DER WEG ZU DEVSECOPS MIT GITHUB ADVANCED SECURITY FÜR AZURE DEVOPS

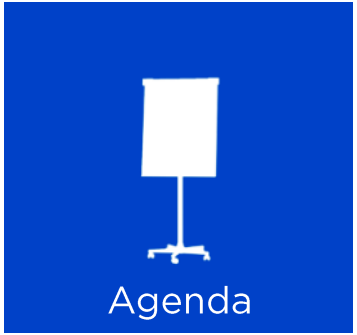
Marc Müller
Principal Consultant

www.4tecture.ch

4tecture
empower your software solutions

Agenda

- Intro
- Applied DevSecOps
- DevSecOps Process
- Questions



About me:

Marc Müller
Principal Consultant
@muellermarc

MVP
Microsoft Most Valuable Professional

4tecture©
empower your software solutions

Our Products:

Multi-Tenant OpenID Connect Identity Provider

ProAuth

www.proauth.net

Enterprise Application Framework for .NET

ReaFx

www.reafx.net

Slide Download

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


Intro

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.

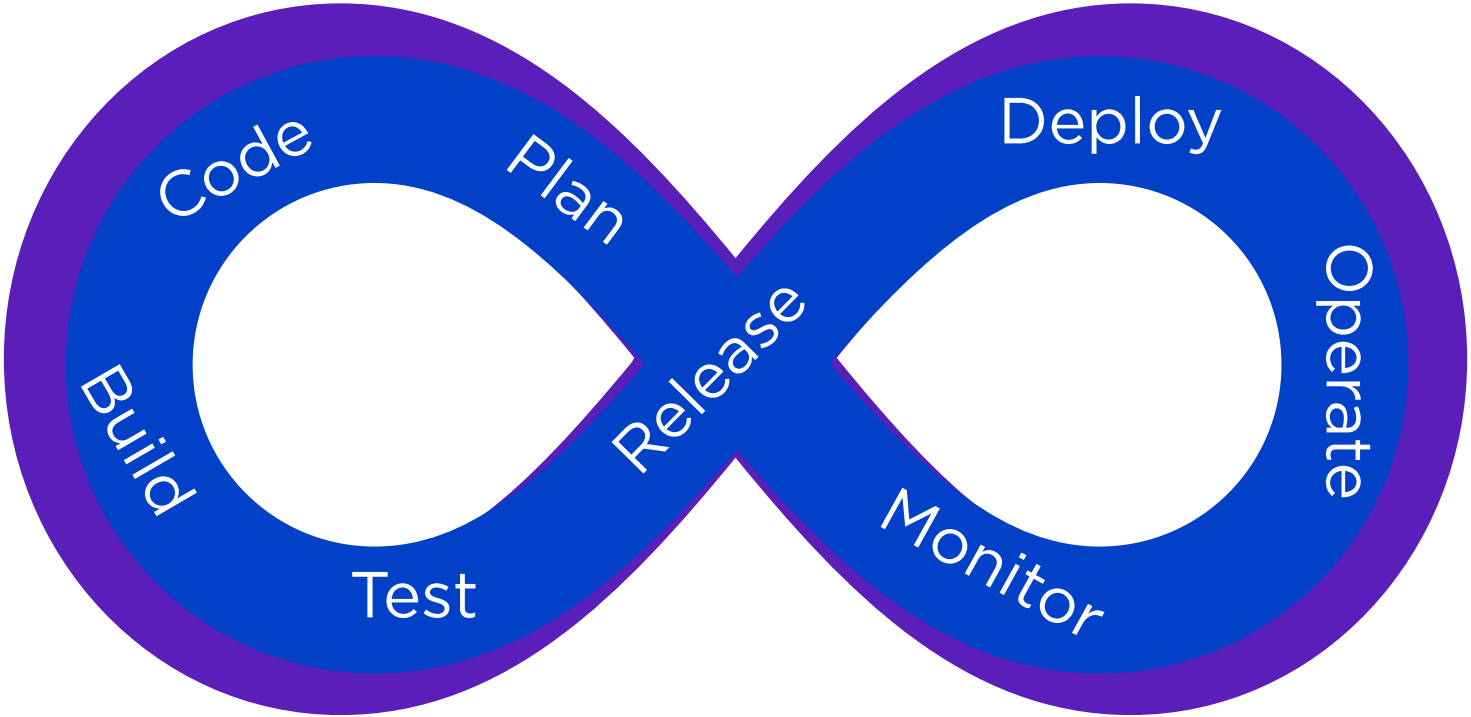
Contrary to a traditional centralized security team model, each delivery team is empowered to factor in the correct security controls into their software delivery.

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

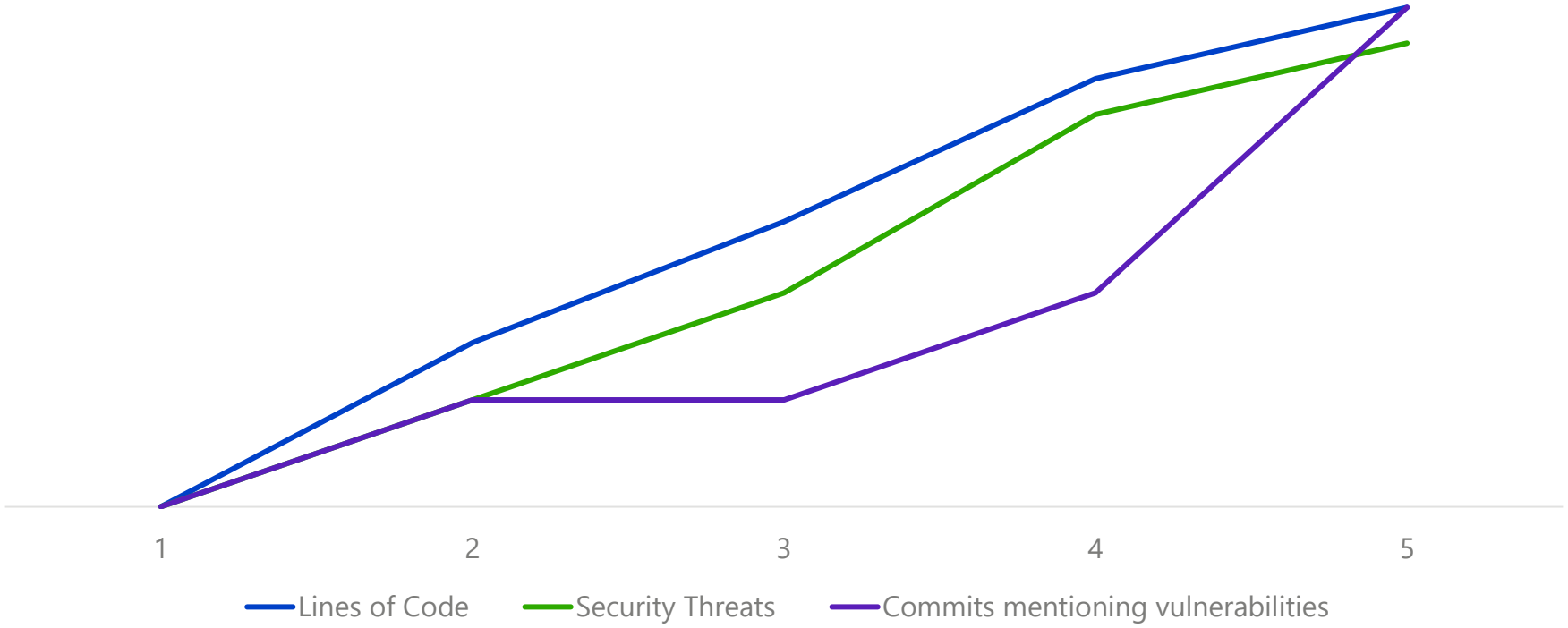
Security is tested in three main areas: static, software composition, and dynamic.

