

Project Overview

This project involves setting up a **Secure CI/CD pipeline** for a chosen open-source application. The pipeline integrates multiple security tools to ensure that best practices in security and software development are followed. The pipeline utilizes tools such as **GitHub Actions**, **SAST**, **DAST**, **SCA**, **Container Security**, and **IaC security scanninG , SIEM & Vulnerability Management** to perform static and dynamic analysis, enforce secure coding practices, and ensure the security of the deployed containers.

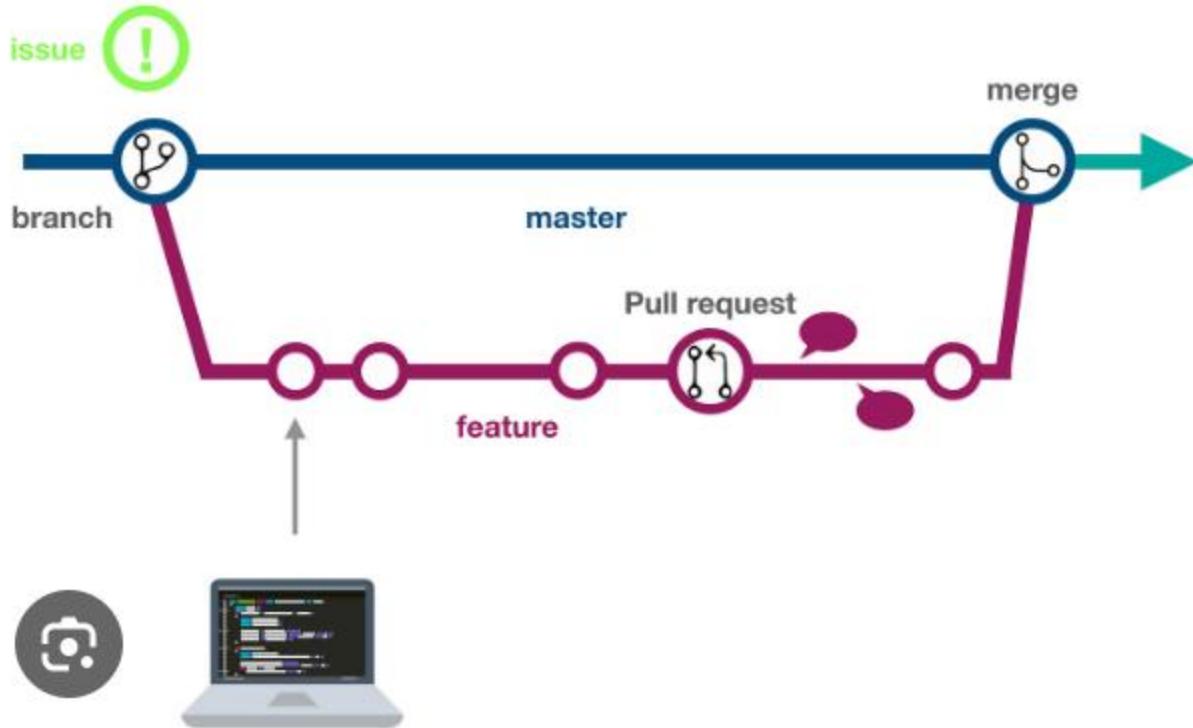
1. Repo : Owasp - Juice Shop :

https://github.com/BilalBaree/Owasp_JuiceShop



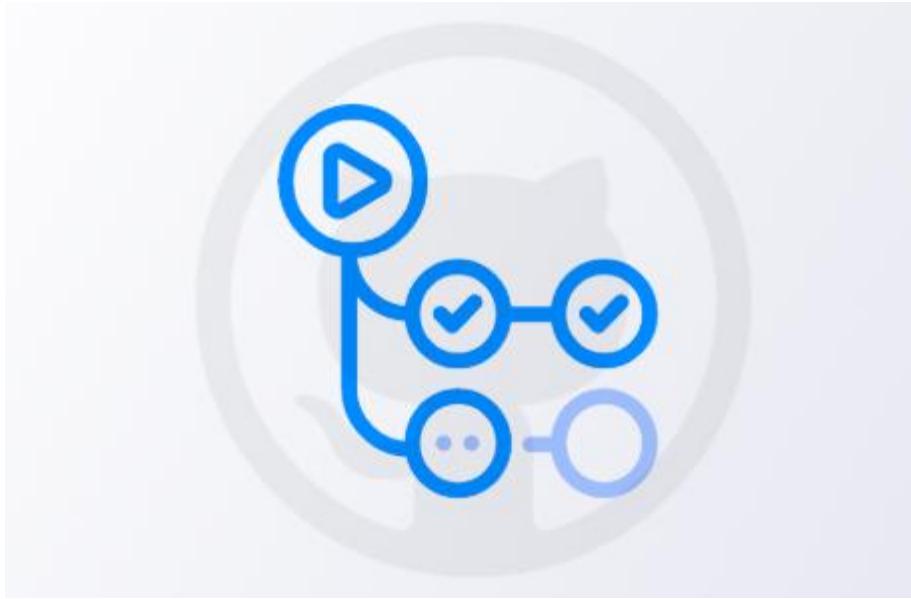
2. Git Branching Strategy

- **Git Strategy Chosen: GitHub Flow**
 - **Justification:** GitHub Flow is simple, and its focus on pull requests aligns well with the goals of enforcing security checks and code review mechanisms. This strategy ensures that every change to the codebase goes through a review process before being merged, which allows for better control over the quality and security of the code.



