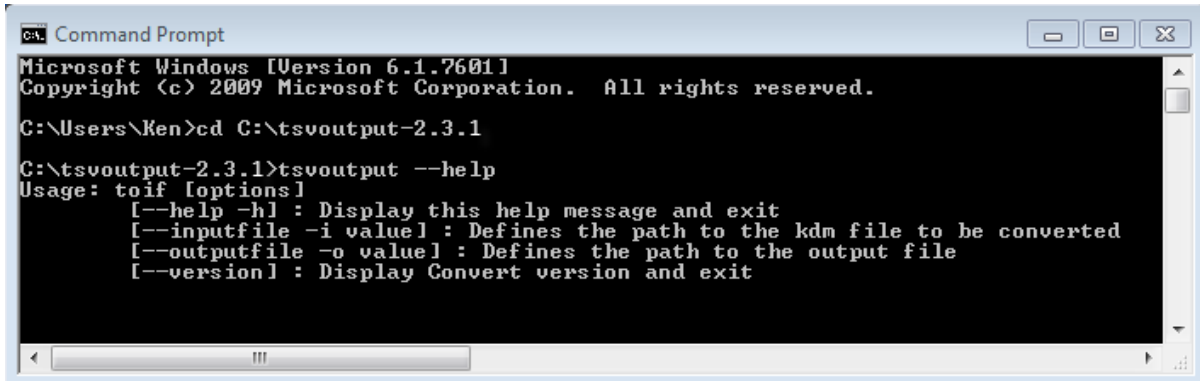
The terminal or command prompt displays `Version=2.3.1`.

3) Do one of the following:

a) Windows: type `tsvoutput --help`

b) Linux: type `./tsvoutput --help`

The terminal or command prompt displays the following:



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Ken>cd C:\tsvoutput-2.3.1

C:\tsvoutput-2.3.1>tsvoutput --help
Usage: toif [options]
  [--help -h] : Display this help message and exit
  [--inputfile -i value] : Defines the path to the kdm file to be converted
  [--outputfile -o value] : Defines the path to the output file
  [--version] : Display Convert version and exit
```

Installing Open Source Software (OSS) Static Code Analysis (SCA) Tools

Install the following open source software (OSS) static code analysis (SCA) tools according to their own instructions. The supported input file types are listed beside each OSS SCA tool.

- ▶▶ **Cppcheck-1.60.1:** .c and .cpp files
- ▶▶ **Findbugs-3.0.0:** .class files
 - ▶ **Find Security Bugs Plugin-1.2.1:** .class files

Note: The Find Security Bugs Plugin is installed by placing the `findsecbugs-plugin-1.2.1.jar` in the Findbugs 3.0.0 plugins folder.

- ▶▶ **Jlint-3.0 (Ubuntu and Windows only):** .class files
- ▶▶ **Rats-2.3:** .c and .cpp files
- ▶▶ **Splint-3.1.2:** .c files

Make sure that you are using the release 2.3.1 of the Blade TOIF OSS application with these versions of the OSS SCA tools. The executable for each tool should be on the system path. A simple check would be to open a command prompt and type the following for each respective OSS SCA tool: `cppcheck --version`, `splint --version`, `findbugs --version`, `jlint -help`, and `rats`. The system will respond after each command with either a version number, for example, `Cppcheck 1.60.1`, or in the case of `jlint -help` some information that describes available options and message categories and in the case of `rats` some run information such as entries and Total lines analyzed.

Chapter 4

Running the TOIF Adaptor

This section will discuss the possible ways of running the TOIF Adaptor:

- ▶ Running the TOIF Adaptor from command line
- ▶ Integrating with a C or Java project's build

Running the TOIF Adaptor from the command line

To run the Adaptor from a command line, perform the following steps.

- 1) Open a command prompt.
- 2) Make sure that the windows or linux *Blade TOIF OSS "toif"* command is on the PATH. Also, ensure that the OSS SCA tools are correctly installed.
- 3) Type `toif --adaptor=<Adaptor Name> --inputfile=<full path to input file> --outputdirectory=<path to output directory> --housekeeping=<path to housekeeping file> -- [Additional arguments]`

Where:

- ▶ `<Adaptor Name>` defines the name of the adaptor class. This is the adaptor that is to be used with the input source file. From this class, the framework is able to discover housekeeping facts about the adaptor as well as which OSS SCA tool to call and what options to use.
- ▶ `<full path to input file>` defines the full path to the input source file. In order for the adaptors to create all the facts for this file, a full path must be provided.
- ▶ `<path to output directory>` defines the path to the output directory. This is the directory where the TOIF XML file will be written.
- ▶ `<path to housekeeping file>` defines the path to the file containing the facts about the project's housekeeping. This file is specific to each adaptor and each project. This is because it is down to the user to provide the project details as well as which OSS SCA tool is running on the system. An example Housekeeping file is located in the *Examples* directory.
- ▶ `[Additional Arguments]` defines any additional arguments that you may want. These must be entered after the TOIF Adaptor's required arguments and after a "--". These arguments can be included files or compilation options, and they will vary from tool to tool. For example, splint can take -I and -D options:

```
./toif --adaptor=splint --inputfile=/home/user/foo.c --  
outputdirectory=/home/user/toifFiles --  
housekeeping=housekeepingFile.txt -- -I./includes -D_U_ =
```

Warning: Do not copy the commands from the manual as the "--" may be pasted as "-" in the command prompt window and will cause an "unexpected argument" error.

Integrating with C project's build

The best way to integrate the adaptors into the build is by wrapping the compiler and the adaptors into a script. When the compiler is called, the adaptors will be run for every source file used. To get the build process to use this wrapper instead of the compiler on its own, the compiler flag needs to be set during the make:

```
./configure
```

The make can then be continued with configuration as:

```
▶▶ make CC=<path to myGccWrapper>  
▶▶ make install
```

An example script has been provided in the Examples folder. However, the following directories will need to be changed within the script to suit your system.

```
▶▶ HOUSE_KEEPING = <the location of the housekeeping file>  
▶▶ OUTPUT_DIR = <the output directory for the toif files>
```

Integrating with Java project's build

It may be possible to integrate into a Java project's build by adding the following to the *build.xml* file.

```
<!-- my target -->  
<target name="mytarget" depends="build">  
  <apply executable="python">  
    <fileset dir="${build.dir}">  
      <patternset>  
        <include name="**/*.class"/>  
      </patternset>  
    </fileset>  
    <arg value="/TOIF/javaAdaptors.py"/>  
    <srcfile/>  
  </apply>  
</target>
```

This creates a new target which will find all the *.class* files in the destination directory of the project. For each file, the *javaAdaptors* python script will be run with the arguments that are specified. An example script is provided in the Examples folder in the *Blade TOIF OSS* installation directory. The script is for reference only and you will have to write your own to be compatible with your system and project.

Chapter 5

Running the TOIF Assimilator

The *TOIF Assimilator* merges TOIF findings (toif files created by running the TOIF Adaptor) and/or KDM data into a common fact-orientated repository or file.

To run the *TOIF Assimilator*, perform the following steps.

- 1) Open a command prompt.
- 2) Make sure that the windows or linux Blade TOIF OSS “toif” is on the PATH.

Do the following:

- ▶ Type `toif --merge --kdmfile=<output destination> --inputfile=[files or directories to merge...]`.

Note: The file extension for the output destination must be `.kdm`.

Where:

- ▶ `<output destination>` The path and filename of the `.kdm` file should be specified as the destination.
- ▶ `[files or directories to merge...]` defines the toif files to be merged. Any number of toif files can be entered here. Alternatively, a directory can be specified which contains the toif files.