Wikipedia
https://en.wikipedia.org/wiki/DevOps#DevSecOps,_Shifting_Security_Left

Dev Sec Ops

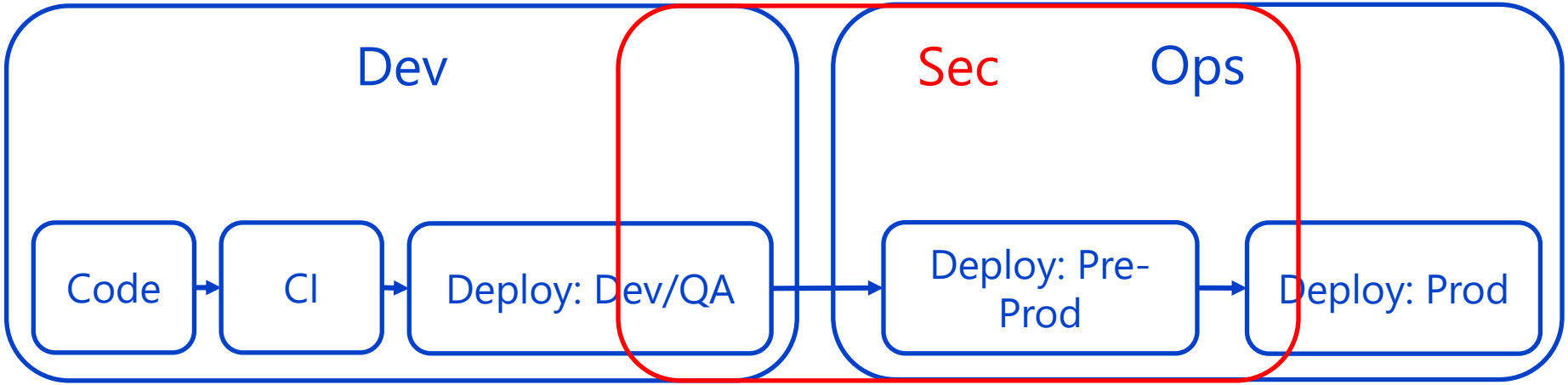


More Code = more technical dept and exposure



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

- Scan for coding errors
- Scan for weaknesses
- Scan for design flaws
- Whitebox Approach

Software Composition Analysis

- Scan dependencies for vulnerabilities
- Opensource and third-party components
- Decompose full dependency tree

Dynamic Application Security Testing

- “hack your application”
- Tools which act like pen tester
- Web and API scanner
- Blackbox approach

Interactive application Security Testing

- Detect runtime vulnerabilities
- Observe application request/response interactions, behavior and dataflow
- During manual or automated functional testing
- Active / Passive IAST
- Combining Whitebox and Runtime analysis


GHAzDO
GitHub Advanced Security for Azure DevOps

GitHub Advanced Security for Azure DevOps

Overview

4tecture®
empower your software solutions

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

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

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

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

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

Alert	Introduced
Azure DevOps personal access token (PAT) `…ztqfva` Critical #6 in README.md:158 (+1)	Dec 2, 2023

Microsoft Defender for Cloud | DevOps security

Showing subscription 'Microsoft Azure Sponsorship'

Search
General
Overview
Getting started
Recommendations
Attack path analysis
Security alerts
Inventory
Cloud Security Explorer
Workbooks
Community
Diagnose and solve problems
Cloud Security
Security posture
Regulatory compliance
Workload protections
Data security
Firewall Manager
DevOps security
Management
Environment settings
Security solutions
Workflow automation

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.

