CodeQL Guidance

Background

CodeQL is a code analysis platform owned by Semmle, now a subsidary of GitHub. It provides value by using extractors to construct a database representing the codebase, then providing a query language to perform sematic analysis. CodeQL instruments the build of compiled languages, and directly analyzes source code for interpreted languages.

CodeQL is required as part of Microsoft's Security Developent Lifecycle (SDL) requirements. .NET Engineering Services supports CodeQL via the Guardian toolset with scan results published to Trust Services Automation (TSA).

CodeQL adds a significant time to builds. We therefore recommend creating a new, seperate pipeline instead of incorporating CodeQL scans into existing PR or testing pipelines.

TL;DR: Quickstart

CodeQL 3000:

1. Ensure your repository has the latest Arcade version
2. Copy Arcade's CodeQL pipeline definition file `azure-pipelines-codeql.yml` to your repository
3. Modify the pipeline definition to work for your repository's needs. For projects using compiled languages, like C#, update the section between 'CodeQL Initialize' and 'CodeQL Finalize' to execute your build. The CodeQL engine executes these in an instrumented environment to enable analysis. Note that if not provided, the engine may use heuristics to build.
4. CodeQL 3000 requires a specific configuration file to convey TSA Bug Filing configuration. To use TSA Bug Filing, copy Arcade's `tsaoptions.json`, updating the `areaPath`, `notificationAliases`, `repositoryName` and `codebaseName` to values appropriate for your repository. Note that your repository is likely already using TSA for other compliance tooling, and those values may be used here.
5. Create a new Pipeline executing this newly-created definition.

6. Ensure this pipeline runs on some cadence (weekly suggested) and successfully submits results (see output of the 'Finalize' task, which will link to results)

See https://aka.ms/codeql3000 for documentation. Users with builds in the dnceng/internal org may now stand up a pipeline using the CodeQL 3000 tasks installed at the organization level

Since this involves wrapping the user's specific build with Azure Pipelines tasks, the Arcade repo does not provide a template for this; however, an example of how this is done in dotnet/arcade can be seen at https://github.com/dotnet/arcade/blob/main/azure-pipelines-codeql.yml.

```
variables:
- name: Codeql.Enabled
  value: true
- name: Codeql.Cadence
  value: 0
- name: Codeql.TSAEnabled
  value: true

# then in the pipeline:

  - task: CodeQL3000Init@0
    displayName: CodeQL Initialize

  # ... build as normal

  - task: CodeQL3000Finalize@0
    displayName: CodeQL Finalize
```

Use with Arcade

NOTE: Arcade previously provided a Guardian-based job and step template that executed CodeQL using the Guardian toolset. This does not meet the requirements for CodeQL reporting and should not be used, as while it may produce scans it does not satisfy organizational requirements for CodeQL execution.

Much of https://github.dev/dotnet/arcade/blob/main/azure-pipelines-codeql.yml can be copied and used as-is for most repositories. With CodeQL3000, as long as the build completes and exercises the codebase to be scanned, all languages should be reported by the same tasks at the same time.

Alert suppression

Suppression may be done using inline comments (in whatever comment form is appropriate for the language). The comment must appear on the same line as the alert, or the first line if the alert spans multiple lines.

A suppression comment is made of:

1. The string "lgtm[1]" (case insensitive)
2. A query ID surrounded by square brackets
3. A justification string of at least 25 characters

[1] At the time of this writing, tools expect `lgtm` instead of `codeql`. This may change as versions evolve and the new name propagates.

For example, in C#,

```
// lgtm [cs/weak-crypto] Algorithm needed per standard and contained safely here
```

Language-specific examples and some variations may be found in LGTM's Alert Suppression document.

Further Reading

GitHub's official CodeQL documentation: CodeQL documentation

Was this helpful? 👍 👎