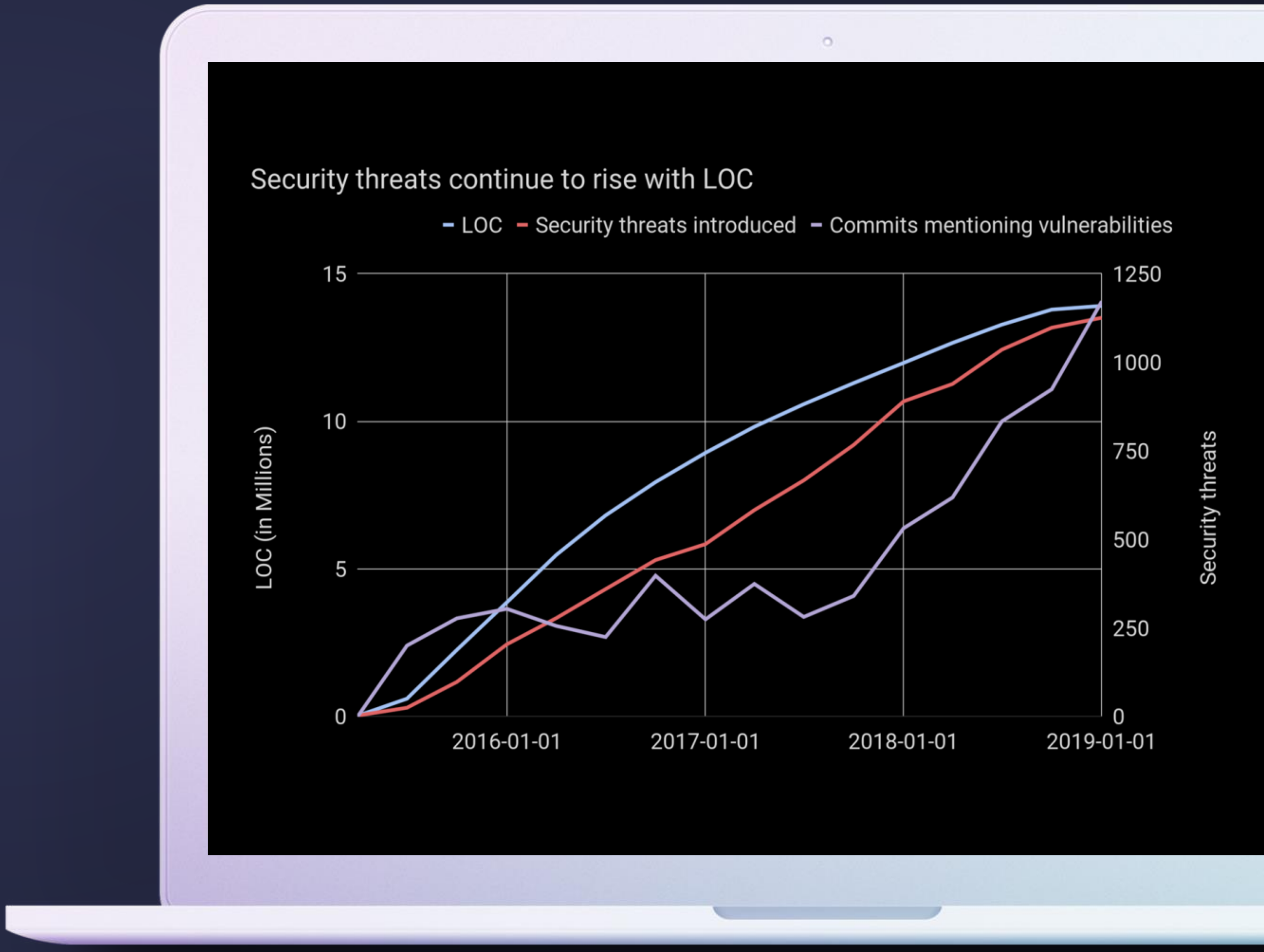
DevSecOps

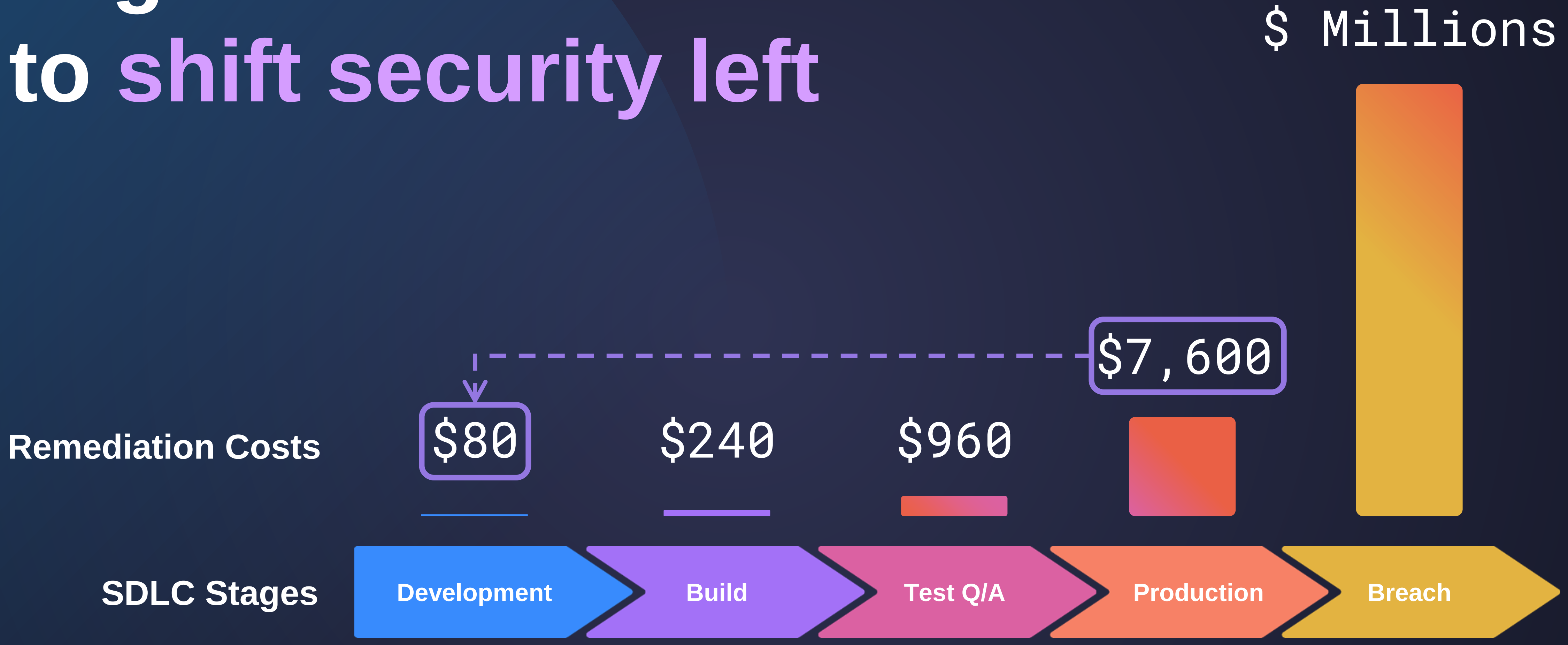
with Github Advanced Security

Houssem Dellai, CSA at Microsoft

Despite increasing developer awareness, security threats continue to rise



Organizations want to shift security left



Sources: NIST, Ponemon Institute

GitHub Advanced Security

Dependency graph
View your dependencies

Advisory database
Canonical database of dependency vulnerabilities

Security alerts and updates
Notifications for vulnerabilities in your dependencies, and pull requests to fix them

Dependency review
Identify new dependencies and vulnerabilities in a PR

Secret scanning
Find API tokens or other secrets exposed anywhere in your git history.

Code scanning
Static analysis of every git push, integrated into the developer workflow and powered by CodeQL

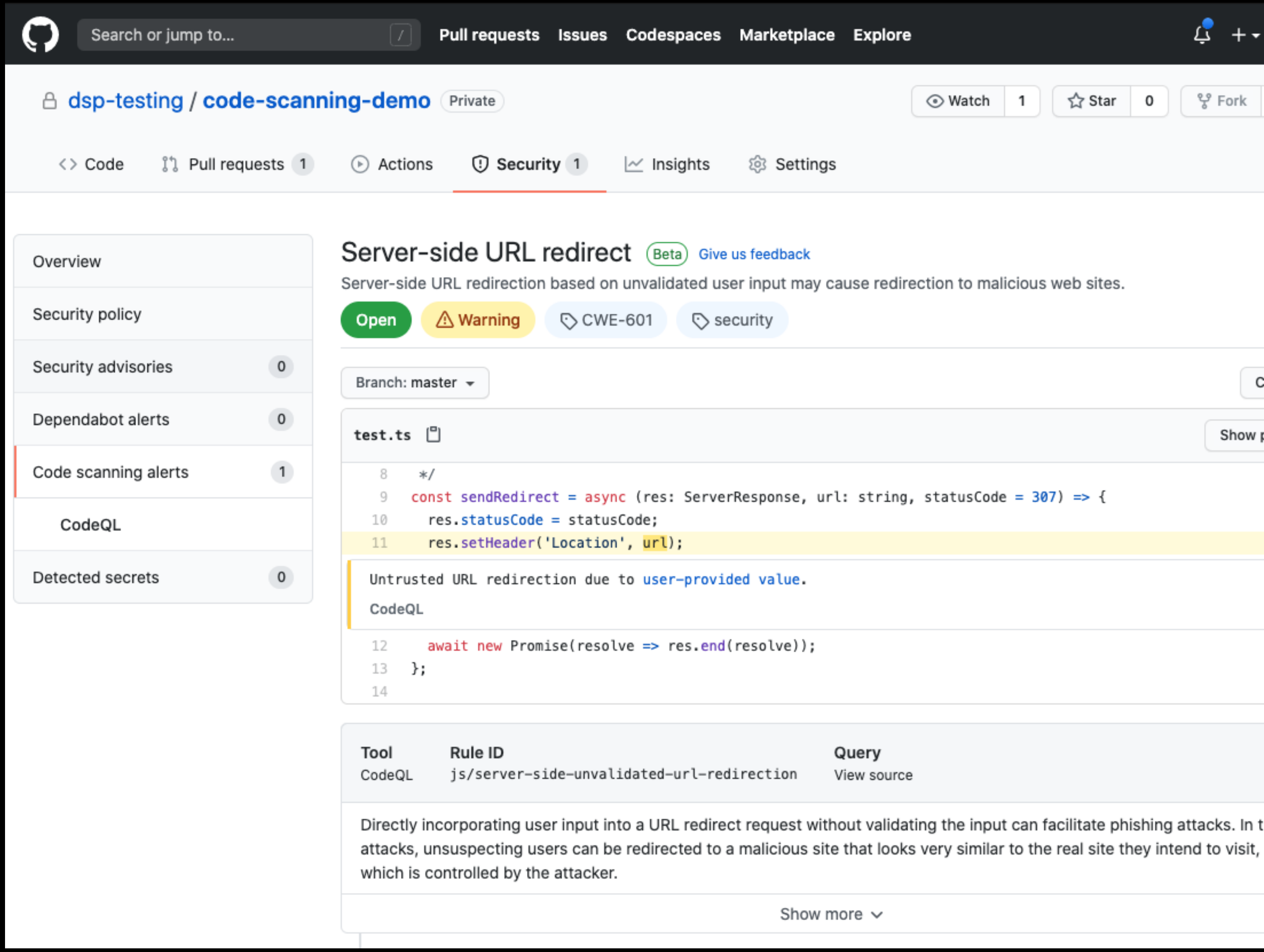
Branch protection
Enforce requirement for pushing to a branch or merging PRs

Commit signing
Enforce requirement that all commits are signed

Security overview
View security results of all kinds across your organization

Code scanning

- Find vulnerabilities before they are merged into the code base with automated CodeQL scans
- Integrate results directly into the developer workflow
- Run custom queries and the community-powered GitHub query set
- Extensible, with support for other SAST tools



Server-side URL redirect

Server-side URL redirection based on unvalidated user input may cause redirection to malicious web sites.

test.ts

```
  */
const sendRedirect = async (res: ServerResponse, url: string, statusCode = 307) => {
  res.statusCode = statusCode;
  res.setHeader('Location', url);
```

Untrusted URL redirection due to user-provided value.

