CodeQL scanned fewer lines than expected

In this article

If CodeQL analyzed less code than than you expected, you may need to use a custom build command.

About analysis of compiled languages

For compiled languages like C/C++, C#, Go, and Java, CodeQL only scans files that are built during the analysis. Therefore the number of lines of code scanned will be lower than expected if some of the source code isn't compiled correctly. This can happen for several reasons:

1. The CodeQL `autobuild` feature uses heuristics to build the code in a repository. However, sometimes this approach results in an incomplete analysis of a repository. For example, when multiple `build.sh` commands exist in a single repository, the analysis may not be complete since the `autobuild` step will only execute one of the commands, and therefore some source files may not be compiled.
2. Some compilers do not work with CodeQL and can cause issues while analyzing the code. For example, most vendor-specific C compilers will not be recognized by CodeQL. C code will need to be compiled with a recognized compiler (for example GCC, Clang or MSVC) in order to be analyzed.

If your CodeQL analysis scans fewer lines of code than expected, you can try replacing the `autobuild step`, or inspecting the copy of the source files in the CodeQL database.

Replace the `autobuild` step

Replace the `autobuild` step with the same build commands you would use in production. This makes sure that CodeQL knows exactly how to compile all of the source files you want to scan. For more information, see "CodeQL code scanning for compiled languages."

Inspect the copy of the source files in the CodeQL database

You may be able to understand why some source files haven't been analyzed by inspecting the copy of the source code included with the CodeQL database. To obtain the database from your Actions workflow, modify the `init` step of your CodeQL workflow file and set `debug: true`.

```yaml
- name: Initialize CodeQL
  uses: github/codeql-action/init@v2
  with:
    debug: true
```

This uploads the database as an actions artifact that you can download to your local machine. For more information, see "Storing workflow data as artifacts."

The artifact will contain an archived copy of the source files scanned by CodeQL called *src.zip*. If you compare the source code files in the repository and the files in *src.zip*, you can see which types of file are missing. Once you know what types of file are not being analyzed, it is easier to understand how you may need to change the workflow for CodeQL analysis.