



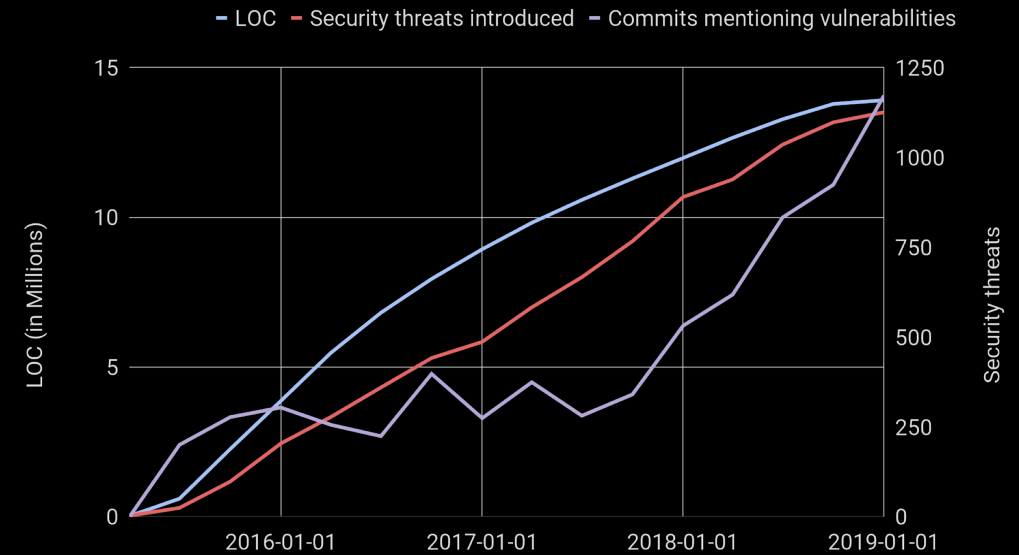
# DevSecOps with Github Advanced Security