For example,

```
./toif --merge --kdmfile=/home/user/outputFile.kdm --  
inputfile=/home/user/toifFiles/
```

Warnings:

Ensure that the output file is not produced where the input files are being read from.

If running the assimilator on large projects with many files, the default toif memory setting may need to be increased. Memory allocation can be updated in the `toif.ini` file, located in the toif install directory, by modifying the “-Xmx” parameter.

We recommend that you do not copy the commands from the manual as the “--” may be pasted as “—” in the command prompt window and will cause “unexpected argument” errors.

The resulting assimilated output data, from running the *TOIF Assimilator*, is used as the input to the *TOIF Findings View*.

Note: The output of the TOIF Assimilator, `<output file name>.kdm`, is a zip file (the `.zip` extension may not be visible) and you will need to extract it to a `<output file name>` directory and use the `*.kdm` file that resides in the

directory to complete the subsequent steps. For example, if your output file is *outputFile.kdm.zip* you would extract it to a directory named *outputFile* and in this directory you would have an *outputFile.kdm* file

Chapter 6

TSV Output

The *TSV Output* provides an alternative to viewing the assimilated output results without having to install the TOIF Findings View in Eclipse. The *TSV Output* allows the assimilated output data (vulnerability findings) to be converted to a tab-separated-values format so that it can be viewed by text editors or spreadsheets. Once imported into a text editor or spreadsheet the vulnerability finding data can be used by analysts who are performing vulnerability citations on a project.

Running the TSV Output

To run the *TSV Output*, perform the following steps.

- 1) Open a terminal or command prompt window.
- 2) Navigate to the location on your file system where you installed the *TSV Output*.
- 3) Do one of the following:
 - a) Windows: type `tsvoutput -i <full path to the input file> -o <full path to output file>`
 - b) Linux: type `./tsvoutput -i <full path to the input file> -o <full path to output file>`

Note: The input file must be *.kdm file

Where:

- ▶ `<full path to input file>` defines the full path to the input source file. The input file must be a *.kdm created by running the TOIF Assimilator.
- ▶ `<path to output file>` defines the path to the output file. This is the directory where the *.tsv file will be written.

For example,

```
tsvoutput -i test.kdm -o test.tsv
```

- 4) Import the *.tsv into a text editor or spreadsheet to view the vulnerability findings.

The following image is an example of a tsvoutput (*.tsv) file in Microsoft Excel. The line 4 is magnified to show details.

SFP	CWE	Citing Status	Confidence	Resource	Line Number	KDM Line Number	SCA tool	Weakness Description
1	SFP--1	CWE-227	0	/TOIF_1.16.0/tests/WebGoatTest/AbstractLesson.java	185	185	Findbugs + Security Plugin	NP_EQUALS_SHOULD_HANDLE_NULL_ARGUMENT: This implementation of equals(Object) violates the contract defined by java.lang.
2	SFP--1	CWE-398	0	/TOIF_1.16.0/tests/WebGoatTest/AbstractLesson.java	185	185	Findbugs + Security Plugin	BC_EQUALS_METHOD_SHOULD_WORK_FOR_ALL_OBJECTS: The equals(Object) method shouldn't make any assumptions about the
3	SFP-24	CWE-172	0	/TOIF_1.16.0/tests/WebGoatTest/AbstractLesson.java	388	388	Findbugs + Security Plugin	DM_DEFAULT_ENCODING: Found a call to a method which will perform a byte to String (or String to byte) conversion, and will assum
4	SFP-24	CWE-172	0	/TOIF_1.16.0/tests/WebGoatTest/AbstractLesson.java	506	506	Findbugs + Security Plugin	DM_DEFAULT_ENCODING: Found a call to a method which will perform a byte to String (or String to byte) conversion, and will assum
5	SFP-16	CWE-22	0	/TOIF_1.16.0/tests/WebGoatTest/AbstractLesson.java	506	506	Findbugs + Security Plugin	DM_TRAVERSAL_IN_SECURITY_PLUGIN: Potential Path Traversal (File Read)
6	SFP-24	CWE-172	0	/TOIF_1.16.0/tests/WebGoatTest/AbstractLesson.java	524	524	Findbugs + Security Plugin	DM_DEFAULT_ENCODING: Found a call to a method which will perform a byte to String (or String to byte) conversion, and will assum
7	SFP-16	CWE-22	0	/TOIF_1.16.0/tests/WebGoatTest/AbstractLesson.java	524	524	Findbugs + Security Plugin	DM_TRAVERSAL_IN_SECURITY_PLUGIN: Potential Path Traversal (File Read)
8	SFP-24	CWE-172	0	/TOIF_1.16.0/tests/WebGoatTest/AbstractLesson.java	470	470	Findbugs + Security Plugin	DM_DEFAULT_ENCODING: Found a call to a method which will perform a byte to String (or String to byte) conversion, and will assum
9	SFP-16	CWE-22	0	/TOIF_1.16.0/tests/WebGoatTest/AbstractLesson.java	470	470	Findbugs + Security Plugin	DM_DEFAULT_ENCODING: Found a call to a method which will perform a byte to String (or String to byte) conversion, and will assum
10	SFP-24	CWE-172	0	/TOIF_1.16.0/tests/WebGoatTest/AbstractLesson.java	699	699	Findbugs + Security Plugin	DM_DEFAULT_ENCODING: Found a call to a method which will perform a byte to String (or String to byte) conversion, and will assum
11	SFP-16	CWE-22	0	/TOIF_1.16.0/tests/WebGoatTest/AbstractLesson.java	699	699	Findbugs + Security Plugin	DM_DEFAULT_ENCODING: Found a call to a method which will perform a byte to String (or String to byte) conversion, and will assum
12	SFP-1	CWE-398	0	/TOIF_1.16.0/tests/WebGoatTest/AbstractLesson.java	124	124	Findbugs + Security Plugin	DM_NUMBER_CTOR: Using new Integer(int) is guaranteed to always result in a new object whereas Integer.valueOf(int) allows cachi
13	SFP-1	CWE-398	0	/TOIF_1.16.0/tests/WebGoatTest/AbstractLesson.java	124	124	Findbugs + Security Plugin	DM_NUMBER_CTOR: Using new Integer(int) is guaranteed to always result in a new object whereas Integer.valueOf(int) allows cachi
14	SFP-1	CWE-398	0	/TOIF_1.16.0/tests/WebGoatTest/AbstractLesson.java	143	143	Findbugs + Security Plugin	DM_NUMBER_CTOR: Using new Integer(int) is guaranteed to always result in a new object whereas Integer.valueOf(int) allows cachi
15	SFP-14	CWE-772	0	/TOIF_1.16.0/tests/WebGoatTest/AbstractLesson.java	620	620	Findbugs + Security Plugin	SQL_NONCONSTANT_STRING_PASSED_TO_EXECUTE: The method invokes the execute method on an SQL statement with a String tha
16	SFP-24	CWE-89	0	/TOIF_1.16.0/tests/WebGoatTest/AbstractLesson.java	622	622	Findbugs + Security Plugin	SQL_NONCONSTANT_STRING_PASSED_TO_EXECUTE: The method invokes the execute method on an SQL statement with a String tha
17	SFP-7	CWE-476	0	/TOIF_1.16.0/tests/WebGoatTest/Course.java	299	299	Findbugs + Security Plugin	NP_LOAD_OF_KNOWN_NULL_VALUE: The variable referenced at this point is known to be null due to an earlier check against null. A
18	SFP-16	CWE-22	0	/TOIF_1.16.0/tests/WebGoatTest/Course.java	98	98	Findbugs + Security Plugin	DM_TRAVERSAL_IN_SECURITY_PLUGIN: Potential Path Traversal (File Read)