3. CI Tool: GitHub Actions

- **CI Tool Chosen: GitHub Actions**
 - **Justification:** GitHub Actions provides seamless integration with GitHub repositories, allowing for automatic workflows for testing, building, and deploying code. It is highly customizable and allows the integration of various security tools into the CI/CD pipeline, which fits the project requirements.



-

4. Security Checks in CI

The following security checks are integrated into the CI pipeline to ensure that the code adheres to secure practices:

a. Code Reviews and Signed Commits

- **Enforce Code Reviews:** The pipeline ensures that code reviews are mandatory for pull requests before merging into the main branch.
- **Enforce Signed Commits:** Each commit in the pipeline must be signed to guarantee authenticity. This check ensures that only trusted contributors can make changes.

b. Role-Based Access Control (RBAC)

- **Access Control Enforcement:** The pipeline includes a check to verify that role-based access is configured correctly in the GitHub repository. It ensures that only authorized users have write access to critical branches.

c. Audit Logging

- **Pipeline Execution Logs:** Each pipeline run logs its execution details to an audit log, helping track security-related activities and actions.

```
#      auto-approve:
# runs-on: ubuntu-latest
# if: github.event_name == 'pull_request'
# steps:
#   - name: Check if PR author is trusted and target is main
#     id: check
#     run: |
#       echo "PR author: ${{ github.event.pull_request.user.login }}"
#       echo "Target branch: ${{ github.event.pull_request.base.ref }}"

#       if [[ "${{ github.event.pull_request.user.login }}" ==
"trusted-user" && "${{ github.event.pull_request.base.ref }}" == "main" ]];
then
#         echo "approve=true" >> $GITHUB_OUTPUT
#       else
#         echo "approve=false" >> $GITHUB_OUTPUT
#       fi
```

```

- name: Enforce code reviews (Only on pull request)
  if: github.event_name == 'pull_request'
  uses: hmarr/auto-approve-action@v2
  with:
    github-token: ${{ secrets.GITHUB_TOKEN }}

- name: Enforce signed commits
  run: |
    git log ${github.event.before}..${github.sha} --pretty=format:'%h %G? %an' | while read line; do
      commit_hash=$(echo $line | cut -d ' ' -f 1)
      commit_status=$(echo $line | cut -d ' ' -f 2)
      if [[ "$commit_status" != "G" ]]; then
        echo "✗ Commit $commit_hash is not signed"
        exit 1
      fi
    done

# 2. Role-based Access Control
access_control:
  runs-on: ubuntu-latest
  steps:
    - name: Check role-based access control
      run: |
        if [[ "$(curl -H "Authorization: Bearer ${secrets.GITHUB_TOKEN}" \
          "https://api.github.com/repos/${github.repository}/collaborators" | jq \
          '.[] | select(.permissions.push == true)' | wc -l)" -lt 1 ]]; then
          echo "✗ Role-based access is not configured properly."
          exit 1
        fi
  audit_log:
    runs-on: ubuntu-latest
    steps:
      - name: Log pipeline run
        run: |
          echo "$(date) - CI/CD pipeline executed for repo: \
            ${github.repository}" >> ./audit_log.txt
          cat ./audit_log.txt

      - name: Upload Audit Log

```

```
uses: actions/upload-artifact@v4
with:
  name: audit-log
  path: audit_log.txt
```

5. SAST (Static Application Security Testing)

The pipeline integrates two key **SAST** tools to identify vulnerabilities in the codebase:

a. TruffleHog (Secrets Scanning)

- **Purpose:** Detect hardcoded secrets, such as API keys or passwords.
- **Positive Case:** No secrets are found in the codebase, and the pipeline proceeds without issues.
- **Negative Case:** Secrets are found, and the pipeline fails, preventing potentially sensitive data from being exposed.

b. Bandit (SQL Injection Detection)

- **Purpose:** Detect potential SQL injection vulnerabilities within the code.
- **Positive Case:** No issues are found, and the pipeline proceeds.
- **Negative Case:** SQL injection vulnerabilities are detected, and the pipeline halts, requiring a fix before proceeding.

```
sast_analysis:
  runs-on: ubuntu-latest
  steps:
    - name: Checkout code
      uses: actions/checkout@v3

    - name: Set up Python
      uses: actions/setup-python@v5
      with:
        python-version: '3.x'

    - name: Install tools
      run: pip install bandit trufflehog jq

    - name: Secrets Scanning (TruffleHog)
      run:
        trufflehog --json . > secrets_scan_results.json || true
        if [[ $(jq length secrets_scan_results.json) -gt 0 ]]; then
```

```
    echo "✗ Secrets found!"
    cat secrets_scan_results.json
    exit 1
else
    echo "☑ No secrets found."
fi

- name: Upload TruffleHog Report
  uses: actions/upload-artifact@v4
  with:
    name: secrets-scan
    path: secrets_scan_results.json

- name: SQL Injection Detection (Bandit)
  run: |
    bandit -r . -f json -o bandit_report.json -t B107
    if grep -q "issue" bandit_report.json; then
        echo "✗ SQL Injection issues found!"
        cat bandit_report.json
        exit 1
    fi

- name: Upload Bandit Report
  uses: actions/upload-artifact@v4
  with:
    name: bandit-report
    path: bandit_report.json
```

