GitHub

GitHub
代碼安全培訓

Session 學習目標

掌握配置和使用 GitHub Code Security 的實用技能

- 理解並配置基於 CodeQL 的代碼掃描。
- 使用 Autofix 高效修復工作流程中的漏洞。
- 優化 CodeQL 以適應複雜代碼庫，包括單代碼庫。
- 將第三方工具整合到代碼掃描中。

我們的議程

安全現狀

當前全球應用安全的現狀如何

什麼是 GHAS?

GHAS 在其中扮演什麼角色

使能

在組織層面激活代碼安全功能

代碼安全

深入解析 GitHub Code Security 的所有產品和功能

第三方集成

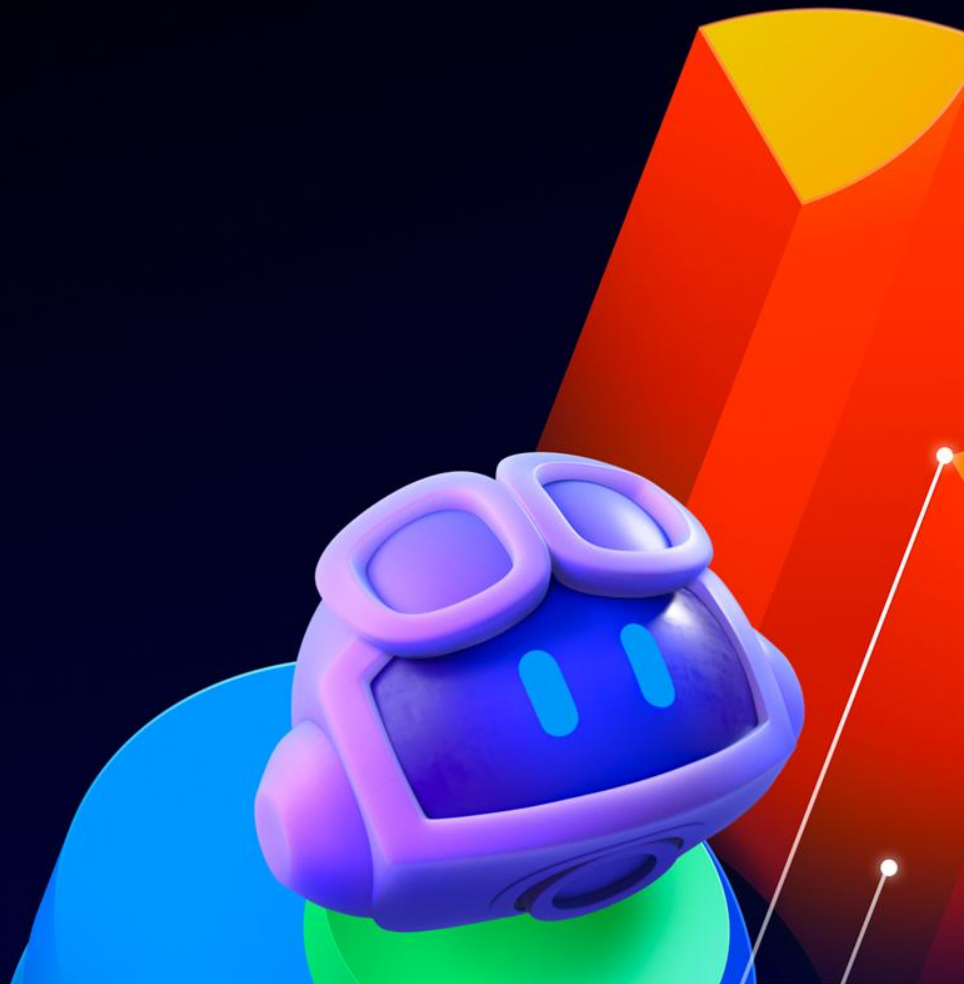
數據匯出與同步

回顧

我們今天所學總結

GitHub 高級安全

安全現狀



DevSecOps 現狀

安全風險

應用程式是首要攻擊載體

80%的入侵事件是通過網頁應用漏洞攻擊實現的

來源： Verizon數據洩露調查報告2023

停滯的進展

3個月後仍有65%的漏洞存在

只有33%的洩露是由組織的團隊或工具發現的

來源：Veracode 2023年安全現狀報告

增幅器

組織認為人工智慧提供了更高的投資回報率

84%的高管計劃優先考慮生成式人工智慧網路安全解決方案，而非傳統網路安全解決方案

來源：IBM CEO生成式人工智慧指南，2023年

展望未來...

我們的攻擊面
正以前所未有的速度增長

我們如今生活在一個完全被軟體所吞噬的世界。 每個組織都是軟體組織，必須學會如何在數位化領域蓬勃發展並實現創新。

700M

未來五年內將誕生更多應用程式

這比過去40年加起來還多

GitHub

人工智慧驅動的開發者平臺

GitHub 進階安全

GitHub 高級安全提供一套 AppSec 工具，說明您的組織免受安全風險。

供應鏈安全

保護您的應用程式免受第三方依賴帶來的風險，並實現您創建的軟體的可驗證來源。

代碼安全

識別第一方代碼中的易受攻擊的編碼模式，並自動修復生成式人工智慧的問題。

機密保護

檢測硬編碼的機密，防止開發者誤上傳憑證到倉庫。

License 功能清單

GitHub Enterprise

- Dependency Graph
- Dependency Insights
- Software Bill of Materials (SBOM) generation
- Dependabot Alerts
- Dependabot Security Updates
- Dependabot Version Updates
- Security Overview
- Secret risk assessment
- All Code Security features for *public repositories*
- All Secret Protection features for *public repositories*

代碼安全許可

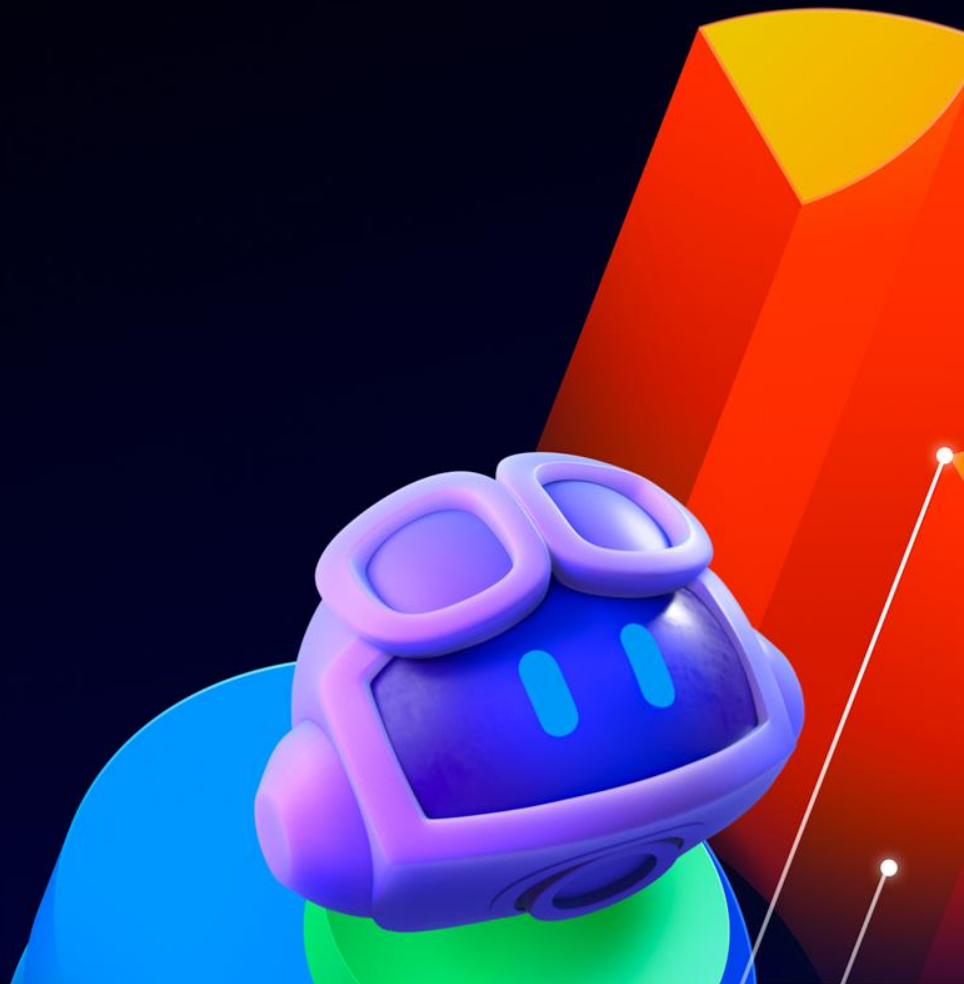
- CodeQL
- Code scanning
- Copilot Autofix
- Security campaigns
- Dependency Review
- Dependabot auto triage rules

機密保護許可

- Secret scanning
- Push protection
- Custom patterns
- AI-pattern generation
- Copilot secret scanning