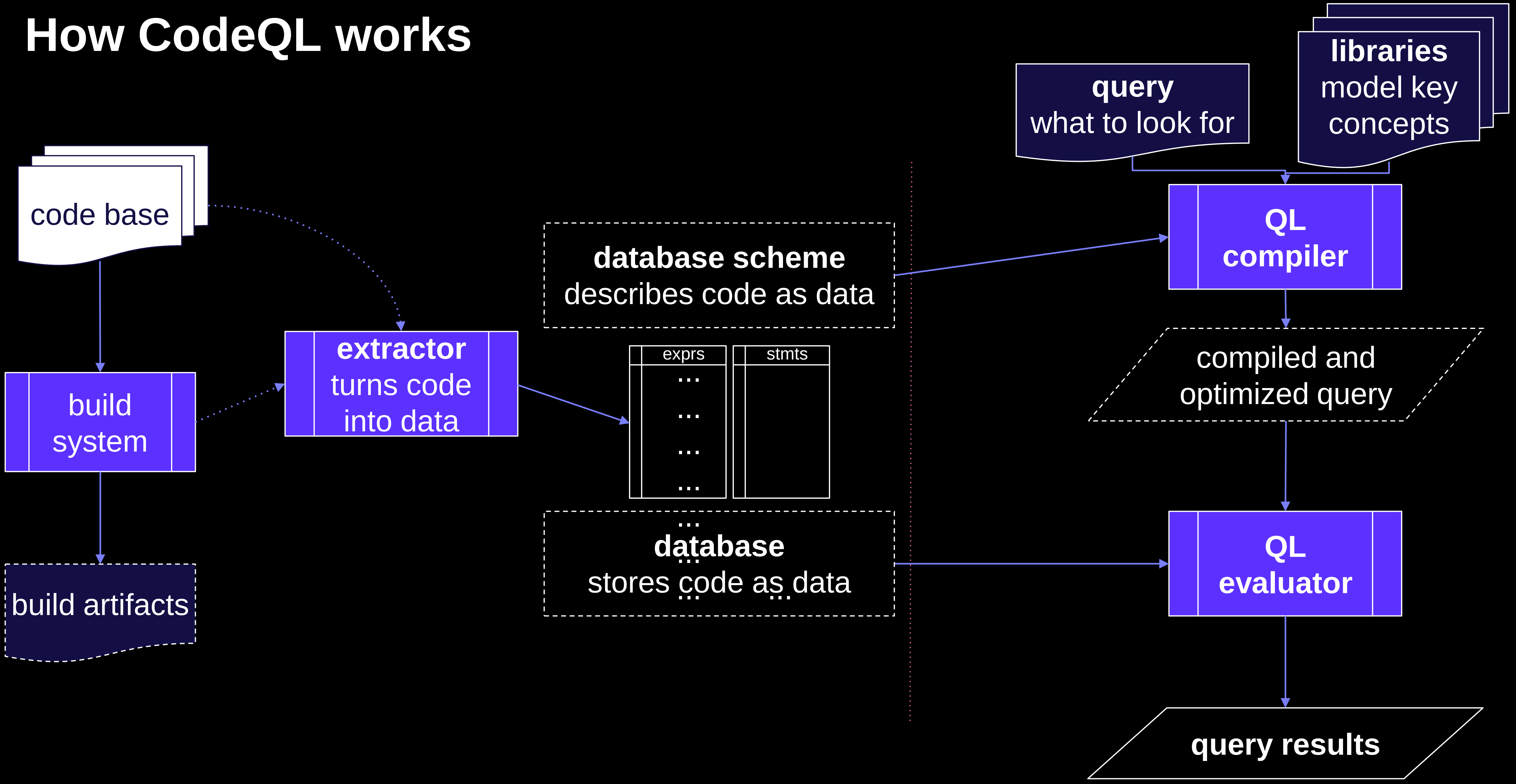
CodeQL

```
  await new Promise(resolve => res.end(resolve));
};
```

Tool	Rule ID	Query
CodeQL	js/server-side-unvalidated-url-redirection	View source

Directly incorporating user input into a URL redirect request without validating the input can facilitate phishing attacks. In t attacks, unsuspecting users can be redirected to a malicious site that looks very similar to the real site they intend to visit, which is controlled by the attacker.