6. DAST (Dynamic Application Security Testing)

Although DAST tools are not fully implemented, **SQLMap** and **OWASP ZAP** are integrated into the pipeline to check for runtime vulnerabilities:

a. SQLMap (SQL Injection Detection)

- **Purpose:** Identify SQL injection vulnerabilities in the deployed application.
- **Positive Case:** No SQL injection issues are found, and the pipeline proceeds.
- **Negative Case:** SQL injection vulnerabilities are identified, and the pipeline fails, preventing deployment.

b. OWASP ZAP (Cross-Site Scripting Detection)

- **Purpose:** Scan for cross-site scripting (XSS) vulnerabilities in web applications.
- **Positive Case:** No XSS vulnerabilities are detected.
- **Negative Case:** XSS vulnerabilities are found, and the pipeline fails until the issues are resolved.

```
dast_scan:
  runs-on: ubuntu-latest
  steps:
    - name: Checkout code
      uses: actions/checkout@v3

    - name: Set up Docker Buildx
      uses: docker/setup-buildx-action@v2

    - name: Pull and start Juice Shop container
      run: |
        docker pull bkimminich/juice-shop
        docker run --rm -d -p 3000:3000 bkimminich/juice-shop
        # Wait for the application to be ready
        for i in {1..60}; do
          if curl --silent --fail http://localhost:3000; then
            echo "Juice Shop is ready."
            break
          fi
        done
```

```
        else
            echo "Waiting for Juice Shop to start..."
            sleep 5
        fi
    done

- name: Install tools
  run: |
    pip install sqlmap
    sudo apt-get update
    sudo apt-get install -y openjdk-11-jre python3
    wget
https://github.com/zaproxy/zaproxy/releases/download/v2.16.1/ZAP\_2.16.1\_Lin
ux.tar.gz
    tar -xzf ZAP_2.16.1_Linux.tar.gz
    sudo mv ZAP_2.16.1 /opt/zaproxy
    sudo ln -s /opt/zaproxy/zap.sh /usr/local/bin/zap

- name: Run SQLMap for SQLi
  run: sqlmap -u "http://localhost:3000" --batch --risk=3 --level=5

- name: Run OWASP ZAP for XSS
  run: zap -cmd -quickurl http://localhost:3000 -quickout
zap_report.html

- name: Upload ZAP Report
  uses: actions/upload-artifact@v4
  with:
    name: zap-report
    path: zap_report.html
```

7. SCA (Software Composition Analysis)

The pipeline uses **Snyk** to scan for vulnerable third-party dependencies:

a. Snyk (Vulnerable Packages Scan)

- **Purpose:** Scan the code for insecure third-party dependencies.
- **Positive Case:** No critical vulnerabilities are found, allowing the pipeline to proceed.
- **Negative Case:** Critical vulnerabilities are found, and the pipeline halts to fix the issues.

b. Signed Libraries Verification

- **Purpose:** Ensure that libraries used in the project are signed to prevent tampering.
- **Positive Case:** All libraries are signed, and the pipeline proceeds.
- **Negative Case:** Unsigned libraries are detected, and the pipeline fails.

```
# 5. SCA Tool: Snyk
sca_scan:
  runs-on: ubuntu-latest
  steps:
    - name: Checkout code
      uses: actions/checkout@v3

    - name: Set up Node.js
      uses: actions/setup-node@v3
      with:
        node-version: '18.19.0'

    - name: Install dependencies
      run: npm install

    - name: Scan for vulnerable packages (Snyk)
      uses: snyk/actions/node@master
      with:
        args: test . --severity-threshold=high
      env:
        SNYK_TOKEN: ${{ secrets.SNYK_TOKEN }}
```

```
- name: Use signed libraries verification
  run: bash scripts/verify-signatures.sh
  continue-on-error: true
```

8. Container Security

The pipeline integrates security scanning for containers, specifically focusing on Docker images:

a. Clair (Root Access Check)

- **Purpose:** Scan the Docker image for root access privileges.
- **Positive Case:** No root access is found, and the pipeline continues.
- **Negative Case:** Root access is detected, and the pipeline fails.

b. Trivy (Read-Only File System)