19	SFP-33	CWE-259	0	/TOIF_1.16.0/tests/WebGoatTest/DatabaseUtilities.java	112	112	Findbugs + Security Plugin	DM_EMPTY_DB_PASSWORD: This code creates a database connect using a blank or empty password. This indicates that the database
20	SFP-24	CWE-78	0	/TOIF_1.16.0/tests/WebGoatTest/Exec.java	289	289	Findbugs + Security Plugin	COMMAND_INJECTION: SECURITY_PLUGIN: Potential Command Injection
21	SFP-24	CWE-78	0	/TOIF_1.16.0/tests/WebGoatTest/Exec.java	107	107	Findbugs + Security Plugin	COMMAND_INJECTION: SECURITY_PLUGIN: Potential Command Injection
22	SFP-24	CWE-172	0	/TOIF_1.16.0/tests/WebGoatTest/Exec.java	360	360	Findbugs + Security Plugin	DM_DEFAULT_ENCODING: Found a call to a method which will perform a byte to String (or String to byte) conversion, and will assum
23	SFP-24	CWE-172	0	/TOIF_1.16.0/tests/WebGoatTest/Exec.java	308	308	Findbugs + Security Plugin	DM_DEFAULT_ENCODING: Found a call to a method which will perform a byte to String (or String to byte) conversion, and will assum
24	SFP-24	CWE-172	0	/TOIF_1.16.0/tests/WebGoatTest/Exec.java	178	178	Findbugs + Security Plugin	DM_DEFAULT_ENCODING: Found a call to a method which will perform a byte to String (or String to byte) conversion, and will assum
25	SFP-24	CWE-172	0	/TOIF_1.16.0/tests/WebGoatTest/Exec.java	126	126	Findbugs + Security Plugin	DM_DEFAULT_ENCODING: Found a call to a method which will perform a byte to String (or String to byte) conversion, and will assum
26	SFP-24	CWE-172	0	/TOIF_1.16.0/tests/WebGoatTest/Exec.java	92	92	Findbugs + Security Plugin	DM_DEFAULT_ENCODING: Found a call to a method which will perform a byte to String (or String to byte) conversion, and will assum
27	SFP-24	CWE-172	0	/TOIF_1.16.0/tests/WebGoatTest/Exec.java	91	91	Findbugs + Security Plugin	DM_DEFAULT_ENCODING: Found a call to a method which will perform a byte to String (or String to byte) conversion, and will assum
28	SFP-24	CWE-172	0	/TOIF_1.16.0/tests/WebGoatTest/Exec.java	91	91	Findbugs + Security Plugin	DM_DEFAULT_ENCODING: Found a call to a method which will perform a byte to String (or String to byte) conversion, and will assum
29	SFP-23	CWE-319	0	/TOIF_1.16.0/tests/WebGoatTest/Interceptor.java	88	88	Findbugs + Security Plugin	UNENCRYPTED_SOCKET: SECURITY_PLUGIN: Unencrypted Socket
30	SFP-16	CWE-22	0	/TOIF_1.16.0/tests/WebGoatTest/LegacyLoader.java	70	70	Findbugs + Security Plugin	DM_TRAVERSAL_IN_SECURITY_PLUGIN: Potential Path Traversal (File Read)
31	SFP-5	CWE-396	0	/TOIF_1.16.0/tests/WebGoatTest/LessonAdapter.java	196	196	Findbugs + Security Plugin	REC_CATCH_EXCEPTION: This method uses a try-catch block that catches Exception objects, but Exception is not thrown within the tr
32	SFP-14	CWE-772	0	/TOIF_1.16.0/tests/WebGoatTest/LessonAdapter.java	82	82	Findbugs + Security Plugin	SQL_OPEN_STREAM: The method creates an IO stream object, does not assign it to any fields, pass it to other methods that might clos
33	SFP-24	CWE-172	0	/TOIF_1.16.0/tests/WebGoatTest/LessonAdapter.java	82	82	Findbugs + Security Plugin	DM_DEFAULT_ENCODING: Found a call to a method which will perform a byte to String (or String to byte) conversion, and will assum
34	SFP-16	CWE-22	0	/TOIF_1.16.0/tests/WebGoatTest/LessonAdapter.java	82	82	Findbugs + Security Plugin	DM_DEFAULT_ENCODING: Found a call to a method which will perform a byte to String (or String to byte) conversion, and will assum

Chapter 7

TOIF Findings View

The *TOIF Findings* view allows the assimilated output data to be viewed in *Eclipse*. The output data can be used by analysts who are performing vulnerability citing's on a project. To install *TOIF Findings* view perform one of the following procedures.

Upgrading the TOIF Findings View from 2.2.0 to 2.3.1 in Eclipse

This section describes how to upgrade from *TOIF Findings* view 2.2.0 to 2.3.1, if installing in the same instance of eclipse.

- 1) Download the TOIF Findings view package, `com.kdmanalytics.toif.p2-2.3.1.zip`.
- 2) Open Eclipse.
- 3) From the Eclipse main menu select **HELP->INSTALL NEW SOFTWARE...**

An *Install* dialog opens.

- 4) Click the **"Available Software Site"** link.

The *Preferences* dialog opens to display the *Available Software Sites* pane. The table lists all available software sites for the current instance of Eclipse.

- 5) From the table select the software site category that was used to install the previous version of TOIF Findings View.

For example, *TOIF Findings*.

- 6) Click the **Edit** button.

The *Edit Site* dialog opens to display the Name and Location of the current software site.

- 7) Click in the **LOCATION:** text field and enter the path to the `com.kdmanalytics.toif.p2-2.3.1.zip` file.

The location of the category related to the *TOIF Findings* View is updated.

- 8) Click **OK** in the **Edit Site** dialog.

The *Edit Site* dialog closes and under the *Location* column of the category related to *TOIF Findings View* is the path to the new *TOIF Findings View* release package.

- 9) Click check box in front of category related to *TOIF Findings View* in the **Available Software** panel.

A check mark appears to indicate that category related to *TOIF Findings View* has been selected.

- 10) Click **OK** in the **Preferences** dialog.

The *Preferences* dialog closes.

- 11) Click the down arrow in the **WORK WITH:** text box and select the software site category related to *TOIF Findings View*.

- 12) Go to step 10) in the **New Install of TOIF Findings View in Eclipse** procedure and follow the remaining steps to complete the installation.

New Install of TOIF Findings View in Eclipse

To install the *TOIF Findings* view in *Eclipse 4.4.2 (Luna)* perform the following steps.

- 1) Start Eclipse.

- 2) Choose the workspace location.

This is home to the eclipse user/session data and also any projects which are created.

- 3) Close the welcome screen.

- 4) Click **HELP** in the tool bar menu and from the drop down menu select **INSTALL NEW SOFTWARE...**

The *Install* dialog opens.

- 5) Click **Add...**

The *Add Repository* dialog opens.

- 6) Click in the **NAME:** text field and enter a name for the new software source.

For example, Blade *TOIF OSS*.

- 7) Click **Archive...**

The *Repository Archive* dialog opens.

- 8) Navigate the directories to find the location of the `com.kdmanalytics.toif.p2-2.3.1.zip` file and then click **Open**.

The *Repository Archive* dialog closes and the *Location:* text field in the *Add Repository* dialog is populated.

- 9) Click **OK** in the **Add Repository** dialog.

The *Add Repository* dialog closes and the *SFP/CWE* category appears under the *Name* column in the *Available Software* panel.

Note: The category is displayed only if the *Group items in category* check box is selected. Otherwise, the single feature, *TOIF* is displayed.

- 10) Click **SFP/CWE** to expand the category in the Available **Software** panel.

The feature, *TOIF* is displayed.

- 11) Click to select the *TOIF* feature.

- 12) Click **Next >**.

