



# DER WEG ZU DEVSECOPS MIT GITHUB ADVANCED SECURITY FÜR AZURE DEVOPS

Marc Müller  
Principal Consultant



[www.4tecture.ch](http://www.4tecture.ch)

4tecture®  
empower your software solutions

# Agenda

- Intro
- Applied DevSecOps
- DevSecOps Process
- Questions





About me:

Marc Müller  
Principal Consultant  
@muellermarc



4tecture<sup>®</sup>  
empower your software solutions

Our Products:

Multi-Tenant OpenID  
Connect Identity Provider



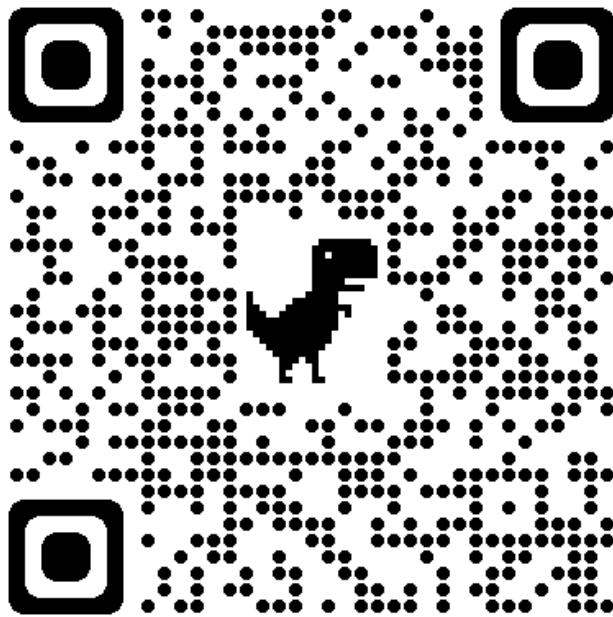
[www.proauth.net](http://www.proauth.net)

Enterprise Application  
Framework for .NET



[www.reafx.net](http://www.reafx.net)

# Slide Download



<https://www.4tecture.ch/events/bastaspring2024ghazdo>

# Intro



A close-up, low-angle shot of several rowers in a racing shell. Their hands are gripping yellow and black oars, which are angled downwards. The water is visible at the bottom, showing ripples and spray. The rowers are wearing blue and red athletic gear.

GitHub Advanced Security for Azure DevOps

# DevSecOps

4tecture®  
empower your software solutions

# DevSecOps Definition

DevSecOps is an augmentation of DevOps to allow for security practices to be integrated into the DevOps approach.

Contrary to a traditional centralized security team model, each delivery team is empowered to factor in the correct security controls into their software delivery.

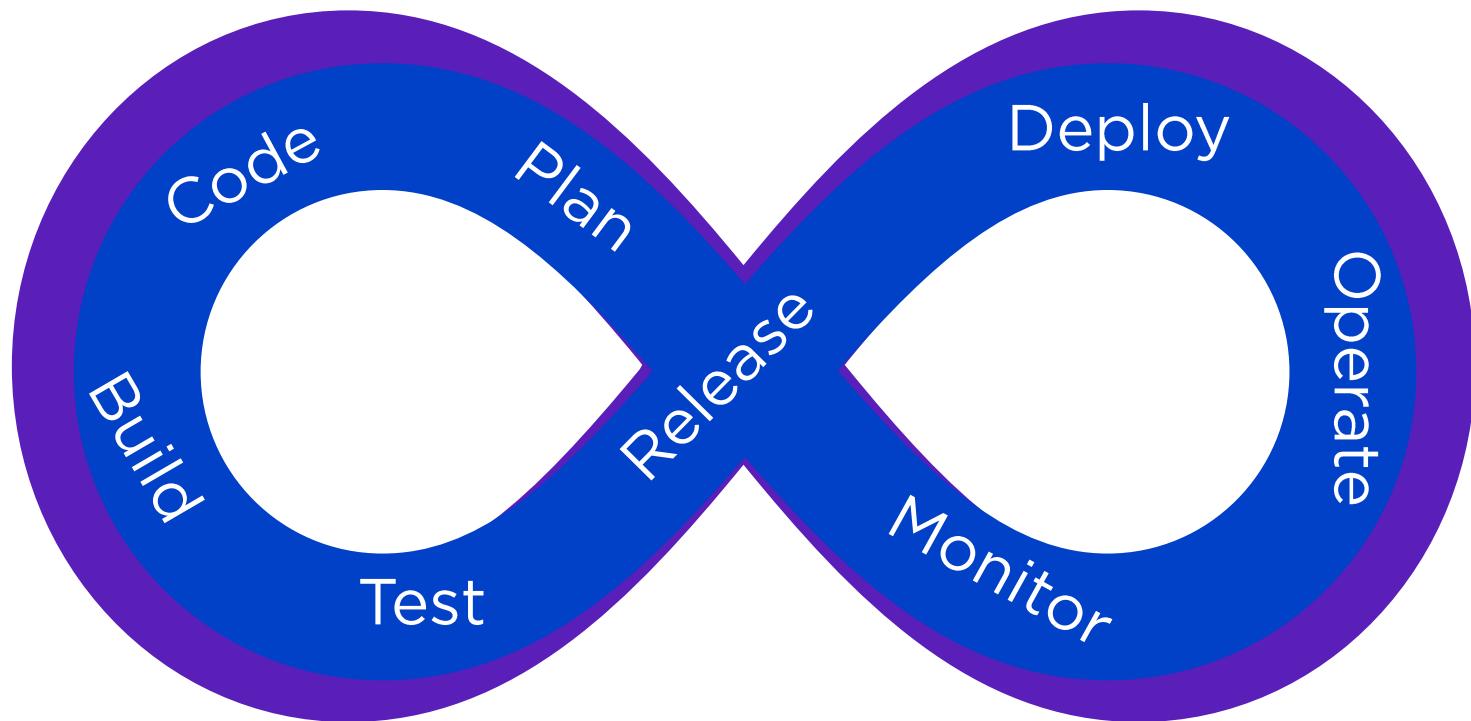
Security practices and testing are performed earlier in the development lifecycle, hence the term "shift left" can be used.

Security is tested in three main areas: static, software composition, and dynamic.

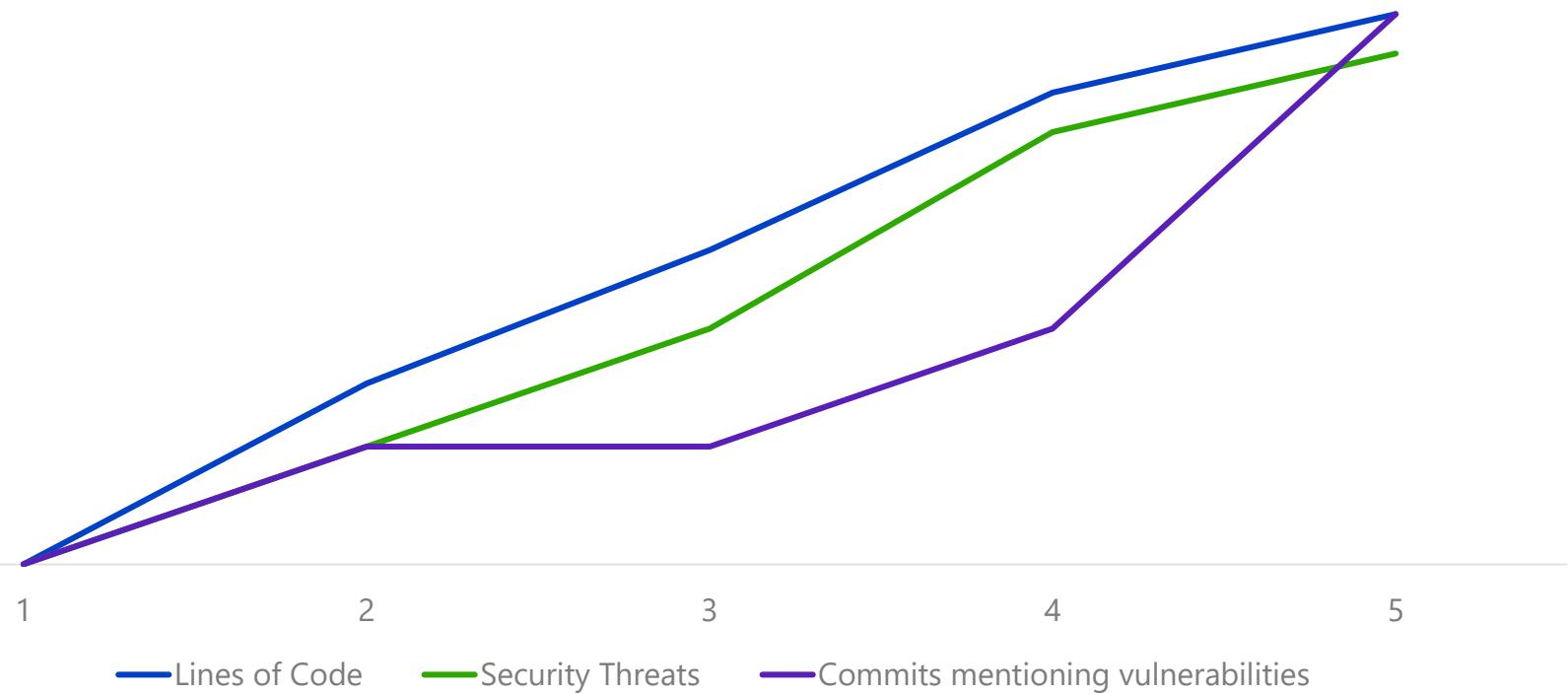
[Wikipedia](#)