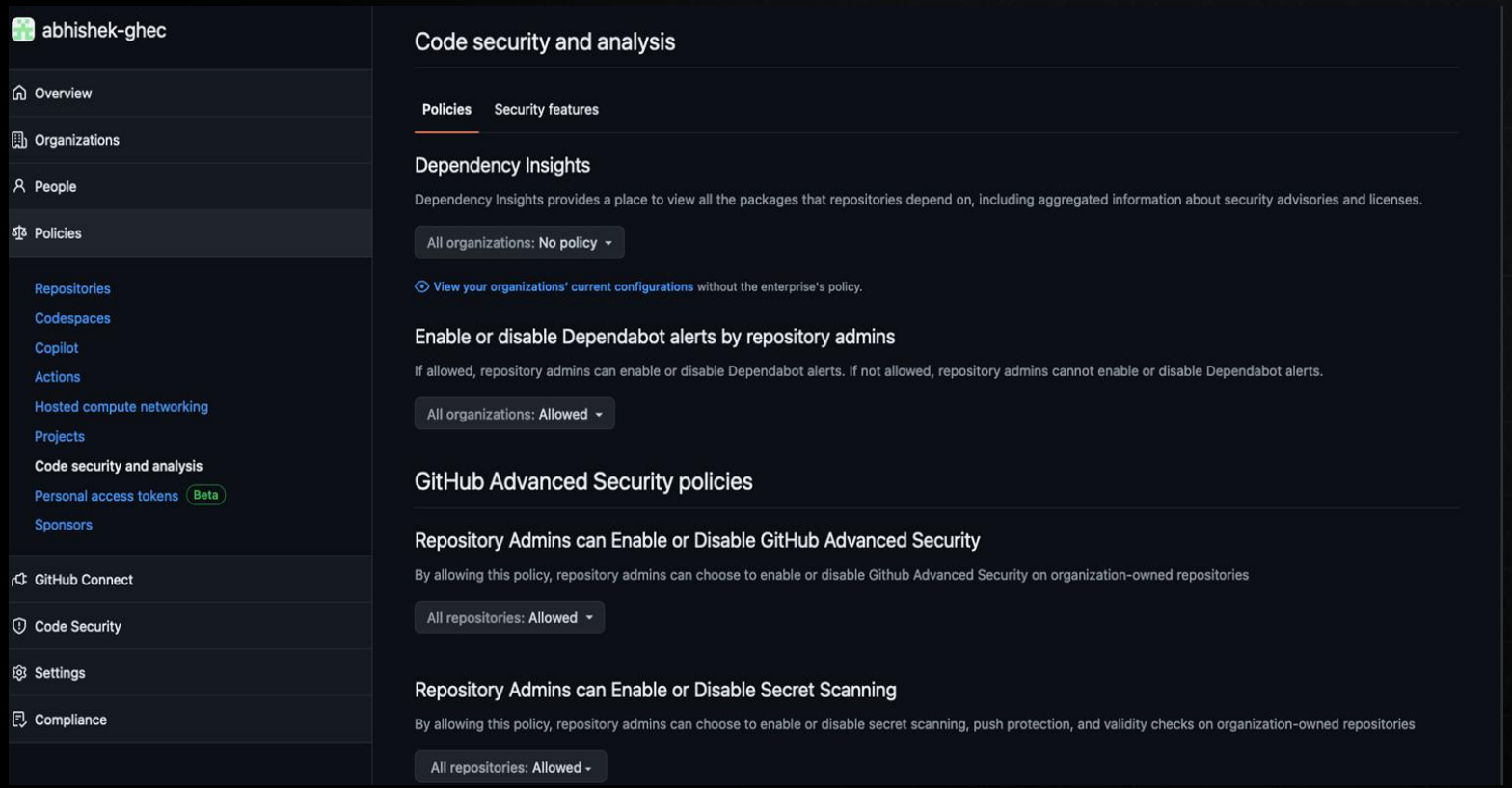
GitHub 高級安全

策略與配置



GHAS 政策

- GHAS 策略可以在企業層級進行管理
- 這些與 GHAS 安全配置不同，後者可在企業或組織層級進行管理
- 在策略推行之初就確立策略定義至關重要，以避免策略偏離


abhishek-ghec
Overview
Organizations
People
Policies
Repositories
Codespaces
Copilot
Actions
Hosted compute networking
Projects
Code security and analysis
Personal access tokens
Beta
Sponsors
GitHub Connect
Code Security
Settings
Compliance
Code security and analysis
Policies
Security features
Dependency Insights
Dependency Insights provides a place to view all the packages that repositories depend on, including aggregated information about security advisories and licenses.
All organizations: No policy
View your organizations' current configurations without the enterprise's policy.
Enable or disable Dependabot alerts by repository admins
If allowed, repository admins can enable or disable Dependabot alerts. If not allowed, repository admins cannot enable or disable Dependabot alerts.
All organizations: Allowed
GitHub Advanced Security policies
Repository Admins can Enable or Disable GitHub Advanced Security
By allowing this policy, repository admins can choose to enable or disable Github Advanced Security on organization-owned repositories
All repositories: Allowed
Repository Admins can Enable or Disable Secret Scanning
By allowing this policy, repository admins can choose to enable or disable secret scanning, push protection, and validity checks on organization-owned repositories
All repositories: Allowed

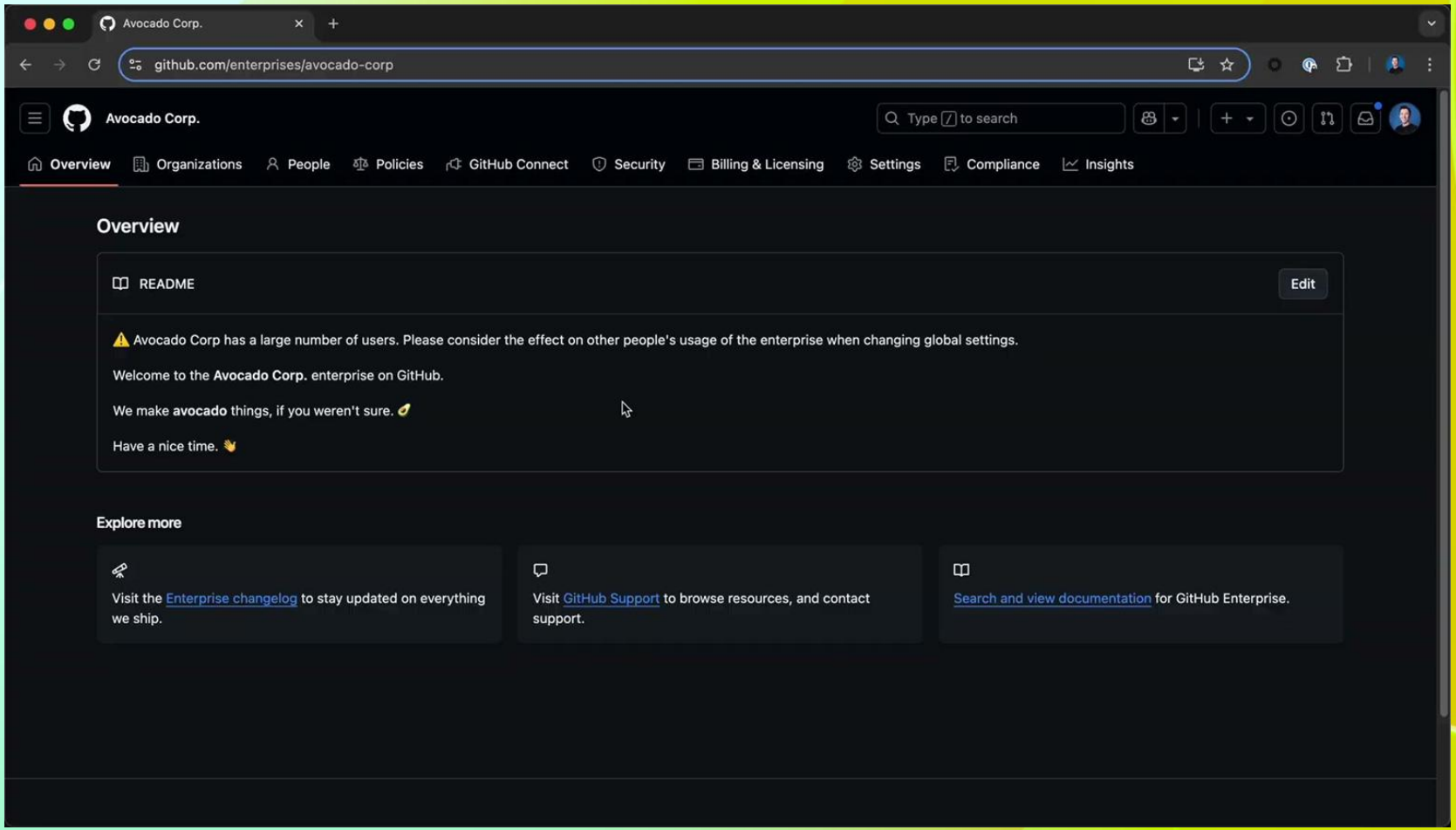
安全配置

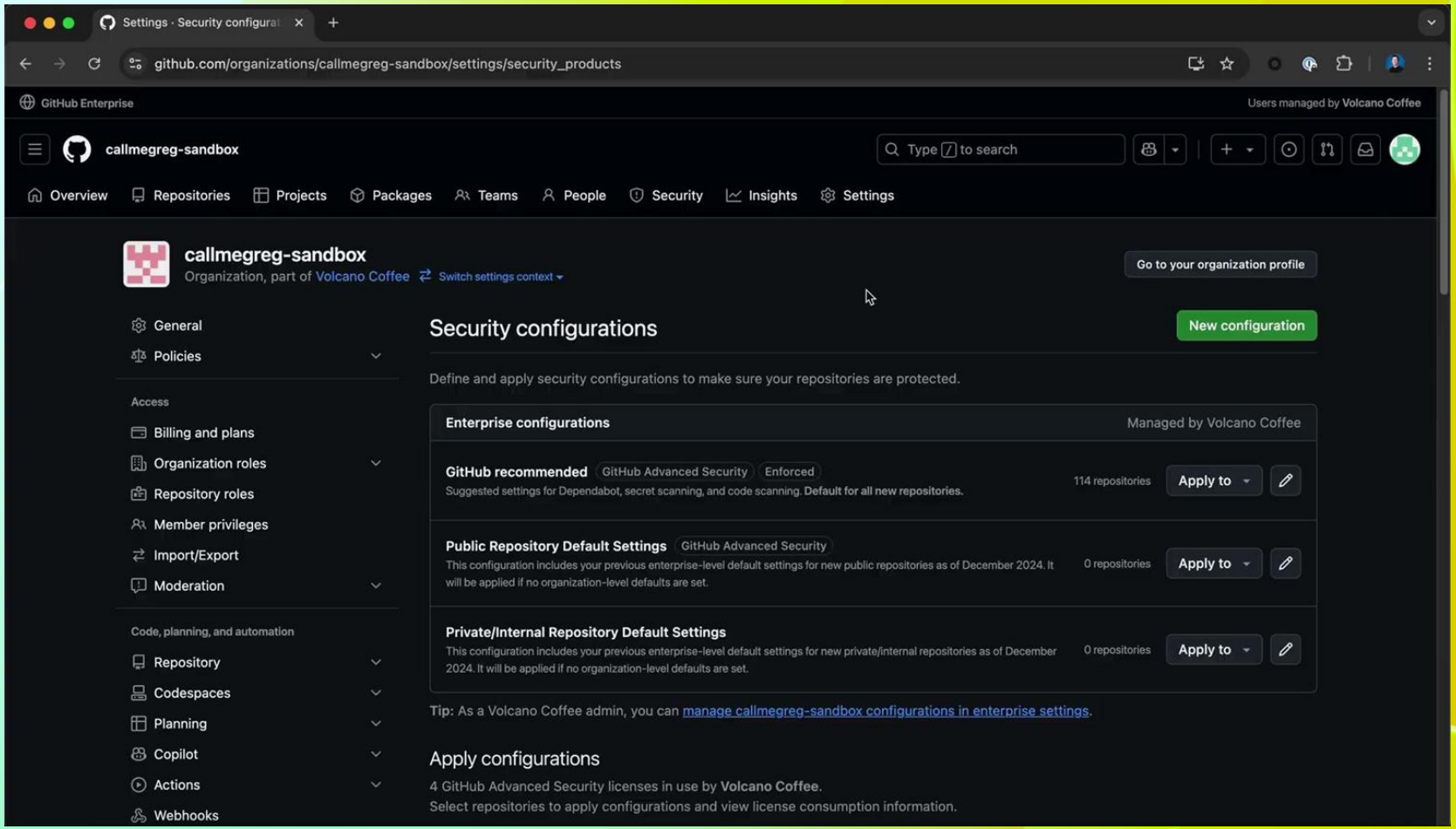
- GHAS 設置也可以在 GitHub 的多個層級應用——企業級、組織級和倉庫層面
- 作為安全經理和 GitHub 企業管理員，瞭解你希望在哪個層級執行這些設置非常重要，以便正確管理下游使用者的期望
- 通過合理規劃，你可以優化機密保護許可證的利用率