- **Purpose:** Ensure that containers use read-only file systems for security.
- **Positive Case:** The container adheres to the read-only file system policy.
- **Negative Case:** Violations are found, and the pipeline fails until addressed.

```
# 7. Container Security Scanning
container-security-scan:
  runs-on: ubuntu-latest
  steps:
    - name: Checkout code
      uses: actions/checkout@v3

    - name: Set up Docker Buildx
      uses: docker/setup-buildx-action@v2

    - name: Build Docker image
      run: |
        docker build -t ${{ secrets.DOCKERHUB_USERNAME }}/juice-
shop:latest .

    - name: Scan Docker image for Root Access (Clair)
      run: |
        curl -sL
        https://github.com/coreos/clair/releases/download/v2.0.9/clairctl-linux-
        amd64 -o clairctl
        chmod +x clairctl
```

```
sudo mv clairctl /usr/local/bin/
clairctl analyze --local --image
${{ secrets.DOCKERHUB_USERNAME }}/juice-shop:latest
    clairctl report ${{ secrets.DOCKERHUB_USERNAME }}/juice-
shop:latest
        continue-on-error: true

    - name: Scan image for Read-Only File System violations (Trivy)
      run: |
        docker run --rm -v /var/run/docker.sock:/var/run/docker.sock -v
$(pwd):/workspace aquasecurity/trivy fs --severity CRITICAL,HIGH --exit-
code 1 --no-progress --ignore-unfixed .
        continue-on-error: true

    - name: Push Docker image to DockerHub
      if: success()
      run: |
        docker login -u ${{ secrets.DOCKERHUB_USERNAME }} -p
${{ secrets.DOCKERHUB_TOKEN }}
        docker push ${{ secrets.DOCKERHUB_USERNAME }}/juice-shop:latest

# 8. Generate and Upload Reports
upload-reports:
  runs-on: ubuntu-latest
  steps:
    - name: Upload Clair report
      uses: actions/upload-artifact@v4
      with:
        name: clair-report
        path: clair-report.html

    - name: Upload Trivy report
      uses: actions/upload-artifact@v4
      with:
        name: trivy-report
        path: trivy-report.json
```

9. IaC (Infrastructure as Code) Security

The pipeline integrates **Checkov** and **Conftest** to enforce best practices and security for infrastructure as code:

a. Checkov (Misconfigurations in IaC)

- **Purpose:** Scan Terraform configurations for misconfigurations.
- **Positive Case:** No misconfigurations are found.
- **Negative Case:** Misconfigurations are found, and the pipeline fails.

b. Conftest (Compliance and Least Privilege)

- **Purpose:** Ensure compliance with policies such as least privilege.
- **Positive Case:** Policy compliance is confirmed, and the pipeline continues.
- **Negative Case:** Policy violations are found, and the pipeline halts for remediation.

```
# 9. IaC Security Scan
iac-security-scan:
  runs-on: ubuntu-latest
  steps:
    - name: Checkout code
      uses: actions/checkout@v3

    - name: Set up Python
      uses: actions/setup-python@v4
      with:
        python-version: '3.x'

    - name: Install Terraform
      uses: hashicorp/setup-terraform@v2
      with:
        terraform_version: 1.4.6

    - name: Install Checkov
      run: pip install checkov
```

```
- name: Install Conftest
  run: |
    CONFTAG=$(curl -s https://api.github.com/repos/open-policy-
agent/conftest/releases/latest | grep tag_name | cut -d '"' -f 4)
    wget https://github.com/open-policy-
agent/conftest/releases/download/${CONFTAG}/conftest_${CONFTAG#v}_Linux_x86
_64.tar.gz
    tar -xzf conftest_${CONFTAG#v}_Linux_x86_64.tar.gz
    sudo mv conftest /usr/local/bin

- name: Run Checkov for misconfigurations
  run: |
    checkov -d ./terraform --skip-check CKV_AWS_355 --skip-check
CKV_SECRET_1

- name: Run Conftest with least privilege policy
  run: |
    conftest test terraform/main.tf --policy policies

- name: Check for secrets in Terraform using Checkov
  run: |
    checkov -d ./terraform --check CKV_SECRET_1
```

Integration Of DataDog (SIEM TOOL)

```
- name: Capture test results and send to Datadog
  env:
    DD_API_KEY: ${{ secrets.DATADOG_API_KEY }}
  run: |
    OUTPUT=$(npm test || true)
    curl -X POST "https://http-intake.logs.datadoghq.eu/v1/input" \
      -H "Content-Type: application/json" \
      -H "DD-API-KEY: $DD_API_KEY" \
      -d "{\"message\": \"$OUTPUT\", \"ddsource\": \"github-actions\", \
\"service\": \"ci-tests\"}"
```

Integration Of Vulnerability Management Tools (CodeQL & Dependabot)

CodeQL

```
name: "CodeQL Scan"

on:
  push:
  pull_request:

jobs:
  analyze:
    name: Analyze
    runs-on: ubuntu-latest
    permissions:
      actions: read
      contents: read
      security-events: write
    strategy:
      fail-fast: false
      matrix:
        language: [ 'javascript-typescript' ]
    steps:
      - name: Checkout repository
        uses: actions/checkout@11bd71901bbe5b1630ceea73d27597364c9af683
#v4.2.2
      - name: Initialize CodeQL
        uses: github/codeql-action/init@v3
        with:
          languages: ${{ matrix.language }}
          queries: security-extended
          config: |
            paths-ignore:
              - 'data/static/codfixes'
      - name: Autobuild
        uses: github/codeql-action/autobuild@v3
      - name: Perform CodeQL Analysis
        uses: github/codeql-action/analyze@v3
```

Dependabot

```
version: 1
update_configs:
  - package_manager: "javascript"
    directory: "/"
    update_schedule: "live"
    target_branch: "develop"
    default_reviewers:
      - "bkimminich"
    default_labels:
      - "dependencies"
  ignored_updates:
    - match:
        dependency_name: "express-jwt"
        version_requirement: "0.1.3"
    - match:
        dependency_name: "sanitize-html"
        version_requirement: "1.4.2"
    - match:
        dependency_name: "unzipper"
        version_requirement: "0.9.15"
    - match:
        dependency_name: "jsonwebtoken"
        version_requirement: "0.4.0"
  - package_manager: "javascript"
    directory: "/frontend"
    update_schedule: "live"
    target_branch: "develop"
    default_reviewers:
      - "bkimminich"
    default_labels:
      - "dependencies"
```

ScreenShots

Pipeline Execution:

dast_scan
succeeded now in 10m 45s

Upload ZAP Report

- Run actions/upload-artifact@v4
- Warning: No files were found with the provided path: zap_report.html. No artifacts will be uploaded.