Housseem Dellai, CSA at Microsoft

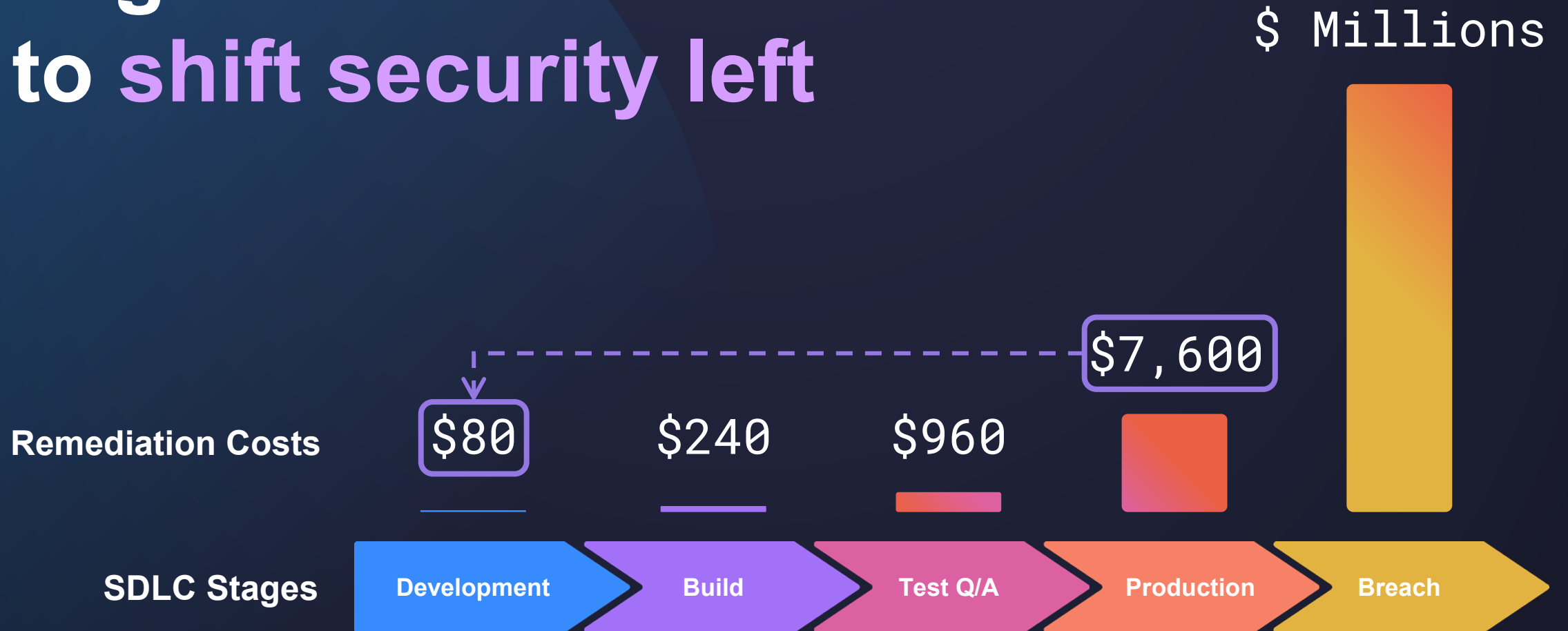


**Despite increasing developer awareness, security threats continue to rise**

Security threats continue to rise with LOC



# 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

The screenshot displays the GitHub Code Scanning interface for a repository named `dsp-testing / code-scanning-demo`. The interface is divided into a sidebar on the left and a main content area on the right.

**Sidebar:**

- Overview
- Security policy
- Security advisories (0)
- Dependabot alerts (0)
- Code scanning alerts (1) - **Selected**
- CodeQL
- Detected secrets (0)

**Main Content Area:**

**Server-side URL redirect** (Beta) [Give us feedback](#)

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

**Alert Details:**

- Open** (Green button)
- Warning** (Yellow button)
- CWE-601** (Blue button)
- security** (Blue button)

Branch: master

**test.ts**

```
8  */
9  const sendRedirect = async (res: ServerResponse, url: string, statusCode = 307) => {
10    res.statusCode = statusCode;
11    res.setHeader('Location', url);
12
13    await new Promise(resolve => res.end(resolve));
14  };
15
```