Security Overview

DevOps security findings

3 High
4 Medium
0 Low

DevOps security results

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

DevOps environment posture management recommendations

0 High severity recommendations, on 0 resources

Recommendations results. Open>

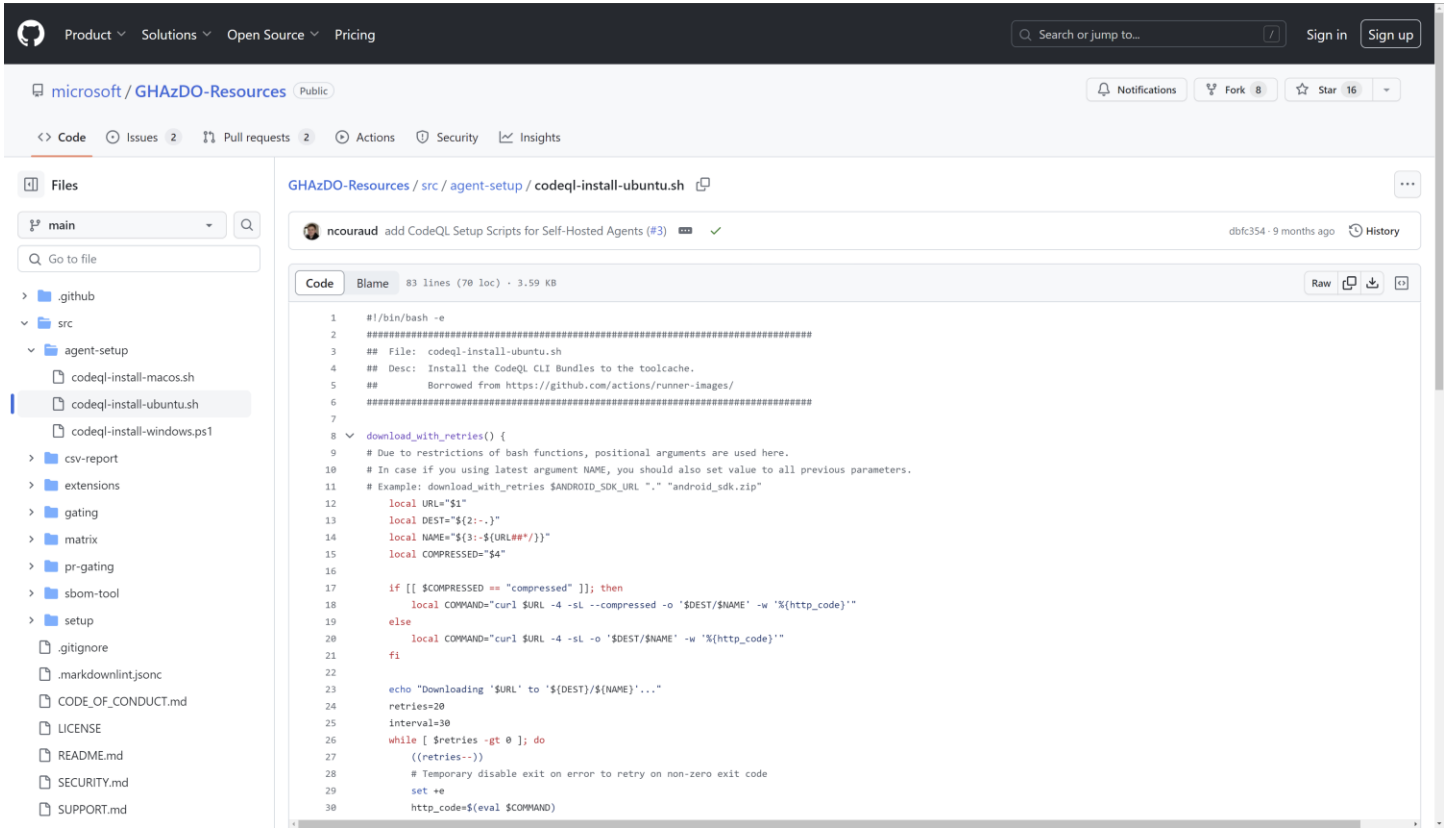
Learn more >

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

Name	DevOps environment	Advanced security status	Pull request status	Findings
4tecture-demo	Azure DevOps	Off	Partially enabled	7

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

Private Agent Deployment



Source: https://github.com/microsoft/GHAzDO-Resources/blob/main/src/agent-setup/codeql-install-ubuntu.sh

Demo

Enabling GitHub Advanced Security

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

+ Create

Repositories | Settings | Policies | Security

Advanced Security

Enable all | Disable all

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.

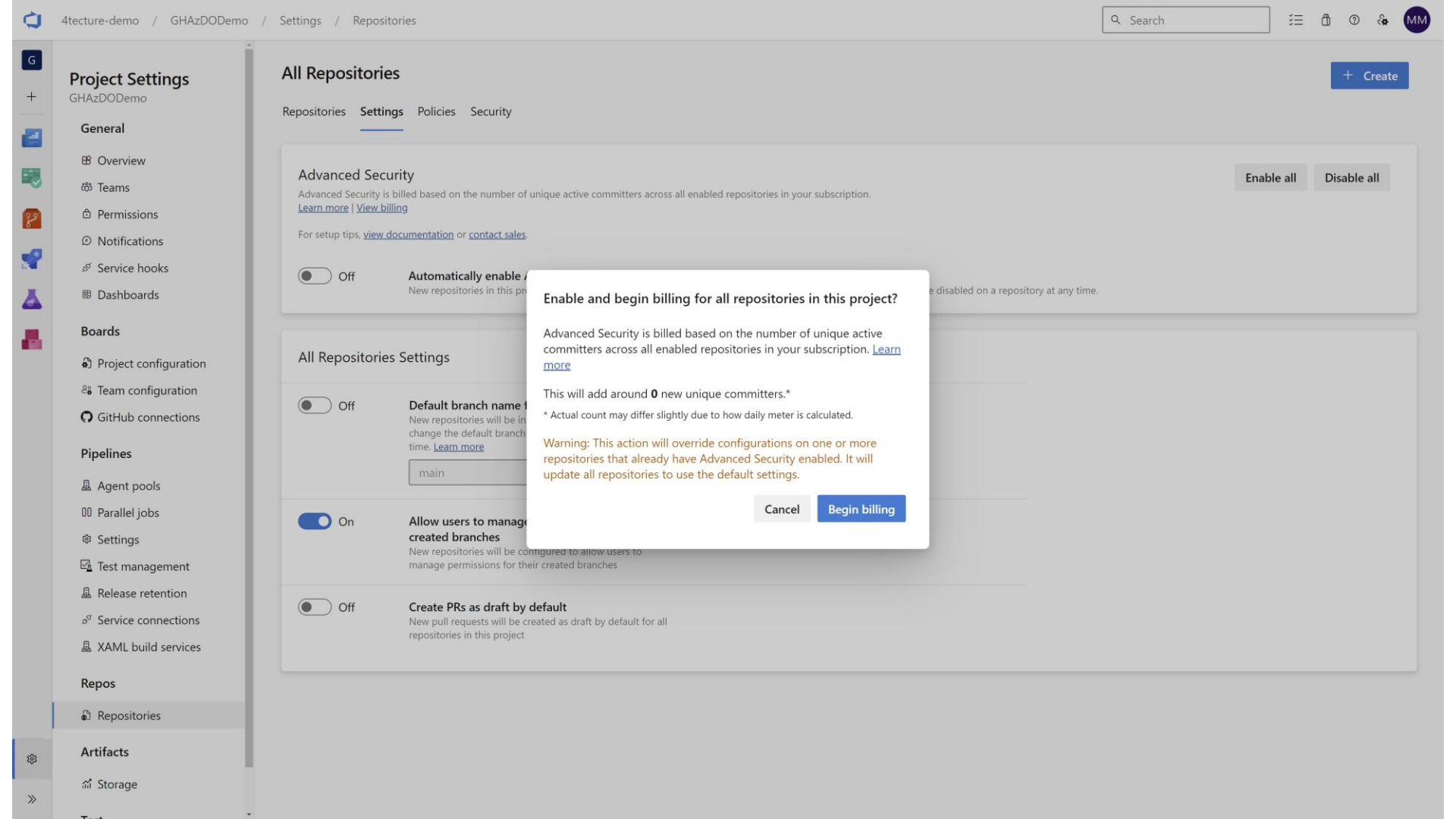
All Repositories Settings

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


4tecture-demo / GHAzDODemo / Settings / Repositories
Search
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
All Repositories
Create
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.
Enable all
Disable all
Off
Automatically enable
New repositories in this pr
e disabled on a repository at any time.
All Repositories Settings
Off
Default branch name
New repositories will be in
change the default branch
time. Learn more
main
On
Allow users to manag
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
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

