Trivy-SBOM

Introduction to Trivy and SBOM

What is Trivy?

Trivy is an open-source security scanner developed by Aqua Security. It is widely used for scanning container images, file systems, and repositories for vulnerabilities, misconfigurations, and other security risks. Trivy is known for its simplicity, speed, and ability to integrate seamlessly into CI/CD pipelines. It supports multiple scan targets, including:

- Container images (Docker, Podman, Kubernetes)
- Filesystems and local directories
- Git repositories
- Infrastructure as Code (IaC) configurations
- Software Bill of Materials (SBOM)

One of Trivy's key features is its ability to generate an SBOM, which provides a detailed inventory of software components and their dependencies.

What is an SBOM (Software Bill of Materials)?

A **Software Bill of Materials (SBOM)** is a structured list of all the components, libraries, and dependencies present in a software application. It acts as an inventory, providing visibility into the software supply chain. SBOMs are crucial for improving security, compliance, and risk management in software development.

Key benefits of an SBOM include:

- **Security:** Identifies vulnerabilities in third-party dependencies.
- **Compliance:** Helps organizations comply with security regulations (e.g., NIST, ISO, and Executive Order 14028).
- **Transparency:** Provides insight into software components and their origins.
- **Risk Management:** Assists in mitigating supply chain attacks.

Trivy can generate SBOMs in industry-standard formats such as **CycloneDX** and **SPDX**, making it easy to integrate with security tools and compliance workflows.

Importance of SBOM in Software Development

A **Software Bill of Materials (SBOM)** is essential in modern software development because it provides a transparent inventory of all software components, dependencies, and libraries

used in an application. This visibility is crucial for security, compliance, and risk management. Here's why SBOMs are important:

1. Enhancing Software Security

- **Vulnerability Identification:** SBOMs help security teams detect and address vulnerabilities in third-party libraries by mapping them to known CVEs (Common Vulnerabilities and Exposures).
- **Supply Chain Security:** With increasing supply chain attacks (e.g., Log4j, SolarWinds), an SBOM helps identify compromised dependencies before they cause harm.
- **Proactive Patching:** Developers can track outdated or insecure components and apply patches quickly.

2. Ensuring Compliance and Regulatory Requirements

- **Regulatory Standards:** Many regulations and industry standards (e.g., NIST, ISO 27001, EU Cyber Resilience Act) require organizations to maintain SBOMs for transparency.
- **Government Mandates:** The U.S. Executive Order 14028 mandates SBOM usage for federal software procurement to enhance national cybersecurity.
- **Open Source Licensing Compliance:** SBOMs help organizations track and comply with open-source licenses like MIT, Apache, and GPL.

3. Improving Software Quality and Transparency

- **Component Transparency:** Developers and organizations know exactly what software components they are using, reducing the risk of unverified or unauthorized code.
- **Dependency Management:** Helps track indirect dependencies (transitive dependencies) that may introduce security or performance issues.
- **Easier Debugging & Audits:** If a bug is found, an SBOM makes it easier to trace the affected components.

4. Risk Management and Incident Response

- **Quick Incident Response:** If a critical vulnerability is discovered (e.g., zero-day attacks), organizations can quickly identify and remediate affected components.
- **Reducing Technical Debt:** By maintaining an updated SBOM, organizations can prevent the accumulation of outdated dependencies that pose security risks.
- **Business Continuity:** Helps assess risks and ensure that software can be safely used in critical systems.

GitHub Actions workflow for SBOM Generation:

GitHub Repository: https://github.com/SumathiD20/Wednesday_Adventures/

Location: .github/workflows/Main_CI_Pipeline.yml

This GitHub Actions workflow is part of the Wednesday Wicked Project's development pipeline. It ensures security and compliance by generating a Software Bill of Materials (SBOM) using Trivy and uploading the report as an artifact. This process helps in identifying vulnerabilities and managing software dependencies effectively.

The workflow is named **sbom_scan** and is executed in GitHub Actions as a job. Below is a step-by-step breakdown:

Installing Trivy

```
- name: Install Trivy
  run: |
    curl -sfL https://raw.githubusercontent.com/aquasecurity/trivy/main/contrib/install.sh | sh
    sudo mv ./bin/trivy /usr/local/bin/trivy
```

- **Trivy** is an open-source security scanner used for generating SBOMs and scanning vulnerabilities in containers, filesystems, and repositories.
- The script downloads and installs **Trivy** using a command from Aqua Security's official repository.
- The **binary is moved** to /usr/local/bin/trivy, making it accessible system-wide.

Generating the SBOM Report

```
- name: Generate SBOM Report
  run: trivy fs --format spdx-json -o sbom.json .
```

- The trivy fs command scans the filesystem (.), which includes all project files.
- The output is formatted as **SPDX JSON** (a widely accepted SBOM format).
- The generated SBOM report is saved as sbom.json.
- This report contains a list of dependencies, libraries, and their versions, helping teams track and analyze software components.
- The parameter fs is used here to scan the repository where the Code resides and GitHub action is implemented.