[https://en.wikipedia.org/wiki/DevOps#DevSecOps,\\_Shifting\\_Security\\_Left](https://en.wikipedia.org/wiki/DevOps#DevSecOps,_Shifting_Security_Left)

# Dev Sec Ops

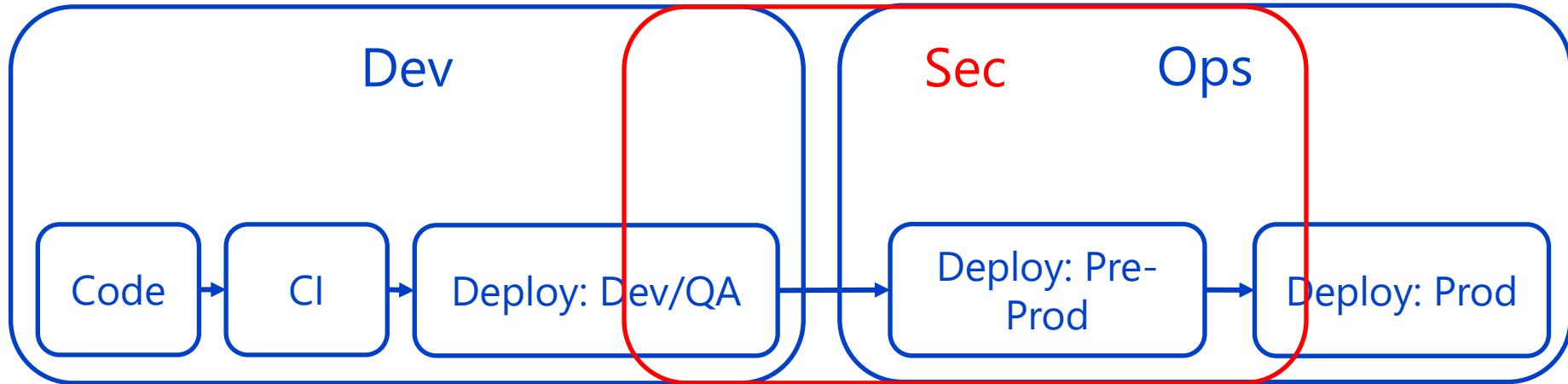


# More Code = more technical dept and exposure



Flaws in applications are consistently the #1 attack vector for breaches

# Problem: Dev-Sec-Ops Divide



No Silos!

# Security in your software lifecycle

## Supply Chain

- Open-Source Dependencies
- Alerts on vulnerabilities

## Code

- Deep scanning for vulnerabilities
  - i.e. XSS / SQL Injection

## Development Lifecycle

- Higher level insights
- Across entire organization

# DevSecOps Goal

- Bridge between security team and developers
  - Fast and safe delivery
  - Remove silos – shared responsibility of application security
- empower team to factor in the correct security controls into their software delivery

# Vulnerability Scanning

Static Application Security Testing	Software Composition Analysis	Dynamic Application Security Testing	Interactive application Security Testing
<ul style="list-style-type: none"><li>• Scan for coding errors</li><li>• Scan for weaknesses</li><li>• Scan for design flaws</li><li>• Whitebox Approach</li></ul>	<ul style="list-style-type: none"><li>• Scan dependencies for vulnerabilities</li><li>• Opensource and third-party components</li><li>• Decompose full dependency tree</li></ul>	<ul style="list-style-type: none"><li>• “hack your application”</li><li>• Tools which act like pen tester</li><li>• Web and API scanner</li><li>• Blackbox approach</li></ul>	<ul style="list-style-type: none"><li>• Detect runtime vulnerabilities</li><li>• Observe application request/response interactions, behavior and dataflow</li><li>• During manual or automated functional testing</li><li>• Active / Passive IAST</li><li>• Combining Whitebox and Runtime analysis</li></ul>

# GHAZDO

GitHub Advanced Security for Azure DevOps



A dynamic photograph of a rowing team in action. The rowers are wearing blue and red athletic gear. Their oars are extended, creating a sense of motion and teamwork. The water is visible at the bottom.

GitHub Advanced Security for Azure DevOps

# Overview

4tecture®  
empower your software solutions

**G** GHAzDODemo +

Overview

Boards

**R** Repos

Files

Commits

Pushes

Branches

Tags

Pull requests

**A** Advanced Security

Pipelines

Test Plans

Artifacts

## Advanced Security

Dependencies   Code scanning   Secrets

Filter by keywords

Branch: main ▾ State: Open ▾ Pipeline ▾ Package ▾ Severity ▾ X

Alert

Vulnerable package

First detected

Azure Identity SDK Remote Code Execution Vulnerability (CVE-2023-36414) High

#5 in HelloWorld/HelloWorld/HelloWorld.csproj

azure.identity (1.7.0) (1 root dep.)

NuGet

Dec 2, 2023

.NET Information Disclosure Vulnerability (CVE-2022-41064) Medium

#1 in HelloWorld/HelloWorld/HelloWorld.csproj

system.data.sqlclient (4.8.3)

NuGet

Sep 18, 2023

**G** GH4zDODemo +

Overview

Boards

Repos

Files

Commits

Pushes

Branches

Tags

Pull requests

Advanced Security

Pipelines

Test Plans

Artifacts

## Advanced Security

Dependencies   **Code scanning**   Secrets

Filter by keywords

Branch: main ▾ State: Open ▾ Pipeline ▾ Tool ▾ Rule ▾ Severity ▾ X

Alert

First detected

Weak encryption (cs/weak-encryption) High

#4 in HelloWorld/HelloWorld/Services/BadEncryptionService.cs:11 (+1)

Sep 18, 2023

SQL query built from user-controlled sources (cs/sql-injection) High

#3 in HelloWorld/HelloWorld/Controllers/SqlInjectionController.cs:34 (+1)

Sep 18, 2023

G GHAzDODemo +

- Overview
- Boards
- Repos
- Files
- Commits
- Pushes
- Branches
- Tags
- Pull requests
- Advanced Security
- Pipelines
- Test Plans
- Artifacts

## Advanced Security

Dependencies Code scanning Secrets

Filter by keywords

State: Open Type X

Alert

Introduced

Azure DevOps personal access token (PAT) `lztqfvva` Critical  
#6 in README.md:158 (+1)

Dec 2, 2023

Microsoft Azure

Search resources, services, and docs (G+)

mmueller@4tecture.ch  
4TECTURE DEMO ENVIRONMEN...

Home > Microsoft Defender for Cloud

# Microsoft Defender for Cloud | DevOps security

Showing subscription 'Microsoft Azure Sponsorship'

Search Add environment Refresh DevOps workbook Getting Started Manage resources Guides & Feedback

Configuration updates take up-to 5 minutes to reflect. Please refresh the dashboard to see the latest status.

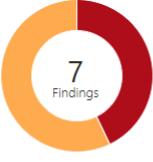
## General

- Overview
- Getting started
- Recommendations
- Attack path analysis
- Security alerts
- Inventory
- Cloud Security Explorer
- Workbooks
- Community
- Diagnose and solve problems

## Cloud Security

- Subscription == Microsoft Azure Sponsorship
- Resource type : All
- Finding type : All
- Severity : All
- Resource hierarchy view

### DevOps security findings



Severity	Count
High	3
Medium	4
Low	0

### DevOps security results

Category	Count
Code findings	0
Infrastructure as Code findings	7
Secret findings	0
Dependency findings	0

### DevOps environment posture management recommendations

0 High severity recommendations, on 0 resources

Recommendations results. Open>  
Learn more >

## Management

- Environment settings
- Security solutions
- Workflow automation

< Previous Page 1 of 2 Next > Showing 1 to 30 of 46 results.

# Private Agent Deployment

The screenshot shows a GitHub repository page for `microsoft/GHAzDO-Resources`. The repository is public and contains several scripts under the `src/agent-setup` directory, specifically `codeql-install-macos.sh`, `codeql-install-ubuntu.sh`, and `codeql-install-windows.ps1`. The `codeql-install-ubuntu.sh` file is selected and displayed in the main pane. The code is a bash script for installing CodeQL CLI Bundles to the toolcache. It includes a function `download_with_retries()` for handling downloads with retries. The script uses curl to download files from URLs and handles compressed files. A commit by `ncouraud` is visible, adding this script. The repository has 8 forks and 16 stars.

```
#!/bin/bash -e
#####
## File: codeql-install-ubuntu.sh
## Desc: Install the CodeQL CLI Bundles to the toolcache.
## Borrowed from https://github.com/actions/runner-images/
#####

download_with_retries() {
    # Due to restrictions of bash functions, positional arguments are used here.
    # In case if you using latest argument NAME, you should also set value to all previous parameters.
    # Example: download_with_retries $ANDROID_SDK_URL ./"$ANDROID_SDK.zip"
    local URL="$1"
    local DEST="${2:-}"
    local NAME="${3:-${URL##*/}}"
    local COMPRESSED="$4"

    if [[ $COMPRESSED == "compressed" ]]; then
        local COMMAND="curl \$URL -4 -sL --compressed -o '\$DEST/\$NAME' -w '%{http_code}'"
    else
        local COMMAND="curl \$URL -4 -sL -o '\$DEST/\$NAME' -w '%{http_code}'"
    fi

    echo "Downloading '\$URL' to '\$DEST/\$NAME'..."
    retries=20
    interval=30
    while [ $retries -gt 0 ]; do
        ((retries--))
        # Temporary disable exit on error to retry on non-zero exit code
        set +e
        http_code=$(eval \$COMMAND)
        set -e
    done
}
```

# Demo

Enabling GitHub Advanced Security



4tecture-demo / GHAzDODemo / Settings / Repositories

Search

G + MM

**Project Settings**  
GHAzDODemo

**General**

- Overview
- Teams
- Permissions
- Notifications
- Service hooks
- Dashboards

**Boards**

- Project configuration
- Team configuration
- GitHub connections

**Pipelines**

- Agent pools
- Parallel jobs
- Settings
- Test management
- Release retention
- Service connections
- XAML build services

**Repos**

**Repositories**

**Artifacts**

Storage

Test

## All Repositories

Repositories **Settings** Policies Security

### Advanced Security

Advanced Security is billed based on the number of unique active committers across all enabled repositories in your subscription.  
[Learn more](#) | [View billing](#)

For setup tips, [view documentation](#) or [contact sales](#).