安全配置

- 通過定義可應用於多個倉庫組的安全設置集合，簡化 GitHub 安全產品的大規模推廣
- 應用“GitHub 推薦”的安全配置，或者創建自定義安全配置
- 根據不同風險配置檔或倉庫自定義屬性管理安全設置
- 查看應用配置所需的額外許可證數量，或通過禁用選定存儲庫中的功能來釋放許可證數量

Demo


Avocado Corp.
github.com/enterprises/avocado-corp
Avocado Corp.
Type / to search
Overview
Organizations
People
Policies
GitHub Connect
Security
Billing & Licensing
Settings
Compliance
Insights
Overview
README
Edit
⚠️ Avocado Corp has a large number of users. Please consider the effect on other people's usage of the enterprise when changing global settings.
Welcome to the Avocado Corp. enterprise on GitHub.
We make avocado things, if you weren't sure.
Have a nice time.
Explore more
Visit the Enterprise changelog to stay updated on everything we ship.
Visit GitHub Support to browse resources, and contact support.
Search and view documentation for GitHub Enterprise.


Settings · Security configurat
github.com/organizations/callmegreg-sandbox/settings/security_products
GitHub Enterprise
Users managed by Volcano Coffee
callmegreg-sandbox
Type / to search
Overview
Repositories
Projects
Packages
Teams
People
Security
Insights
Settings
callmegreg-sandbox
Organization, part of Volcano Coffee
Switch settings context
Go to your organization profile
General
Policies
Access
Billing and plans
Organization roles
Repository roles
Member privileges
Import/Export
Moderation
Code, planning, and automation
Repository
Codespaces
Planning
Copilot
Actions
Webhooks
Security configurations
New configuration
Define and apply security configurations to make sure your repositories are protected.
Enterprise configurations
Managed by Volcano Coffee
GitHub recommended
GitHub Advanced Security
Enforced
Suggested settings for Dependabot, secret scanning, and code scanning. Default for all new repositories.
114 repositories
Apply to
Public Repository Default Settings
GitHub Advanced Security
This configuration includes your previous enterprise-level default settings for new public repositories as of December 2024. It will be applied if no organization-level defaults are set.
0 repositories
Apply to
Private/Internal Repository Default Settings
This configuration includes your previous enterprise-level default settings for new private/internal repositories as of December 2024. It will be applied if no organization-level defaults are set.
0 repositories
Apply to
Tip: As a Volcano Coffee admin, you can manage callmegreg-sandbox configurations in enterprise settings.
Apply configurations
4 GitHub Advanced Security licenses in use by Volcano Coffee.
Select repositories to apply configurations and view license consumption information.

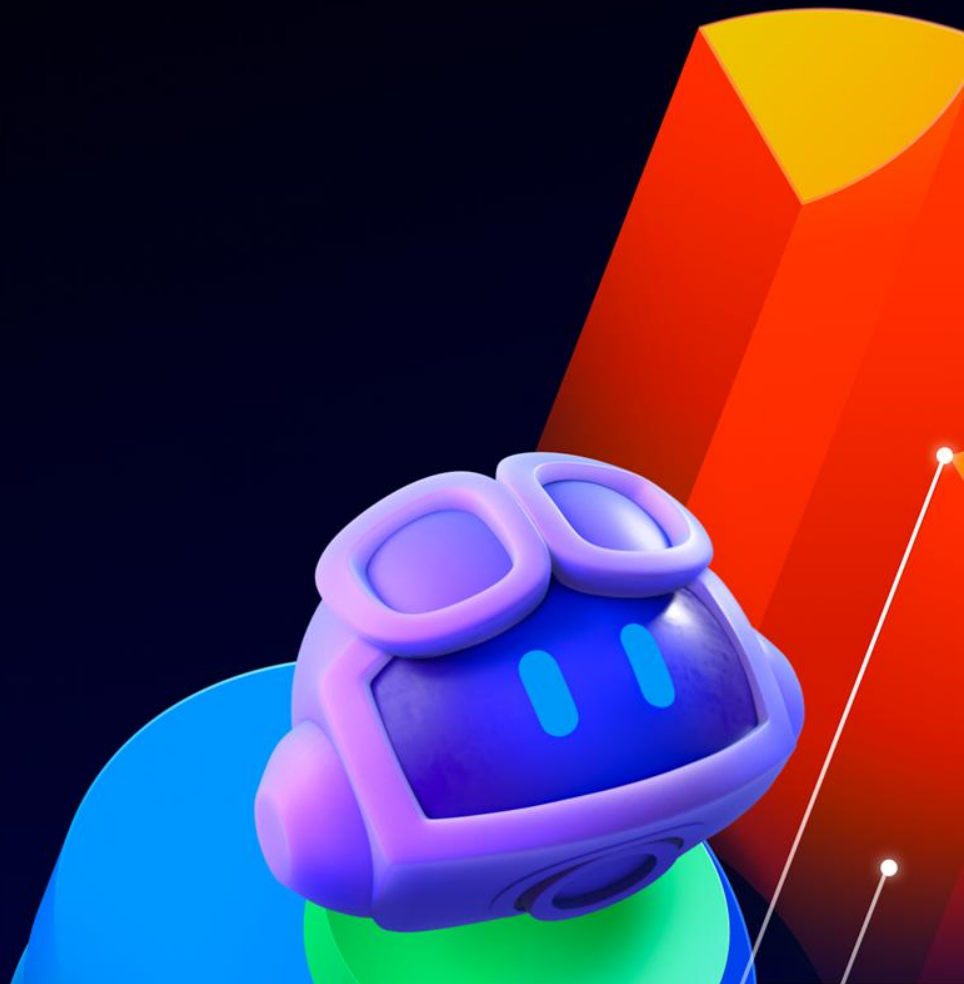
Module 0: Lab exercises

Questions?

Break!