The *Install Details* panel is displayed.

- 13) Click **Finish**.

The *Install* dialog closes and the *TOIF Findings View* installation starts. Once the installation is complete a dialog will appear requesting that Eclipse be re-started.

Note: If a dialog appears with a warning that the content is unsigned simply click *OK*. This allows the *TOIF Findings View* installation to continue.

- 14) Click **Yes**.

Eclipse session will close and then re-open.

- 15) Click **HELP** in the Eclipse menu bar and from the drop down menu select **ABOUT ECLIPSE**.

The *About Eclipse* dialog opens.

- 16) Click the Installation **Details** button.

The Eclipse Installation Details dialog opens.

- 17) Click the Installed **Software** tab in the **Eclipse Installation Details** dialog to ensure that the TOIF is listed in the table.

TOIF Findings view has been installed in Eclipse.

- 18) Click **CLOSE** in the Eclipse menu bar.

The *Eclipse Installation Details* dialog closes.

- 19) Click **Ok** in the **ABOUT ECLIPSE** dialog

The *About Eclipse* dialog closes.

Opening the TOIF Findings View in Eclipse

The *TOIF Findings* view displays the vulnerability findings from a KDM file generated by running Blade TOIF OSS. To open the *Blade TOIF Findings* view, perform the following steps

- 1) From the Eclipse main menu, select **WINDOW -> SHOW VIEW -> OTHER....**

The Show View dialog opens.

- 2) Navigate to and expand the **SFP/CWE** folder.

The SFP/CWE folder displays its contents, including the *TOIF Findings* view

- 3) Select the **TOIF FINDINGS** view click **OK**.

The *Show View* dialog closes and the *TOIF Findings* view appears in the List panel.

Importing Vulnerability Findings from a KDM File to a General Project in Eclipse

Blade TOIF OSS provides the ability to import findings from a KDM file generated by Blade TOIF OSS 2.3.1 as well as open source TOIF 2.2.0, 2.1.0, or 1.16.0 and view vulnerabilities in the TOIF Findings view. To do this, perform the following steps.

- 1) Right-click in the Project Explorer and select **NEW->PROJECT** from the drop down menu.

A *New Project* dialog is displayed.

- 2) Expand the **General** folder and select **Project**.

The *Next>* button becomes enabled.

- 3) Click **Next>**.

- 4) Type the name of the project inside the **PROJECT NAME:** text field.

The *Finish* button becomes enabled.

- 5) Click **Finish**.

The *New Project* dialog closes and the project appears in the *Project Explorer*.

- 6) (Optional) Add related source files to the project.

The source files related to the KDM file can be imported into the project but they must be from an equivalent environment (source files, includes, defines, and so on). This allows you to **analyze** (see "**Analyzing and Citing Vulnerability Findings in the Findings View**" on page 34) the source files related to vulnerability findings captured in the .kdm file for the project.

- 7) Right-click on the project and select **IMPORT...** from the drop-down menu.

An *Import* dialog opens.

- 8) Expand the **SFP/CWE** folder and select **Import Integrated SFP/CWE File**.

The *Next>* button becomes enabled.

- 9) Click **Next>**.

The *File Import Wizard* appears.

- 10) Select the project from the *Select target project:* list.

- 11) Click the **Browse** button and navigate to and select the KDM file.

The location to the KDM file populates the *Select TOIF Data:* text field and the *Finish* button becomes enabled.

- 12) Click **Finish**.

Note: If step 6) was not performed and your project does not contain the source files related to the KDM file, an *Empty project* dialog will appear indicating that importing sources after KDM file will require the KDM data to be re-imported. Click *Yes* to continue with the import of the KDM file.

13) Select the **TOIF Findings** view tab.

The vulnerability findings can be displayed in the *TOIF Findings* view in the following ways:

- ▶ Group multiple instances of a reported vulnerability type at a specific location by 2 or more vulnerability detection tools. The group is identified by the term “MULTIPLE” under the Tool column. Clicking the arrow by the file name expands the group
- ▶ Group multiple instances of a reported vulnerability type at a specific location by one vulnerability detection tool. The group is identified by the name of the vulnerability detection tool under the Tool Column. Clicking the arrow by the file name expands the group
- ▶ A single instance of a reported vulnerability type by one vulnerability detection tool

The following image is an example of the vulnerability findings in the TOIF Findings view.

File	Location	Tool	SFP	CWE	Confidence	Description
zvse.c	66	MULTIPLE	SFP1	CWE457	0	
zvse.c	66	Cppcheck	SFP1	CWE457	0	uninitvar: Uninitialized variable: load1
zvse.c	66	Splint	SFP1	CWE457	0	usedef: Variable load1 used before definition
zvse.c	66	Splint	SFP1	CWE457	0	usedef: Variable load1 used before definition
xymond_client.c	66	Cppcheck	SFP1	CWE457	0	
xymond_client.c	66	Cppcheck	SFP1	CWE457	0	uninitvar: Uninitialized variable: load1
xymond_client.c	66	Cppcheck	SFP1	CWE457	0	uninitvar: Uninitialized variable: load1
md5.c	226	Cppcheck	SFP1	CWE457	0	uninitvar: Uninitialized variable: X
md5.c	227	Cppcheck	SFP1	CWE457	0	uninitvar: Uninitialized variable: X
md5.c	228	Cppcheck	SFP1	CWE457	0	uninitvar: Uninitialized variable: X

Note: To remove the vulnerability findings from the TOIF Findings view, select the project in the Project Explorer and right click. From the drop down menu select *Configure-> Disable TOIF Import Nature*. All of the vulnerabilities disappear as the KDM data that was previously imported into the project has been removed.

Prioritizing Vulnerability Findings Reported in the TOIF Findings view

The *TOIF Findings Configuration Preferences* table contains a list of “mapped” vulnerability types (SFP/CWE) that can be reported by each supported vulnerability detection tool. The table provides a default listing of the vulnerability types by order-of-importance and vulnerability type/vulnerability detection tool confidence level. All vulnerability types are set to “Yes” by default indicating that if reported on, these vulnerabilities are displayed in the *TOIF Findings View*.

An integer value in the table cell intersecting the vulnerability type (SFP/CWE) and vulnerability detection tool indicates not only that the vulnerability detection tool reports the vulnerability type but also the level of confidence; level of certainty that this vulnerability tool correctly reports the vulnerability type (true-positive). A “-” in the table cell intersecting the vulnerability type (SFP/CWE) and vulnerability detection tool indicates that the vulnerability detection tool DOES NOT report the related vulnerability type.

The *TOIF Findings Configuration Preferences* table can be customized to prioritize how vulnerability types and their associated confidence values are or are not reported in the TOIF Findings view. Prioritization can be applied in the following ways:

- ▶ Re-order the vulnerability type (SFP/CWE) to set order of importance
- ▶ Setting vulnerability detection tool confidence level per vulnerability type (SFP/CWE)
- ▶ Show/Hide vulnerability type (SFP/CWE) from being

Prioritizing can be performed before or after **you scope vulnerability findings in the TOIF Findings View** (see page 31) or **you analyze and cite** (see page 34) your vulnerability findings.

Re-order vulnerability type (SFP/CWE) to set order of importance

The *TOIF Findings Configuration* preferences table provides a default setting of the order of importance in which vulnerabilities (SFP/CWE vulnerability type) are reported in the TOIF Finding view. To re-order the order of importance in which vulnerabilities are reported in the TOIF Findings view , perform the following steps:

- 1) Select **WINDOWS->PREFERENCES** from the main menu .

The *Preferences* dialog opens.

- 2) Click **TOIF>TOIF Findings Configuration**, located in the sidebar.