Off **Automatically enable Advanced Security for new repositories**  
New repositories in this project will be initialized with Advanced Security enabled by default. Advanced Security can be disabled on a repository at any time.

Off **Default branch name for new repositories**  
New repositories will be initialized with this branch. You can change the default branch for a particular repository at any time. [Learn more](#)  
main

On **Allow users to manage permissions for their created branches**  
New repositories will be configured to allow users to manage permissions for their created branches

Off **Create PRs as draft by default**  
New pull requests will be created as draft by default for all repositories in this project

+ Create

Enable all Disable all

4tecture-demo / GHAzDODemo / Settings / Repositories

Search

All Repositories

Repositories Settings Policies Security

Advanced Security

Advanced Security is billed based on the number of unique active committers across all enabled repositories in your subscription.

[Learn more](#) | [View billing](#)

For setup tips, [view documentation](#) or [contact sales](#).

Off **Automatically enable /**  
New repositories in this project will automatically be enabled.

On **Manually enable /**  
New repositories in this project must be manually enabled by an administrator.

**Enable all** **Disable all**

Enable and begin billing for all repositories in this project?

Advanced Security is billed based on the number of unique active committers across all enabled repositories in your subscription. [Learn more](#)

This will add around **0** new unique committers.\*

\* Actual count may differ slightly due to how daily meter is calculated.

Warning: This action will override configurations on one or more repositories that already have Advanced Security enabled. It will update all repositories to use the default settings.

**Cancel** **Begin billing**

All Repositories Settings

Off **Default branch name**  
New repositories will be initialized with the default branch name. [Learn more](#)

On **Allow users to manage created branches**  
New repositories will be configured to allow users to manage permissions for their created branches.

Off **Create PRs as draft by default**  
New pull requests will be created as draft by default for all repositories in this project.

Project Settings

GHAzDODemo

General

Overview Teams Permissions Notifications Service hooks Dashboards

Boards

Project configuration Team configuration GitHub connections

Pipelines

Agent pools Parallel jobs Settings Test management Release retention Service connections XAML build services

Repos

Repositories

Artifacts

Storage

Create

4tecture-demo / GHAzDODemo / Settings / Repositories

Search

Project Settings  
GHAzDODemo

General

- Overview
- Teams
- Permissions
- Notifications
- Service hooks
- Dashboards

Boards

- Project configuration
- Team configuration
- GitHub connections

Pipelines

- Agent pools
- Parallel jobs
- Settings
- Test management
- Release retention
- Service connections
- XAML build services

Repos

Repositories

Artifacts

Storage

All Repositories

demo-devsecops

GHAzDODemo

Filter by keywords

Settings Policies Security Approvals and checks

On Advanced Security  
Advanced Security is billed based on the number of unique active committers across all enabled repositories in your subscription.  
[View billing](#) | [View alerts](#)

For setup tips, [view documentation](#) or [contact sales](#).

Block secrets on push  
Scan all pushes to the repository and block pushes containing secrets.

Repository Settings

On Forks  
Allow users to create forks from this repository.

Off Commit mention linking  
Automatically create links for work items mentioned in a commit comment.

Off Commit mention work item resolution  
Allow mentions in commit comments to close work items (e.g. "Fixes #123").

On Work item transition preferences  
Remember user preferences for completing work items with pull requests.

On Permissions management  
Allow users to manage permissions for the branches they created

Off Strict Vote Mode  
Enable Strict Vote Mode for repository which requires Contribute permission to vote in Pull Requests.

On Inherit PR creation mode  
When enabled, new pull request creation mode is inherited from project. When disabled, new pull request creation mode is set at the current level and may differ from project setting.

# Demo

Pipeline Tasks



G GHAzDODemo +

- Overview
- Boards
- Repos
- Pipelines
- Pipelines
- Environments
- Releases
- Library
- Task groups
- Deployment groups
- Test Plans
- Artifacts

Project settings <

## ← GHAzDO with AutoBuild

main ▾

demo-devsecops / .azure-pipelines/advanced-security.yml

```
1 trigger:
2   - main
3
4 schedules:
5   - cron: "0 0 * * *"
6     displayName: 'Nightly build'
7     branches:
8       - master
9       - always: true
10
11   pool:
12     vmImage: ubuntu-latest
13
14 steps:
15   - task: AdvancedSecurity-Codeql-Init@1
16     inputs:
17       languages: 'csharp'
18       querysuite: 'security-extended'
19   - task: AdvancedSecurity-Codeql-Autobuild@1
20   - task: AdvancedSecurity-Dependency-Scanning@1
21   - task: AdvancedSecurity-Codeql-Analyze@1
22   - task: AdvancedSecurity-Publish@1
23
24
```

Tasks

Search tasks

-  .NET Core  
Build, test, package, or publish a dotnet applicatio...
-  A wrapper for the AzureDevOps.WikiPDF...  
A wrapper for the [AzureDevOps.WikiPDFExport](...)...
-  Advanced Security AutoBuild  
Attempts to build the repository by finding and b...
-  Advanced Security Dependency Scanning  
Scan for open source dependency vulnerabilities i...
-  Advanced Security Initialize CodeQL  
Initializes the CodeQL database in preparation for...
-  Advanced Security Perform CodeQL analysis  
Finalizes the CodeQL database and runs the analy...
-  Advanced Security Publish Results  
Combines SARIF file(s) produced by code scannin...
-  Android signing  
Sign and align Android APK files
-  Ant  
Build with Apache Ant
-  App Center distribute  
Distribute app builds to testers and users via Visu...
-  App Center test  
Test app packages with Visual Studio App Center
-  Archive files  
Compress files into .7z, .tar.gz, or .zip

G GHazDODemo +

- Overview
- Boards
- Repos
- Pipelines
- Pipelines
- Environments
- Releases
- Library
- Task groups
- Deployment groups

Test Plans

Artifacts

Project settings

## ← HelloWorld CI

Variables Run

Show assistant

main ▾

demo-devsecops / .azure-pipelines/helloworld-ci.yml

```
1 # Trigger the pipeline on changes to the main branch
2 trigger:
3   - main
4
5 # Define the build environment
6 pool:
7   vmImage: 'ubuntu-latest'
8
9 # Define variables used in the pipeline
10 variables:
11   buildConfiguration: 'Release'
12   workingDirectory: '${Build.SourcesDirectory}/HelloWorld'
13   # DependencyScanning.Timeout: 600 # default is 300, anything under 300 has no effect
14   # DependencyScanning.Skip: true # break-glass scenario to skip dependency scanning from blocking the build
15
16 # Pipeline steps
17 steps:
18   # Initialize Advanced Security CodeQL
19   - task: AdvancedSecurity-Codeql-Init@1
20     inputs:
21       languages: 'csharp'
22       querysuite: 'security-extended'
23       displayName: 'Initialize CodeQL'
24
25   # Install .NET Core SDK
26   - task: UseDotNet@2
27     inputs:
28       packageType: 'sdk'
29       version: '8.x'
30       installationPath: $(Agent.ToolsDirectory)/dotnet
31       displayName: 'Install .NET Core SDK'
32
33   # Restore dependencies using dotnet command
34   - script: dotnet restore $(workingDirectory)/HelloWorld.sln
35     displayName: 'Restore dependencies'
36     workingDirectory: $(workingDirectory)
37
38   # Build the solution using dotnet command
39   - script: dotnet build $(workingDirectory)/HelloWorld.sln --configuration $(buildConfiguration) --no-restore
40     displayName: 'Build solution'
```



G GHazDODemo +

- Overview
- Boards
- Repos
- Pipelines
- Pipelines
- Environments
- Releases
- Library
- Task groups
- Deployment groups

