User Input Security - Using FlawExplorer to speed up result check

Introduction

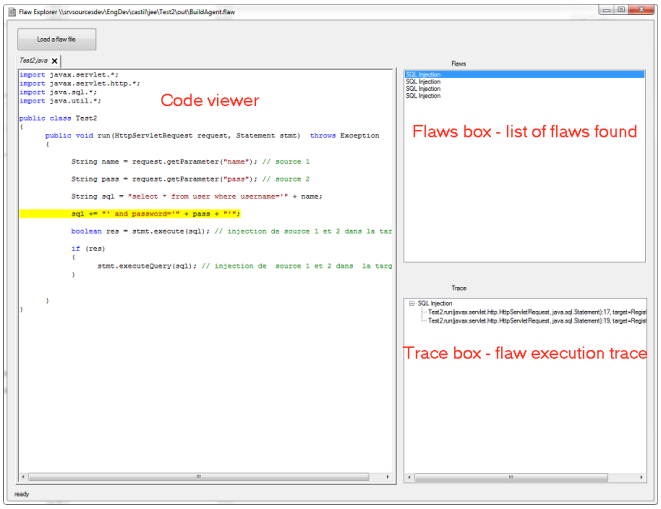
If it is necessary to fine tune elements of the User Input Security configuration without running an entire snapshot or if it is necessary to browse the User Input Security results without using the **CAST Engineering Dashboard**, it is possible to use an unsupported CAST tool called the **FlawExplorer** to visualize the flaws. It takes in input "**.flaw**" files, and lets you browse the flaws that have been generate during the snapshot. In addition, since the FlawExplorer is an "offline" tool, you can browse flaw files without having to connect to a CAST Management Service.

You can also use it during a snapshot generation - ensure the "Dataflow Security" task has been completed successfully before attempting to use it.

Using FlawExplorer

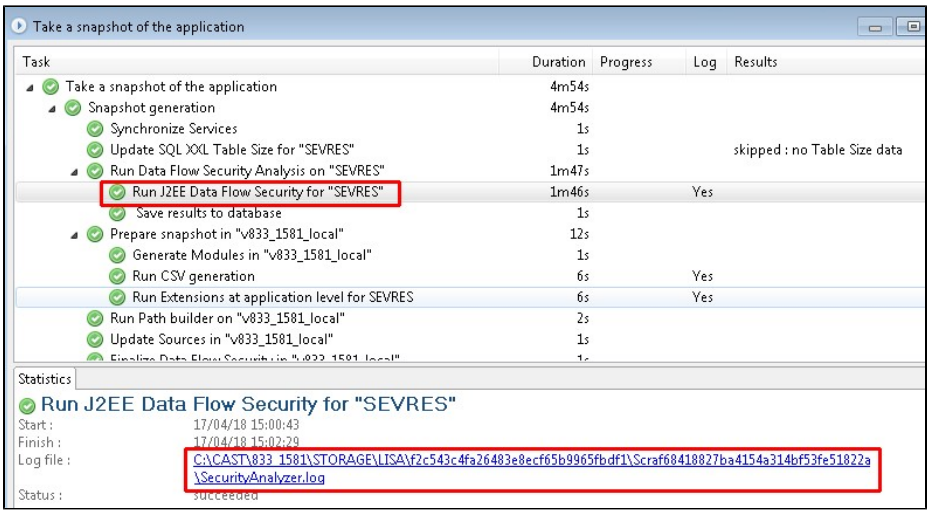
flawExplorer.exe is located in the CAST AI installation folder - double click the file to run it:¨

Click to enlarge



Use the **Load a flaw file** option (located in the top lefthand corner of the GUI) to load up a .flaw file. **.flaw** files are stored in the **Large Intermediate Storage Area** (LISA). The path to the .flaw file will also be logged in the **SecurityAnalyzer.log** which logs all User Input Security actions. You can open the SecurityAnalyzer.log file by clicking the link in the CAST Management Studio log window under the step **Run DataFlow Security**:

Click to enlarge



Typical usage

- In the **Flaws box**: select a flaw in the list. The **Trace box** will be refreshed and contains the corresponding execution trace.
- In the **Trace box**: select a line in the execution trace. The **Code viewer** will be refreshed and contains the bookmark (highlighted in yellow) of the corresponding executed statement.

Hints:

- In the **Flaws box**: You can use keyboard arrows to switch between flaws
- In the **Trace box**: You can use keyboard arrows to play the execution trace.

Note that the flaw names displayed in the FlawExplorer are not exactly the same as those presented in the CAST Engineering Dashboard. For example, the flaw called "Http Response Splitting" as displayed in the FlawExplorer is reported as "Cross-site scripting" in the CAST Engineering Dashboard.