WEB SECURITY

November 2025

NAHCO3.FR

TLP:AMBER+STRICT

CI/CD SECURITY - WHAT IS CI/CD?

- CI (CONTINUOUS INTEGRATION): FREQUENT INTEGRATION OF CODE INTO A SHARED REPOSITORY, FOLLOWED BY AUTOMATED BUILDS AND TESTS
- CD (CONTINUOUS DEPLOYMENT): AUTOMATIC RELEASE OF VALIDATED CHANGES TO PRODUCTION
- AUTOMATE THE PROCESS OF BUILDING, TESTING, AND DEPLOYING SOFTWARE

CI/CD SECURITY - PROACTIVE SECURITY

- TRADITIONAL PATCHING IS SLOW → DETECTION → TICKET → PATCH → VERIFY
- CI/CD PIPELINES = DETECT → PATCH → TEST → DEPLOY → VERIFY - ROLLBACK POSSIBLE
- EXAMPLE: EQUIFAX BREACH (2017) – PATCH WAS AVAILABLE 2 MONTHS BEFORE THE ATTACK (LINK TO APACHE STRUTS)
- SECURITY CI/CD = FASTER MITIGATION, CONTINUOUS VALIDATION

CI/CD SECURITY - TOOLS WE WILL USE

- GITHUB ACTIONS: AUTOMATE SECURITY TESTINGS, DEPLOYMENT OF DETECTION RULES, INCIDENT RESPONSE WORKFLOWS - LINK TO A GITHUB REPO
- GITHUB PAGES FOR HOSTING OUR WEB PAGE
- DOCKER FOR CONTAINERIZATION
- TRIVY: SCAN CODE, CONTAINER IMAGES, AWS, KUBERNETES AND FILE SYSTEMS FOR VULNS

CI/CD SECURITY - ANATOMY OF A GITHUB ACTIONS

- WORKFLOW: AUTOMATION RECIPE (.GITHUB/WORKFLOWS/)
- TRIGGERS: DEFINES WHEN THE WORKFLOW RUNS
- JOBS: MADE OF STEPS
- STEPS: PERFORM AN ACTION LIKE CHECKOUT, SCAN, DEPLOY
- RUNNERS: ENVIRONMENTS WHERE THE JOBS ARE EXECUTED LIKE UBUNTU

```yaml
on:
  push:
    branches:
      - main
  pull_request:
    branches:
      - main

jobs:
  hello-world:
    runs-on: ubuntu-latest
    steps:
      - name: Hello world
        run: echo "Hello world"
```

CI/CD SECURITY - GITHUB ACTIONS CONTEXT

- GITHUB.EVENT_NAME: NAME OF THE EVENT THAT TRIGGERED THE WORKFLOW
- GITHUB.EVENT.PULL_REQUEST.NUMBER: NUMBER OF THE PULL REQUEST THAT TRIGGERED THE WORKFLOW
- GITHUB.REPOSITORY: OWNER AND REPOSITORY NAME

EXPRESSION EXAMPLE:

```
- NAME: CHECK PULL REQUEST
  IF: ${{ GITHUB.EVENT_NAME == 'PULL_REQUEST' &&
GITHUB.EVENT.PULL_REQUEST.DRAFT == FALSE }}
  RUN: ECHO "THIS IS A NON-DRAFT PULL REQUEST."
```