Test Plans

Artifacts

Project settings

## ← HelloWorld CI

Variables Run

Show assistant

main ▾

demo-devsecops / .azure-pipelines/helloworld-ci.yml

```
38 # Build the solution using dotnet command
39 - script: dotnet build $(workingDirectory)/HelloWorld.sln --configuration $(buildConfiguration) --no-restore
40   displayName: 'Build solution'
41   workingDirectory: $(workingDirectory)
42
43 # Run unit tests using dotnet command
44 - script: dotnet test $(workingDirectory)/HelloWorld.sln --configuration $(buildConfiguration) --no-build --logger trx
45   displayName: 'Run unit tests'
46   workingDirectory: $(workingDirectory)
47
48 # Publish the web application to the artifact staging directory
49 - script: dotnet publish $(workingDirectory)/HelloWorld/HelloWorld.csproj --configuration $(buildConfiguration) --output $(Build.ArtifactStagingDirectory)
50   displayName: 'Publish web app'
51   workingDirectory: $(workingDirectory)
52
53 # Create an SBOM and add it to the pipeline artifact
54 - script: |
55   curl -Lo $(Agent.TempDirectory)/sbom-tool https://github.com/microsoft/sbom-tool/releases/latest/download/sbom-tool-linux-x64
56   chmod +x $(Agent.TempDirectory)/sbom-tool
57   mkdir -p $(Build.ArtifactStagingDirectory)/sbom
58   $(Agent.TempDirectory)/sbom-tool generate -b $(Build.ArtifactStagingDirectory)/sbom -bc $(Build.SourcesDirectory) -pn HelloWorld -pv $(Build.BuildNumber) -ps
59   displayName: Generate SBOM
60
61 Settings
62 - task: PublishPipelineArtifact@1
63   inputs:
64     targetPath: '$(Build.ArtifactStagingDirectory)/sbom'
65     artifact: 'sbom'
66     publishLocation: 'pipeline'
67     displayName: Publish SBOM artifact
68
69 # Publish the web application as a pipeline artifact
70 Settings
71 - task: PublishPipelineArtifact@1
72   inputs:
73     targetPath: '$(Build.ArtifactStagingDirectory)'
74     artifact: 'webapp'
75     publishLocation: 'pipeline'
76     displayName: 'Publish Pipeline Artifact'
77
78 # Run Advanced Security Dependency Scanning
```

G GHAzDODemo +

- Overview
- Boards
- Repos
- Pipelines
- Pipelines
- Environments
- Releases
- Library
- Task groups
- Deployment groups

Test Plans

Artifacts

Project settings

## ← HelloWorld CI

Variables Run

Show assistant

main ▾

demo-devsecops / .azure-pipelines/helloworld-ci.yml

```
75 # Run Advanced Security Dependency Scanning
    Settings
76 - task: AdvancedSecurity-Dependency-Scanning@1
    ...# inputs:
78 ...#   directoryExclusionList: 'sampledir/ignoreddirectory1;sampledir/ignoreddirectory'
79   displayName: 'Dependency Scanning'
80
81 # Run CodeQL Analysis
    Settings
82 - task: AdvancedSecurity-Codeql-Analyze@1
    ...displayName: 'CodeQL Analysis'
83
84 # Publish CodeQL Analysis Results
    Settings
85 - task: AdvancedSecurity-Publish@1
    ...displayName: 'Publish CodeQL Results'
86
87 # Add a CSV report for Advanced Security alerts
    Settings
88 - task: PowerShell@2
89   displayName: 'Generate a csv report based on Advanced Security alerts'
90   ...inputs:
91     ...targetType: filePath
92     ...filePath: $(Build.SourcesDirectory)/.azure-pipelines/ghazdo-csv-report.ps1
93     ...pwsh: true
94     ...env:
95       ...MAPPED_ADO_PAT: $(GHAZDO_PAT)
96
97 # Conditional step for Pull Requests: Run a PowerShell script for additional PR validation
98 - ${{ if eq(variables['Build.Reason'], 'PullRequest') }}:
    Settings
99 - task: PowerShell@2
100   displayName: 'PR Gating - Additional Checks for Pull Requests'
101   ...inputs:
102     ...targetType: filePath
103     ...filePath: $(Build.SourcesDirectory)/.azure-pipelines/CIGate.ps1
104     ...pwsh: true
105     ...env:
106       ...MAPPED_ADO_PAT: $(GHAZDO_PAT)
```



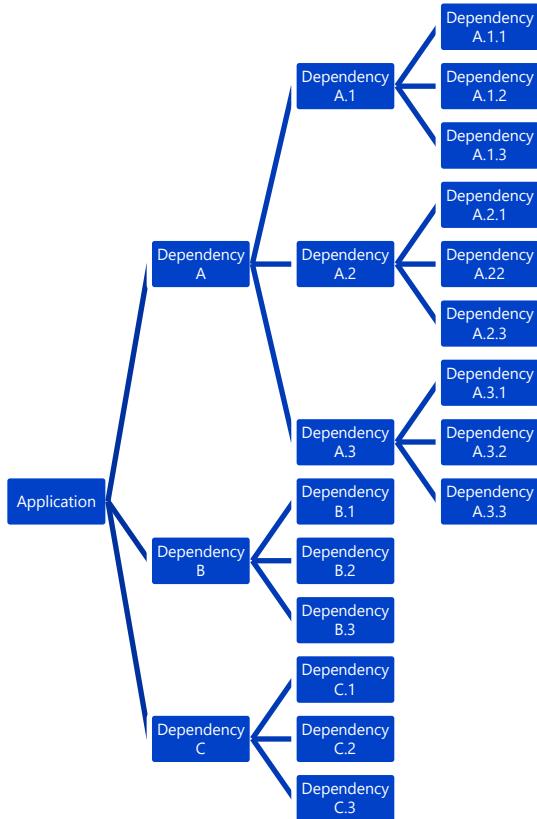
A close-up, slightly blurred photograph of several rowers in a boat, focusing on their oars and the water. The oars have yellow and black grips.

GitHub Advanced Security for Azure DevOps

# Dependency Scanning

4tecture®  
empower your software solutions

# Application Dependencies



# Current threats

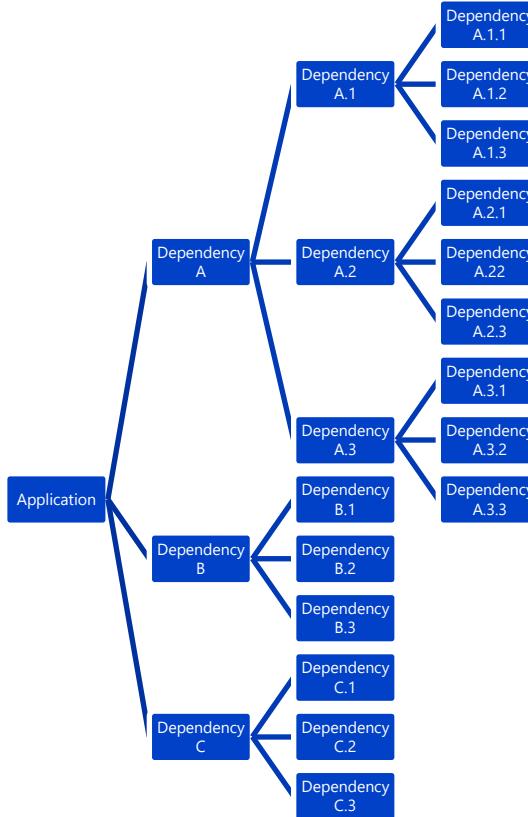
The image shows a composite of three web pages related to cybersecurity threats:

- CISA.gov:** An official website of the United States government. It features the CISA logo and navigation links for Alerts and Tips, Resources, CISA.gov, Services, and Report.
- Sonatype:** A news article titled "Popular npm Project Used by Millions Hijacked in Supply-Chain Attack" by Ax Sharma, dated October 25, 2021. The article discusses a supply-chain attack on the npm registry involving packages like klow, kloon, and oksha.
- BleepingComputer.com:** A news article titled "Popular npm Project Used by Millions Hijacked in Supply-Chain Attack" by Ax Sharma, dated October 25, 2021. This page also includes a sidebar with "TOPIC POSTS" and a "POPULAR STORIES" section.

Source: <https://www.cisa.gov/uscert/apache-log4j-vulnerability-guidance>, <https://blog.sonatype.com/npm-project-used-by-millions-hijacked-in-supply-chain-attack>, <https://www.bleepingcomputer.com/news/security/big-sabotage-famous-npm-package-deletes-files-to-protest-ukraine-war/>

# Dependency Graph

- Inventories your dependencies
- Scans for JavaScript, Java, .NET, PHP, Python and Ruby dependencies
- Connects all of GitHub.com as the world's largest code graph



# Supported package ecosystem

Package manager	Languages	Supported formats
Cargo	Rust	Cargo.toml, Cargo.lock
CocoaPods	Swift	Podfile.lock
Go modules	Go	go.mod, go.sum
Gradle	Java	*.lockfile
Maven	Java	pom.xml
npm	JavaScript	package-lock.json, package.json, npm-shrinkwrap.json, lerna.json
NuGet	C#	*.packages.config, *.project.assets (*.csproj)
pip	Python	setup.py, requirements.txt
pnpm	JavaScript	package.json
RubyGems	Ruby	Gemfile.lock
Yarn	JavaScript	package.json

# GitHub Advisory Database

- Powers GitHub's vulnerable dependency alerts
- Supports Composer, Maven, npm, NuGet, pip, RubyGems
- Includes advisories submitted directly to GitHub by maintainers who are requesting a CVE
- Aggregates additional data from NVD, Community sources, WhiteSource
- Curated by a dedicated team

The screenshot shows the GitHub Advisory Database interface. At the top, there is a navigation bar with links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the navigation bar is a search bar with placeholder text 'Search or jump to...'. The main content area is titled 'GitHub Advisory Database' and describes it as a 'Security vulnerability database inclusive of CVFs and GitHub originated security advisories from the world of open source software'. A search bar at the top of this section contains the query 'type:reviewed ecosystem:nugget'. Below the search bar, there are two sections: 'GitHub reviewed advisories' and 'Unreviewed advisories'. The 'GitHub reviewed advisories' section has a table with columns for 'Ecosystem' and 'Count'. The 'NuGet' row is highlighted with a blue background. The 'Unreviewed advisories' section shows a count of 170,890. Below these sections is a table of specific security advisories, each with a title, severity, and a link to the details. The first few rows include:

- Potential leak of NuGet.org API key (High) - CVE-2022-38184 was published for NuGetCommunity (NuGet) 2 days ago.
- Cross site scripting in SCCMS (Moderate) - CVE-2021-38348 was published for SCCMS (NuGet) 14 days ago.
- Weak private key generation in SSH.NET (Moderate) - CVE-2022-38460 was published for SSH.NET (NuGet) 15 days ago.
- Cross-site Scripting in ZKEACMS (Moderate) - CVE-2022-28962 was published for ZKEACMS-Publisher (NuGet) 22 days ago.
- Cross site scripting in SiteServer CMS (Moderate) - CVE-2021-42826 was published for SiteServer (NuGet) 23 days ago.
- SQL Injection in SiteServer CMS (High) - CVE-2021-43553 was published for SiteServer (NuGet) 23 days ago.
- Improper Input Validation in IgmaMatcher (Moderate) - CVE-2021-31818 was published for IgmaMatcher (NuGet) on May 17.
- Code Injection in Massif.Tools.Core (High) - CVE-2022-21976 was published for Massif.Tools.Core (NuGet) on May 3.
- Exposure of Sensitive Information to an Unauthorized Actor in DisCatSharp (Moderate) - CVE-2022-28488 was published for DisCatSharp (NuGet) on Apr 22.
- YARP Denial of Service Vulnerability (High) - CVE-2022-28604 was published for YarpDotNet (NuGet) on Apr 22.
- Improper Certificate Validation (High) - CVE-2021-37770 was published for Microsoft.NETCore.App (NuGet) on Apr 12.
- Infinite Loops in .Net React (High)

# Demo

Dependency Scanning



G GHazDODemo +

- Overview
- Boards
- Repos
- Files
- Commits
- Pushes
- Branches
- Tags
- Pull requests
- Advanced Security
- Pipelines
- Test Plans
- Artifacts

Project settings ≪

## Advanced Security

Dependencies Code scanning Secrets

Filter by keywords

Branch: main ▾ State: Open ▾ Pipeline ▾ Package ▾ Severity ▾ X

Alert

Vulnerable package

First detected

Azure Identity SDK Remote Code Execution Vulnerability (CVE-2023-36414) High

#5 in HelloWorld/HelloWorld/HelloWorld.csproj

azure.identity (1.7.0) (1 root dep.)

NuGet

Dec 2, 2023

.NET Information Disclosure Vulnerability (CVE-2022-41064) Medium

#1 in HelloWorld/HelloWorld/HelloWorld.csproj

system.data.sqlclient (4.8.3)

NuGet

Sep 18, 2023

G GHazDODemo +

- Overview
- Boards
- Repos
- Files
- Commits
- Pushes
- Branches
- Tags
- Pull requests
- Advanced Security
- Pipelines
- Test Plans
- Artifacts

## ← Azure Identity SDK Remote Code Execution Vulnerability (CVE-2023-36414)

#5 [Open](#) in [main](#) • detected [Fri at 2:10 PM](#)

Overview Detections

### Recommendation

Upgrade Azure.Identity from 1.7.0 to 1.10.2 to fix the vulnerability.

### Location

[HelloWorld/HelloWorld>HelloWorld.csproj](#) @ 28f03d41

### Description

Azure Identity SDK is vulnerable to remote code execution.

### Resources

- [See advisory for vulnerability details](#)
- [Learn how to assess and remediate vulnerabilities](#)

Severity

High

First detected

Dec 2, 2023 at 7:54 AM

Finding details

Vulnerable package  
azure.identity (1.7.0)

Root dependency  
microsoft.entityframeworkcore.sqlserver (8.0.0)

**G** GHAzDODemo +

Overview

Boards

Repos

Files

Commits

Pushes

Branches

Tags

Pull requests

Advanced Security

Pipelines

Test Plans

Artifacts

## ← Azure Identity SDK Remote Code Execution Vulnerability (CVE-2023-36414)

Close alert ▾

#5 Open in main • detected Fri at 2:10 PM

Overview Detections

Filter by keyword

Last scanned	Pipeline	First detected
<span style="color: red;">✖</span> Friday Detected	HelloWorld CI (Job)	<u>Dec 2, 2023</u>
<span style="color: red;">✖</span> Friday Detected	GHAzDO with AutoBuild (Job)	<u>Dec 2, 2023</u>

G GHAzDODemo +

- Overview
- Boards
- Repos
- Files
- Commits
- Pushes
- Branches
- Tags
- Pull requests
- Advanced Security
- Pipelines
- Test Plans
- Artifacts

## ← Azure Identity SDK Remote Code Execution Vulnerability (CVE-2023-36414)

#5 [Open](#) in [main](#) • detected [Fri at 2:10 PM](#)[Overview](#) [Detections](#)

### Recommendation

Upgrade Azure.Identity from 1.7.0 to 1.10.2 to fix the vulnerability.

### Location

[HelloWorld/HelloWorld>HelloWorld.csproj](#) @ 28f03d41

### Description

Azure Identity SDK is vulnerable to remote code execution.

### Resources

- [See advisory for vulnerability details](#)
- [Learn how to assess and remediate vulnerabilities](#)

### Reason

 Risk accepted

Risk is tolerable or irrelevant (e.g., only used in test or not exploitable in your implementation).

 False positive

This alert is inaccurate or incorrect.

### Comment (optional)

[Cancel](#)[Close](#)

A close-up, slightly blurred photograph of a team of rowers in a boat. The rowers are wearing blue and red athletic gear. Their yellow oars are visible, dipping into the water. The background is a bright, overexposed sky.

GitHub Advanced Security for Azure DevOps

# Code Scanning

4tecture®  
empower your software solutions

# CodeQL

## A revolutionary semantic code engine

- Advanced code analysis engine based on 13 years of research by a 30 person team from Oxford University
- Query your code's logic to find vulnerabilities
- Queries can be quickly customized to adapt to your specific threat topology
- Community-driven query set

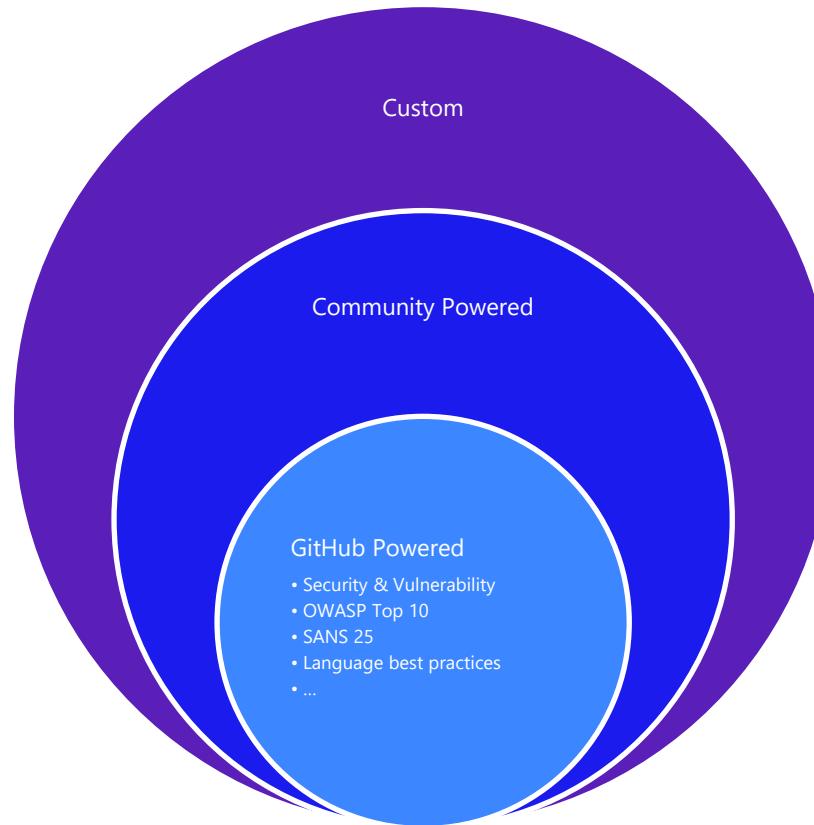
# Supported Languages

- C/C++
- C#
- Go
- Java
- JavaScript/TypeScript
- Kotlin (beta)
- Python
- Ruby
- Swift

# CodeQL Queries

- GitHub experts, security researchers, and community contributors write and maintain the default CodeQL queries used for code scanning.
- Queries: <https://github.com/github/codeql>

# Shared security expertise



# Community Queries Example

The screenshot shows a Microsoft Security blog post. At the top, there's a navigation bar with links for Microsoft, Microsoft Security, Solutions, Products, Services, Partners, Resources, Contact sales, and a 'Start free trial' button. To the right are links for All Microsoft and a search bar. The main content area has a header 'February 25, 2021 • 6 min read' and a title 'Microsoft open sources CodeQL queries used to hunt for Solorigate activity'. Below the title is the author 'Microsoft Security Team'. A 'Share' button is visible. A gray box contains an 'UPDATE' section: 'Microsoft continues to work with partners and customers to expand our knowledge of the threat actor behind the nation-state cyberattacks that compromised the supply chain of SolarWinds and impacted multiple other organizations. Microsoft previously used 'Solorigate' as the primary designation for the actor, but moving forward, we want to place appropriate focus on the actors behind the sophisticated attacks, rather than one of the examples of malware used by the actors. Microsoft Threat Intelligence Center (MSTIC) has named the actor behind the attack against SolarWinds, the SUNBURST backdoor, TEARDROP malware, and related components as NOBELIUM. As we release new content and analysis, we will use NOBELIUM to refer to the actor and the campaign of attacks.' At the bottom, there's a paragraph about the SolarWinds attack.