**Alert Message:** Untrusted URL redirection due to `user-provided value`.

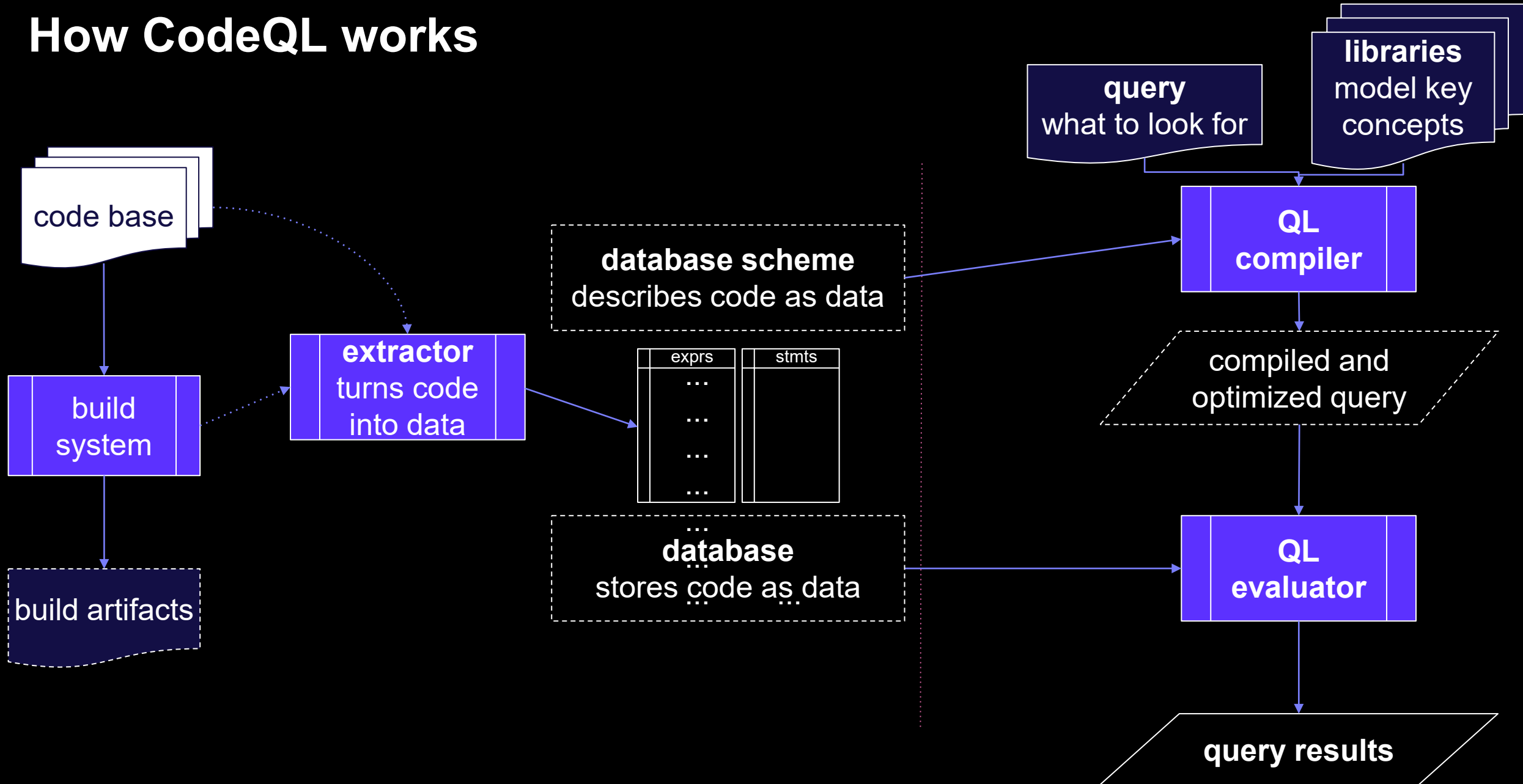
**CodeQL**

Tool	Rule ID	Query
CodeQL	js/server-side-unvalidated-url-redirection	<a href="#">View source</a>

Directly incorporating user input into a URL redirect request without validating the input can facilitate phishing attacks. In these 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.

[Show more](#)

# 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

The screenshot shows the GitHub Actions interface for a repository named 'eShopOnContainers' by user 'HoussemDellai'. The 'Actions' tab is active, displaying a 'Choose a workflow' page. A sidebar on the left lists categories: Deployment, Security (selected), Continuous integration, Automation, and Pages. The main area shows 'Found 70 workflows' and a grid of workflow cards. Each card includes a title, author, description, a 'Configure' button, and a 'Code scanning' status indicator.

Workflow Name	Author	Description	Configure	Code scanning
CodeQL Analysis	By GitHub	Security analysis from GitHub for C, C++, C#, Go, Java, JavaScript, TypeScript, Python, Ruby, Kotlin and Swift developers.	Configure	●
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	●
Codacy Security Scan	By Codacy	Free, out-of-the-box, security analysis provided by multiple open source static analysis tools.	Configure	●
Snyk Security	By Snyk	Detect vulnerabilities across your applications and infrastructure with the Snyk platform.	Configure	●
SonarCloud	By Sonar	Static analysis of code for vulnerability detection, covering 26+ languages. Start cleaning your code in minutes!	Configure	●
SonarQube	By Sonar	Static analysis of code for vulnerability detection, covering 26+ languages. Start cleaning your code in minutes!	Configure	●
Sysdig Inline Scan	By Sysdig	Performs analysis on locally built container image and posts the results in SARIF report	Configure	●
DevSkim	By Microsoft CST-E	DevSkim is a security linter that highlights common security issues in source code.	Configure	●
APIsec Scan	By APIsec	APIsec provides the industry's only automated and continuous API testing platform that uncovers security vulnerabilities and logic flaws in APIs.	Configure	●
EthicalCheck	By APIsec	EthicalCheck provides the industry's only free & automated API security testing service that uncovers security vulnerabilities using OWASP API list.	Configure	●
Mayhem for API	By ForAllSecure	Automatically test your REST APIs with your OpenAPI specs and Postman collections.	Configure	●
NeuraLegion	By NeuraLegion	Scans any target, whether Web Apps, (REST, & SOAP, GraphQL & more), Web sockets or mobile, providing actionable reports	Configure	●

