com.idtus.contest.winter2017.framework.jar Documentation

Team ICT-2

INTRO

This document is IDT team ICT-2's documentation for its black-box jar testing software. The software tests java jar files, and attempts to find security vulnerabilities. These may exist in the form of exceptions, or risky security permissions.

The built software may be found at the following link:

https://github.com/J3698/IDTICT2/releases

The source can be downloaded at the following link:

https://github.com/J3698/IDTICT2

CONTENTS

FRAMEWORK REQUIREMENTS

Windows 7 or higher – The framework may work on machines which run the latest version of Java such as Mac or Linux may work, however such operation is untested.

Java SE 1.8 or higher – The framework was developed using the Java 8 API, thus the framework may not work on older versions of java.

Java bin appended to system path – To run the framework from the command line, the java bin folder must be appended to the system path.

Java set as the default program to open Jars – To run the framework by clicking on the Jar, Java must be the default program to open Java Jars.

*JDK 1.8 or higher – To compile the framework, JDK 1.8 is required.

*Eclipse 4.4 or higher – To open the framework in an IDE, Eclipse 4.4 is recommended.

*Maven2Eclipse plugin – To open the framework in Eclipse, the Maven2Eclipse plugin is required.

*These requirements only apply if the framework is to be extended

INSTRUCTIONS TO RUN TESTS FROM THE COMMAND LINE

This application has five basic modes of operation which can be accessed from the command line. The modes are testing mode, help mode, permission info mode, permission list mode, and GUI mode. If the arguments for one of those modes are supplied, the modes further down the list are ignored.

Testing Mode - The first operation mode is testing mode. In this mode required arguments are the path to the executable jar, the path to the directory for Jacoco output, and the path to the Jacoco agent jar. Optional parameters include bbTests for the number of tests to run (default 1000), timeGoal to set a goal time in minutes to run tests in (default 5), and toolchain, which limits application output. Command usage (type on one line):

java -jar com.idtus.contest.winter2017.framework.jar jarToTestPath jarToTest.jar

-jacocoOutputPath ./ -jacocoAgentJarPath jacocoagent.jar -bbTests 100

Help Mode - The second mode of operation is help mode. In this mode help is supplied to the user. Command usage:

java -jar com.idtus.contest.winter2017.framework.jar -help

Permission Info Mode - The third mode of operation is permission info mode. This mode supplies the security risks associated with the given java permissions to the user. Command usage:

java -jar com.idtus.contest.winter2017.framework.jar -permissionInfo filePermission

Permission List Mode - The fourth mode of operation is permission list mode. This mode lists the different sorts of permissions. Command usage:

java -jar com.idtus.contest.winter2017.framework.jar -permissionInfo FilePermission

User Interface Mode - The fifth mode of operation is user interface mode. In this mode, a JavaFX application is run, which exposes much of the functionality in the other modes. Command usage:
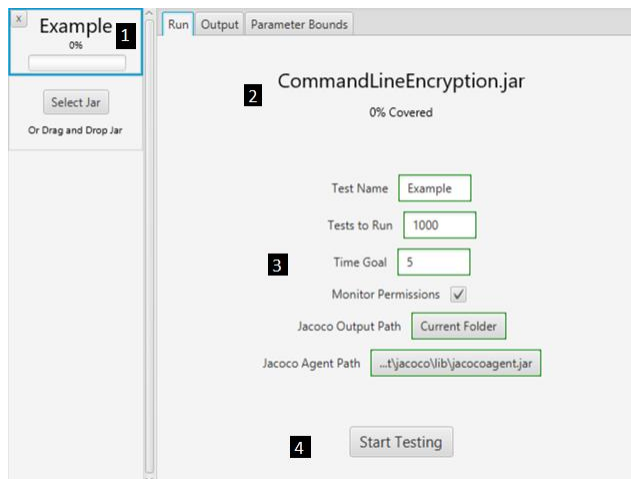
java -jar com.idtus.contest.winter2017.framework.jar -gui

INSTRUCTIONS TO RUN TESTS FROM THE USER INTERFACE

The user interface has several different screens. The photos and information below explain these.
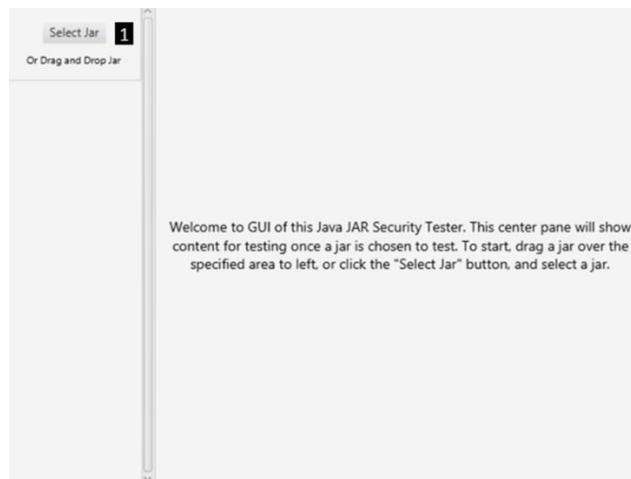
1 – Drag a jar to test here, or click the button to select a jar to test.

1 – View the test name and testing progress of a jar under test.

2 – View the name of the jar under test and how much of the jar has been covered.

3 – Change test settings for the jar.

4 – Start, pause, resume testing for the test.

INSTRUCTIONS TO RUN TESTS FROM THE USER INTERFACE

1 – View the input commands of tests run.

2 – View the standard out of tests run.

1 – View the standard error of tests run.

2 – View permissions the jar under test used during testing.

1 – View the toolchain YAML output when testing has completed.

2 – View the count, risks, and allowances of permissions used by a jar under test.

INSTRUCTIONS TO CHANGE THE FRAMEWORK

Importing the Framework - To import the framework to Eclipse, go to file > import, then Maven > Existing Maven Projects. Progress until asked to select a directory for the existing project. Select the folder named com.idtus.contest.winter2017.framework from the framework source (see the intro for where to download the source), then finish. In package explorer, right click JRE System Library, go to properties and make sure version 1.8 is being used.

Extending the Framework – There are several ways to contribute to this framework. They include creating new methods of generating tests, changing the user interface, or adding new modes.

Generating Tests – To create new methods of generating tests, create a class which extends TestGenerator, and implement nextTest. Read the Javadoc comments for TestGenerator for further guidelines.

Changing the User Interface – To change the skin, consider JavaFX skins such as JMetro, AeroFX, or AquaFX. To change existing code, see the package contest.winter2017.gui. The application entry point is GUIMain.java. Code for the side bar resides in TestListPane.java, and in TestInfo, a class in GUITestPackage.java. The functional center pane is MainPane.java, and the placeholder text when the user interface is started can be found in IntroPane, a class which resides in GUIMain.java

Adding new Modes – To add new modes, look in Main.java. This is the entry point for the whole application, and parses command line arguments.