The *TOIF Findings Configuration* preferences table appears in the right pane of the Preferences dialog.


- 3) Click on any SFP/CWE vulnerability type or press **CTRL** click to select multiple or **CTRL+SHIFT** to select a range of rows you want to move and drag the row(s) to another location.

The TOIF Findings Configuration table updates to reflect the changes.

Note: Clicking **Cancel** closes the Preferences dialog. If you click **Cancel** after you have selected **Apply** then your changes are saved. If you click **Cancel** prior to selecting **Apply** then your changes are lost.

- 4) Click **Apply**, in the TOIF Findings Configuration preferences pane.

The vulnerability findings displayed TOIF Findings view are re-sorted to reflect the updated order of SFP/CWE vulnerability types set in the TOIF Findings Configuration table.

Note 1: If the vulnerability findings are not automatically re-ordered in the TOIF Findings view click  **User defined sort order** icon in the TOIF Findings View toolbar.

Note 2: You can click **Restore Defaults** to restore the default rows and their sorting order.

Setting the SFP/CWE Confidence levels per Vulnerability Detection Tool

The *TOIF Findings Configuration* preferences table provides a default confidence value setting of zero (0) for each vulnerability type detected by a detection tool. The confidence level value is an indication of how much faith the analyst has in the vulnerability detection tools ability to accurately detect the vulnerability; true positive indicator. The confidence level value is propagated throughout the data set, marking any vulnerability finding with the same CWE from the same vulnerability detection tool with the specified value

To change the confidence level value, perform the following steps:

Note: A “-“ in the table cell intersecting the SFP/CWE vulnerability type and vulnerability detection tool indicates that the vulnerability detection tool DOES NOT report or detect the related vulnerability type and therefore the confidence level value cannot be changed from a “-“ to an integer value.

- 1) Select **WINDOWS->PREFERENCES** from the main menu .

The *Preferences* dialog opens.

- 2) Click **TOIF>TOIF Findings Configuration**, located in the sidebar.

The TOIF Findings Configuration preferences table appears in the right pane of the Preferences dialog.

- 3) Click on any table cell that intersects SFP/CWE vulnerability type and vulnerability detection tool name.

The current confidence level value is highlighted.

Note: Clicking **Cancel** closes the Preferences dialog. If you click **Cancel** after you have selected **Apply** then your changes are saved. If you click **Cancel** prior to selecting **Apply** then your changes are lost.

- 4) Enter a value between 0-100.

The confidence level changes to display the entered value.

- 5) Click **Apply**, in the TOIF Findings Configuration preferences pane.

The confidence level value for all vulnerability findings related to the SFP/CWE vulnerability type and vulnerability detection tool displayed in the *TOIF Findings* view are updated to reflect the Confidence level value set in the *TOIF Findings Configuration* preferences table.

Note 2: You can click **Restore Defaults** to restore the default confidence level values.

Show or Hide Vulnerability Type (SFP/CWE) from being reported in TOIF Findings View

The *TOIF Findings Configuration* preferences table by default is set to report or show (Yes) all supported vulnerability types in the *TOIF Findings* view. If you decide that one or more vulnerability types do not need to be reported to the TOIF Findings view you can set the vulnerability type to be hidden (No). To do this, perform the following steps:

- 1) Select **WINDOWS->PREFERENCES** from the main menu .

The *Preferences* dialog opens.

- 2) Click **TOIF>TOIF Findings Configuration**, located in the sidebar.

The *TOIF Findings Configuration* preferences table appears in the right pane of the Preferences dialog.

- 3) Click in the respective vulnerability table cell under the **SHOW** column in the **TOIF Findings Configuration** preferences table.

The *Show* value for the selected cell toggles between **Yes** and **No**.

So, if the Show value is **“Yes”** then the vulnerability will be reported or displayed in the TOIF Findings view. If the Show value is **“No”** then the vulnerability will be not reported or displayed in the TOIF Findings view.

Note: Clicking **Cancel** closes the Preferences dialog. If you click **Cancel** after you have selected **Apply** then your changes are saved. If you click **Cancel** prior to selecting **Apply** then your changes are lost.

- 4) Click **Apply**, in the **TOIF Findings Configuration** preferences pane.

The vulnerability type set as **“Yes”** in the *TOIF Findings Configuration* are reported or displayed in the TOIF Findings view. The vulnerability type(s) set as **“No”** in the *TOIF Findings Configuration* are NOT reported or displayed in the *TOIF Findings* view.

Note: You can click **Restore Defaults** to restore the default rows and their sorting order.

Exporting and Importing the TOIF Findings Configuration Preferences Table Settings

The *TOIF Findings Configuration* preferences table settings can be exported so you can share your settings with other team members or organizations or make a copy that can be used on another project workspace at a later time. The *TOIF Findings Configuration* preferences table values are saved to *.csv file. The imported *.csv file replaces the current state or working copy of the *TOIF Findings Configuration* preferences table; the imported file takes precedence.


When you import a *TOIF Findings Configuration* preferences table *.csv file that contains:

- a partial list or sub-set of the vulnerability types listed in *TOIF Findings Configuration* preferences table the imported vulnerability types and their values are placed in the *TOIF Findings Configuration* preferences table with the default vulnerability types and their values that were not included in the imported file completing the rest of the table
- a complete list of vulnerability types listed in *TOIF Findings Configuration* preferences table then the current state or working copy is overwritten by the imported file

Tip: We recommend that after making changes to the *TOIF Findings Configuration* preferences table that you perform an export so that you save your changes and can always revert back to that state by performing an import.

To export and import a *TOIF Findings Configuration* preferences table *.csv file, perform the following:

Exporting

- 1) Select **WINDOWS->PREFERENCES** from the main menu .
The *Preferences* dialog opens.
- 2) Click **TOIF>TOIF Findings Configuration**, located in the sidebar.
The *TOIF Findings Configuration* preferences table appears in the right pane of the Preferences dialog.
- 3) Perform any priority customizations to the **TOIF Findings Configuration** preferences table .
The *TOIF Findings Configuration* preferences table updates accordingly.
- 4) Click  Export button located in the top right corner of the **TOIF Findings Configuration** preferences pane.
A file system window opens.
- 5) Click in the **FILENAME:** text field and enter the name you want to name the file.
- 6) Navigate the file system to the directory or location in which you want to save the file.
- 7) Click **Save**.

A *<file name>.csv* file is saved in the selected location within the file system.

Importing

- 1) Select **WINDOWS->PREFERENCES** from the main menu .


The *Preferences* dialog opens.

- 2) Click **TOIF>TOIF Findings Configuration**, located in the sidebar.

The *TOIF Findings Configuration* preferences table appears in the right pane of the Preferences dialog.

- 3) Perform any priority customizations to the **TOIF Findings Configuration** preferences table .

The *TOIF Findings Configuration* preferences table updates accordingly.

- 4) Click  Import button located in the top right corner of the **TOIF Findings Configuration** preferences pane.

A file system window opens.

- 5) Select the *<file name>.csv* file you want to import from the file system window.

- 6) Click **Open** in the file system window.

The *TOIF Findings Configuration* preferences table updates based on the value settings in the imported *<file name>.csv* file.

- 7) Click **Apply**, in the **TOIF Findings Configuration** preferences pane.

The *TOIF Findings* view is updated to reflect the value settings of the *TOIF Findings Configuration* preferences table which were imported from the *<file name>.csv* file.