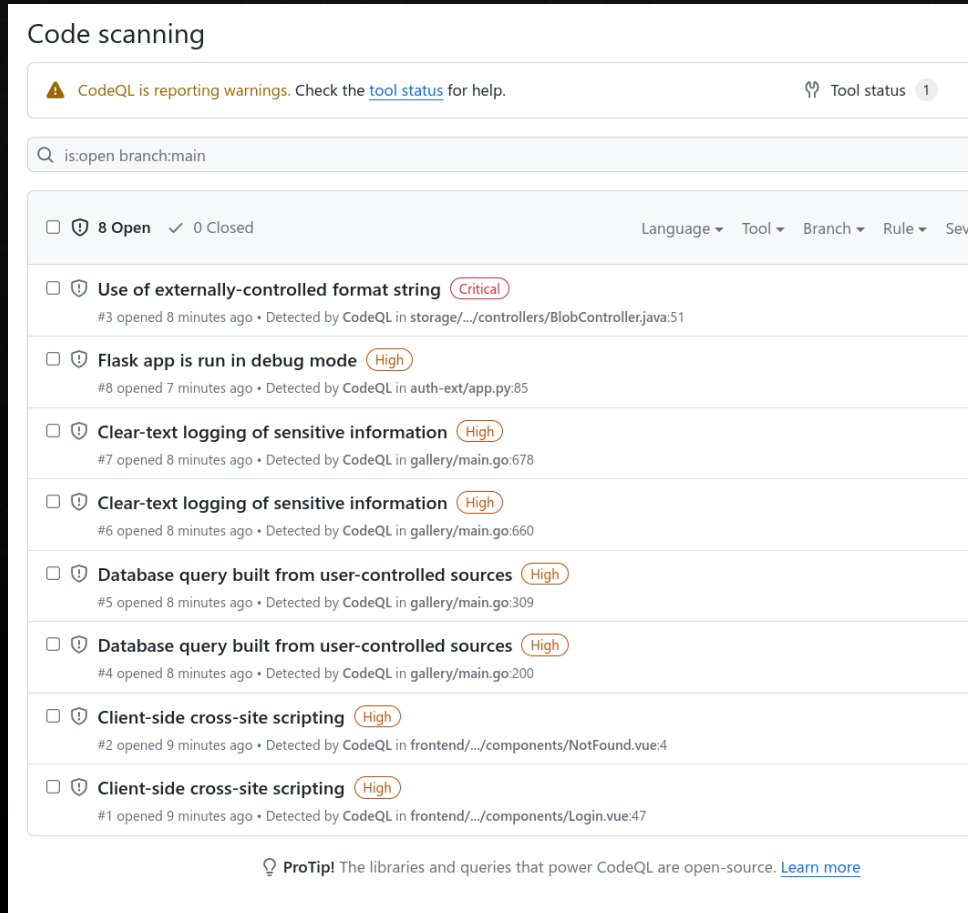
GitHub 高級安全

代碼安全

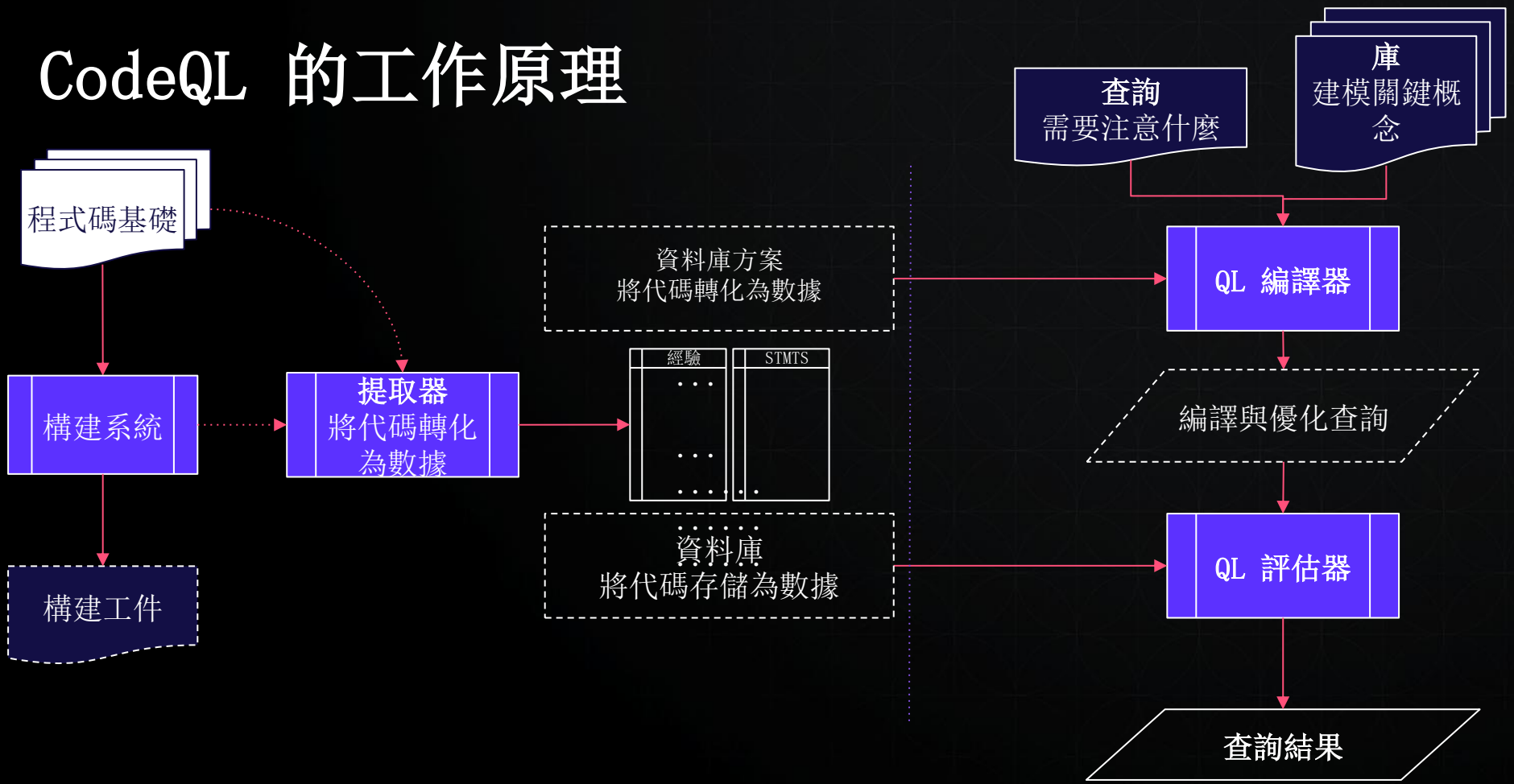


代碼掃描

- 靜態分析在不執行代碼的情況下檢查代碼問題
- GitHub 代碼掃描通過自動化原始程式碼檢查實現漏洞早期發現
- 警報會顯示在 GitHub 的專用標籤頁和拉取請求中
- CodeQL 通過語義分析為 GitHub 代碼掃描提供技術支援



CodeQL 的工作原理



SQL 注入範例

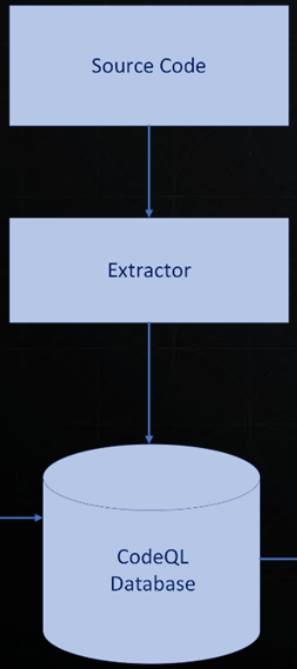
src/main/java/org/owasp/webgoat/lessons/sqlinjection/introduction/**SqlInjectionLesson10.java**:56

```
53          Statement statement =
54              connection.createStatement(
55                  ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);
56          ResultSet results = statement.executeQuery(query);
```

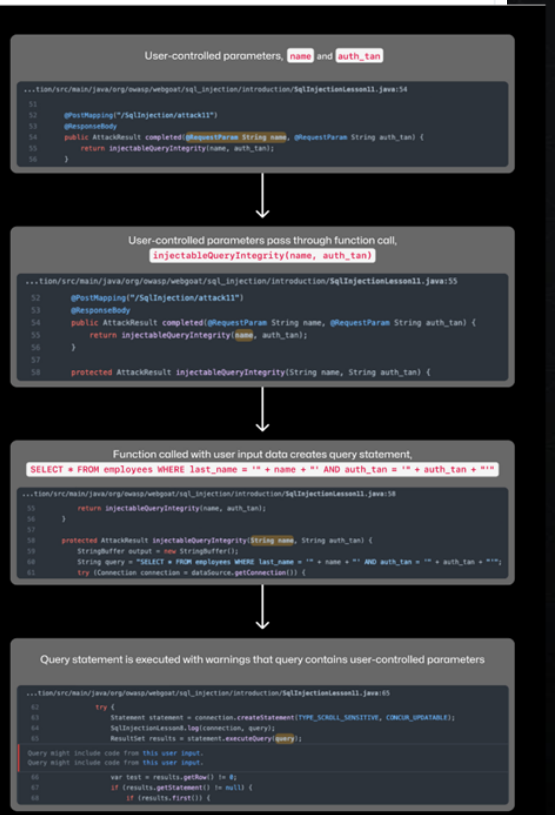
This query depends on a user-provided value.

CodeQL Show paths

```
57
58          if (results.getStatement() != null) {
59            results.first();
```