Search

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

All Repositories

Filter by keywords

- demo-devsecops
- GHAzDODemo

demo-devsecops

Browse | Rename

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

GHAzDODemo

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Releases

Library

Task groups

Deployment groups

Test Plans

Artifacts

Project settings

GHAzDO with AutoBuild

Variables | Run

main | demo-devsecops / .azure-pipelines/**advanced-security.yml**

```yaml
trigger:
- main

schedules:
- cron: "0 0 * * *"
  displayName: 'Nightly build'
  branches:
    include:
    - master
  always: true

pool:
  vmImage: ubuntu-latest

steps:
- task: AdvancedSecurity-Codeql-Init@1
  inputs:
    languages: 'csharp'
    querysuite: 'security-extended'
- task: AdvancedSecurity-Codeql-Autobuild@1
- task: AdvancedSecurity-Dependency-Scanning@1
- task: AdvancedSecurity-Codeql-Analyze@1
- task: AdvancedSecurity-Publish@1

```

Tasks

Search tasks

- .NET Core — Build, test, package, or publish a dotnet applicatio...
- A wrapper for the AzureDevOps.WikiPDFE... — 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

GHAzDODemo

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Releases

Library

Task groups

Deployment groups

Test Plans

Artifacts

Project settings

HelloWorld CI

Variables | Run

main | demo-devsecops / .azure-pipelines/**helloworld-ci.yml** | Show assistant

```yaml
# Trigger the pipeline on changes to the main branch
trigger:
- main

# Define the build environment
pool:
  vmImage: 'ubuntu-latest'

# Define variables used in the pipeline
variables:
  buildConfiguration: 'Release'
  workingDirectory: '$(Build.SourcesDirectory)/HelloWorld'
  # DependencyScanning.Timeout: 600 # default is 300, anything under 300 has no effect
  # DependencyScanning.Skip: true # break-glass scenario to skip dependency scanning from blocking the build

# Pipeline steps
steps:
# Initialize Advanced Security CodeQL
Settings
- task: AdvancedSecurity-Codeql-Init@1
  inputs:
    languages: 'csharp'
    querysuite: 'security-extended'
  displayName: 'Initialize CodeQL'

# Install .NET Core SDK
Settings
- task: UseDotNet@2
  inputs:
    packageType: 'sdk'
    version: '8.x'
    installationPath: $(Agent.ToolsDirectory)/dotnet
  displayName: 'Install .NET Core SDK'

# Restore dependencies using dotnet command
- script: dotnet restore $(workingDirectory)/HelloWorld.sln
  displayName: 'Restore dependencies'
  workingDirectory: $(workingDirectory)

# Build the solution using dotnet command
- script: dotnet build $(workingDirectory)/HelloWorld.sln --configuration $(buildConfiguration) --no-restore
```

GHAzDODemo

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Releases

Library

Task groups

Deployment groups

Test Plans

Artifacts

Project settings

HelloWorld CI

Variables | Run

main | demo-devsecops / .azure-pipelines/**helloworld-ci.yml** | Show assistant

```yaml
# Build the solution using dotnet command
- script: dotnet build $(workingDirectory)/HelloWorld.sln --configuration $(buildConfiguration) --no-restore
  displayName: 'Build solution'
  workingDirectory: $(workingDirectory)

# Run unit tests using dotnet command
- script: dotnet test $(workingDirectory)/HelloWorld.sln --configuration $(buildConfiguration) --no-build --logger trx
  displayName: 'Run unit tests'
  workingDirectory: $(workingDirectory)

# Publish the web application to the artifact staging directory
- script: dotnet publish $(workingDirectory)/HelloWorld/HelloWorld.csproj --configuration $(buildConfiguration) --output $(Build.ArtifactStagingDirectory)
  displayName: 'Publish web app'
  workingDirectory: $(workingDirectory)

# Create an SBOM and add it to the pipeline artifact
- script:
    curl -Lo $(Agent.TempDirectory)/sbom-tool https://github.com/microsoft/sbom-tool/releases/latest/download/sbom-tool-linux-x64
    chmod +x $(Agent.TempDirectory)/sbom-tool
    mkdir $(Build.ArtifactStagingDirectory)/sbom
    $(Agent.TempDirectory)/sbom-tool generate -b $(Build.ArtifactStagingDirectory)/sbom -bc $(Build.SourcesDirectory) -pn HelloWorld -pv $(Build.BuildNumber) -ps
  displayName: Generate SBOM
Settings
- task: PublishPipelineArtifact@1
  inputs:
    targetPath: '$(Build.ArtifactStagingDirectory)/sbom'
    artifact: 'sbom'
    publishLocation: 'pipeline'
  displayName: Publish SBOM artifact

# Publish the web application as a pipeline artifact
Settings
- task: PublishPipelineArtifact@1
  inputs:
    targetPath: '$(Build.ArtifactStagingDirectory)'
    artifact: 'webapp'
    publishLocation: 'pipeline'
  displayName: 'Publish Pipeline Artifact'

# Run Advanced Security Dependency Scanning
```

