Changes or new features - 8.3.20

Summary: CAST AIP 8.3.20 introduces a number of features and changes as listed below.

Analysis result save process has been optimized

The internal mechanism that is used to save analysis results in the CAST AIP schemas has been optimized and improved in this release of AIP. The goal of this optimization has primarily been to introduce more rigorous controls on the data that is saved to reduce inconsistencies and therefore to increase the overall accuracy of CAST AIP. In addition, performance has been stabilized. As a result of this optimization, some small changes in analysis results are to be expected when performing a new analysis/snapshot post-upgrade on unchanged source code, for example:

- Some objects, links, properties and bookmarks that flag the location of rule violations in the source code are now more rigorously checked and therefore some items may no longer be saved, notably for Universal Importer jobs, improving accuracy
- Results of metrics calculated by the Metrics Assistant may be impacted (values may be higher or lower), improving accuracy
- 7962 - Avoid direct or indirect remote calls inside a loop: previously some violations of this rule were not saved - these are now saved, improving accuracy

CAST Server Manager - CLI

The **-MODIFY_COMBINED** command has been optimized to improve performance when using the command to **Install new extensions**, **upgrade existing extensions** or **deactivate existing extensions** to an existing combined installation (Management, Analysis and Dashboard Service schemas) - equivalent to the **Manage Extensions** option in the GUI.

Mainframe

IMS/DC - support for links between Cobol Programs and IMS Transactions

CAST AIP 8.3.20 introduces support for links between Cobol Programs and IMS Transactions for **IMS/DC** (Data Communications). See **Mainframe - IMS DC support**.

IMS/DB - link type changes

Links between Cobol paragraphs/sections and DB/GSAM/ALT PCB when using DLI function have been updated as follows:

- OPEN/CLSE: (only for GSAM): accessOpenLink, accessCloseLink.
- DLET : UseDeleteLink
- GU/GHU, GN/GHN, GNP/GHNP, INQY: UseSelectLink
- ISRT: UseInsertLink
- REPL: UseUpdateLink
- Other dli calls: useLink

Rule updates

Avoid unchecked return code (SQLCODE) after EXEC SQL query (7690)

Several fixes have been applied to the rule **Avoid unchecked return code (SQLCODE) after EXEC SQL query (7690)** to reduce the number of false violations reported:

- MAINFRAME-379 - Correcting a situation where the rule is falsely violated when a paragraph contains multiple paragraphs called via IF clauses and where each of the called paragraphs contains SQL statements and where the SQL statement is checked from the parent paragraph.
- MAINFRAME-361 - Correcting a situation where the rule is falsely violated when the SQL statement is contained in parentheses.
- MAINFRAME-360 - Correcting a situation where the rule is falsely violated when the SQL statement is contained inside an IF statement of a PERFORM paragraph.

MAINFRAME-373 - Never truncate data in MOVE statements (7688)

Fixes have been applied to the rule **Never truncate data in MOVE statements (7688)** to reduce the number of false violations reported.

MAINFRAME-348 - CICS Return code should be checked (8162)

Fixes have been applied to the rule **CICS Return code should be checked (8162)** to reduce the number of false violations reported when the check statement is called via an IF statement in a variable.

JCL Symbol coverage

Improved coverage for JCL Symbol resolution:

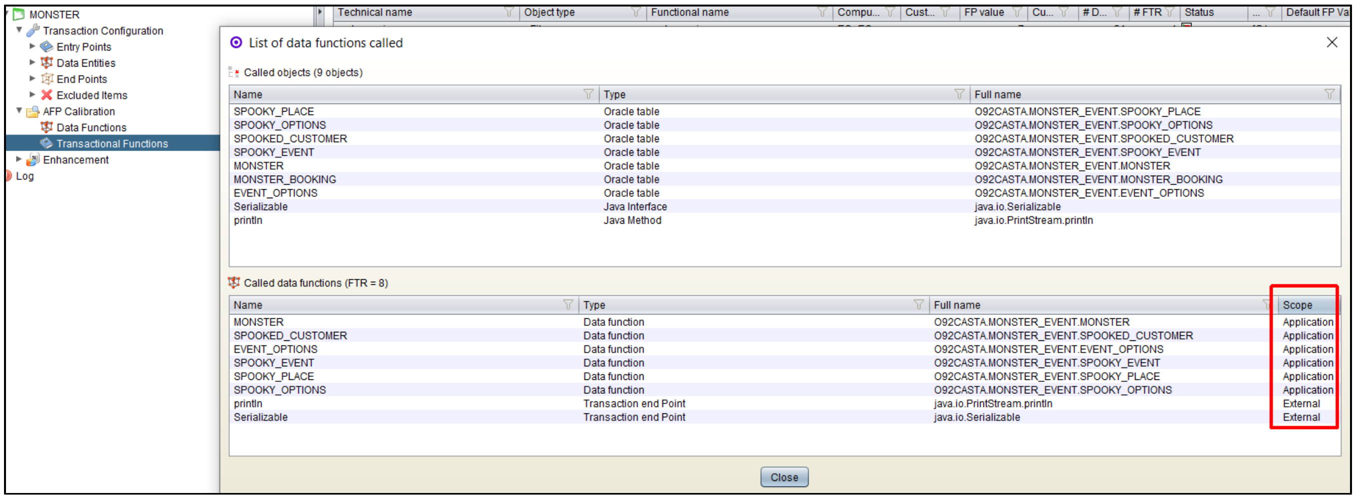
- Support for EXEC Procedures or PGMs containing "&" such as "NAME1&VAR2"
- Support for the following situation:
    - JOBs containing VAR1 = X
    - and PROCs containing VAR1 = DEFAULT VALUE
    - X will not be overridden, and the value in JOB is taken as a priority

CAST Transaction Configuration Center

GUI update for external end-points

An update has been made to the GUI of the CAST Transaction Configuration Center to allow users to see if an end point is external when checking the **dat afunction called by a transaction** in a new column called **Scope**:

Click to enlarge



Called objects (9 objects)

Name	Type	Full name
SPOOKY_PLACE	Oracle table	O92CASTA.MONSTER_EVENT.SPOOKY_PLACE
SPOOKY_OPTIONS	Oracle table	O92CASTA.MONSTER_EVENT.SPOOKY_OPTIONS
SPOOKED_CUSTOMER	Oracle table	O92CASTA.MONSTER_EVENT.SPOOKED_CUSTOMER
SPOOKY_EVENT	Oracle table	O92CASTA.MONSTER_EVENT.SPOOKY_EVENT
MONSTER	Oracle table	O92CASTA.MONSTER_EVENT.MONSTER
MONSTER_BOOKING	Oracle table	O92CASTA.MONSTER_EVENT.MONSTER_BOOKING
EVENT_OPTIONS	Oracle table	O92CASTA.MONSTER_EVENT.EVENT_OPTIONS
Serializable	Java Interface	java.io.Serializable
println	Java Method	java.io.PrintStream.println

Called data functions (FTR = 8)

Name	Type	Full name	Scope
MONSTER	Data function	O92CASTA.MONSTER_EVENT.MONSTER	Application
SPOOKED_CUSTOMER	Data function	O92CASTA.MONSTER_EVENT.SPOOKED_CUSTOMER	Application
EVENT_OPTIONS	Data function	O92CASTA.MONSTER_EVENT.EVENT_OPTIONS	Application
SPOOKY_EVENT	Data function	O92CASTA.MONSTER_EVENT.SPOOKY_EVENT	Application
SPOOKY_PLACE	Data function	O92CASTA.MONSTER_EVENT.SPOOKY_PLACE	Application
SPOOKY_OPTIONS	Data function	O92CASTA.MONSTER_EVENT.SPOOKY_OPTIONS	Application
println	Transaction end Point	java.io.PrintStream.println	External
Serializable	Transaction end Point	java.io.Serializable	External

For all datafunctions the scope is always **Application**, but for end-points the scope can be **External** or **Application**.

User Input Security

AIPCORE-1239 - improved SecurityAnalyzer.log

The **SecurityAnalyzer.log** has been improved to list the number of distinct flaws found for each analyzed target. For example, **Distinct=** has been added:

```
2020-01-08 14:15:52,238 [1] DEBUG Analyzed target: 369/1941. Found=2, Distinct=1. Steps=128.
```

AIPCORE-1492 - support for bsh.Interpreter.eval

Added support for **bsh.Interpreter.eval** as a target for code injection.

AIPCORE-1538 - improvement to handling of constructors of System.IO.MemoryStream

Constructors of System.IO.MemoryStream are now handled correctly avoiding false positive violations to the rule **Avoid file path manipulation vulnerabilities (7752)**.

AIPCORE-1585 - improved coverage of database access methods from the .NET framework

Access to database methods of the .NET Framework are now handled more accurately. As a consequence, some false positives may be removed and new true positives may be found for the rule **Avoid SQL injection vulnerabilities (7742)**.