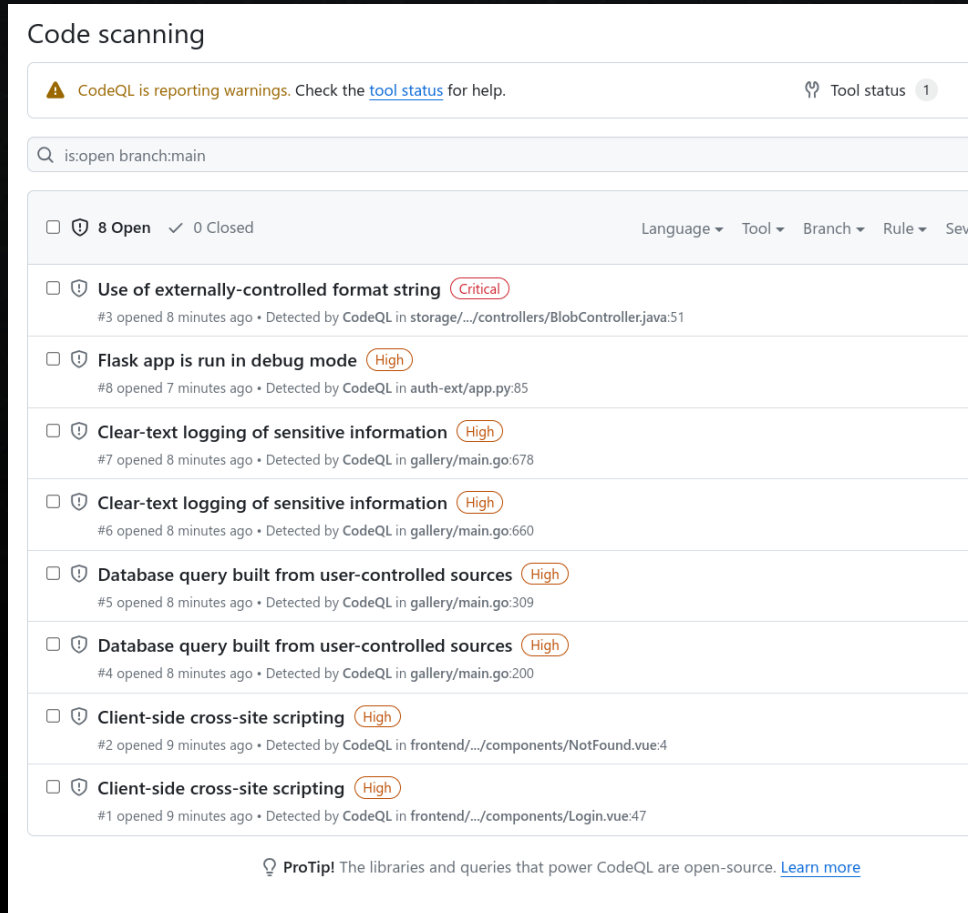


SARIF Output



CodeQL

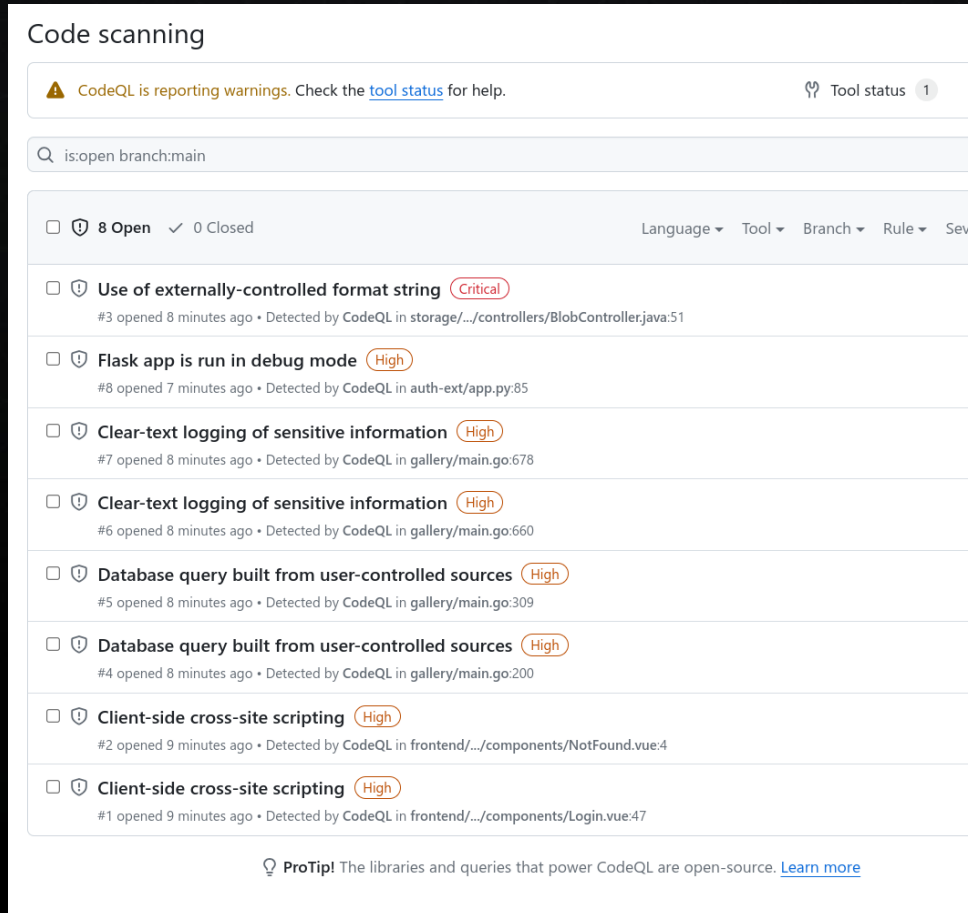
- CodeQL 將代碼視為資料，通過創建代碼資料庫並在其上運行查詢實現處理
- 查詢套件：代碼掃描（預設）、安全擴展型、安全與品質型
- 運行 CodeQL 的三種主要方式：
  - 預設設定（自動偵測語言、查詢與觸發器 ）
  - 進階配置（自定義 GitHub Actions 工作流）
  - 在外部 CI/CD 管道中使用 CodeQL 命令行介面，並將結果上傳至 GitHub



預設設置

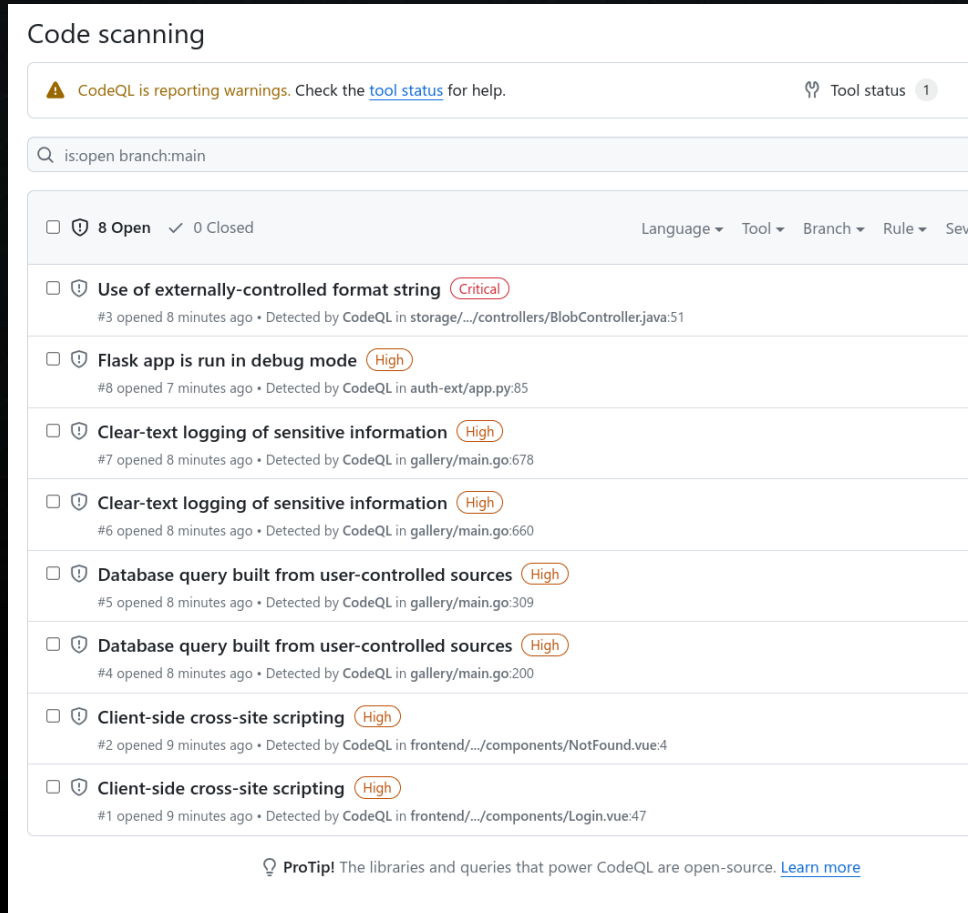
- 預設設定觸發掃描：
  - 每次向預設分支或任何受保護分支推送代碼時
  - 創建或提交基於預設分支或受保護分支的拉取請求時
  - 每周定時執行

- 預設設定適用條件：
  - 倉庫包含至少一種 CodeQL 支援的語言
  - 已啟用 GitHub Actions（支援 GitHub 託管和自託管執行器）
  - 倉庫為公開狀態或已啟用 GitHub 高級安全功能



高級設置

- GitHub Actions 工作流程 - YAML 檔
- 檔中的主要元件
  - 觸發點
    - Push
    - Pull request
    - Schedule（默认分支）
    - 手動觸發
    - Path-ignore（when not what）
  - 操作系統：Linux、Windows
  - CodeQL Action Init - 建立 CodeQL 資料庫、輸入查詢套件和語言
  - 手動輸入自定義構建步驟
  - CodeQL 動作分析 —— 在資料庫上運行 CodeQL 查詢



CodeQL CLI

- 獨立的命令行工具
- 生成分析結果並上傳到 GitHub 的 3 個命令：
  - *codeql database create*
  - *codeql database analyze*
  - *codeql github upload results*

```
# Create CodeQL databases for Java and Python in the 'codeql-dbs' directory
# Call the normal build script for the codebase: 'myBuildScript'

codeql database create codeql-dbs --source-root=src \
    --db-cluster --language=java,python --command=./myBuildScript

# Analyze the CodeQL database for Java, 'codeql-dbs/java'
# Tag the data as 'java' results and store in: 'java-results.sarif'

codeql database analyze codeql-dbs/java java-code-scanning.qls \
    --format=sarif-latest --sarif-category=java --output=java-results.sarif

# Analyze the CodeQL database for Python, 'codeql-dbs/python'
# Tag the data as 'python' results and store in: 'python-results.sarif'

codeql database analyze codeql-dbs/python python-code-scanning.qls \
    --format=sarif-latest --sarif-category=python --output=python-results.sarif

# Upload the SARIF file with the Java results: 'java-results.sarif'
# The GitHub App or personal access token created for authentication
# with GitHub's REST API is available in the `GITHUB_TOKEN` environment variable.

codeql github upload-results \
    --repository=my-org/example-repo \
    --ref=refs/heads/main --commit=deb275d2d5fe9a522a0b7bd8b6b6a1c939552718 \
    --sarif=java-results.sarif

# Upload the SARIF file with the Python results: 'python-results.sarif'

codeql github upload-results \
    --repository=my-org/example-repo \
    --ref=refs/heads/main --commit=deb275d2d5fe9a522a0b7bd8b6b6a1c939552718 \
    --sarif=python-results.sarif
```

優化 CodeQL 性能

- **硬體**
  - 可使用 `--threads` 和 `--ram` 標誌優化運行 CodeQL 掃描的命令。
- **掃描目標**
  - 對於解釋型語言，你可以忽略某些與掃描無關的路徑（例如測試檔）。
  - 對於編譯型語言，檔忽略必須在編譯過程中完成。
- **構建模式**
  - 若應用程式構建耗時較長，編譯型語言分析可考慮使用 `build-mode: none` 以提升掃描效率。
- **計劃掃描**
  - 若在 PR 階段進行掃描過於繁瑣，可考慮使用 GitHub Actions 計劃觸發器 實現夜間定時掃描。
- **查詢選擇**
  - 為了縮短分析時間，考慮在PR掃描中減少查詢數量。

單倉庫分析

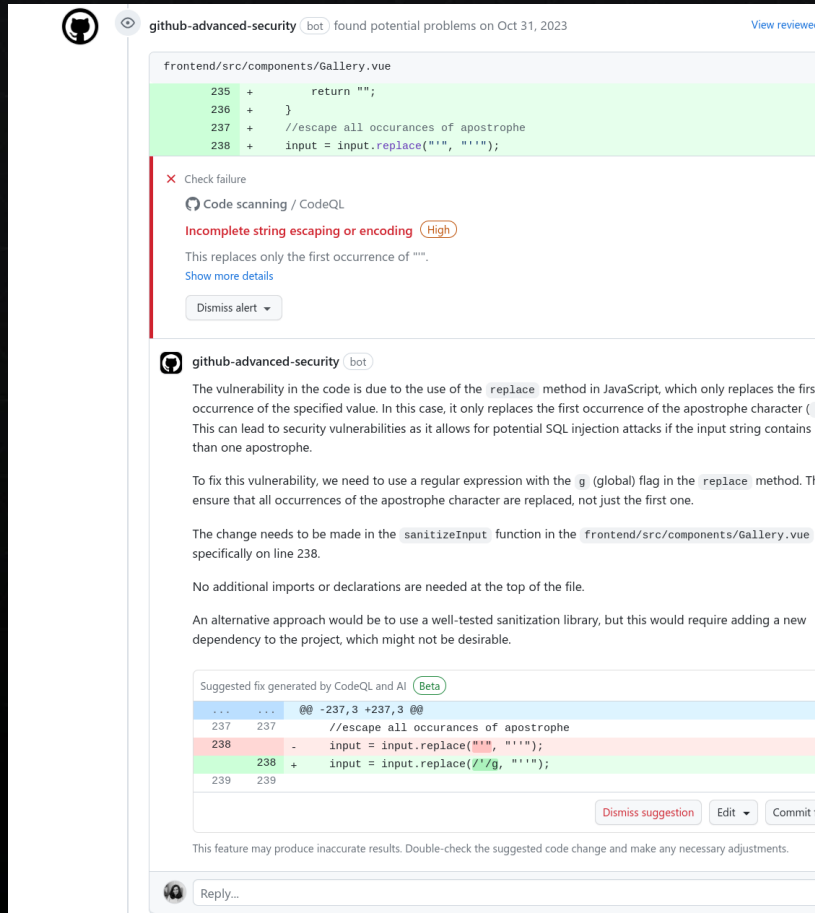
- 使用 *category* 來區分同一工具和提交的多個分析，但這些分析在不同語言或代碼的不同部分上進行。 您在工作流程中指定的類別將包含在 SARIF 結果檔中