GHAzDODemo

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Releases

Library

Task groups

Deployment groups

Test Plans

Artifacts

Project settings

HelloWorld CI

Variables | Run

main | demo-devsecops / .azure-pipelines/**helloworld-ci.yml** | Show assistant

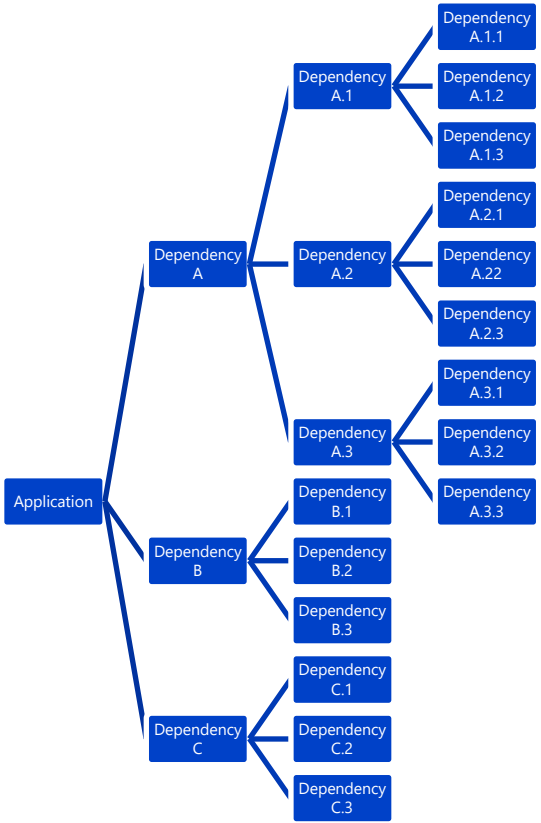
```yaml
# Run Advanced Security Dependency Scanning
Settings
- task: AdvancedSecurity-Dependency-Scanning@1
  # inputs:
  #   directoryExclusionList: 'sampledir/ignoreddirectory1;sampledir/ignoreddirectory'
  displayName: 'Dependency Scanning'

# Run CodeQL Analysis
Settings
- task: AdvancedSecurity-Codeql-Analyze@1
  displayName: 'CodeQL Analysis'

# Publish CodeQL Analysis Results
Settings
- task: AdvancedSecurity-Publish@1
  displayName: 'Publish CodeQL Results'

# Add a CSV report for Advanced Security alerts
Settings
- task: PowerShell@2
  displayName: 'Generate a csv report based on Advanced Security alerts'
  inputs:
    targetType: filePath
    filePath: $(Build.SourcesDirectory)/.azure-pipelines/ghazdo-csv-report.ps1
    pwsh: true
  env:
    MAPPED_ADO_PAT: $(GHAzDO_PAT)

# Conditional step for Pull Requests: Run a PowerShell script for additional PR validation
- ${{ if eq(variables['Build.Reason'], 'PullRequest') }}:
  Settings
  - task: PowerShell@2
    displayName: 'PR Gating - Additional Checks for Pull Requests'
    inputs:
      targetType: filePath
      filePath: $(Build.SourcesDirectory)/.azure-pipelines/CIGate.ps1
      pwsh: true
    env:
      MAPPED_ADO_PAT: $(GHAzDO_PAT)
```

