

# [2019-09-12] Change Log

- [New version of CQM \(2.4.0\) and Kiuwan Engine](#)
  - Support for KOTLIN programming language
  - Added Support for RM/COBOL dialect in COBOL technology.
  - Duplicated Code rules are not longer mandatory

## New version of CQM (2.4.0) and Kiuwan Engine



**Main features** are:

- **Support for KOTLIN programming language**
- **Support for RM/Cobol dialect**
- **Duplicated Code rules are not longer mandatory**

CQM is the default Model (i.e. a concrete set of active and pre-configured rules):

- If you are using **CQM**,
  - **new rules will automatically become active** and will be applied to new analyses
- If you are using your own **custom model**, **your model remains unchanged**, but *you can modify it and activate the new rules* (in case you want to be applied to your code).

You can find new rules by comparing this release of CQM against previous version. A detailed description of the behavior of these new rules is available in rule's description.

This **new version of Kiuwan Engine** incorporates **bug fixes, performance and reliability improvements in rules and parsers**.

Kiuwan Engine is the binary code executed when an analysis is run.

- **If the engine is not blocked** in your Kiuwan account, **the engine will upgrade automatically** to the last version of Kiuwan Engine once a new analysis is run
- **If the engine is blocked**, your kiuwan **engine will not be modified**.



A **new version of Kiuwan Engine** has been released that incorporates **bug fixes, performance and reliability improvements in rules and parsers**.

Kiuwan Engine is the binary code executed when an analysis is run.

- **If the engine is not blocked** in your Kiuwan account, **the engine will upgrade automatically** to the last version of Kiuwan Engine once a new analysis is run
- **If the engine is blocked**, your kiuwan **engine will not be modified**.

## Support for KOTLIN programming language

**Kotlin** is a cross-platform, statically typed, general-purpose programming language with type inference and it's **Google's preferred language for Android app development** since 7 May 2019.

Kotlin is **officially supported by Google for mobile development on Android**. It targets the JVM, but also compiles to JavaScript or native code (via LLVM), and it's an alternative to the standard Java compiler.

As Kotlin is becoming a widely adopted programming language, **Kiuwan** now incorporates **support to analyze Kotlin source files**, thus searching for code and desing conditions that are indicative of **security vulnerabilities**.



Kiuwan provides **100 Rules** specifically suited to **Kotlin** programming language.

You can find these rules going to **Models Management**, select **CQM** and search for **Rules** applying to **Kotlin language**

The screenshot shows the Kiuwan Models Management interface. The top navigation bar includes tabs for CODE SECURITY, CODE ANALYSIS, INSIGHTS, LIFE CYCLE, GOVERNANCE, and MODELS. The MODELS tab is active. Below the navigation bar, there's a header with 'SUMMARY', 'INDICATORS', 'RULES', and 'METRICS'. The 'RULES' tab is selected. A search bar at the top right shows '2019/08/14 - 2019/09/12' and a 'PUBLISH' button. A dropdown menu shows 'CQM > Version 2019/09/12 11:44 - v2.4.0'. Below this, the 'RULES' section has a table with columns: Active, Name or description, Language, Characteristic, Vulnerability type, Priority, Effort, Normative, Framework, and Tag. The 'Language' column is filtered to 'Kotlin'. A table of rules is displayed below, with columns: Active, Name, Language, Characteristic, Vulnerability type, Priority, and Repair difficulty. The rules listed include: Do not write IP address in source code, Insecure transport, Sensitive information exposed through JSONP, Native Code Exposed, Password Management - Password in Redirect, Exposure of Private Information (Privacy Violation), Same Origin Method Execution (SOME), and Deserialization of untrusted data. A '50 of 100' indicator is visible. A 'Help' button is at the bottom right.

## Added Support for RM/COBOL dialect in COBOL technology.

Support for **RM/Cobol** dialect has been added to current list of supported Cobol dialects.

All **versions between 9.x to 12.16** (latest version at the moment of writing this post) have been successfully tested.

Current list of **supported Cobol dialects**:

- Cobol85,
- OS/VS Cobol II
- IBM Enterprise COBOL for z/OS v5
- IBM ILE COBOL 5
- HP COBOL for TNS (Tandem NonStop)
- AcuCOBOL-GT
- Net Express COBOL 5 / Server Express COBOL
- MicroFocus RM/COBOL 9.x to 12.16

## Duplicated Code rules are not longer mandatory

When creating a new model, it was mandatory to activate duplicated code rules (i.e. rules that check for repeated blocks in source code).

**This new Kiuwan release no longer forces duplicated code rules to be activated in your custom model.**

- Standard pseudo-random number generators cannot withstand cryptographic attacks.
- Denial of Service by externally controlled sleep time
- Improper Neutralization of Special Elements used in an SQL Command ("SQL Injection")

- Too broad privileges granted.
- Avoid using an user controlled Primary Key into a query
- Weak cryptographic hashes cannot guarantee data integrity
- Weak symmetric encryption algorithm.