- 支援將單倉庫拆分為清晰元件的場景。如果無法做到，可以考慮使用定時掃描

```yaml
- name: Perform CodeQL Analysis
  uses: github/codeql-action/analyze@v3
  with:
    # Optional. Specify a category to distinguish between multiple analyses
    # for the same tool and ref. If you don't use `category` in your workflow,
    # GitHub will generate a default category name for you
    category: "my_category"
```

自動修正

- Copilot Autofix 由 GitHub Copilot 提供支援，通過提供自動、有針對性的建議來解決代碼掃描警報，從而增強 CodeQL 的功能。
- 它利用大型語言模型 （LLMs） 基於代碼庫數據和 CodeQL 分析生成潛在修復方案，說明預防新的安全漏洞。
- 通過 OpenAI 的 GPT-4o 將警報描述和位置轉換為可操作的代碼更改。
- 默認啟用於使用 CodeQL 的倉庫；可在企業、組織或倉庫級別禁用。
- 自動修复支持 C#、C/C++、Go、Java/Kotlin、Swift、JavaScript/TypeScript、Python 和 Ruby。



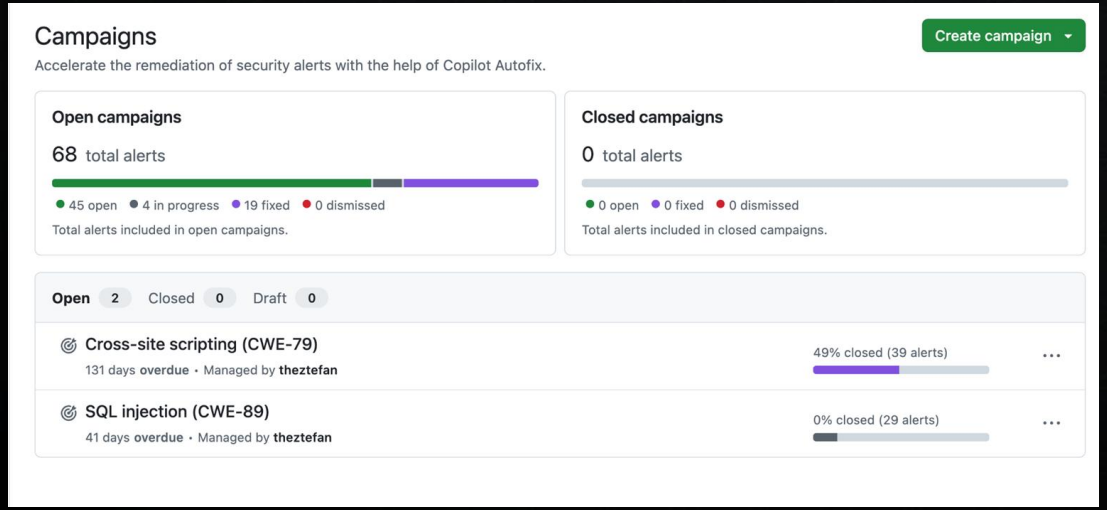
Security campaigns

這是什麼？

允許安全團隊將最高優先順序的安全警報分組，並設定修復時間的預期。

目標

通過促進安全團隊與開發者之間的協作，並藉助 Copilot Autofix，消除歷史上的安全債務。



Demo

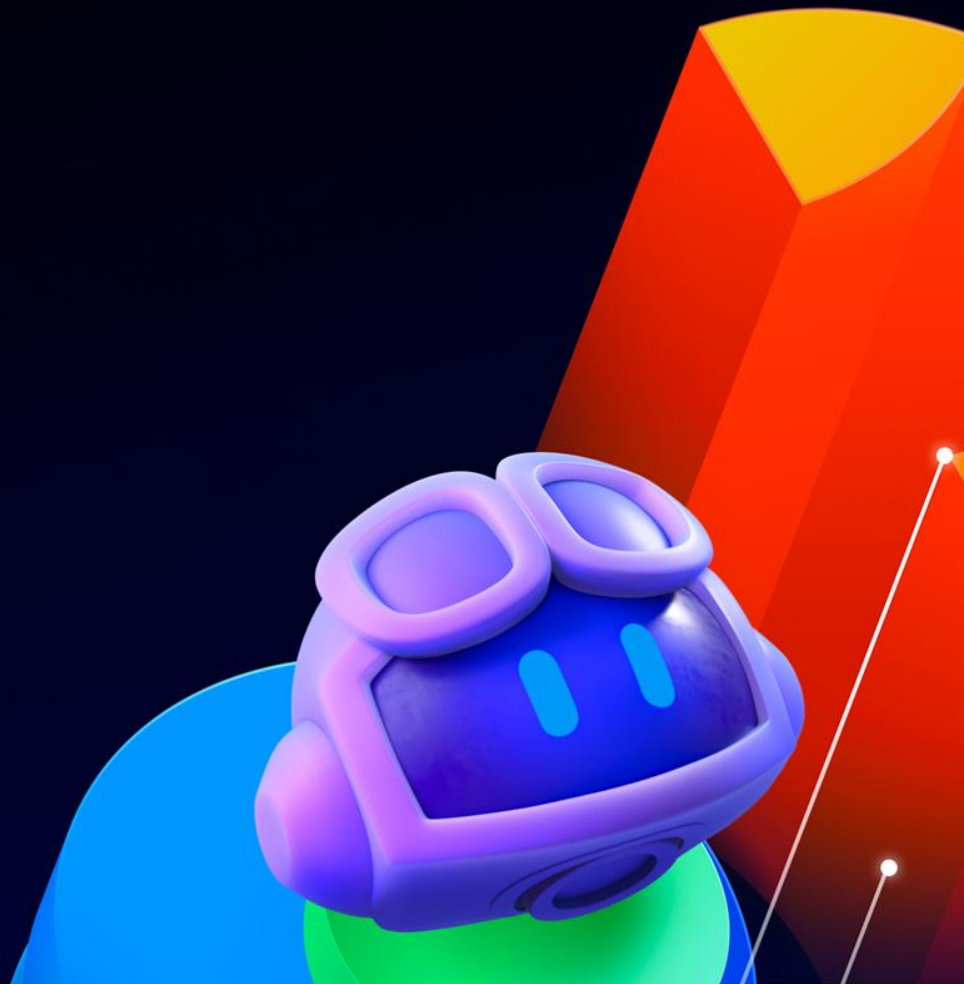
Module 3: Lab exercises

Questions?

Break!