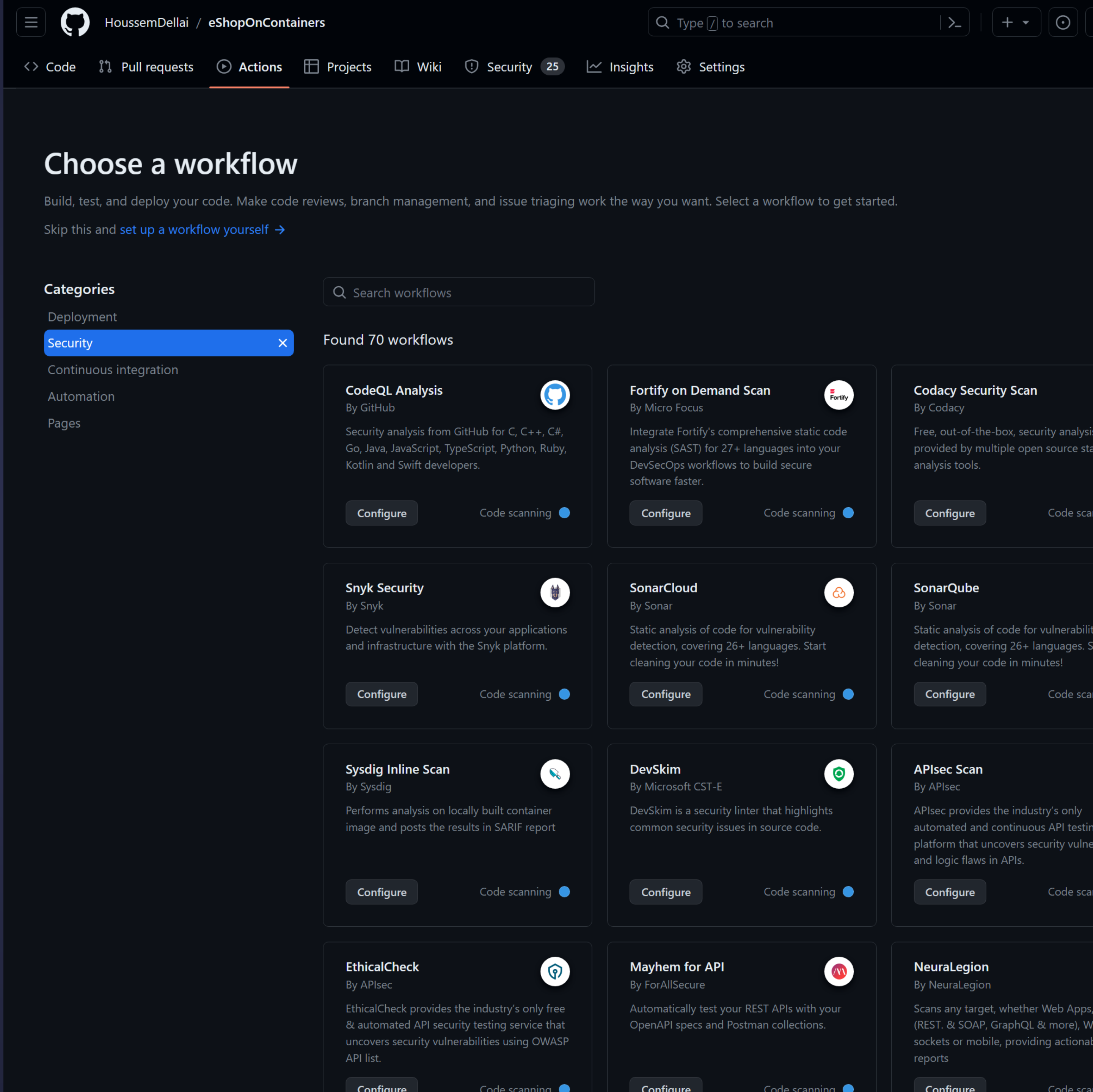
How CodeQL works



Code Scanning

Integrate any static application security testing (SAST) engine.

Use CodeQL, an open source engine, or any commercial third-party SAST tool



Choose a workflow

Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Select a workflow to get started.

Skip this and set up a workflow yourself →

Found 70 workflows

CodeQL Analysis
By GitHub
Security analysis from GitHub for C, C++, C#, Go, Java, JavaScript, TypeScript, Python, Ruby, Kotlin and Swift developers.
Configure — Code scanning

Fortify on Demand Scan
By Micro Focus
Integrate Fortify's comprehensive static code analysis (SAST) for 27+ languages into your DevSecOps workflows to build secure software faster.
Configure — Code scanning

Codacy Security Scan
By Codacy
Free, out-of-the-box, security analysi provided by multiple open source sta analysis tools.
Configure — Code sca

Snyk Security
By Snyk
Detect vulnerabilities across your applications and infrastructure with the Snyk platform.
Configure — Code scanning

SonarCloud
By Sonar
Static analysis of code for vulnerability detection, covering 26+ languages. Start cleaning your code in minutes!
Configure — Code scanning

SonarQube
By Sonar
Static analysis of code for vulnerabilit detection, covering 26+ languages. S cleaning your code in minutes!
Configure — Code sca

Sysdig Inline Scan
By Sysdig
Performs analysis on locally built container image and posts the results in SARIF report
Configure — Code scanning

DevSkim
By Microsoft CST-E
DevSkim is a security linter that highlights common security issues in source code.
Configure — Code scanning

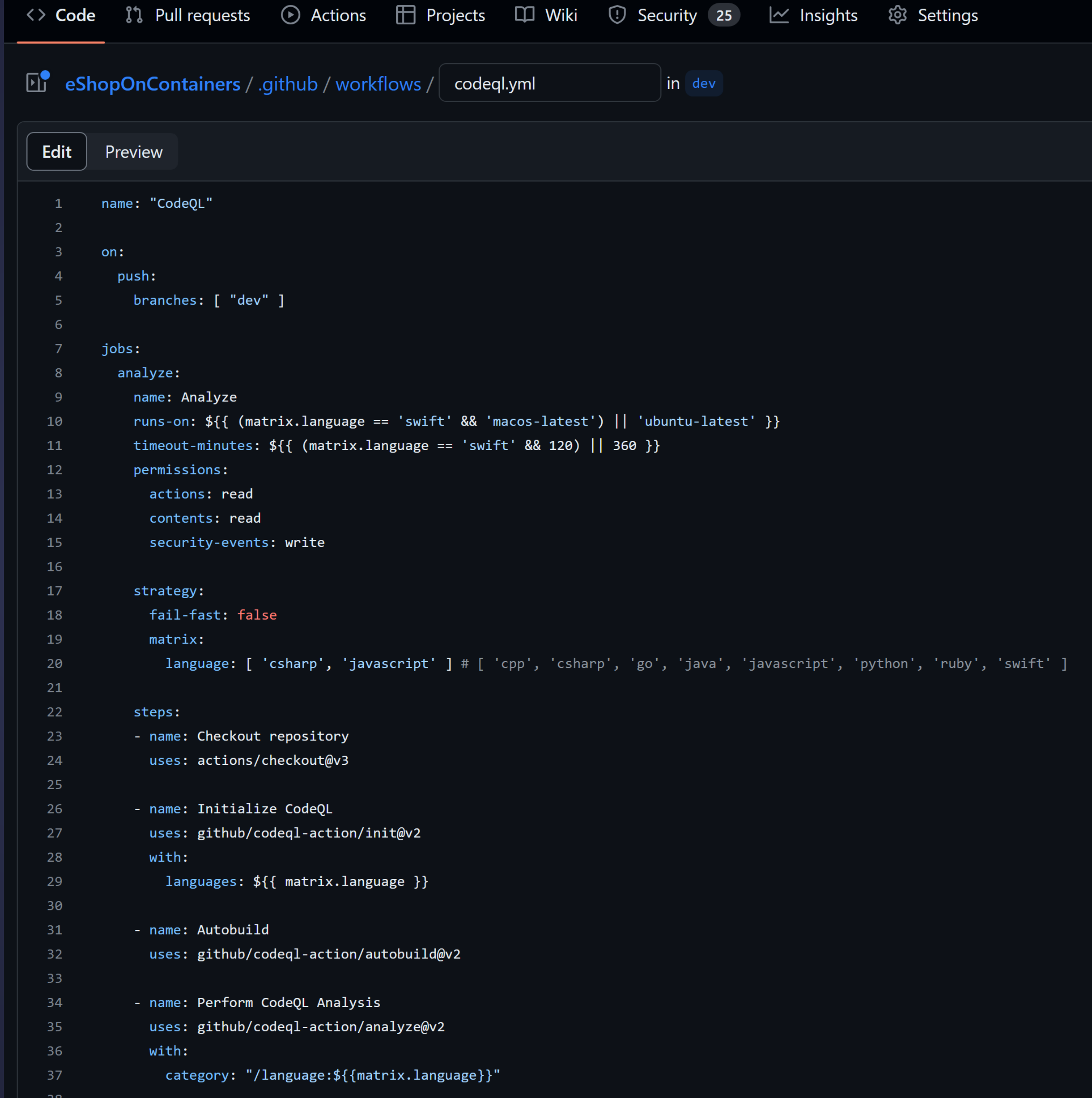
APIsec Scan
By APIsec
APIsec provides the industry's only automated and continuous API testin platform that uncovers security vulne and logic flaws in APIs.
Configure — Code sca