GitHub Advanced Security for Azure DevOps

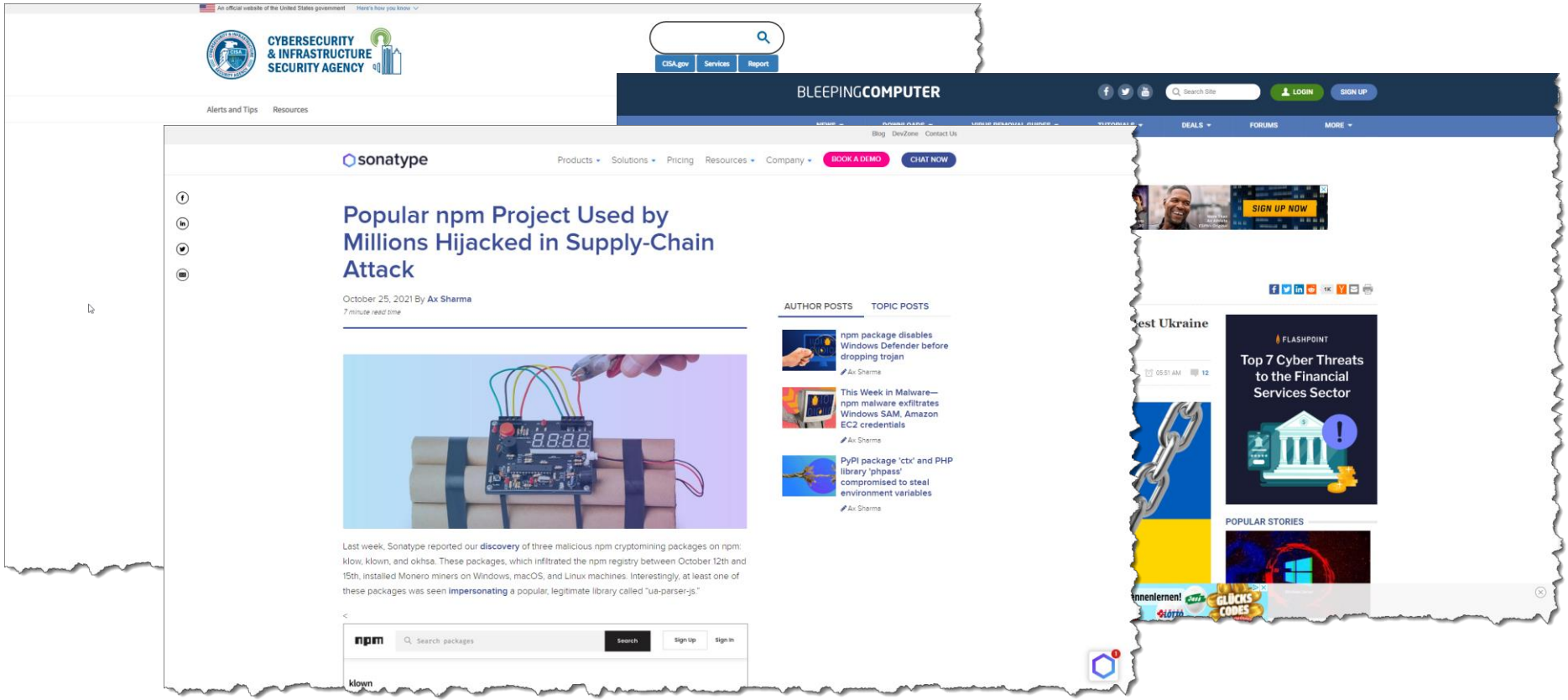
Dependency Scanning

4tecture
empower your software solutions

Application Dependencies



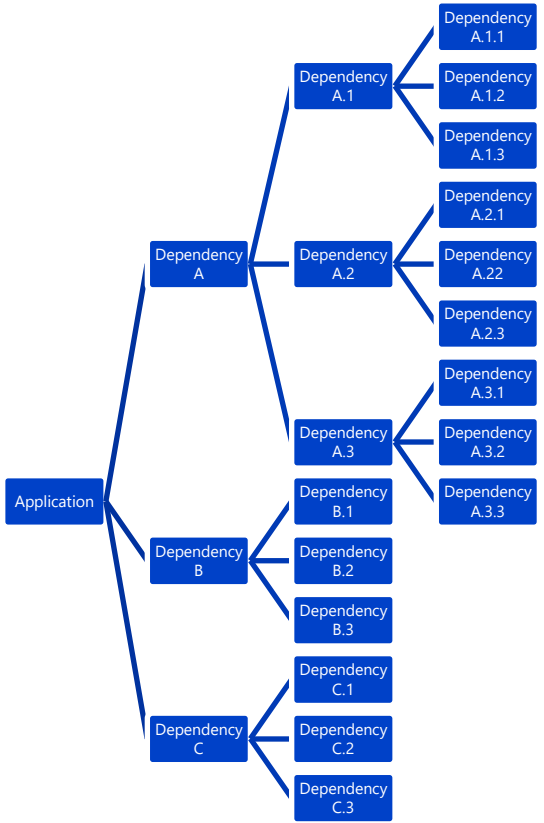
Current threats



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



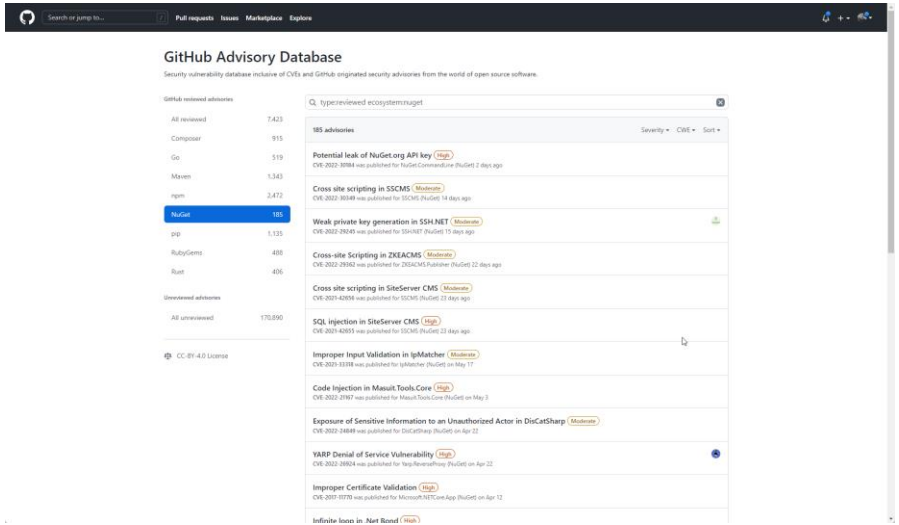
Supported package ecosystem

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

Source: https://learn.microsoft.com/en-us/azure/devops/repos/security/github-advanced-security-dependency-scanning?view=azure-devops

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



Source: https://github.com/advisories

Demo

Dependency Scanning

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

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

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

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

Close alert

#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)

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

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
Friday Detected	HelloWorld CI (Job)	Dec 2, 2023
Friday Detected	GHAzDO with AutoBuild (Job)	Dec 2, 2023

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

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

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

Close alert

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

er (8.0.0)