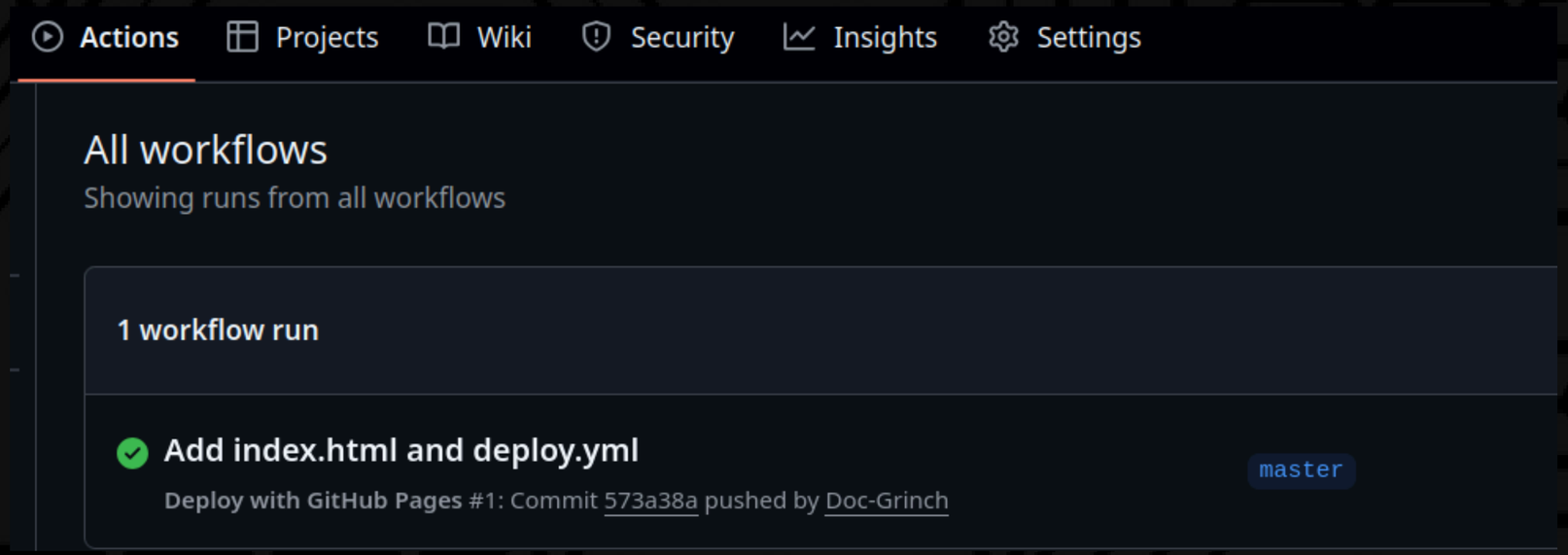
CI/CD SECURITY - GITHUB TOKENS

- GITHUB ACTIONS WORKFLOW: ACCESS TO A GITHUB_TOKEN GENERATED WHEN WORKFLOW RUNS
- CAN AND NEED TO ADJUST PRIVILEGES
- GIVES WORKFLOW PERMISSIONS TO ACCESS AND MODIFY PARTS OF THE REPOSITORY

```yaml
permissions:
  contents: read # Read-only access to repository content
  pull-requests: write # Write access to pull requests
  actions: none # No access to actions
```

CI/CD SECURITY - GITHUB ACTIONS - EXAMPLE

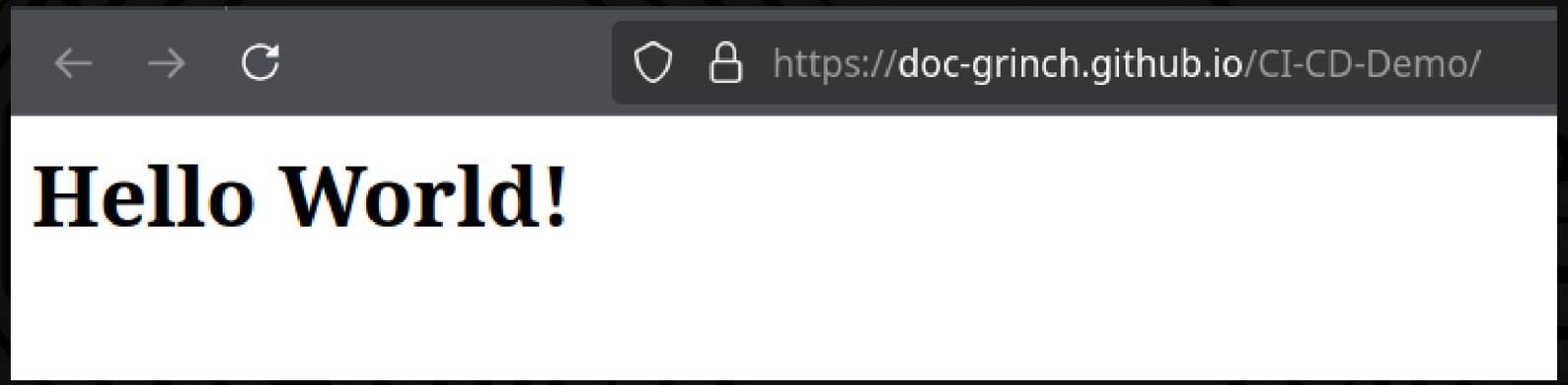
NEED TO ACTIVATE THE GITHUB PAGES
GO TO THE SETTINGS TAB AND SCROLL DOWN TO THE GITHUB PAGES SECTION. THEN SELECT UNDER "BUILD AND DEPLOYMENT", THE "SOURCE": "GITHUB ACTIONS"