February 25, 2021 • 6 min read

## Microsoft open sources CodeQL queries used to hunt for Solorigate activity

Microsoft Security Team

Share

**UPDATE:** Microsoft continues to work with partners and customers to expand our knowledge of the threat actor behind the nation-state cyberattacks that compromised the supply chain of SolarWinds and impacted multiple other organizations. Microsoft previously used 'Solorigate' as the primary designation for the actor, but moving forward, we want to place appropriate focus on the actors behind the sophisticated attacks, rather than one of the examples of malware used by the actors. Microsoft Threat Intelligence Center (MSTIC) has named the actor behind the attack against SolarWinds, the SUNBURST backdoor, TEARDROP malware, and related components as NOBELIUM. As we release new content and analysis, we will use NOBELIUM to refer to the actor and the campaign of attacks.

A key aspect of the Solorigate attack is the supply chain compromise that allowed the attacker to modify binaries in SolarWinds' Orion product. These modified binaries were distributed via previously legitimate update channels and allowed the attacker to remotely perform malicious activities, such as credential theft, privilege escalation, and lateral movement, to steal sensitive information. The incident has reminded organizations to reflect not just on their readiness to respond to sophisticated attacks, but also the resilience of their own codebases.

# Custom Query

```
name: "Run custom queries"

# When using a configuration file, if you do not disable default queries,
# then the default CodeQL queries in the `code-scanning` query suite will also execute upon analysis.
disable-default-queries: true

# To reference local queries saved to your repository,
# the path must start with `./` followed by the path to the custom query or queries.
# Names for each query referenced is optional.
queries:
  - name: Use security-extended query suite
    uses: security-extended
  - name: Use local custom query (single query)
    uses: ./customQueries/javascript/FindTestFunctions.ql
  - name: Use local custom query (directory of queries)
    uses: ./customQueries/javascript/MemoryLeakQueries

packs:
  - mygithubborg/mypackname

paths:
  - src

paths-ignore:
  - src/node_modules
  - '**/*.test.js'

query-filters:
  - include:
      kind: problem
  - include:
      precision: medium
  - exclude:
      id:
        - js/angular/disabling-sce
        - js/angular/insecure-url-allowlist
```

```
trigger: none

pool:
  vmImage: windows-latest

# You can either specify your CodeQL variables in a variable block...
variables:
# `configfilepath` must be an absolute file path relative to the repository root
advancedsecurity.codeql.configfilepath: '${build.sourcesDirectory}/.pipelines/steps/configfile.yml'

# Or you can specify variables as variables for the task. You do not need both definitions.
steps:
  - task: AdvancedSecurity-Codeql-Init@1
    displayName: Initialize CodeQL
    inputs:
      languages: 'javascript'
      loglevel: '2'
      configfilepath: '${build.sourcesDirectory}/.pipelines/steps/configfile.yml'
    # If downloading a pack from GitHub,
    # you must include a GitHub access token with the scope of `read:packages`.
    env:
      GITHUB_TOKEN: $(githubtoken)

  - task: AdvancedSecurity-Codeql-Autobuild@1
    displayName: AutoBuild

  - task: AdvancedSecurity-Codeql-Analyze@1
    displayName: Perform CodeQL Analysis
```

# Demo

Code Scanning



G GHazDODemo +

- Overview
- Boards
- Repos
- Files
- Commits
- Pushes
- Branches
- Tags
- Pull requests
- Advanced Security
- Pipelines
- Test Plans
- Artifacts

Project settings <<

## Advanced Security

Dependencies **Code scanning** Secrets

Filter by keywords

Branch: main State: Open Pipeline Tool Rule Severity X

Alert

First detected

Weak encryption (cs/weak-encryption) **High**

#4 in HelloWorld/HelloWorld/Services/BadEncryptionService.cs:11 (+1)

Sep 18, 2023

SQL query built from user-controlled sources (cs/sql-injection) **High**

#3 in HelloWorld/HelloWorld/Controllers/SqlInjectionController.cs:34 (+1)

Sep 18, 2023

Azure DevOps 4tecture-demo / GHazDODemo / Repos / Advanced Security / demo-devsecops

Search

G GHazDODemo +

Overview Boards Repos Files Commits Pushes Branches Tags Pull requests Advanced Security Pipelines Test Plans Artifacts

← SQL query built from user-controlled sources (cs/sql-injection)

#3 Open in main • detected Fri at 2:12 PM

Overview Detections

Locations

>HelloWorld/HelloWorld/Controllers/SqlInjectionController.cs:34 @ 28f03d41

```
27     Ienumerable<Person> persons;
28
29     try
30     {
31         await conn.OpenAsync();
32         persons = await conn.QueryAsync<Person>(query);
33     }
34     finally
35     {
36         conn.Close();
37     }
38 }
```

> HelloWorld/HelloWorld/Controllers/SqlInjectionController.cs:34 @ 28f03d41

Severity High

First detected Sep 18, 2023 at 5:45 PM

Finding details

Tool CodeQL

Rule ID cs/sql-injection

Weakness CWE-089

## Description

If a SQL query is built using string concatenation, and the components of the concatenation include user input, a user is likely to be able to run malicious database queries.

## Recommendation

Usually, it is better to use a prepared statement than to build a complete query with string concatenation. A prepared statement can include a parameter, written as either a question mark (?) or with an explicit name (@parameter), for each part of the SQL query that is expected to be filled in by a different value each time it is run. When the query is later executed, a value must be supplied for each parameter in the query.

It is good practice to use prepared statements for supplying parameters to a query, whether or not any of the parameters are directly traceable to user input. Doing so avoids any need to worry about quoting and escaping.

## Example

In the following example, the code runs a simple SQL query in three different ways.

The first way involves building a query, `query1`, by concatenating a user-supplied text box value with some string literals. The text box value can include special characters, so this code allows for SQL injection attacks.

The second way uses a stored procedure, `ItemsStoredProcedure`, with a single parameter (`@category`). The parameter is then given a value by calling `Parameters.Add`. This version is immune to injection attacks, because any special characters are not given any special treatment.

Project settings <

Azure DevOps 4tecture-demo / GHazDODemo / Repos / Pull requests / demo-devsecops

Search

G GHazDODemo +

Overview Boards Repos Files Commits Pushes Branches Tags Pull requests Advanced Security Pipelines Test Plans Artifacts Project settings

## Extend SQL Injection Controller

Active I284 MM Marc Müller proposes to merge feature/extendSqlInjectionController into main 0/1 comments resolved

Overview Files Updates Commits Conflicts

All Changes Filter SqlInjectionController.cs +20 /HelloWorld/HelloWorld/Controllers/SqlInjectionController.cs

Side-by-side

Marc Mül... Just now Active

SQL query built from user-controlled sources (cs/sql-injection)

Building a SQL query from user-controlled sources is vulnerable to insertion of malicious SQL code by the user. See details [here](#)

Write a reply... Resolve

```
1 using System.Collections.Generic;
2 using System.Threading.Tasks;
3 using Microsoft.AspNetCore.Mvc;
4 using HelloWorld.Database;
5 using Microsoft.EntityFrameworkCore;
6 using Dapper;
7
8 namespace HelloWorld.Controllers
9 {
10     public class SqlInjectionController : Controller
11     {
12         private DemoContext _context;
13
14         public SqlInjectionController(DemoContext context)
15         {
16             _context = context;
17         }
18
19         public IActionResult Index()
20         {
21             return View();
22         }
23
24         [HttpGet("SqlInjection/SearchPersonUnsecure/{name}")]
25         public async Task SearchPersonUnsecure(string name)
26         {
27             var conn = _context.Database.GetDbConnection();
```

```
1 using System.Collections.Generic;
2 using System.Threading.Tasks;
3 using Microsoft.AspNetCore.Mvc;
4 using HelloWorld.Database;
5 using Microsoft.EntityFrameworkCore;
6 using Dapper;
7
8 namespace HelloWorld.Controllers
9 {
10     public class SqlInjectionController : Controller
11     {
12         private DemoContext _context;
13
14         public SqlInjectionController(DemoContext context)
15         {
16             _context = context;
17         }
18
19         public IActionResult Index()
20         {
21             return View();
22         }
23
24         [HttpGet("SqlInjection/SearchPersonUnsecure/{name}")]
25         public async Task SearchPersonUnsecure(string name)
26         {
27             var conn = _context.Database.GetDbConnection();
```

G GHazDODemo +

Overview Boards Repos Files Commits Pushes Branches Tags Pull requests Advanced Security Pipelines Test Plans Artifacts

Project settings <>

## ← SQL query built from user-controlled sources (cs/sql-injection)

#10 Open in [refs/pull/284/merge](#) • detected Just now[Close alert](#)[Overview](#) Detections

### Location

>HelloWorld/HelloWorld/Controllers/SqlInjectionController.cs:34 @ b135cc58

```
29    string query = "SELECT * FROM Person WHERE PersonID = " + personId;
30
31    try
32    {
33        await conn.OpenAsync();
34        persons = await conn.QueryAsync<Person>(query);
35    }
36
37    finally
38    {
39        conn.Close();
40    }
41}
```

### Severity

[High](#)

### First detected

Just now

### Finding details

Tool  
CodeQLRule ID  
[cs/sql-injection](#)Weakness  
[CWE-089](#)

### Description

If a SQL query is built using string concatenation, and the components of the concatenation include user input, a user is likely to be able to run malicious database queries.