Post Set up Docker Buildx

- Post: Job cleanup.
- Removing builder
- Cleaning up certificates

Post Checkout code

- Post job cleanup.
- git version 2.40.0
- Temporarily overriding HOME='/home/runnerrunner/work/_temp/5fe45e6-ad79-4ed7-bd87-c765b896679' before making global git config changes
- Adding repository directory to the temporary git global config as a safe directory
- /usr/bin/git config --global --add safe.directory /home/runnerrunner/work/0wasp_JuiceShop/0wasp_JuiceShop
- /usr/bin/git config --local --name-only --get-regexp core\\.sshCommand
- /usr/bin/git config --local --name-only --get-regexp http\\(https\\)://github\\.com/\\.extraheader
- /usr/bin/git config --local --name-only --get-regexp http\\(https\\)://github\\.com/\\.extraheader
- http:\\https://github.com/.extraheader
- /usr/bin/git config --local --unset-all http:\\https://github.com/.extraheader
- /usr/bin/git config --local --name-only --get-regexp 'git config --local --name-only --get-regexp 'core\\.sshCommand' && git config --local --unset-all 'core\\.sshCommand' || ;'
- 9 /usr/bin/git config --local --name-only --get-regexp http\\(https\\)://github\\.com/\\.extraheader
- 10 http:\\https://github.com/.extraheader
- 11 /usr/bin/git config --local --unset-all http:\\https://github.com/.extraheader
- 12 /usr/bin/git config --local --name-only --get-regexp sh -c "git config --local --name-only --get-regexp 'http\\.https:\\/\\/github\\.com/\\.extraheader' && git config --local --unset-all 'http\\.https://github\\.com/.extraheader' || ;"

> Complete job

Security Check:

Jobs

- security-scan**
- access_control
- sast_analysis
- dast_scan
- sca_scan
- audit_log
- container-security-scan
- upload-reports
- iac-security-scan
- SIEM

security-scan
succeeded 37 minutes ago in 3m 27s

Set up job

Checkout code

Set up Node.js

Install dependencies

Enforce code reviews (Only on pull request)

Enforce signed commits

Post Set up Node.js

Post Checkout code

Complete job

SAST Analysis (Bandit & Trufflehog):

The screenshot shows the GitHub Actions interface for a repository. On the left, a sidebar lists various jobs: security-scan, access_control, sast_analysis, dast_scan, sca_scan, audit_log, container-security-scan, upload-reports, iac-security-scan, and SIEM. The sast_analysis job is currently selected, indicated by a blue vertical bar. The main area displays the results of the sast_analysis run, which succeeded 41 minutes ago. A prominent message indicates that a TruffleHog report was uploaded. Below this, the log output shows the execution of actions/upload-artifact@v4, detailing the upload of a zip artifact containing secrets-scan results. The SHA256 digest of the uploaded artifact is provided. The final artifact ID is 3057522893, and the download URL is a GitHub Actions link. Another section titled "SQL Injection Detection (Bandit)" is also visible, showing the execution of bandit_report.json and its output.

Bandit & Trufflehog SAST Analysis Reports:

```

11  "generated_at": "2025-05-04T10:26:01Z",
12  "metrics": {
13    "./test/files/decrypt.py": {
14      "loc": 4,
15      "nosec": 0,
16      "skipped_tests": 0
17    },
18    "./test/files/decrypt_bruteforce.py": {
19      "loc": 8,
20      "nosec": 0,
21      "skipped_tests": 0
22    },
23    "./test/files/encrypt.py": {
24      "CONFIDENCE.HIGH": 0,
25      "CONFIDENCE.LOW": 0,
26      "CONFIDENCE.MEDIUM": 0,
27      "CONFIDENCE.UNDEFINED": 0,
28      "SEVERITY.HIGH": 0,
29      "SEVERITY.LOW": 0,
30      "SEVERITY.MEDIUM": 0,
31      "SEVERITY.UNDEFINED": 0,
32      "loc": 7,
33      "nosec": 0,
34      "skipped_tests": 0
35    },
36    "totals": {
37      "CONFIDENCE.HIGH": 0,
38      "CONFIDENCE.LOW": 0,
39      "CONFIDENCE.MEDIUM": 0,
40      "CONFIDENCE.UNDEFINED": 0,
41      "SEVERITY.HIGH": 0,
42      "SEVERITY.LOW": 0,
43      "SEVERITY.MEDIUM": 0,
44      "skipped_tests": 0
45    }
46  }
47}

```

Ln 1, Col 1 | Spaces: 2 | UTF-8 | LF | ↵ JSON | ✅ Prettier | ⌂

DAST Analysis(SQLMap & Owasp ZAP):

dast scan
succeeded 34 minutes ago in 10m 58s

Search logs

Run details

Jobs

- security-scan
- access_control
- sast_analysis
- dast_scan**
- sca_scan
- audit_log
- container-security-scan
- upload-reports
- iac-security-scan
- SIEM