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

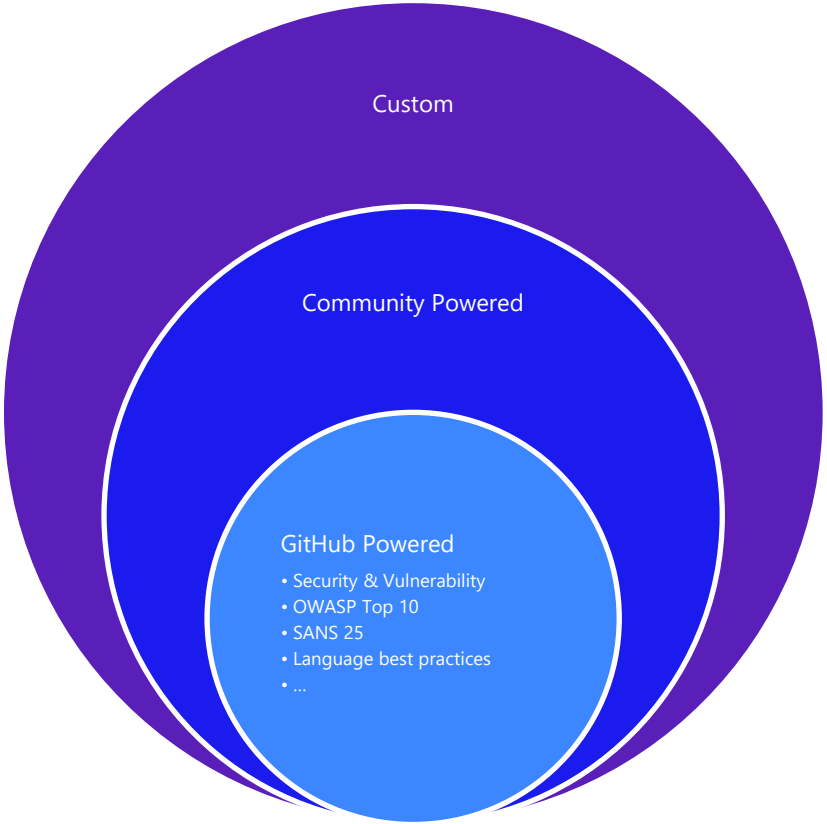
Supported Languages

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

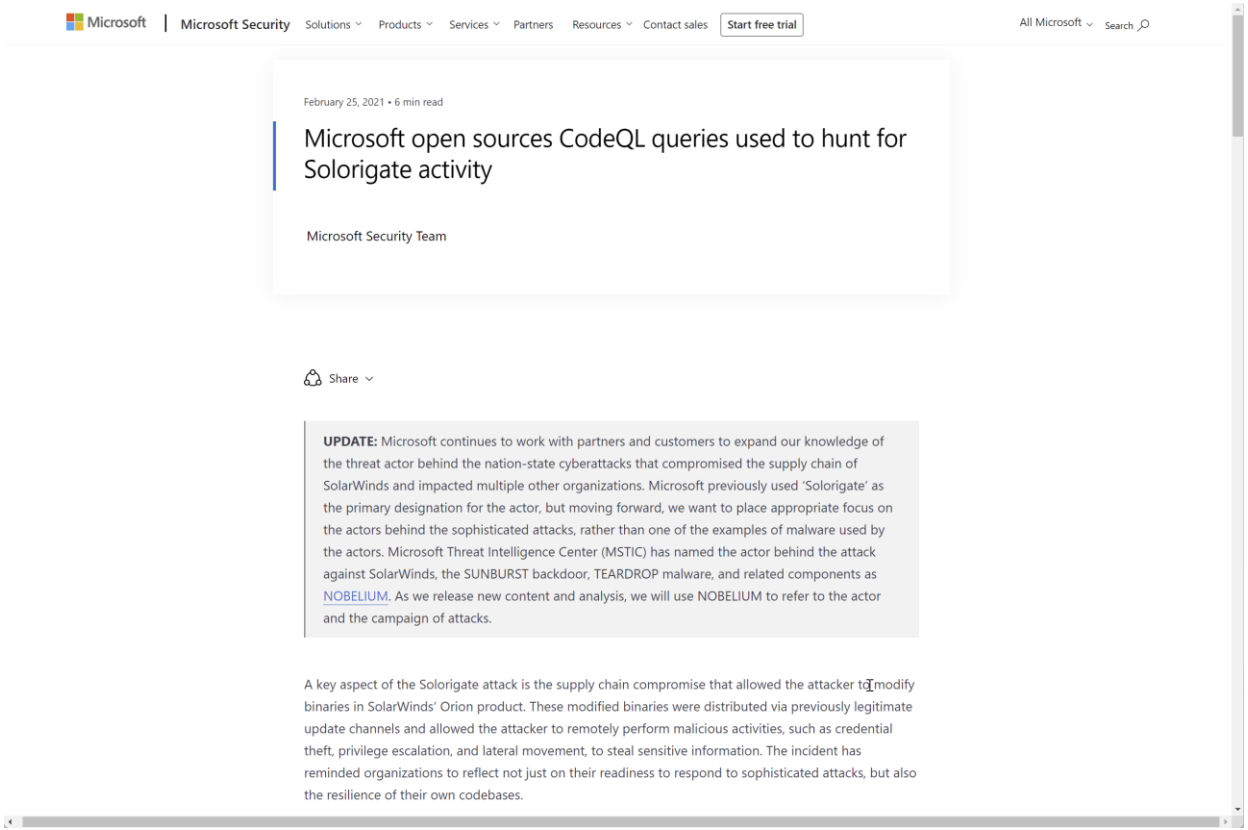
CodeQL Queries

- 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



Source: https://www.microsoft.com/security/blog/2021/02/25/microsoft-open-sources-codeql-queries-used-to-hunt-for-solorigate-activity/

Custom Query

```yaml
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:
 - mygithuborg/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
```

```yaml
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
```

Source: https://learn.microsoft.com/en-us/azure/devops/repos/security/github-advanced-security-code-scanning?view=azure-devops

Demo

Code Scanning

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

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

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)

Close alert

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

Overview Detections

Locations

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

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

GHAzDODemo

Overview

Boards

Repos

Files

Commits

Pushes

Branches

Tags

Pull requests

Advanced Security

Pipelines

Test Plans

Artifacts

Project settings

Extend SQL Injection Controller

Approve | Set auto-complete

Active !284 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

demo-devsecops

HelloWorld/HelloWorld/Controll...

SqlInjectionController.cs

**SQL query built from use...

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

```
using System.Collections.Generic;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using HelloWorld.Database;
using Microsoft.EntityFrameworkCore;
using Dapper;

namespace HelloWorld.Controllers
{
    public class SqlInjectionController : Controller
    {
        private DemoContext _context;

        public SqlInjectionController(DemoContext context)
        {
            _context = context;
        }

        public IActionResult Index()
        {
            return View();
        }

        [HttpGet("SqlInjection/SearchPersonUnsecure/{name}")]
        public async Task<IActionResult> SearchPersonUnsecure(string name)
        {
            var conn = _context.Database.GetDbConnection();
```

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)

Close alert

#10 Open in refs/pull/284/merge • detected Just now

Overview Detections

Location

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

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

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.

Severity
High

First detected
Just now

Finding details

Tool
CodeQL

Rule ID
cs/sql-injection

Weakness
CWE-089

Demo

Custom Queries

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

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

authorizedcontrollers.ql

Edit

Contents History Compare Blame

```
/**
 * @name Secure API controllers without Authorize attribute
 * @description Secure controllers within an 'api-secure' route must have an Authorize attribute.
 * @kind problem
 * @problem.severity warning
 * @id demo/api-controller-without-authorize
 * @tags security
 *       api
 */

import csharp
import semmle.code.csharp.Attribute

from Class controllerClass, Attribute routeAttribute
where controllerClass = routeAttribute.getTarget()
    and routeAttribute.getType().hasName("RouteAttribute")
    and routeAttribute.getArgument(0).getValue().matches("%api-secure/%")
    and not exists(Attribute authorizeAttribute |
        authorizeAttribute.getTarget() = controllerClass and
        authorizeAttribute.getType().hasName("AuthorizeAttribute")
    )
select controllerClass, "This controller serving API a secret route does not have the 'Authorize' attribute."

```

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

demo-devsecops
.azure-pipelines
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

feature/customquery / .azure-pipelines / codeqlconfiguration.yml

codeqlconfiguration.yml

Commit Revert

Contents Highlight changes

```yaml
name: "Run custom queries"

disable-default-queries: false

queries:
  - name: Autorized controllers
    uses: ./.customQueries/authorizedcontrollers.ql

paths:
- HelloWorld