### Recommendation

Usually, it is better to use a prepared statement than to build a complete query with string concatenation. A prepared statement can include a parameter, written as either a question mark (?) or with an explicit name (@parameter), for each part of the SQL query that is expected to be filled in by a different value each time it is run. When the query is later executed, a value must be supplied for each parameter in the query.

It is good practice to use prepared statements for supplying parameters to a query, whether or not any of the parameters are directly traceable to user input. Doing so avoids any need to worry about quoting and escaping.

### Example

In the following example, the code runs a simple SQL query in three different ways.

The first way involves building a query, `query1`, by concatenating a user-supplied text box value with some string literals. The text box value can include special characters, so this code allows for SQL injection attacks.

The second way uses a stored procedure, `ItemsStoredProcedure`, with a single parameter (`@category`). The parameter is then given a value by calling `Parameters.Add`. This version is immune to injection attacks, because any special characters are not given any special treatment.

The third way builds a query, `query2`, with a single string literal that includes a parameter (`@category`). The parameter is then given a value by calling `Parameters.Add`. This version is immune to injection attacks, because any special characters are not given any special treatment.

# Demo

Custom Queries



- G GHazDODemo +
- Overview
- Boards
- Repos
- Files Selected
- Commits
- Pushes
- Branches
- Tags
- Pull requests
- Advanced Security
- Pipelines
- Test Plans
- Artifacts

- demo-devsecops
- .azure-pipelines
- .customQueries
- authorizedcontrollers.sql
- codeql-pack.lock.yml
- codeql-pack.yml
- .devcontainer
- .github
- .vscode
- HelloWorld
- Infrastructure
- .gitignore
- LICENSE
- microsoftsecuritydevops.yml
- README.md

## authorizedcontrollers.sql

[Contents](#) [History](#) [Compare](#) [Blame](#)

```
1 /**
2  * @name Secure API controllers without Authorize attribute
3  * @description Secure controllers within an 'api-secure' route must have an Authorize attribute.
4  * @kind problem
5  * @problem.severity warning
6  * @id demo/api-controller-without-authorize
7  * @tags security
8  *   api
9 */
10
11 import csharp
12 import semmle.code.csharp.Attribute
13
14 from Class controllerClass, Attribute routeAttribute
15 where controllerClass = routeAttribute.getTarget()
16     and routeAttribute.getType().hasName("RouteAttribute")
17     and routeAttribute.getArgument(0).getValue().matches("%api-secure%")
18     and not exists(Attribute authorizeAttribute |
19         authorizeAttribute.getTarget() = controllerClass and
20         authorizeAttribute.getType().hasName("AuthorizeAttribute"))
21
22 select controllerClass, "This controller serving API a secret route does not have the 'Authorize' attribute."
23
```

[Edit](#)

Azure DevOps 4ecture-demo / GHazDODemo / Repos / Files / demo-devsecops

Search

GHazDODemo + demo-devsecops

Overview Boards Repos Files Commits Pushes Branches Tags Pull requests Advanced Security Pipelines Test Plans Artifacts

feature/customquery .azure-pipelines / codeqlconfiguration.yml

codeqlconfiguration.yml

Commit Revert

Contents Highlight changes

```
1 name: "Run custom queries"
2
3 disable-default-queries: false
4
5 queries:
6   - name: Authorized controllers
7     uses: ./customQueries/authorizedcontrollers.ql
8
9 paths:
10  - HelloWorld
11
```

advanced-security.yml  
CIGate.ps1  
codeqlconfiguration.yml  
ghazdo-csv-report.ps1  
helloworld-ci.yml  
.customQueries  
authorizedcontrollers.ql  
codeql-pack.lock.yml  
codeql-pack.yml  
.devcontainer  
.github  
.vscode  
HelloWorld  
Infrastructure  
.gitignore  
LICENSE  
microsoftsecuritydevops.yml  
README.md

Project settings

Azure DevOps 4ecture-demo / GHazDODemo / Repos / Files / demo-devsecops

Search

G GHazDODemo + demo-devsecops : feature/customquery / .azure-pipelines / helloworld-ci.yml

Overview Boards Repos Files Commits Pushes Branches Tags Pull requests Advanced Security Pipelines Test Plans Artifacts

Contents History Compare Blame

```
1 # Trigger the pipeline on changes to the main branch
2 trigger:
3 - main
4
5 # Define the build environment
6 pool:
7 vmImage: 'ubuntu-latest'
8
9 # Define variables used in the pipeline
10 variables:
11 buildConfiguration: 'Release'
12 workingDirectory: '${Build.SourcesDirectory}/HelloWorld'
13 # DependencyScanning.Timeout: 600 # default is 300, anything under 300 has no effect
14 # DependencyScanning.Skip: true # break-glass scenario to skip dependency scanning from blocking the build
15
16 # Pipeline steps
17 steps:
18 # Initialize Advanced Security CodeQL
19 - task: AdvancedSecurity-Codeql-Init@1
20 inputs:
21 languages: 'csharp'
22 queriesuite: 'security-extended'
23 configfilepath: '${build.sourcesDirectory}/.azure-pipelines/codeqlconfiguration.yml'
24 displayName: 'Initialize CodeQL'
25
26 # Install .NET Core SDK
27 - task: UseDotNet@2
28 inputs:
29 packageType: 'sdk'
30 version: '8.x'
31 installationPath: ${Agent.ToolsDirectory}/dotnet
32 displayName: 'Install .NET Core SDK'
33
34 # Restore dependencies using dotnet command
35 - script: dotnet restore ${workingDirectory}/HelloWorld.sln
36 displayName: 'Restore dependencies'
37 workingDirectory: ${workingDirectory}
38
39 # Build the solution using dotnet command
40 - script: dotnet build ${workingDirectory}/HelloWorld.sln --configuration ${buildConfiguration} --no-restore
41 displayName: 'Build solution'
42 workingDirectory: ${workingDirectory}
43
44 # Run unit tests using dotnet command
45 - script: dotnet test ${workingDirectory}/HelloWorld.sln --configuration ${buildConfiguration} --no-build --logger trx
46 displayName: 'Run unit tests'
47 workingDirectory: ${workingDirectory}
```

Project settings <

Azure DevOps 4tecture-demo / GHazDODemo / Repos / Pull requests / demo-devsecops

Search

G GHazDODemo +

Add custom CodeQL query

Active I283 MM Marc Müller proposes to merge feature/customquery into main 2/3 comments resolved

Overview Files Updates Commits Conflicts

All Changes Filter AuthorizedAccessControllerA1.cs /HelloWorld/HelloWorld/Controllers/AuthorizedAccessControllerA1.cs

demo-devsecops .azure-pipelines .customQueries HelloWorld/HelloWorld/Controllers

Marc Müller 6m ago

Secure API controllers without Authorize attribute (demo/api-controller-without-authorize)  
Secure controllers within an 'api-secure' route must have an Authorize attribute.  
See details [here](#)

Write a reply... Resolve

```
1 using Microsoft.AspNetCore.Http;
2 using Microsoft.AspNetCore.Mvc;
3
4 namespace HelloWorld.Controllers
5 {
6     [Route("api-secure/[controller]")]
7     [ApiController]
8     public class AuthorizedAccessControllerA1 : ControllerBase
9     {
10         [HttpGet("GetSecretInfo")]
11         public IActionResult GetSecret()
12         {
13             return Ok("Something secret");
14         }
15     }
16 }
17
18 }
```

.gitignore

Approve Set auto-complete

Project settings

A photograph showing a team of rowers in a boat, blurred to suggest motion. The rowers are wearing blue and red athletic gear. The water is visible at the bottom.