Run details

Upload ZAP Report

Set up Docker Buildx

Post Checkout code

Complete job

DAST Analysis Reports:

ZAP Scan Baseline Report #3

[Open](#)

github-actions opened yesterday

Site: <https://cdnjs.cloudflare.com>

Site: <https://preview.owasp-juice.shop>

New Alerts

- Missing Anti-clickjacking Header [10020] total: 12:
 - https://preview.owasp-juice.shop/socket.io/?EIO=4&transport=polling&t=PQM2Kig&sid=4FSBe_j30g0fr04AAms
 - <https://preview.owasp-juice.shop/socket.io/?EIO=4&transport=polling&t=PQM2Le2&sid=GtVLP5i3d2SHEr3AAmu>
 - <https://preview.owasp-juice.shop/socket.io/?EIO=4&transport=polling&t=PQM2MY3&sid=j5hyD24BnKjy8A58AAmw>
 - <https://preview.owasp-juice.shop/socket.io/?EIO=4&transport=polling&t=PQM2O3c&sid=mjCbnEf5Kyf1iVLIAAmy>
 - https://preview.owasp-juice.shop/socket.io/?EIO=4&transport=polling&t=PQM2Owe&sid=Ob_Q1nPJMipYyAAm0
 - ...
- Session ID in URL Rewrite [3] total: 12:
 - https://preview.owasp-juice.shop/socket.io/?EIO=4&transport=polling&t=PQM2Kik&sid=4FSBe_j30g0fr04AAms
 - <https://preview.owasp-juice.shop/socket.io/?EIO=4&transport=polling&t=PQM2Le4&sid=GtVLP5i3d2SHEr3AAmu>
 - <https://preview.owasp-juice.shop/socket.io/?EIO=4&transport=polling&t=PQM2MY5&sid=j5hyD24BnKjy8A58AAmw>
 - <https://preview.owasp-juice.shop/socket.io/?EIO=4&transport=polling&t=PQM2O3h&sid=mjCbnEf5Kyf1iVLIAAmy>
 - https://preview.owasp-juice.shop/socket.io/?EIO=4&transport=websocket&sid=4FSBe_j30g0fr04AAms
 - ...
- Full Path Disclosure [11009] total: 3:

SCA (Snyk):

Summary

Jobs

- [security-scan](#)
- [access_control](#)
- [sast_analysis](#)
- [dast_scan](#)
- [sca_scan](#)
- [audit_log](#)
- [container-security-scan](#)
- [upload-reports](#)
- [iac-security-scan](#)
- [SIEM](#)

Run details

Usage

Workflow file

sca_scan
succeeded 45 minutes ago in 3m 54s

Scan for vulnerable packages (Snyk) - continue on error

```

97 No upgrade or patch available
98
99
100
101 Organization: bilalbaree
102 Package manager: npm
103 Target file: package.json
104 Project name: juice-shop
105 Open source: no
106 Project path: /home/runner/work/Owasp_JuiceShop/Owasp_JuiceShop
107 Licenses: enabled
108
109 Tip: Detected multiple supported manifests (2), use --all-projects to scan all of them at once.
110
111
> Use signed libraries verification 0s
> Post set up Node.js 0s
> Post Checkout code 0s
> Complete job 0s

```

SCA Analysis Reports:

Container & Security (Grype & Trivy):

Grype & Trivvy Reports :

Summary

Jobs

- security-scan
- access_control
- sast_analysis
- dast_scan
- sca_scan
- audit_log
- container-security-scan**
- upload-reports
- iac-security-scan
- SIEM

Run details

Usage

Workflow file

container-security-scan
succeeded 46 minutes ago in 5m 38s

Scan Docker image for vulnerabilities (Grype)

```

10 [0022] ERROR discovered vulnerabilities at or above the severity threshold
11 NAME           INSTALLED   FIXED-IN    TYPE      VULNERABILITY      SEVERITY
12 base64url     0.0.6       3.0.0       npm      GHSA-rvg8-peq2-xj7q Medium
13 braces         2.3.2       3.0.3       npm      GHSA-grv7-fg5c-xoqj High
14 cookie          0.4.2       0.7.0       npm      GHSA-pxg6-pf52-xhBx Low
15 crypto-js        3.3.0       4.2.0       npm      GHSA-xxcc-pe8n-c4vf Critical
16 engine.io        4.1.2       6.2.1       npm      GHSA-r7qp-cfhv-p84w Medium
17 express-jwt       0.1.3       6.0.0       npm      GHSA-6g6m-mh55-wqgf High
18 gcc-12-base     12.2.0-14    deb      CVE-2022-27943   Negligible
19 gcc-12-base     12.2.0-14    deb      CVE-2023-4039   Negligible
20 got             8.3.2       11.8.5      npm      GHSA-pfrx-2q88-qq97 Medium
21 http-cache-semantics 3.8.1       4.1.1       npm      GHSA-rc47-6667-2j5j High
22 ip              2.0.1       4.2.0       npm      GHSA-2p57-rawc-gvfp High
23 jsonwebtoken     0.1.0       4.2.2       npm      GHSA-c7hr-j4mj-j2w6 Critical
24 jsonwebtoken     0.1.0       9.0.0       npm      GHSA-8cf7-32gw-w33 High
25 jsonwebtoken     0.1.0       9.0.0       npm      GHSA-hjrf-2a68-5999 Medium
26 jsonwebtoken     0.1.0       9.0.0       npm      GHSA-qphb-4952-7rx6 Medium
27 jsonwebtoken     0.4.0       4.2.2       npm      GHSA-c7hr-j4mj-j2w6 Critical
28 jsonwebtoken     0.4.0       9.0.0       npm      GHSA-8cf7-32gw-w33 High
29 jsonwebtoken     0.4.0       9.0.0       npm      GHSA-hjrf-2a68-5999 Medium
30 jsonwebtoken     0.4.0       9.0.0       npm      GHSA-qphb-4952-7rx6 Medium
31 jws              0.2.6       3.0.0       npm      GHSA-gjcw-v447-2w7q High
32 libc6            2.36-9+deb12u10   deb      CVE-2010-4756   Negligible
33 libc6            2.36-9+deb12u10   deb      CVE-2018-20796 Negligible
34 libc6            2.36-9+deb12u10   deb      CVE-2019-1010622 Negligible
35 libc6            2.36-9+deb12u10   deb      CVE-2019-1010623 Negligible
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58