EthicalCheck
By APIsec
EthicalCheck provides the industry's only free & automated API security testing service that uncovers security vulnerabilities using OWASP API list.
Configure — Code scanning

Mayhem for API
By ForAllSecure
Automatically test your REST APIs with your OpenAPI specs and Postman collections.
Configure — Code scanning

NeuraLegion
By NeuraLegion
Scans any target, whether Web Apps (REST. & SOAP, GraphQL & more), W sockets or mobile, providing actionab reports
Configure — Code sca

CodeQL pipeline



```yaml
name: "CodeQL"

on:
  push:
    branches: [ "dev" ]

jobs:
  analyze:
    name: Analyze
    runs-on: ${{ (matrix.language == 'swift' && 'macos-latest') || 'ubuntu-latest' }}
    timeout-minutes: ${{ (matrix.language == 'swift' && 120) || 360 }}
    permissions:
      actions: read
      contents: read
      security-events: write

    strategy:
      fail-fast: false
      matrix:
        language: [ 'csharp', 'javascript' ] # [ 'cpp', 'csharp', 'go', 'java', 'javascript', 'python', 'ruby', 'swift' ]

    steps:
    - name: Checkout repository
      uses: actions/checkout@v3

    - name: Initialize CodeQL
      uses: github/codeql-action/init@v2
      with:
        languages: ${{ matrix.language }}

    - name: Autobuild
      uses: github/codeql-action/autobuild@v2

    - name: Perform CodeQL Analysis
      uses: github/codeql-action/analyze@v2
      with:
        category: "/language:${{matrix.language}}"
```

CodeQL pipeline

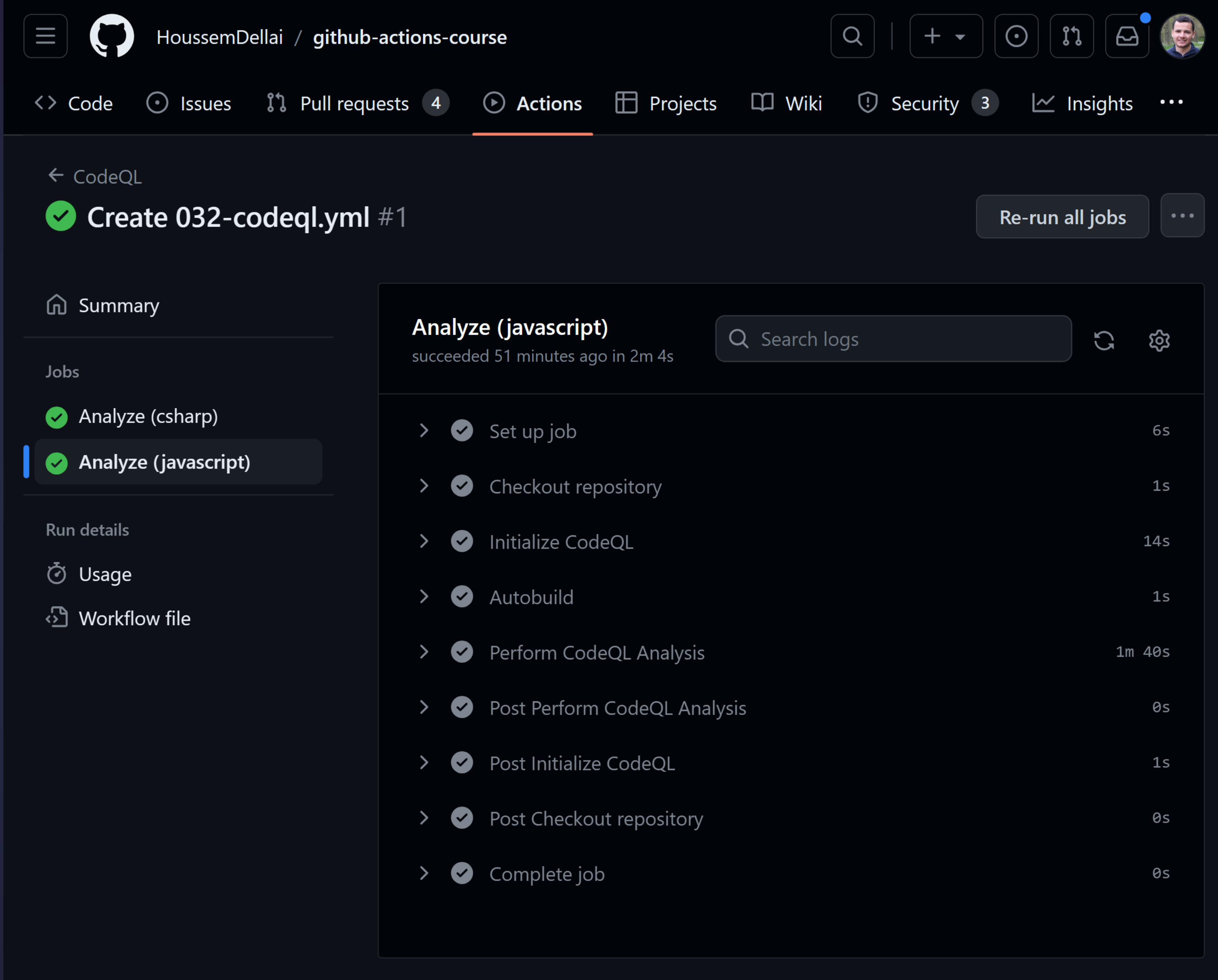
```yaml
steps:
- name: Checkout repository
  uses: actions/checkout@v3

- name: Initialize CodeQL
  uses: github/codeql-action/init@v2
  with:
    languages: 'python'

- name: Autobuild
  uses: github/codeql-action/autobuild@v2

- name: Perform CodeQL Analysis
  uses: github/codeql-action/analyze@v2
  with:
    category: "/language:${{matrix.language}}"
```

CodeQL scan


HoussemDellai / github-actions-course

Code | Issues | Pull requests 4 | Actions | Projects | Wiki | Security 3 | Insights