GitHub Advanced Security for Azure DevOps

# Secret Scanning

4tecture®  
empower your software solutions

# Secret Scanning

- Scanning for secrets in code
- Push protection for secrets

# Handling blocked push

- Remove the secret from the file
- Remove the secret from commit history
  - In previous commit: `git commit --amend`
  - Further back in history:
    - Change `pick` to `edit`
    - Remove the secret from your code
    - Commit change with `git commit --amend`
    - Finish the rebase with `git rebase --continue`
- Push a blocked secret
  - Add `skip-secret-scanning:true` in your commit message

# Supported Secrets

Adafruit IO - Adafruit IO Key

Alibaba Cloud - Alibaba Cloud Access Key ID with Alibaba Cloud Access Key Secret

Amazon - Amazon OAuth Client ID with Amazon OAuth Client Secret

Amazon Web Services (AWS) - Amazon AWS Access Key ID with Amazon AWS Secret Access Key

Amazon Web Services (AWS) - Amazon AWS Session Token with Amazon AWS Temporary Access Key ID and Amazon AWS Secret Access Key

Asana - Asana Personal Access Token

Atlassian - Bitbucket Server Personal Access Token

Chief Tools - Chief Tools Token

Clojars - Clojars Deploy Token

Databricks - Databricks Access Token

DevCycle - DevCycle Client API Key

DevCycle - DevCycle Mobile API Key

DevCycle - DevCycle Server API Key

DigitalOcean - DigitalOcean OAuth Token

DigitalOcean - DigitalOcean Personal Access Token

DigitalOcean - DigitalOcean Refresh Token

DigitalOcean - DigitalOcean System Token

Discord - Discord Bot Token

Doppler - Doppler Audit Token

Doppler - Doppler CLI Token

Doppler - Doppler Personal Token

Doppler - Doppler SCIM Token

Doppler - Doppler Service Token

Dropbox - Dropbox Short Lived Access Token

Duffel - Duffel Live Access Token

EasyPost - EasyPost Production API Key

Figma - Figma Personal Access Token

Flutterwave - Flutterwave Live API Secret Key

FullStory - FullStory API Key

GitHub - GitHub Personal Access Token

GitHub - GitHub App Installation Access Token

GitHub - GitHub OAuth Access Token

GitHub - GitHub Refresh Token

Google - Google OAuth Client ID with Google OAuth Client Secret

Google - Google Cloud Storage Service Account Access Key ID with Google Cloud Storage Access Key Secret

Google - Google Cloud Storage User Access Key ID with Google Cloud Storage Access Key Secret

Grafana - Grafana API Key

HashiCorp - HashiCorp Vault Batch Token

HashiCorp - HashiCorp Vault Root Service Token

HashiCorp - HashiCorp Vault Service Token

Highnote - Highnote RK Live Key

Highnote - Highnote RK Test Key

Highnote - Highnote SK Live Key

Highnote - Highnote SK Test Key

Hubspot - Hubspot API Key

Hubspot - Hubspot API Personal Access Key

Intercom - Intercom Access Token

Ionic - Ionic Personal Access Token

Ionic - Ionic Refresh Token

JFrog - JFrog Platform API Key

JFrog - JFrog Platform Access Token

Linear - Linear API Key

Linear - Linear OAuth Access Token

LogicMonitor - LogicMonitor Bearer Token

LogicMonitor - LogicMonitor LMV1 Access Key

Microsoft - Azure DevOps Personal Access Token

Microsoft - Office/Teams Inbound Webhook

Microsoft - Azure Storage Key Identifiable

Microsoft - Azure Cache for Redis Access Key

Microsoft - Microsoft Entra Application Secret

Microsoft - Azure Functions V4+ KEY

Microsoft - Azure Cosmos DB Key Identifiable

Microsoft - Azure Batch Key Identifiable

Microsoft - Azure Search Query Key

Microsoft - Azure Search Admin Key

Microsoft - Azure Machine Learning studio (classic) web service key

Midtrans - Midtrans Production Server Key

New Relic - New Relic Insights Query Key

New Relic - New Relic Personal API Key

New Relic - New Relic REST API Key

npm - npm Access Token

NuGet - NuGet API Key

Onfido - Onfido Live API Token

OpenAI - OpenAI API Key

PlanetScale - PlanetScale Database Password

PlanetScale - PlanetScale OAuth Token

PlanetScale - PlanetScale Service Token

Postman - Postman API Key

Prefect - Prefect Server API Key

Prefect - Prefect User API Key

Proctorio - Proctorio Secret Key

ReadMe - ReadMe API Access Key

redirect.pizza - redirect.pizza API Token

Samsara - Samsara API Token

Samsara - Samsara OAuth Access Token

SendGrid - SendGrid API Key

Sendinblue - Sendinblue API Key

Sendinblue - Sendinblue SMTP Key

Shippo - Shippo Live API Token

Shopify - Shopify Access Token

Shopify - Shopify App Shared Secret

Slack - Slack API Token

Stripe - Stripe API Key

Tencent Cloud - Tencent Cloud Secret ID

Typeform - Typeform Personal Access Token

Uniwise - WISEflow API Key

WakaTime - WakaTime App Secret

WakaTime - WakaTime OAuth Access Token

WakaTime - WakaTime OAuth Refresh Token

WorkOS - WorkOS Production API Key

Zuplo - Zuplo Consumer API Key

# Demo

Secret Scanning



G GHAzDODemo +

- Overview
- Boards
- Repos
- Files
- Commits
- Pushes
- Branches
- Tags
- Pull requests
- Advanced Security
- Pipelines
- Test Plans
- Artifacts

Project settings 

## Advanced Security

Dependencies Code scanning Secrets

 Filter by keywords

State: Open Type X

Alert

Introduced

Azure DevOps personal access token (PAT) ...ztqfvva Critical

#6 in README.md:158 (+1)

Dec 2, 2023

## G GHazDODemo +

Overview

Boards

Repos

Files

Commits

Pushes

Branches

Tags

Pull requests

Advanced Security

Pipelines

Test Plans

Artifacts

## ← Azure DevOps personal access token (PAT)

#6 Open in 810ec583 • detected Today at 2:16 PM

Close alert ▾

## Locations

READMD.md:158 @ a9d892a5

```
154     [ApiController]
155     public class ThirdPartyAccessController : ControllerBase
156     {
157
158         private string api_key = "gqvusw156hy2aztqfva"; //This is a fake API key for demonstration purposes
159
160         //Simulated sensitive data (AWS credentials)
161         private string aws_access_key_id = "AK>LE";
162         private string aws_secret_access_key = "wJalrXutriPLEKEY";
```

&gt; HelloWorld/HelloWorld/Controllers/ThirdPartyAccessController.cs:11 @ f1975ac4

## Severity

Critical

## First detected

Dec 2, 2023 at 8:49 AM

## Finding details

Type  
AdoPatID  
SEC101/102

## Recommendation

Review evidence of possible plaintext (or base64-encoded plaintext) secrets in versioned engineering content.

## Remediation steps

Follow the steps below before you close this alert:

1. Rotate the secret and store securely if it's in use to prevent breaking workflows.
2. Revoke this Azure DevOps personal access token (PAT) to prevent unauthorized access.
3. Check security logs for potential breaches.
4. Close the alert as revoked.

A close-up, slightly blurred photograph of a team of rowers in a boat. The rowers are wearing blue and red athletic gear. Their yellow and black oars are visible, dipping into the water. The background is a bright, overexposed sky.

GitHub Advanced Security for Azure DevOps

# Azure Pipelines

4tecture®  
empower your software solutions

# GHAzDo Resources

<https://github.com/microsoft/GHAzDO-Resources>

The screenshot shows the GitHub repository page for 'microsoft/GHAzDO-Resources'. The repository is public and contains 10 branches and 0 tags. The main branch has 22 commits. The repository includes files like .github/workflows, src, .gitignore, .markdownlint.jsonc, CODE\_OF\_CONDUCT.md, LICENSE, README.md, SECURITY.md, and SUPPORT.md. The README file contains a section titled 'GitHub Advanced Security on Azure DevOps Resources'.

**About**

Resources, Scripts, etc. for GitHub Advanced Security on Azure DevOps

**Code**

main · 10 Branches · 0 Tags · Go to file · Code

**Commits**

tjcorr and felickz Fix PR annotations when CodeQL finds error on a single line (#... 66fcfa8 · 5 days ago 22 Commits

.github/workflows Initial Setup for Sample Repo (#1) 9 months ago

src Fix PR annotations when CodeQL finds error on a single line ... 5 days ago

.gitignore add .gitignore to avoid pushing awkward files 9 months ago

.markdownlint.jsonc Initial Setup for Sample Repo (#1) 9 months ago

CODE\_OF\_CONDUCT.md CODE\_OF\_CONDUCT.md committed 10 months ago

LICENSE Initial Setup for Sample Repo (#1) 9 months ago

README.md Initial Setup for Sample Repo (#1) 9 months ago

SECURITY.md Initial Setup for Sample Repo (#1) 9 months ago

SUPPORT.md Initial Setup for Sample Repo (#1) 9 months ago

**Activity**

Custom properties

16 stars

3 watching

8 forks

Report repository

**Releases**

No releases published

**Packages**

No packages published

**Contributors** 6

**Languages**

JavaScript

# Demo

PR Gating



# Demo

SBOM



# Demo

GHAzDO Exports



A close-up, slightly blurred photograph of several rowers in a boat, focusing on their oars and the water. The oars have yellow and black grips.

GitHub Advanced Security for Azure DevOps

# Defender for Cloud

4tecture®  
empower your software solutions

# Demo

Defender for Cloud



A close-up, slightly blurred photograph of several rowers in a boat, focusing on their legs and the oars. The oars have yellow handles and black blades. The water is visible at the bottom.

GitHub Advanced Security for Azure DevOps

# Q & A

4tecture®  
empower your software solutions

# Recap

- Out-of-the-box security features for SCA, SAST, and Secret Scanning
- Built on top of GitHub Advisories database and CodeQL queries
- CodeQL queries can be extended with custom queries
- Easy to setup (enable and extend pipelines), but alerts need to be managed

# Thank you for your attention!

If you have any questions do not hesitate to contact us:

4tecture GmbH  
Industriestrasse 25  
CH-8604 Volketswil

+41 44 508 37 00  
[info@4tecture.ch](mailto:info@4tecture.ch)  
[www.4tecture.ch](http://www.4tecture.ch)

Marc Müller  
Principal Consultant

[www.powerofdevops.com](http://www.powerofdevops.com)

**4tecture**<sup>®</sup>  
empower your software solutions



A close-up photograph of several hands reaching towards a central wooden puzzle piece. The puzzle piece is light-colored wood with a dark green and red section. The hands belong to different people, suggesting collaboration. The background is blurred.

4 tecture<sup>©</sup>  
empower your software solutions