```

Search logs

container-security-scan
succeeded 47 minutes ago in 5m 38s

Scan image for Read-Only File System violations (Trivy)

```

33 Report Summary
34
35
36 | Target          | Type | Vulnerabilities | Secrets |
37 |-----|-----|-----|-----|
38 | 61601f9d01b8 (alpine 3.21.3) | alpine | 0 | - |
39 |
40 | workspace/frontend/src/app/app.guard.spec.ts | text | - | 0 |
41 |
42 | workspace/frontend/src/app/last-login-ip/last-login-ip.component.spec.ts | text | - | 0 |
43 |
44 | workspace/lib/insecurity.ts | text | - | 1 |
45 |
46 Legend:
47 - -: Not scanned
48 - '0': Clean (no security findings detected)
49
50
51 workspace/lib/insecurity.ts (secrets)
52 =====
53 Total: 1 (HIGH: 1, CRITICAL: 0)
54
55 HIGH: AsymmetricPrivateKey (private-key)
56 -----
57 Asymmetric Private Key
58

```

Search logs

DockerHub Image Upload:

The screenshot shows the Docker Hub interface for the user 'bilalbaree'. The left sidebar shows the user's personal space with options like Repositories, Settings, Default privacy, Notifications, Billing, Usage, Pulls, and Storage. The main area is titled 'Repositories' and lists all repositories within the 'bilalbaree' namespace. A red box highlights the first repository, 'bilalbaree/juice-shop', which was last pushed 'about 1 hour ago' and is a public image. Other repositories listed are 'bilalbaree/sublimephpcompanion', 'bilalbaree/devsec_a33', and 'bilalbaree/devsec_a3', all of which are public and inactive.

IaC Security (Terraform):

I cannot run EC2 instances (not free just uploading workflow file)

The screenshot shows the logs for an 'iac-security-scan' job. The left sidebar lists various security scan jobs: security-scan, access_control, sast_analysis, dast_scan, sca_scan, audit_log, container-security-scan, upload-reports, iac-security-scan (which is selected), and SIEM. The main area displays the log entries for the 'iac-security-scan' job, which succeeded 54 minutes ago in 42s. The log details the execution steps: Set up job, Checkout code, Set up Python, Install Terraform, Install Checkov, Install Conftest, Run Checkov for misconfigurations, Run Conftest with least privilege policy, Check for secrets in Terraform using Checkov, Post Set up Python, Post Checkout code, and Complete job. Each step is accompanied by a timestamp and a small icon indicating its status.

Conftest and Checkov for OPA and Policy Review/Enforcement:

Summary

Jobs

- ✓ [security-scan](#)
- ✓ [access_control](#)
- ✓ [sast_analysis](#)
- ✓ [dast_scan](#)
- ✓ [sca_scan](#)
- ✓ [audit_log](#)
- ✓ [container-security-scan](#)
- ✓ [upload-reports](#)
- ✓ [iac-security-scan](#)
- ✓ [SIEM](#)

[Run details](#)

[Usage](#)

[Workflow file](#)

iac-security-scan
succeeded 1 hour ago in 42s

Run Checkov for misconfigurations 8s

```
50    PASSED for resource: aws_iam_policy.secure_limited_policy
51    File: /main.tf:37-59
52    Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-iam-policies/bc-aws-iam-45
53    Check: CKV_AWS_41: "Ensure no hard coded AWS access key and secret key exists in provider"
54    PASSED for resource: aws.default
55    File: /main.tf:1-3
56    Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/secrets-policies/bc-aws-secrets-5
57    Check: CKV_AWS_40: "Ensure AWS IAM policy does not allow full IAM privileges"
58    PASSED for resource: aws_iam_policy.secure_limited_policy
59    File: /main.tf:37-59
60    Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-iam-policies/bc-aws-2-40
61
```

Run Conftest with least privilege policy 0s

```
1  ► Run conftest test terraform/main.tf --policy policies
12
13  0 tests, 0 passed, 0 warnings, 0 failures, 0 exceptions
```

> Check for secrets in Terraform using Checkov 3s

> Post Set up Python 0s

> Post Checkout code 0s

> Complete job 0s

Siem Tool Integration (DataDog):