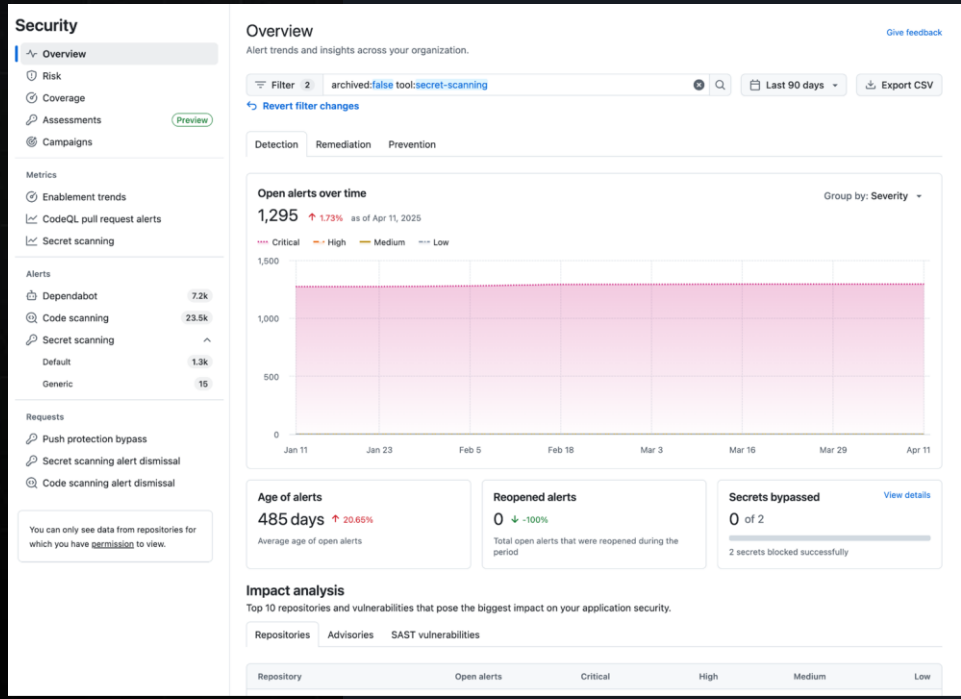
GitHub 高級安全

安全概述



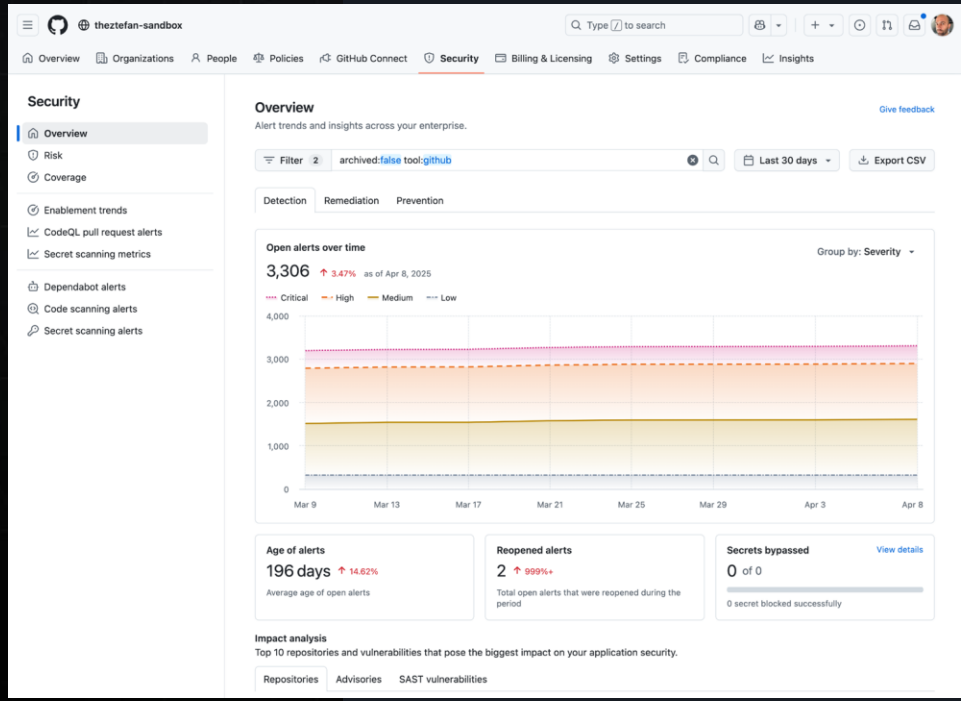
安全概述

- 提供組織或企業安全態勢的高級概覽，便於快速識別需要干預的代碼庫
- 警報訪問基於許可權控制，需具備寫入代碼庫許可權方可查看代碼掃描警報



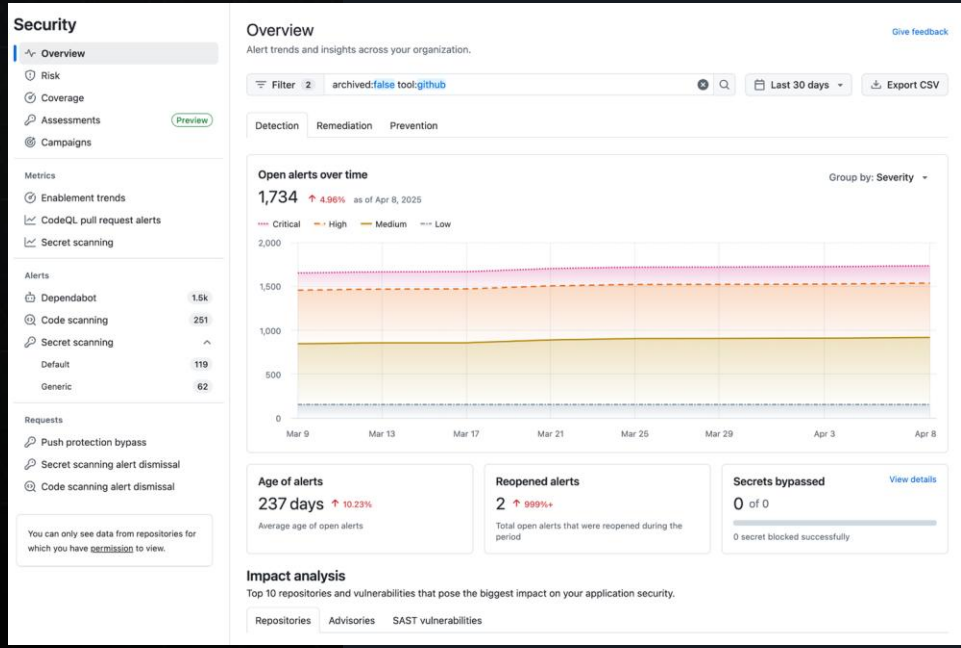
企業層級

- 企業安全選項卡包含與高級安全相關的原生儀錶板
- “CodeQL 拉取請求警報”選項卡顯示有關 Copilot 自動修復有效性的更多詳細資訊
- “代碼掃描警報”選項卡顯示所有組織中所有警報的匯總資訊
- 你只能看到你已經有許可權訪問的警報數據



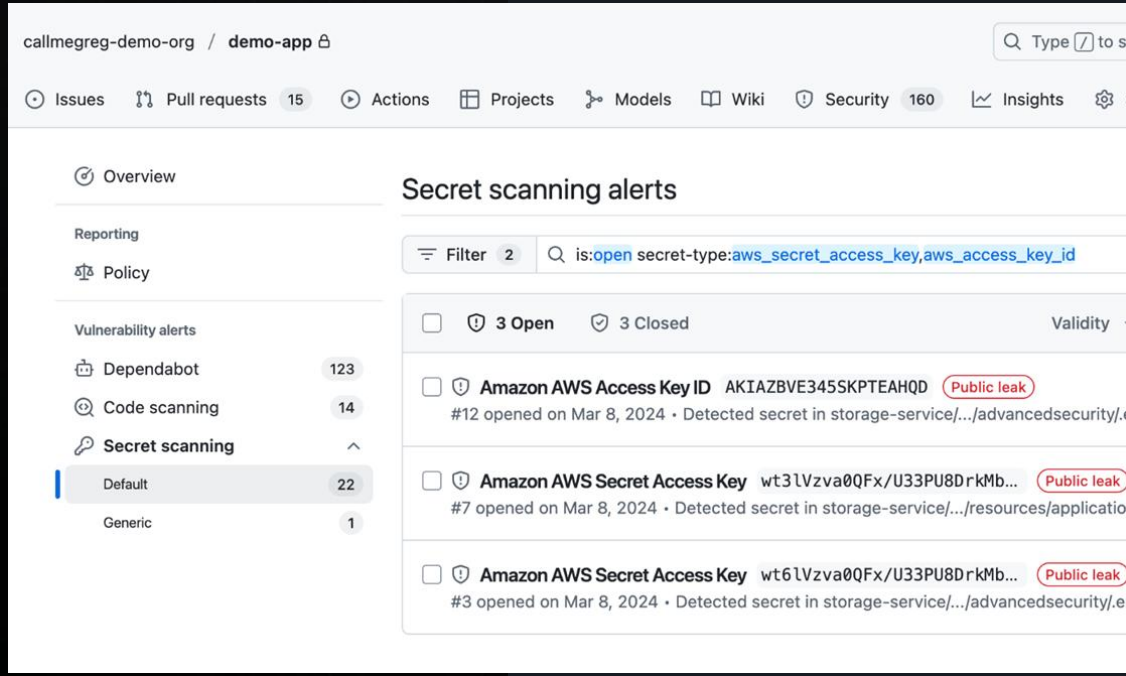
組織層級

- 在組織層面，查看整體風險狀況及跨代碼庫的賦能覆蓋範圍
- “Campaigns”選項卡展示當前安全活動的進展情況
- “Code scanning alert dismissal”標籤顯示了關於請求警報狀態變更的詳細資訊



倉庫級別

- 在存儲庫層級，根據有效性檢查、金鑰類型或供應商對警報進行審查和篩選。
- 默認情況下，僅存儲庫管理員、組織擁有者和安全管理員可見密鑰掃描選項卡。



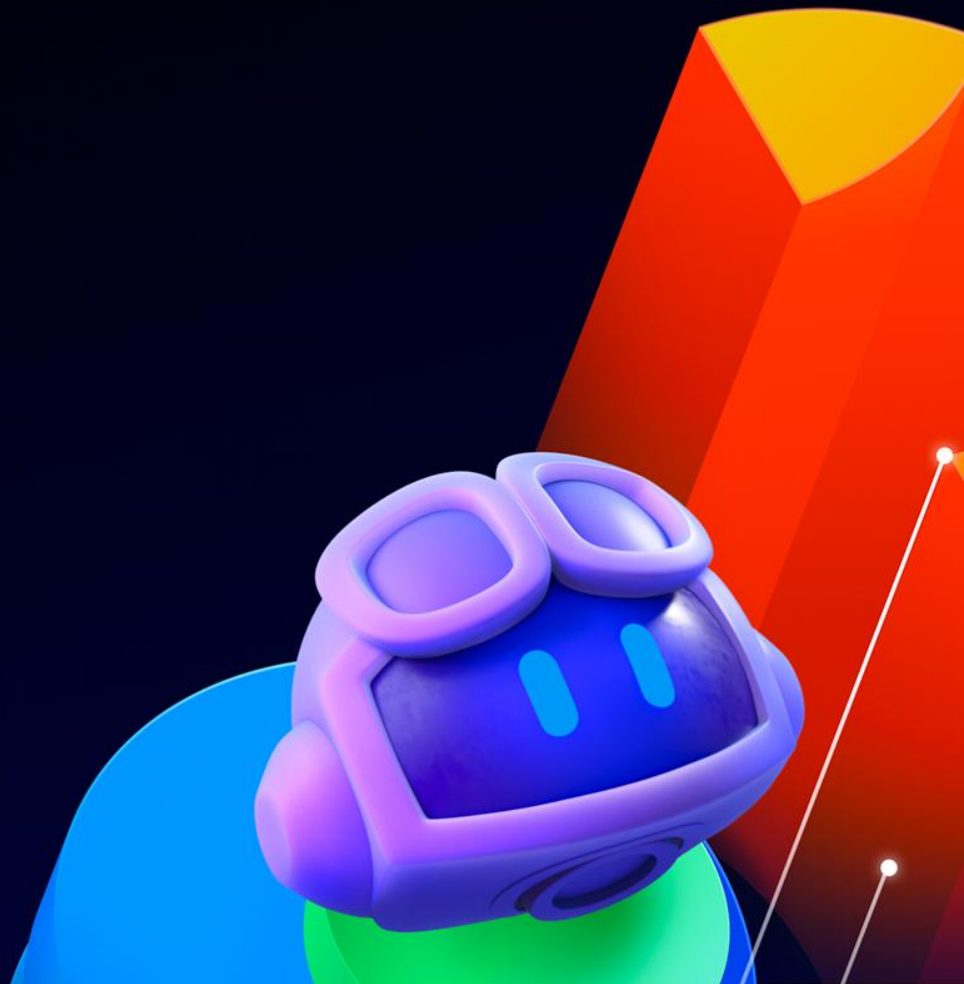
Demo

Module 3: Lab exercises

Questions?