```

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

demo-devsecops
.azure-pipelines
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

helloworld-ci.yml

Edit

Contents History Compare Blame

```yaml
# Trigger the pipeline on changes to the main branch
trigger:
- main

# Define the build environment
pool:
  vmImage: 'ubuntu-latest'

# Define variables used in the pipeline
variables:
  buildConfiguration: 'Release'
  workingDirectory: '$(Build.SourcesDirectory)/HelloWorld'
  # DependencyScanning.Timeout: 600 # default is 300, anything under 300 has no effect
  # DependencyScanning.Skip: true # break-glass scenario to skip dependency scanning from blocking the build

# Pipeline steps
steps:
# Initialize Advanced Security CodeQL
- task: AdvancedSecurity-Codeql-Init@1
  inputs:
    languages: 'csharp'
    querysuite: 'security-extended'
    configfilepath: '$(build.sourcesDirectory)/.azure-pipelines/codeqlconfiguration.yml'
  displayName: 'Initialize CodeQL'

# Install .NET Core SDK
- task: UseDotNet@2
  inputs:
    packageType: 'sdk'
    version: '8.x'
    installationPath: $(Agent.ToolsDirectory)/dotnet
  displayName: 'Install .NET Core SDK'

# Restore dependencies using dotnet command
- script: dotnet restore $(workingDirectory)/HelloWorld.sln
  displayName: 'Restore dependencies'
  workingDirectory: $(workingDirectory)

# Build the solution using dotnet command
- script: dotnet build $(workingDirectory)/HelloWorld.sln --configuration $(buildConfiguration) --no-restore
  displayName: 'Build solution'
  workingDirectory: $(workingDirectory)

# Run unit tests using dotnet command
- script: dotnet test $(workingDirectory)/HelloWorld.sln --configuration $(buildConfiguration) --no-build --logger trx
  displayName: 'Run unit tests'
  workingDirectory: $(workingDirectory)
```

GHAzDODemo

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

Project settings

Add custom CodeQL query

Approve | Set auto-complete

Active | !283 | 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
  - codeqlconfiguration.yml
  - helloworld-ci.yml
- .customQueries
  - authorizedcontrollers.ql
  - codeql-pack.lock.yml
  - codeql-pack.yml
- .vscode
  - settings.json
- HelloWorld/HelloWorld/Controll...
  - AuthorizedAccessContro...
    - **Secure API controllers wit...
  - AuthorizedAccessContro...
- .gitignore

Marc Müller 6m ago — Active

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
```

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: `git rebase –I [commit ID before secret]~1`
    - 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

Source: https://learn.microsoft.com/en-us/azure/devops/repos/security/github-advanced-security-secret-scanning?view=azure-devops

Demo

Secret Scanning

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

Alert	Introduced
Azure DevOps personal access token (PAT) `…ztqfva` Critical #6 in README.md:158 (+1)	Dec 2, 2023

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

Azure DevOps personal access token (PAT)

Close alert

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

Locations

README.md:158 @ a9d892a5

```
154 [ApiController]
155 public class ThirdPartyAccessController : ControllerBase
156 {
157
158     private string api_key = "qqvu5wi56[redacted]hy2aztqfva";  //This is a fake API key for demonstration purposes
159
160     //Simulated sensitive data (AWS credentials)
161     private string aws_access_key_id = "AK[redacted]LE";
162     private string aws_secret_access_key = "wJalrXUtr[redacted]LEKEY";
```

HelloWorld/HelloWorld/Controllers/ThirdPartyAccessController.cs:11 @ f1975ac4

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.

Severity
Critical

First detected
Dec 2, 2023 at 8:49 AM

Finding details

Type
AdoPat

ID
SEC101/102

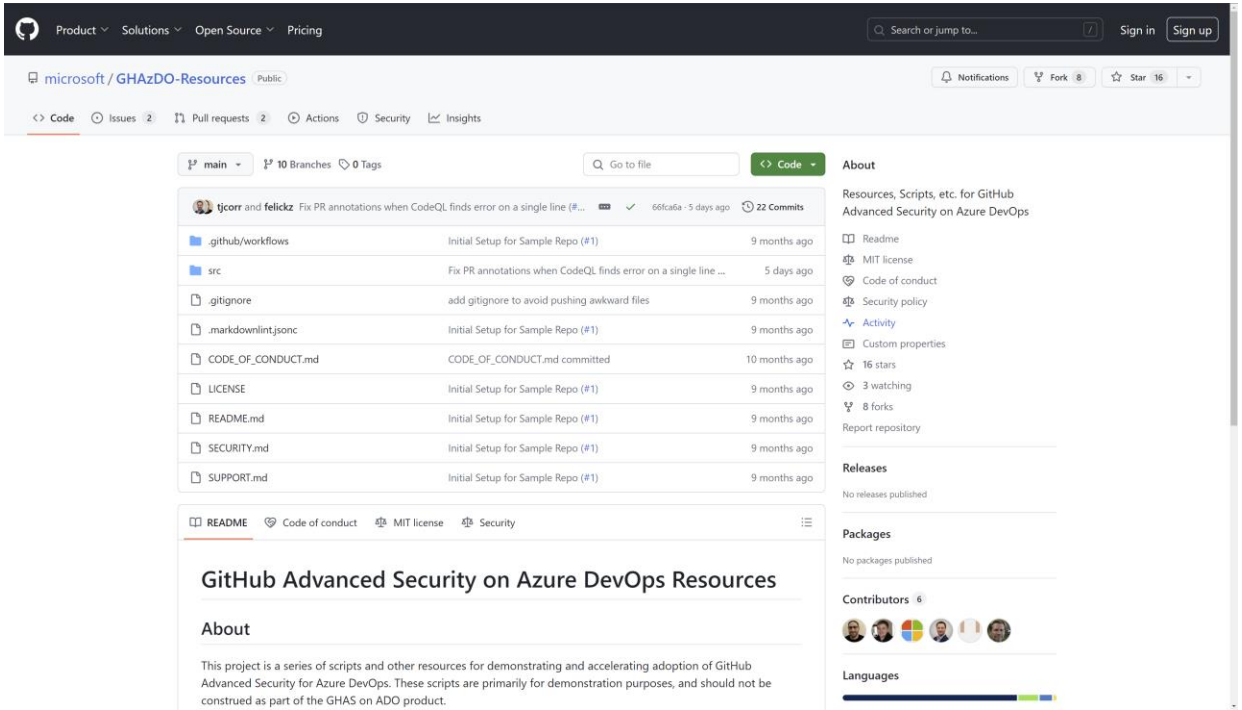
GitHub Advanced Security for Azure DevOps

Azure Pipelines

4tecture®
empower your software solutions

GHAzDo Resources

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



Demo

PR Gating

Demo

SBOM

Demo

GHAzDO Exports

GitHub Advanced Security for Azure DevOps

Defender for Cloud

4tecture
empower your software solutions

Demo

Defender for Cloud

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
www.4tecture.ch

Marc Müller
Principal Consultant

www.powerofdevops.com

4tecture©
empower your software solutions

ProAuth

ReaFx


4tecture©
empower your software solutions