The screenshot shows a GitHub Actions pipeline for the 'Secure CI/CD Pipeline' in the 'Owasp_JuiceShop' repository. The 'Actions' tab is selected, displaying the log for the 'Update ci.yml #43' job. The 'SIEM' step is highlighted in green, indicating success. The log output shows the command used to send logs to Datadog via API:

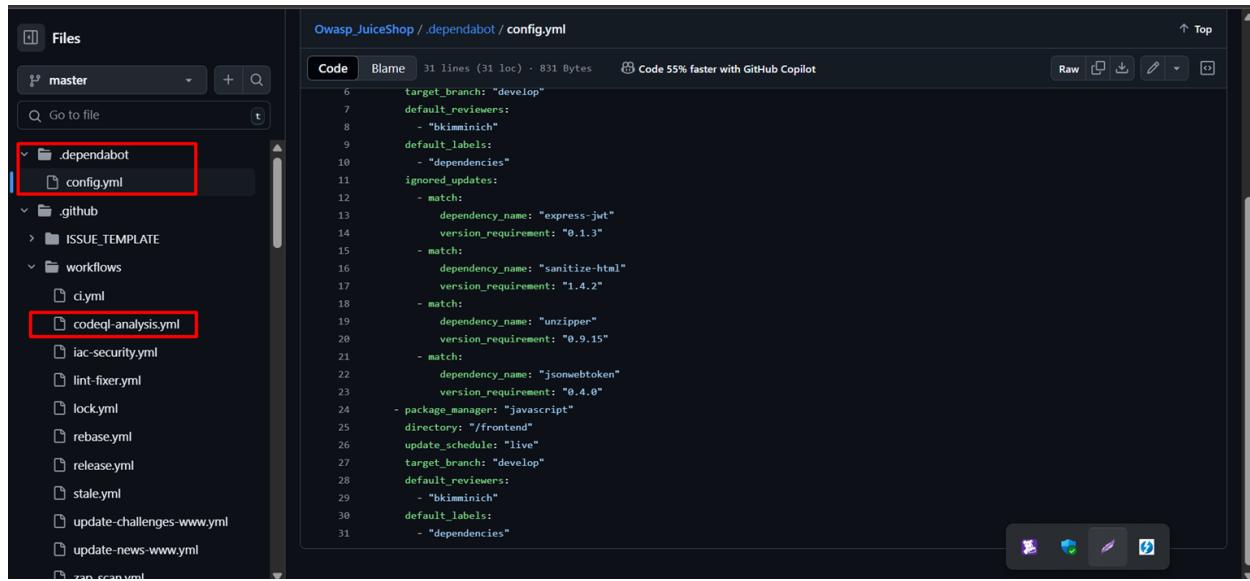
```
▶ Run curl -X POST "https://http-intake.logs.datadoghq.eu/v1/input" \
13 % Total    % Received % Xferd  Average Speed   Time     Time  Current
14                               Dload  Upload Total Spent    Left  Speed
15
16      0     0     0     0     0     0     0     0 ---::--- ::---::--- 0
17 100  207 100   2 100 205     8  830 ---::--- ::---::--- 841
18 {
```

DataDog Log Explorer :

The screenshot shows the DataDog Log Explorer interface. The timeline at the top displays log entries from 12:30 to 16:15 UTC+05:00, with four logs found in the last 4 hours. The facets sidebar on the left shows filters for CORE, Index, Source, Host, Service (selected), and Status (Error, Warn, Info). The main area displays 'Watchdog Insights' which states 'Watchdog did not detect any insights in this scope.' Below this is a table of log entries:

	DATE	HOST	SERVICE	CONTENT
1	May 04 15:25:51.012	github-runner	ci-cd	Security scan passed for commit df16fafc442c613f992a107ae55ab572f481ba17 by Bil...
1	May 04 14:31:10.020	github-runner	ci-cd	Security scan passed for commit df16fafc442c613f992a107ae55ab572f481ba17 by Bil...
1	May 04 14:13:21.999	github-runner	ci-cd	Security scan passed for commit c3389a8f1ca639014fe5eb00c78eb2ef6174ebdc by Bil...
1	May 04 13:51:28.102	github-runner	ci-cd	Security scan passed for commit c3389a8f1ca639014fe5eb00c78eb2ef6174ebdc by Bil...

Vulnerability Management (CodeQL & Dependabot):



The screenshot shows the GitHub Files interface for the repository "Owasp_JuiceShop / dependabot". The left sidebar displays a tree view of files, with two specific files highlighted by red boxes: ".dependabot/config.yml" and "codeql-analysis.yml". The right panel shows the content of the ".dependabot/config.yml" file.

```
target_branch: "develop"
default_reviewers:
- "bkimminich"
default_labels:
- "dependencies"
ignored_updates:
- match:
  dependency_name: "express-jwt"
  version_requirement: "0.1.3"
- match:
  dependency_name: "sanitize-html"
  version_requirement: "1.4.2"
- match:
  dependency_name: "unzipper"
  version_requirement: "0.9.15"
- match:
  dependency_name: "jsonwebtoken"
  version_requirement: "0.4.0"
- package_manager: "javascript"
directory: "/frontend"
update_schedule: "live"
target_branch: "develop"
default_reviewers:
- "bkimminich"
default_labels:
- "dependencies"
```