GitHub 高級安全

第三方集成



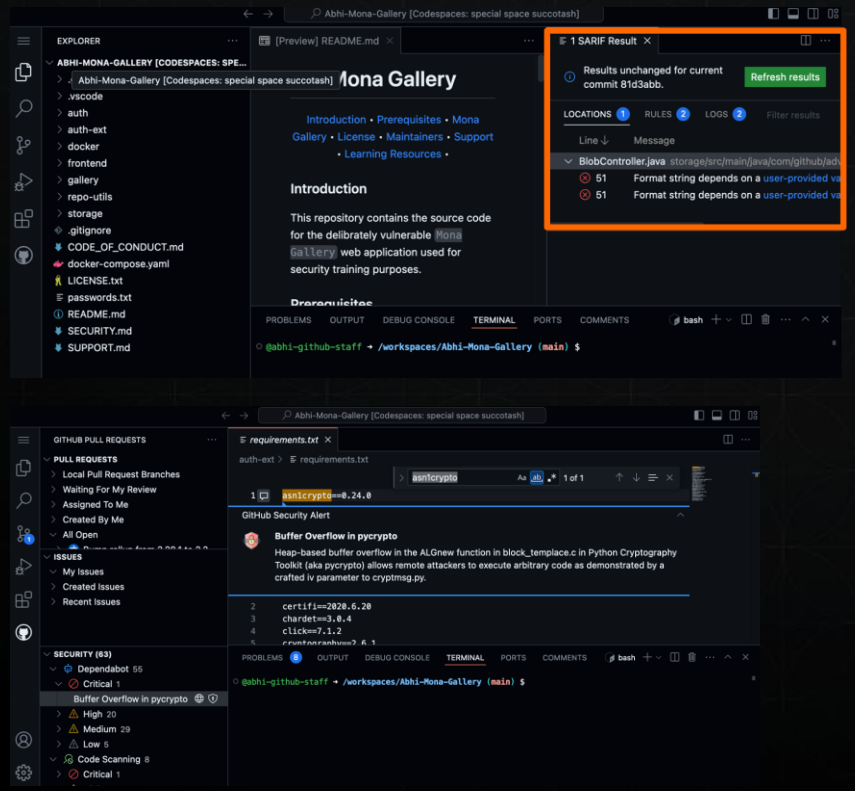
VS Code 集成

Sarif Viewer

開發者可以通過 SARIF Viewer 外掛程式在他們的 VS Code IDE 中查看 CodeQL 漏洞

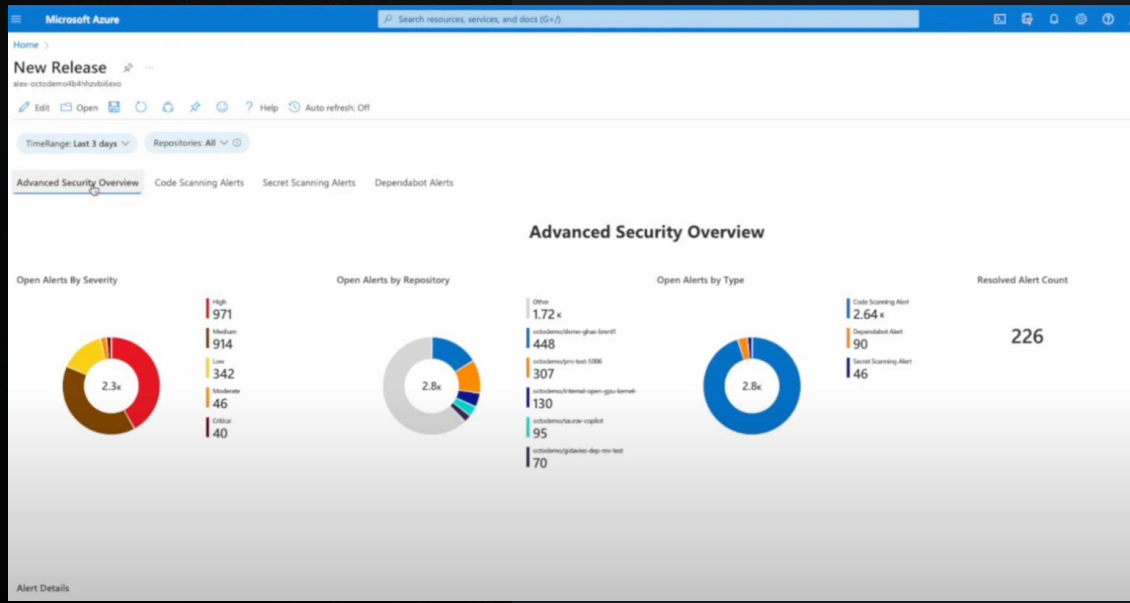
GitHub Security Alerts

另一款第三方 VS Code 擴展是 GitHub Security Alerts 外掛程式，可將 GHAS 漏洞資訊導入 VS Code IDE 中



SIEM 集成

- Splunk
- Microsoft Sentinel
- DataDog
- Sumo Logic
- Elastic Security
- Panther



Jira 集成

Jira 現已原生支援將 GHAS 漏洞導入系統，並自動在 Jira 中創建相應問題。

Integrate GitHub Advanced Security with Jira

These instructions are for connecting **GitHub Cloud** or **GitHub Enterprise Cloud** to Jira. Show me how to connect GitHub Enterprise Server

The security feature in Jira allows you to view, triage, and track security vulnerabilities from GitHub Advanced Security. To get this feature working, you'll need to:

1. Install the GitHub for Jira app.
2. Connect a GitHub organization.
3. Add GitHub Advanced Security to your Jira project.
4. Connect security containers to your project.

Before you begin

To install and set up the GitHub for Jira app, you need:

- Site administrator permission for your Jira site.
- Organization owner permission for your GitHub organization.

For some organizations, the task of integrating GitHub Advanced Security might involve multiple team members:

- A Jira site admin will install the GitHub for Jira app.
- A GitHub organization owner will connect a GitHub organization to your Jira site.
- A Jira project admin will add GitHub Advanced Security to a project and connect security containers.

第三方工具

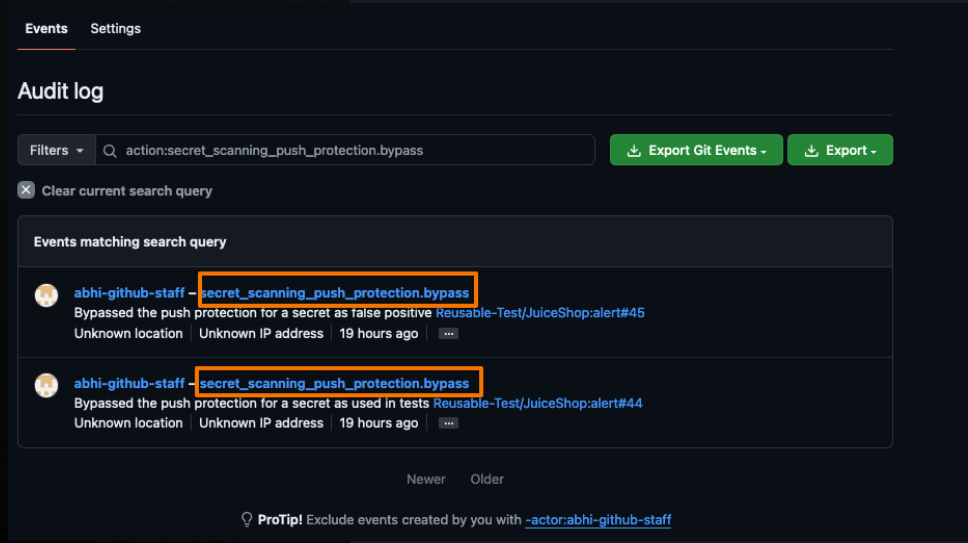
- 代碼掃描允許您整合其他靜態分析安全工具的結果。
- 輸出數據格式應為 SARIF
- 如果在 GitHub Actions 中運行掃描（例如 IaC），請使用 *upload-sarif*
- 上傳也可以通過 REST API 完成
- 漏洞可在 GitHub 的安全概述中查看

Code Blame 43 lines (37 loc) · 1.09 KB

```yaml
name: scan with KICS and upload SARIF

on:
  push:
    branches: [master]

jobs:
  kics-job:
    runs-on: ubuntu-latest
    name: kics-action
    strategy:
      fail-fast: false
    steps:
      - name: Checkout repo
        uses: actions/checkout@v3
      - name: Mkdir results-dir
        # make sure results dir is created
        run: mkdir -p results-dir

      - name: Run KICS Scan with SARIF result
        uses: checkmarx/kics-github-action@v2.1.0
        with:
          path: 'terraform'
          output_path: results-dir
          platform_type: terraform
          output_formats: 'json,sarif'
          ignore_on_exit: results

      - name: Show results
        run: |
          cat results-dir/results.sarif
          cat results-dir/results.json

      - name: Archive code coverage results
        uses: actions/upload-artifact@v4
        with:
          name: result
          path: results-dir/results.sarif

      - name: Upload SARIF file
        uses: github/codeql-action/upload-sarif@v1
        with:
          sarif_file: results-dir/results.sarif
```

審計日誌

- dependabot_*
- dependency_graph_*
- repository_secret_scanning*
- repository_vulnerability_*
- *secret_scanning_*



代碼安全審計日誌

code_scanning.alert_*
記錄代碼掃描警報的創建、關閉、撤銷、重新打開等操作。

org|repo.code_scanning_*
記錄代碼掃描相關配置的變更或代碼掃描分析的刪除。

security_configuration*
記錄組織或企業層級安全配置的創建、刪除或更新。

org.codeql*
記錄代碼掃描預設設置啟用或禁用時的日誌。

business.code_scanning|code_security*
記錄企業級 GHAS 策略的任何变更。

Webhook

code_scanning_alert
儲存庫、組織、應用程式

dependabot_alert
倉庫、組織、應用程式

secret_scanning_alert
儲存庫、組織、應用程式

secret_scanning_alert_location
儲存庫、組織、應用程式

security_advisory
應用程式

security_and_analysis
儲存庫、組織、應用程式

代碼掃描 Webhook

code_scanning_event
儲存庫中與代碼掃描警報相關的活動事件：創建、關閉、忽略、重新打開、出現在分支中

dismissal_request_code_scanning
使用者請求忽略警報時的事件

security_and_analysis
為存儲庫啟用或禁用代碼安全與分析功能時的事件

APIs

REST

/代碼掃描
企業、組織與儲存庫

/dependabot
企業、組織與儲存庫

/依賴圖
儲存庫

/secret-scanning
企業、組織與儲存庫

GraphQL

Dependabot 更新

安全漏洞

安全公告

預覽：DependencyGraph

代碼掃描 API

REST

- 管理現有警報
- 管理自動修復
- 管理分析
- 管理 CodeQL 資料庫
- 管理 CodeQL 變體分析
- 上傳 SARIF 結果

GraphQL

- 目前沒有代碼掃描支援

Questions?

GitHub 代碼安全培訓

回顧

GitHub 代碼安全

- 安全漏洞短期內不會消失
- 安全配置允許您**自定義部署過程**
- CodeQL 說明你在**漏洞進入生產環境前發現它們**
- 預設設定能讓你最快獲得最多**覆蓋**
- 通過高級設置**提升精度和性能**
- 自動修復和安全活動可說明您**更快解決檢測結果**
- 安全概覽包含**原生儀錶板和洞察分析**
- API、webhook 和審計日誌可**匯出警報數據**

接下來要做的三件事！

- [x] **制定計劃**

  確定何時以及如何為您的代碼庫推出代碼安全功能。

- [x] **與開發者溝通**

  讓開發人員瞭解何時會推出哪些功能，以及對他們的具體要求。

- [x] **當需要時進行自定義**

  開始優化您代碼掃描分析的必要部分。


Thank you