← CodeQL

Create 032-codeql.yml #1

Re-run all jobs

Summary

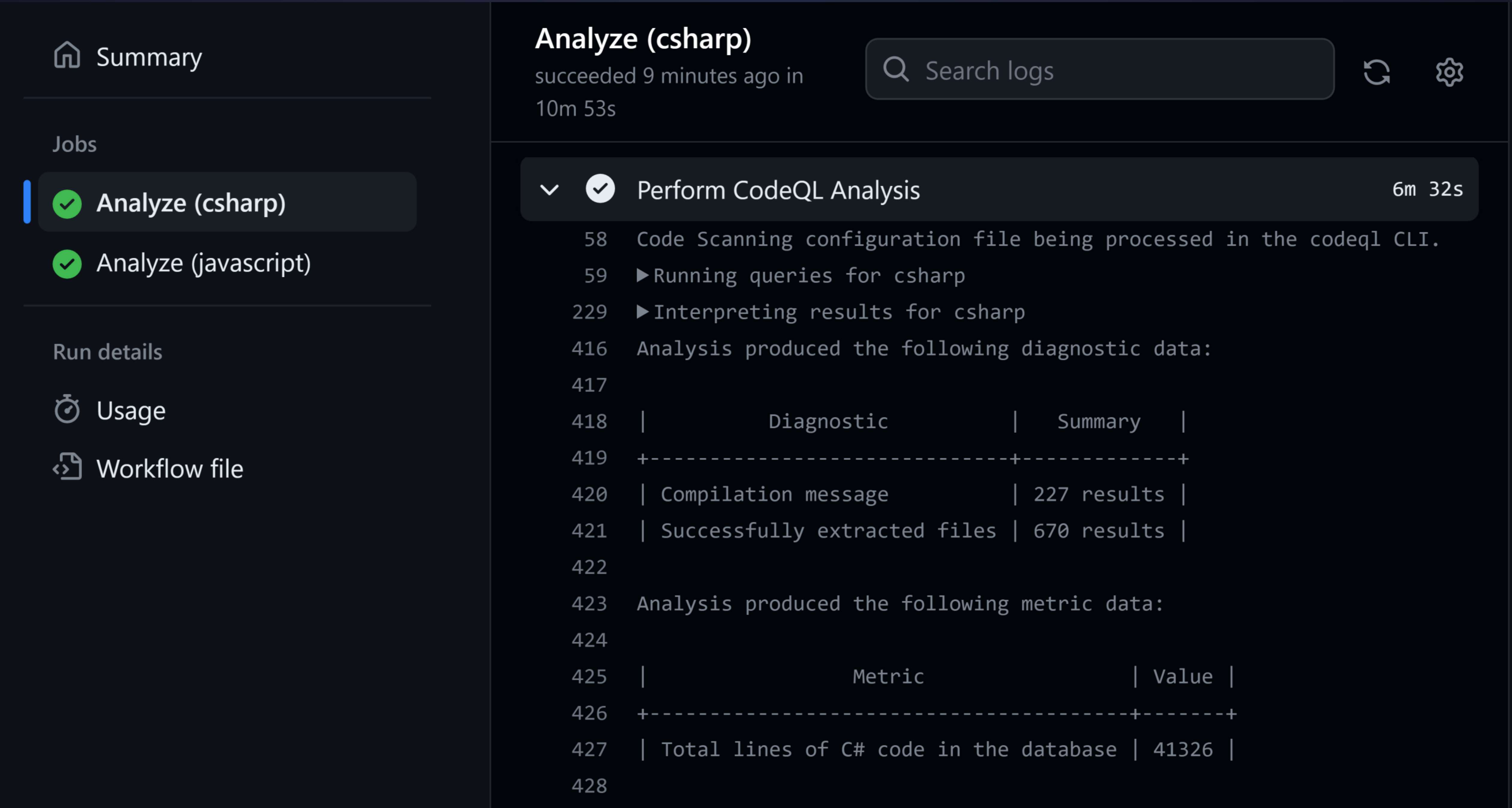
Jobs
- Analyze (csharp)
- Analyze (javascript)

Run details
- Usage
- Workflow file

Analyze (javascript)
succeeded 51 minutes ago in 2m 4s

Search logs

Set up job	6s
Checkout repository	1s
Initialize CodeQL	14s
Autobuild	1s
Perform CodeQL Analysis	1m 40s
Post Perform CodeQL Analysis	0s
Post Initialize CodeQL	1s
Post Checkout repository	0s
Complete job	0s



Analyze (csharp)

succeeded 9 minutes ago in 10m 53s

Perform CodeQL Analysis 6m 32s

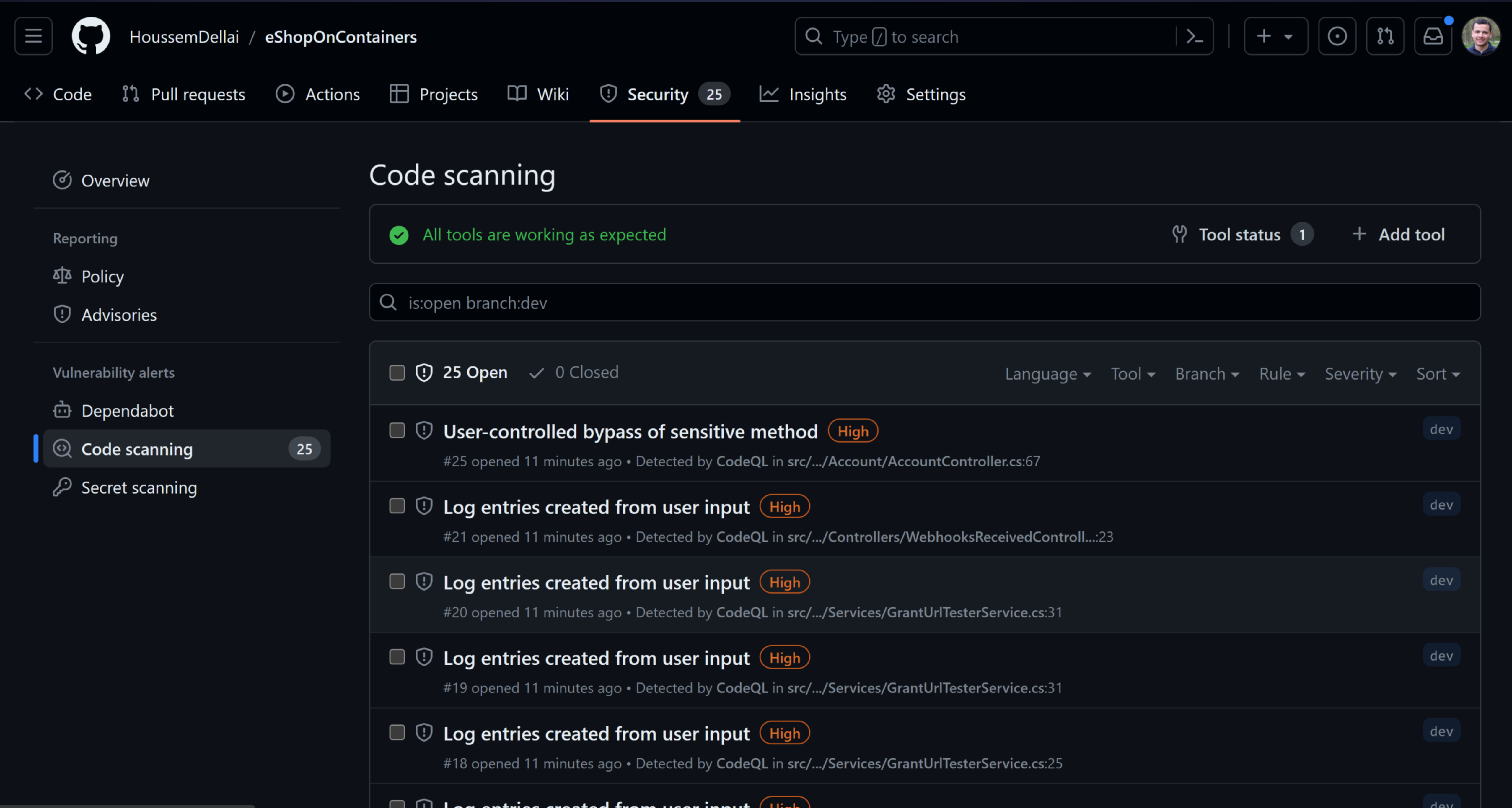
```
Code Scanning configuration file being processed in the codeql CLI.
▶Running queries for csharp
▶Interpreting results for csharp
Analysis produced the following diagnostic data:

|          Diagnostic           |   Summary   |
+-------------------------------+-------------+
| Compilation message           | 227 results |
| Successfully extracted files  | 670 results |

Analysis produced the following metric data:

|                Metric                  | Value |
+----------------------------------------+-------+
| Total lines of C# code in the database | 41326 |