Scoping Vulnerability Findings in the TOIF Findings View

Vulnerability findings currently displayed in the *TOIF Findings View* can be scoped to better to focus your analysis and citing by applying a term search and/or a filter option. Scoping can be applied in the following ways:

- ▶▶ Search and term filter
- ▶▶ Filter options
- ▶▶ Sorting by column headings
- ▶▶ Filter on a selected project source file(s)
- ▶▶ User defined sort
- ▶▶ Sort by Multiple
- ▶▶ Any combination of the above

Scoping can be performed before or after ***you analyze and cite*** (see "***Analyzing and Citing Vulnerability Findings in the Findings View***" on page 34) your vulnerability findings.

When any of the scoping methods are applied the TOIF Findings view will indicate that a filter(s) is applied by displaying a "**(Filter(s) active)**" message in red in the label above the vulnerability findings table. For example, *[MyProject] (Filter(s) active) Number of defects: 2 (500 filtered from view)*.

Search and term filter

To do this, perform the following steps:

- 1) Click on the project that contains defect findings in the Project Explorer.

The project is highlighted to indicate that it is selected and the respective vulnerability findings are displayed in the *TOIF Findings view*.

- 2) Click in the Search and term text field in the *TOIF Findings view* and type a term.

- 3) Click **Search**.

Only findings that contain the specific term(s) will be displayed in the *TOIF Findings View*. Running the search without any terms in the term filter field will show all the findings in the data set.


Note: To clear the Search and term text field click the *Clear* button in the *TOIF Findings view*.

Filter Options

To do this, perform the following steps:

- 1) Click on the project that contains vulnerability findings in the Project Explorer.
-

The project is highlighted to indicate that it is selected and the respective vulnerability findings are displayed in the *TOIF Findings* view

- 2) Click Filter  icon in the *TOIF Findings* view toolbar and from the drop down menu select **FILTERS...**

The Filter window is displayed

- 3) Do any combination of the following:
 - a) Click the check box in front of **2+ TOOLS REPORT SAME LOCATION** to view all vulnerability findings reported on the same file and line number by two or more OSS SCA tools.
 - b) Click the check box in front of **2+ TOOLS REPORT SAME LOCATION. WITH THE SAME CWE** to view all vulnerability findings reported on the same file and line number with the same CWE by two or more OSS SCA tools.
 - c) Click the check box in front of **2+ TOOLS REPORT SAME LOCATION. WITH THE SAME SFP** to view all vulnerability findings reported on the same file and line number with the same SFP by two or more OSS SCA tools.
 - d) Click the check box in front of **TRUST ABOVE:** and enter a value between 0 and 100 in the text field to view all vulnerability findings with a trust level above the value entered in the test field.
 - e) Click the check box in front of **IS VALID** to view all vulnerability findings cited as *Is a Weakness* or "true" citing.
 - f) Click the check box in front of **NOT VALID** to view all vulnerability findings cited as *Not a Weakness* or "false" citing or those that are unmarked.
 - g) Click the check box in front of **NOT SFP--1** to view all security related vulnerability findings

Note: The filtering is applied to the vulnerability findings currently displayed in the *TOIF Findings* view.

- 4) Click **Ok**

The *Filter* window closes and the *TOIF Findings* view updates to display only the vulnerability findings associated with the applied filtering options

Note: If the *Filter* option is applied in combination with the *Search and term* filter the *Filter* option(s) is applied to the vulnerability findings currently displayed in the *TOIF Findings* view.

Sorting by Column Heading

To do this, perform the following steps:

- 1) Click on the project that contains vulnerability findings in the Project Explorer.

The project is highlighted to indicate that it is selected and the respective vulnerability findings are displayed in the *TOIF Findings* view.

- 2) Click on any of the heading columns to sort the vulnerability findings alpha-numerically by column.

Filter on a Selected Project Source File(s)

To do this, perform the following steps:

- 1) Navigate to a project source file(s) that contains vulnerability findings in the Project Explorer.

The project source file(s) is highlighted to indicate that it is selected and the respective vulnerability findings are displayed in the *TOIF Findings* view.

Note: Hold **CTRL** key and click the project source files to choose them.

- 2) Right click on the project source file and select **FILTER TOIF FINDINGS ON SELECTION** in the drop down menu.

The *TOIF Findings* view updates to display vulnerability findings associated with the selected project source file(s).

User Defined Sort

Sorts the order of vulnerabilities displayed in the TOIF Findings as defined in the *TOIF Findings Configuration* preferences table. The sort order criterion when a "User defined" sort is applied is as follows:

- a) SFP/CWE order as defined in the *TOIF Findings Configuration* preferences table
- b) Number of vulnerability detection tools defining vulnerabilities on the same file/line
- c) Confidence value of the vulnerability detection tool as defined in the *TOIF Findings Configuration* preferences table
- d) File path of the detected vulnerability
- e) Line number where the vulnerability was detected

To do this, perform the following steps:

- 1) Click on the project that contains vulnerability findings in the Project Explorer.

The project is highlighted to indicate that it is selected and the respective vulnerability findings are displayed in the *TOIF Findings* view.

- 2) Click on the  **User defined sort order** icon in the **TOIF Findings** view toolbar.

The *TOIF Findings* view updates to sort the vulnerability findings as defined by the settings in *TOIF Findings Configuration* preferences table.

Sort by Multiple

Sorts the order of vulnerabilities in the TOIF Findings view by group or "MULTIPLE" findings first followed by all other findings. The sort order criterion when a group or "Multiple" sort is applied is as follows:

- a) Number of vulnerability detection tools defining vulnerabilities on the same file/line
- b) SFP/CWE order as defined in the TOIF Findings Configuration preferences table
- c) Confidence value of the vulnerability detection tool as defined in the TOIF Findings Configuration preferences table
- d) File path of the detected vulnerability
- e) Line number where the vulnerability was detected

To do this, perform the following steps:

- 1) Click on the project that contains vulnerability findings in the Project Explorer.

The project is highlighted to indicate that it is selected and the respective vulnerability findings are displayed in the *TOIF Findings* view.

- 2) Click on the  **Sort by Multiple Findings** icon in the **TOIF Findings** view toolbar.

The *TOIF Findings* view updates to sort the vulnerability findings by the MULTIPLE findings first followed by all other findings.


Analyzing and Citing Vulnerability Findings in the TOIF Findings View

The source code associated with the finding can be analyzed and then cited in the *TOIF Findings* view. To do this do the following.

- 1) Click on a project that contains vulnerability findings in the Project Explorer.


The project is highlighted to indicate that it is selected and the respective vulnerability findings are displayed in the *TOIF Findings* view.

- 2) (Optional) **Scoping Vulnerability Findings in the Findings View** on page 33 to narrow the focus of your analysis.

- 3) (Optional) Select a vulnerability finding instance in the **TOIF Findings** view and click the **Defect Description** icon  in the **TOIF Findings** view toolbar.

The *Defect Description* tab opens to provide the cluster, SFP, and CWE description of the selected vulnerability instance.

- 4) Double click on a vulnerability finding instance in the *TOIF Findings* view.

The selected file opens to the location of the vulnerability in the Editor panel of Eclipse. Place the cursor over the marker  displayed to the left of the finding to pop up a tool tip with the vulnerability description.

Note: If a “Cannot open file” dialog appears it is because the source file associated with the vulnerability finding does not exist in your project. Please check to ensure that the file exists in your project and has not been deleted.

5) Analyze the vulnerability and do any of the following to cite the vulnerability:

- a) If you decide that the vulnerability is a weakness right click anywhere on the finding instance in the *TOIF Findings* view and select **IS A WEAKNESS** from the drop down menu.

A red “X” appears beside the Tool, SFP, and CWE column cells within the *TOIF Findings* view for the respective finding instance.