TLP:AMBER+STRICT

CI/CD SECURITY - GITHUB ACTIONS - EXAMPLE

HTTPS://DOC-GRINCH.GITHUB.IO/CI-CD-DEMO/



TLP:AMBER+STRICT

CI/CD SECURITY - TRIVY

KEY CAPABILITIES:

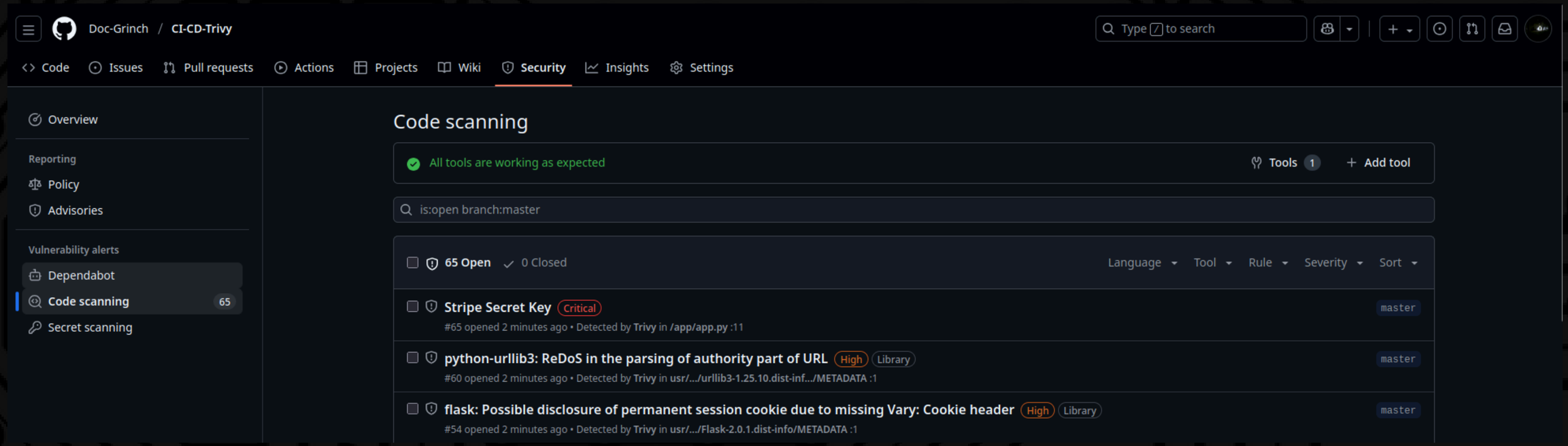
- SCANS CODE, CONTAINERS, IAC, DEPENDENCIES, AND SECRETS
- DETECTS VULNERABILITIES, HARDCODED CREDENTIALS, LICENSE ISSUES

CODE SNIPPET HIGHLIGHTING VULNERABILITIES:

- HARDCODED PASSWORD
- COMMAND INJECTION
- PATH TRAVERSAL
- INSECURE COOKIES
- DEBUG MODE

```yaml
- name: Run Trivy
  uses: aquasecurity/trivy-action@...
  with:
    scan-type: fs
    scan-ref: .
    severity: CRITICAL,HIGH
```

CI/CD SECURITY - TRIVY - EXAMPLE



TLP:AMBER+STRICT

CI/CD SECURITY - BUILDING SECURE PIPELINES

- AUTOMATE TESTING AND PATCHING
- INTEGRATE SECURITY TOOLS LIKE TRIVY
- SET MINIMAL PERMISSIONS (PRINCIPLE OF LEAST PRIVILEGE)
- TEST SAFELY, DEPLOY CONFIDENTLY
- CONTINUOUSLY IMPROVE PIPELINES