```



Code scanning

All tools are working as expected — Tool status 1 — Add tool

is:open branch:dev

25 Open ✓ 0 Closed — Language ▾ Tool ▾ Branch ▾ Rule ▾ Severity ▾ Sort ▾

User-controlled bypass of sensitive method High — dev
#25 opened 11 minutes ago • Detected by CodeQL in src/.../Account/AccountController.cs:67

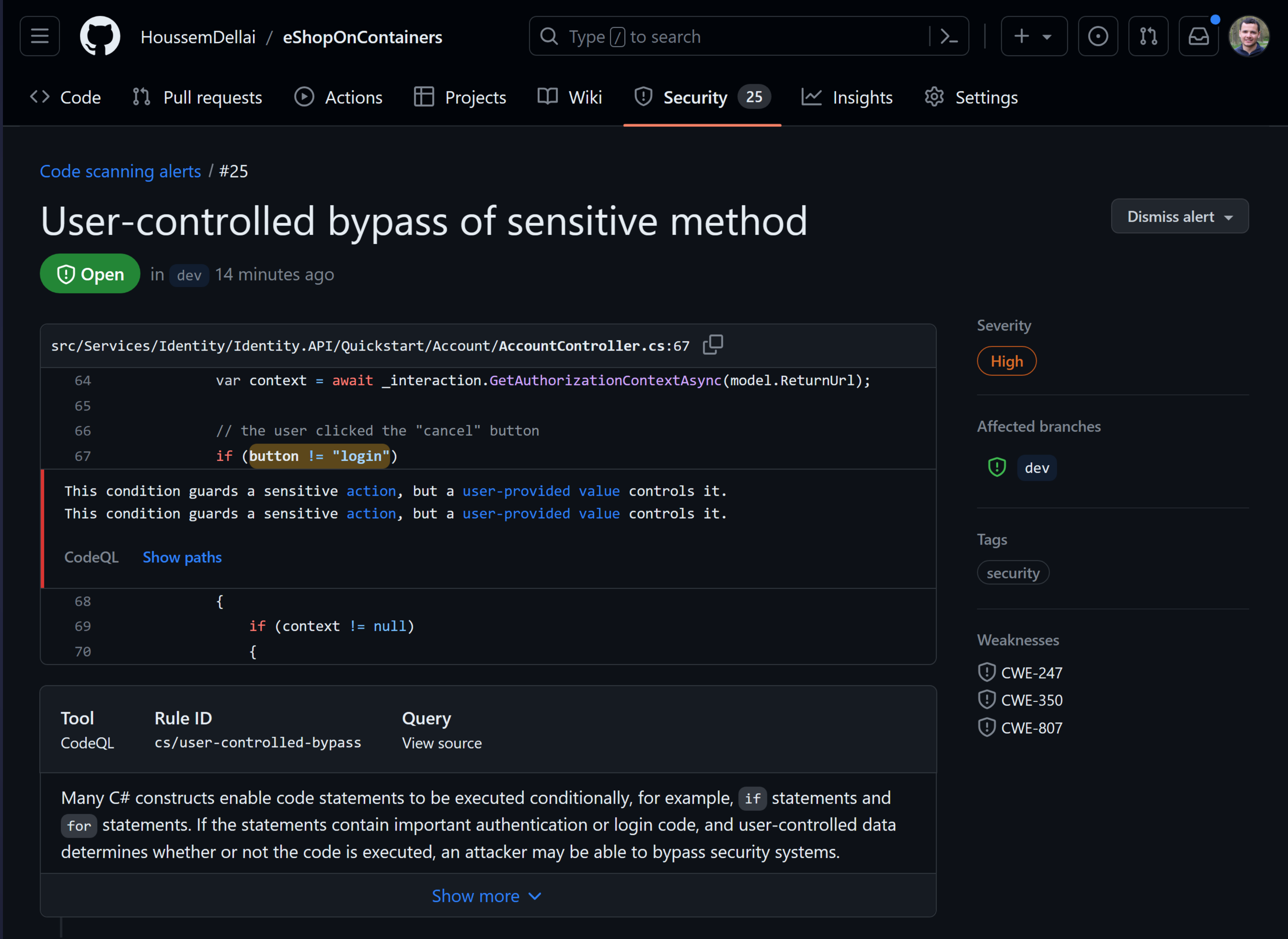
Log entries created from user input High — dev
#21 opened 11 minutes ago • Detected by CodeQL in src/.../Controllers/WebhooksReceivedControll...:23

Log entries created from user input High — dev
#20 opened 11 minutes ago • Detected by CodeQL in src/.../Services/GrantUrlTesterService.cs:31