- b) If you decide that the vulnerability is not a weakness right click anywhere on the finding instance in the *TOIF Findings* view and select **NOT A WEAKNESS** from the drop down menu.

A green check mark appears beside the Tool, SFP, and CWE column cells within the *TOIF Findings* view for the respective finding instance.

- c) If you decide that the OSS SCA tool's ability to accurately detect the vulnerability finding is suspect you can set the trust level value between 0-100 to indicate the level of confidence. This level is propagated throughout the data set, marking any finding with the same CWE from the same tool with the specified trust level value.

Note 1: When a project is cited for the first time a warning dialog appears with the following message:

Warning: Citings are file attributes. Editing or deleting a file will delete its citing information. Daily snapshots of citing information are saved in <project>/.KDM/TOIF/history

Click **Ok** in the Warning dialog. The Warning dialog closes and the initial citing is applied.

Note 2: If you need more information on the vulnerability and you have a connection to the internet right click on the vulnerability finding instance in the **TOIF Findings** view and select **MORE INFORMATION** from the drop down menu. A browser window opens displaying the mitre site for the selected CWE.

Exporting Cited Vulnerability Findings to a *.tsv file


Vulnerability findings, including citing information, can be exported to a *.tsv file. To do this, perform the following steps.

- 1) Click on the project that contains vulnerability findings in the Project Explorer.

The project is highlighted to indicate that it is selected and the respective vulnerability findings are displayed in the *TOIF Findings* view.

- 2) Click on any vulnerability finding instance or press **CTRL+A** to select all instances in the *TOIF Findings* view.

All vulnerability findings in the *TOIF Findings* view are selected.

- 3) Click **Export Selection**  icon in the *TOIF Findings* view tool bar menu.

A *Save As* dialog opens.

- 4) Click in the **FILE NAME:** text field and enter the name of the file.
- 5) Click **Save**.

The *Save As* dialog closes and a *<filename>.tsv* file is saved to the specified directory.

Note: TOIF automatically saves a *<project name>.<yyyy-mm-dd>T<hh-mm-sec>.tsv* file every time citing information is modified. The file is located in *<project workspace directory>/<project name>/KDM/TOIF/history* directory.

Importing Cited Vulnerability Findings from a *.tsv file

Vulnerability findings, including citing information, exported from one TOIF project can be imported to another TOIF project by a *.tsv file. To do this, perform the following steps.

Note: The projects that you are importing the files to must have the cited source files for the citations to be applied.

- 1) Click on the project that contains the same source files as the project the **.tsv file* (see "**Exporting cited vulnerability findings to a *.tsv file**" on page 35) was exported from.

The project appears in the *Project Explorer* and the vulnerability findings are displayed in the *Findings* view.

- 2) Right-click on the project and select **IMPORT...** from the drop-down menu.

An *Import* dialog opens.

- 3) Expand the **SFP/CWE** folder and select **Import Citing File (*.tsv)**.

The *Next>* button becomes enabled.

- 4) Click **Next>**.

The *File Import Wizard* appears.

- 5) Select the project from the *Select target project:* list.

- 6) Click the **Browse** button and navigate to and select the *.tsv file.

The location to the *.tsv file populates the *Select TOIF Data:* text field and the *Finish* button becomes enabled.

- 7) Click **Finish**.

Note: If your project does not contain the same source files related to the *.tsv file then no finding or citing information will be displayed in the *TOIF Findings* view.

- 8) Select the **TOIF Findings** view.

The vulnerability findings and any citing information from the imported *.tsv file is displayed in the *TOIF Findings* view.

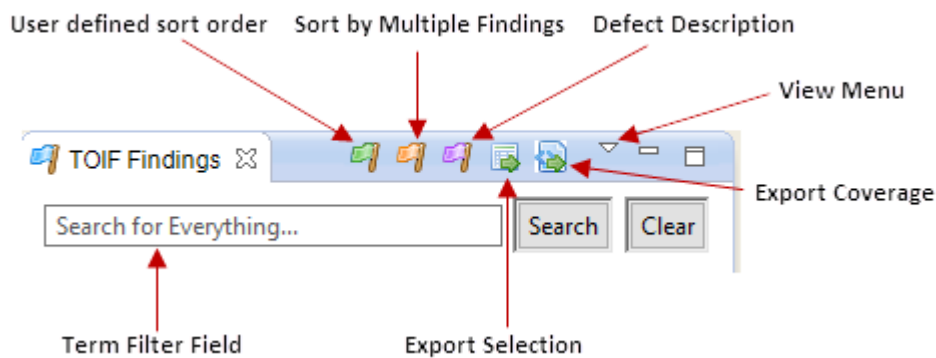
Chapter 8

TOIF Findings View Interface

The following section provides the descriptions of the toolbar buttons and context-sensitive menu options in the *TOIF Report View* that can be used in performing a vulnerability analysis on your project.

TOIF Findings View toolbar

The *TOIF Findings View* consists of a toolbar which is located in the top right corner contains the following buttons and fields:



View Menu

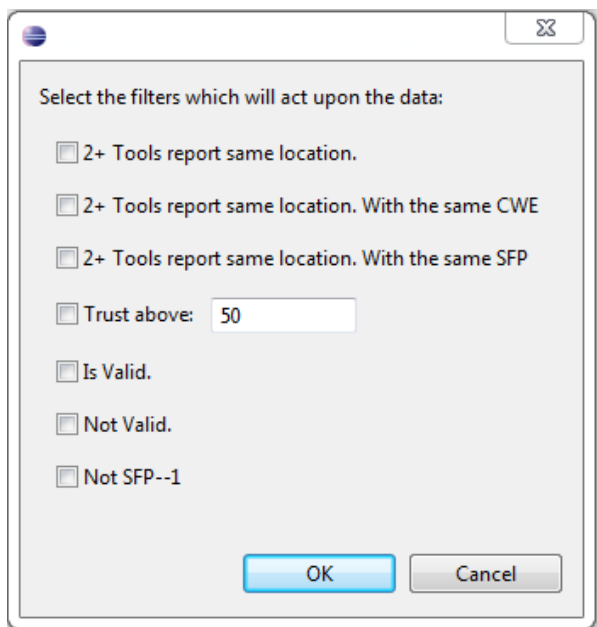
The View Menu contains the following options which can be applied to the *TOIF Findings* view:

- ▶▶ Filters...
- ▶▶ Sort
- ▶▶ Preserve Column order

Filter...

Filters reduce the number of visible vulnerability findings displayed in the *TOIF Findings* view.

- ▶▶ The 2+ Tools filter options only displays findings where two tools found a finding in the same location, same location with the same CWE, or same location with the same SFP.
- ▶▶ The trust filter only shows findings with a trust above the set amount.
- ▶▶ The "Is valid" filter option displays the findings that have been marked as *IS A WEAKNESS*
- ▶▶ The "Not valid" filter option displays the findings that have been marked as either *NOT A WEAKNESS* or are unmarked.
- ▶▶ The Not SFP--1 filter option removes all non-security related (SFP--1) vulnerability findings from the TOIF Findings view.



Clicking in the check box in front of a filter option toggles a check mark; displayed or not displayed. A check mark in the check box denotes that the filter option is selected. No check mark denotes that filter option is not selected. Only selected filter options will be applied.

Sort

The Sort option allows you to sort the vulnerabilities displayed in the TOIF Findings view by User definition (see page 39) or by Multiple (see page 40).

Preserve column order

The Preserve column order allows you to preserve the order of the columns displayed in the *TOIF Findings* view. Click and drag to move a column to the left or right to place it where you want it to be displayed in the TOIF Findings view then once you have selected the order in which you want the columns to be displayed, select *Preserve column order*. The column order will be preserved for that workspace after re-start or starting a new session.