# CodeQL pipeline

eShopOnContainers / .github / workflows / codeql.yml in dev

Edit Preview

```
1  name: "CodeQL"
2
3  on:
4    push:
5      branches: [ "dev" ]
6
7  jobs:
8    analyze:
9      name: Analyze
10     runs-on: ${{ (matrix.language == 'swift' && 'macos-latest') || 'ubuntu-latest' }}
11     timeout-minutes: ${{ (matrix.language == 'swift' && 120) || 360 }}
12     permissions:
13       actions: read
14       contents: read
15       security-events: write
16
17     strategy:
18       fail-fast: false
19     matrix:
20       language: [ 'csharp', 'javascript' ] # [ 'cpp', 'csharp', 'go', 'java', 'javascript', 'python', 'ruby', 'swift' ]
21
22     steps:
23     - name: Checkout repository
24       uses: actions/checkout@v3
25
26     - name: Initialize CodeQL
27       uses: github/codeql-action/init@v2
28       with:
29         languages: ${{ matrix.language }}
30
31     - name: Autobuild
32       uses: github/codeql-action/autobuild@v2
33
34     - name: Perform CodeQL Analysis
35       uses: github/codeql-action/analyze@v2
36       with:
37         category: "/language:${{matrix.language}}"
38
```











# CodeQL pipeline









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

>	 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

 Summary

## Jobs

 Analyze (csharp)

 Analyze (javascript)

## Run details

 Usage

 Workflow file

## Analyze (csharp)

succeeded 9 minutes ago in  
10m 53s

 Search logs



▼  Perform CodeQL Analysis

6m 32s

58 Code Scanning configuration file being processed in the codeql CLI.

59 ▶Running queries for csharp

229 ▶Interpreting results for csharp

416 Analysis produced the following diagnostic data:

417

	Diagnostic	Summary
--	------------	---------

	+-----+-----+
--	---------------

420	Compilation message	227 results
-----	---------------------	-------------

421	Successfully extracted files	670 results
-----	------------------------------	-------------

422

423 Analysis produced the following metric data:

424

	Metric	Value
--	--------	-------

426	+-----+-----+
-----	---------------

427	Total lines of C# code in the database	41326
-----	--	-------

428



Overview

Reporting

Policy

Advisories

Vulnerability alerts

Dependabot

**Code scanning** 25

Secret scanning

# 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

☐ **Log entries created from user input** High dev

# CodeQL alerts

HoussemDellai / eShopOnContainers

Q

Type / to search

>

<

+

⌵

⌂

🔗

📁

👤

<> Code

🔗 Pull requests

⌚ Actions

📁 Projects

📖 Wiki

🛡️ Security

25

📈 Insights

⚙️ Settings

Code scanning alerts / #25

User-controlled bypass of sensitive method

Dismiss alert

🛡️ Open

in dev

14 minutes ago

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

64

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

65

66

// the user clicked the "cancel" button

67

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

68

{

69

if (context != null)

70

{

Tool	Rule ID	Query
CodeQL	cs/user-controlled-bypass	<a href="#">View source</a>

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;
}
```