Log entries created from user input High — dev
#19 opened 11 minutes ago • Detected by CodeQL in src/.../Services/GrantUrlTesterService.cs:31

Log entries created from user input High — dev
#18 opened 11 minutes ago • Detected by CodeQL in src/.../Services/GrantUrlTesterService.cs:25

CodeQL alerts



Code scanning alerts / #25

User-controlled bypass of sensitive method

Open in dev 14 minutes ago

Dismiss alert

src/Services/Identity/Identity.API/Quickstart/Account/**AccountController.cs**:67

```
            var context = await _interaction.GetAuthorizationContextAsync(model.ReturnUrl);

            // the user clicked the "cancel" button
            if (button != "login")
```

```
This condition guards a sensitive action, but a user-provided value controls it.
This condition guards a sensitive action, but a user-provided value controls it.
```

CodeQL Show paths

```
            {
                if (context != null)
                {
```

Tool	Rule ID	Query
CodeQL	cs/user-controlled-bypass	View source

Many C# constructs enable code statements to be executed conditionally, for example, `if` statements and `for` statements. If the statements contain important authentication or login code, and user-controlled data determines whether or not the code is executed, an attacker may be able to bypass security systems.

Show more

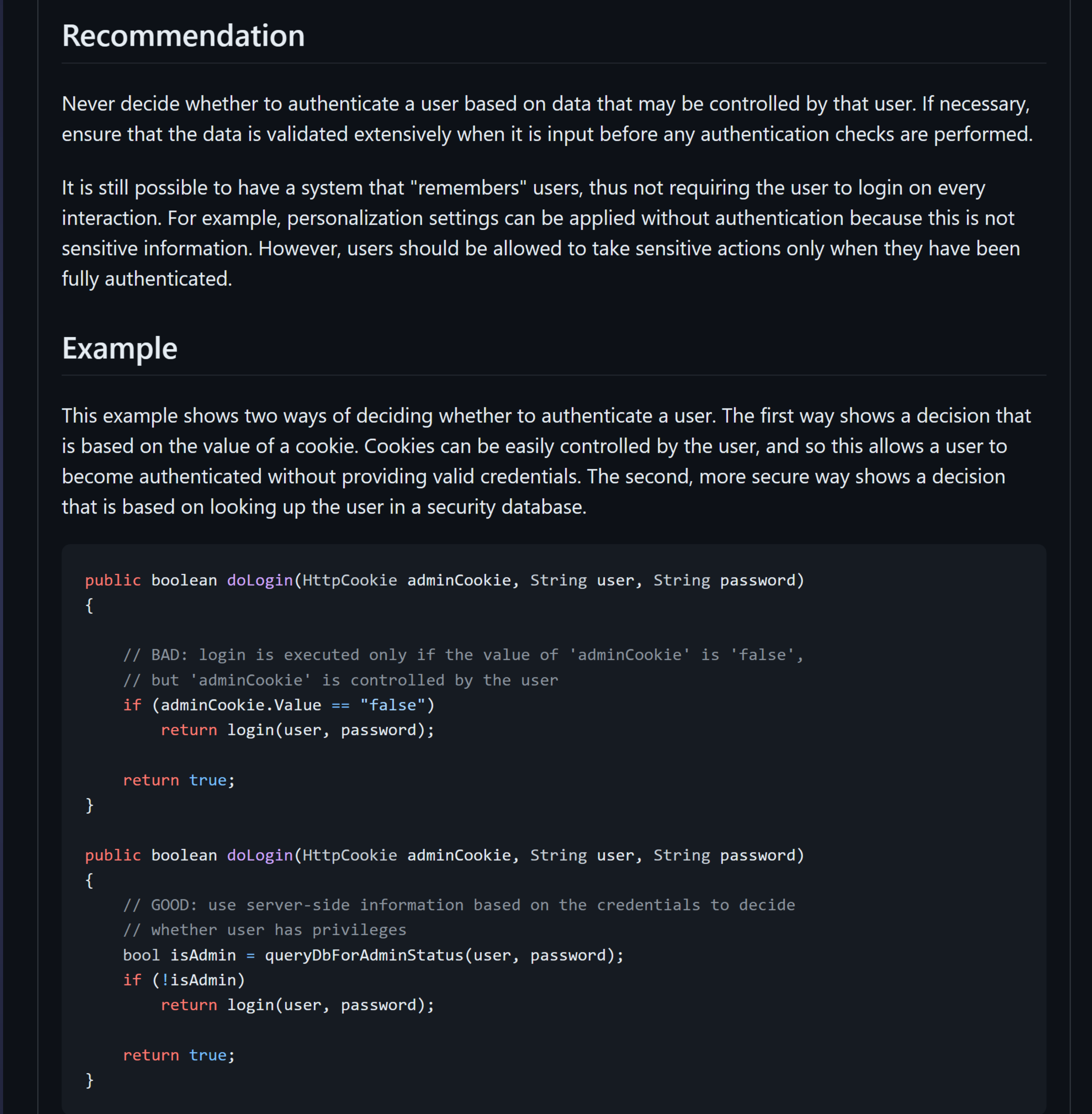
Severity
High

Affected branches
dev

Tags
security

Weaknesses
CWE-247
CWE-350
CWE-807

Alert mitigation /recommendation



Recommendation

Never decide whether to authenticate a user based on data that may be controlled by that user. If necessary, ensure that the data is validated extensively when it is input before any authentication checks are performed.

It is still possible to have a system that "remembers" users, thus not requiring the user to login on every interaction. For example, personalization settings can be applied without authentication because this is not sensitive information. However, users should be allowed to take sensitive actions only when they have been fully authenticated.

Example

This example shows two ways of deciding whether to authenticate a user. The first way shows a decision that is based on the value of a cookie. Cookies can be easily controlled by the user, and so this allows a user to become authenticated without providing valid credentials. The second, more secure way shows a decision that is based on looking up the user in a security database.

```
public boolean doLogin(HttpCookie adminCookie, String user, String password)
{

    // BAD: login is executed only if the value of 'adminCookie' is 'false',
    // but 'adminCookie' is controlled by the user
    if (adminCookie.Value == "false")
        return login(user, password);

    return true;
}

public boolean doLogin(HttpCookie adminCookie, String user, String password)
{
    // GOOD: use server-side information based on the credentials to decide
    // whether user has privileges
    bool isAdmin = queryDbForAdminStatus(user, password);
    if (!isAdmin)
        return login(user, password);

    return true;
}
```