Export Selection

Exports the selected elements to a *.tsv format. This file can then be imported into an Eclipse project containing the same source files as the project used to export the *.tsv or programs such as Excel or Libre Office Calc. It is important to note that this file is tab separated.

Export Coverage

Exports the entire data set as a Coverage Claims Representation (CCR). More information for this coverage report can be found at <http://cwe.mitre.org/compatible/ccr.html>.

Defect Description

Displays the cluster, SFP, and CWE description of the selected vulnerability type.

User Defined Sort Order

Sorts the order of vulnerabilities displayed in the TOIF Findings as defined in the *TOIF Findings Configuration* preferences table. The sort order criterion when a “User defined” sort is applied is as follows:

- a. SFP/CWE order as defined in the *TOIF Findings Configuration* preferences table
- b. Number of vulnerability detection tools defining vulnerabilities on the same file/line
- c. Confidence value of the vulnerability detection tool as defined in the *TOIF Findings Configuration* preferences table
- d. File path of the detected vulnerability
- e. Line number where the vulnerability was detected

Sort by Multiple Findings

Sorts the order of vulnerabilities in the TOIF Findings view by group or “MULTIPLE” findings first followed by all other findings. The sort order criterion when a group or “Multiple” sort is applied is as follows:

- a. Number of vulnerability detection tools defining vulnerabilities on the same file/line
- b. SFP/CWE order as defined in the TOIF Findings Configuration preferences table
- c. Confidence value of the vulnerability detection tool as defined in the TOIF Findings Configuration preferences table
- d. File path of the detected vulnerability
- e. Line number where the vulnerability was detected

Term Filter Field

The *term filter field* is a search box for findings containing a specific term. Clicking the [Search](#) button executes the search on the terms included in CWE/SFP ids, Description contents, line numbers, OSS SCA Tool name, or resource names of the findings. Only findings that contain the specific term(s) will be displayed in the TOIF Findings View. Running the search without any terms in the term filter field will show all the findings in the data set.

Clicking the [Clear](#) button removes the terms from the *term filter field* and updates the *TOIF Findings* view to display the list of vulnerability findings displayed prior to executing the search.

TOIF Findings View Context Sensitive Menu Options

Right clicking on data displayed in the *TOIF Findings View* will display a drop down menu with the following options.

Not a Weakness

This marks that the finding is not actually a weakness.

Is a Weakness

This marks that the finding is a weakness.

Uncite Weakness

This unmarks a finding that was previously marked as either “Not a Weakness” or “Is A Weakness”.

Trace

This option is only available at the “Finding” level. If trace data is present, selecting this option will display the trace back as a dynamic menu which when clicked will take you to the various places within the code; tracing the route all the way to where the finding was generated. If trace data is not present the option is disabled; appears greyed out.

More Information

This will take you to the mitre site for the selected CWE id.

TOIF Findings View Column Sorting

Click on the column headings to alphabetically sort in ascending and descending order.

Chapter 9

Known Limitations

Traceback displays numbers as reported by TOIF Adaptors

The traceback currently shows numbers as reported by the TOIF Adaptors unlike the code locations in the view list (which are normalized to kdm locations).

Jlint not supported on Fedora or RHE platform

Use of Jlint 3.0 OSS SCA tool with Blade TOIF OSS 2.3.1 is not supported on the Fedora-17.x 64-bit or RHE 7.1 64-bit operating system.

Glossary of Terms

C

Common Weakness Enumeration

Common Weakness Enumeration (CWE) is a software community project whose goal is to create a catalog of software weaknesses and vulnerabilities.

Coverage Claims Representation

Coverage Claims Representation (CCR) is an XML document used for representing information about Common Weakness Enumeration.

O

OSS SCA

An acronym for open source software static code analysis (OSS SCA).

S

Software Fault Patterns

Software Fault Patterns (SFP) are a generalized description of an identifiable family of computations:

- ▶▶ Described as patterns with an invariant core and variant parts
- ▶▶ Aligned with injury
- ▶▶ Aligned with operational views and risk through events
- ▶▶ Fully identifiable in code (discernible)
- ▶▶ Aligned with CWE
- ▶▶ With formally defined characteristics

Index

A

- Analyzing and Citing Defect Findings in the Findings View • 31
- Analyzing and Citing Vulnerability Findings in the TOIF Findings View • 34
- Audience • 1

B

- Blade TOIF OSS Components • 6
- Blade TOIF OSS RCP • 12

C

- Client Hardware • 8
- Client Software • 8
- Common Weakness Enumeration • 45
- Coverage Claims Representation • 45

D

- Defect Description • 39

E

- Export Coverage • 39
- Export Selection • 39
- Exporting and Importing the TOIF Findings Configuration Preferences Table Settings • 29
- Exporting cited defect findings to a *.tsv file • 36
- Exporting Cited Vulnerability Findings to a *.tsv file • 35
- Exporting the TOIF Findings Configuration Preferences Table Settings • 29

F

- Filter on a Selected Project Source File(s) • 33
- Filter Options • 31
- Filters... • 38

G

- Getting Help • 3

H

- How does Blade TOIF OSS Work? • 5

I

- Importing Cited Vulnerability Findings from a *.tsv file • 36
- Importing the TOIF Findings Configuration Preferences Table Settings • 30
- Importing Vulnerability Findings from a KDM File to a General Project in Eclipse • 24
- Installation Overview • 7
- Installing Blade TOIF OSS Packages • 11
- Installing Open Source Software Static Code Analysis (SCA) Tools • 13
- Integrating with a C project's build • 16
- Integrating with a Java project's build • 16
- Introduction • 1
- Is a Weakness • 40

J

- Jlint not supported on Fedora platform • 43

K

- Known Limitations • 43

M

- More Information • 41

N

- New Install of the TOIF Findings View in Eclipse • 22
- Not a Weakness • 40

O

- Opening the TOIF Findings View in Eclipse • 23
- OSS SCA • 45

P

- Preparing to install Blade TOIF OSS • 7
- Preserve column order • 39
- Prioritizing Vulnerability Findings Reported in the TOIF Findings View • 26

R

- Re-order Vulnerability type (SFP/CWE) to Set Order-of-Importance • 26
- Running the TOIF Adaptor • 15
- Running the TOIF Adaptor from command the line • 15
- Running the TOIF Assimilator • 17
- Running the TSV Output • 19

S

- Scoping Defect Findings in the Findings View • 34
- Scoping Vulnerability Findings in the TOIF Findings View • 31
- Search and term filter • 31
- Setting the SFP/CWE Confidence Levels per Vulnerability Detection Tool • 27
- Show or Hide Vulnerability Type (SFP/CWE) from being Reported in the TOIF Findings View • 28
- Software Fault Patterns • 45
- Sort • 39
- Sort by Multiple Findings • 40
- Sorting by Column Heading • 32
- Supported Deployment Configurations • 8
- System Requirements • 8

T

- Term Filter Field • 40
- TOIF Finding View Sorting • 41
- TOIF Findings View • 21
- TOIF Findings View Context Sensitive Menu Options • 40
- TOIF Findings View Interface • 37

- TOIF Findings View Toolbar • 37
- Trace • 41
- Traceback Numbers • 43
- TSV Output • 19
- TSV Output • 12
- Typographical Conventions • 2

U

- Uncite Weakness • 40
- Upgrading the TOIF Findings View 2.2.0 to 2.3.1 in Eclipse • 21
- User Defined Sort Order • 39

V

- Verify Blade TOIF OSS and TSV Output Installation • 12
- View Menu • 37

W

- When to contact us • 3