Uploading the SBOM Report as an Artifact

```
- name: Upload SBOM Report
  uses: actions/upload-artifact@v4
  with:
    name: sbom-report
    path: sbom.json
```

- The upload-artifact action stores the generated SBOM file (sbom.json) as a downloadable artifact in the GitHub Actions workflow.
- The uploaded file can be retrieved later for security analysis, compliance audits, or further processing

The file generated by the GitHub Actions for SBOM generation:

sbom-report (3).zip

The GitHub Actions workflow triggers on every push or pull request to the main branch.
It automatically runs security scans using Trivy to generate an SBOM report.
This ensures continuous security monitoring in the L00187746-Trivy-Sumathi repository.
Similarly, it is integrated into l00187927-WA-Jira18-CIPipeline for automated security checks.
By doing this, the project maintains compliance, security, and software transparency

What does this Document contain?

The SBOM report contains frontend, backend, and library dependency information.

Frontend: Identified via frontend/package-lock.json and UI-related npm packages.

Backend: Identified via project-backend/package-lock.json and API-related libraries.

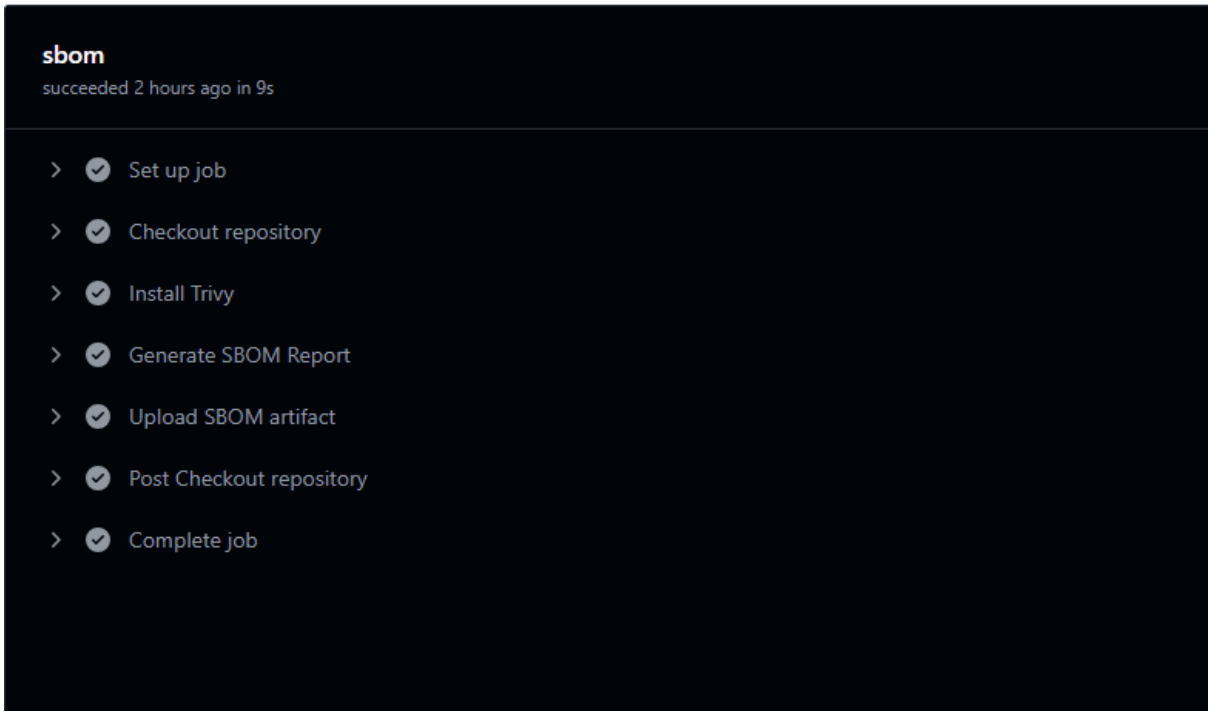
GitHub Actions Code for SBOM:

SumathiD20 Update SBOM_trivy.yaml ✓

Code | Blame | 38 lines (32 loc) · 990 Bytes

```yaml
name: Generate SBOM with Trivy

on:
  push:
    branches:
      - L00187746-Trivy-Sumathi
      - 100187927-WA-Jira18-CIPipeline
      - main
  pull_request:
    branches:
      - L00187746-Trivy-Sumathi
      - 100187927-WA-Jira18-CIPipeline
      - main

jobs:
  sbom:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout repository
        uses: actions/checkout@v3


      - name: Install Trivy  # ✅ Ensure Trivy is installed
        run: |
          curl -sfL https://raw.githubusercontent.com/aquasecurity/trivy/main/contrib/install.sh | sh
          sudo mv ./bin/trivy /usr/local/bin/trivy
          trivy --version  # ✅ Verify installation

      - name: Generate SBOM Report
        run: |
          # trivy sbom --format spdx-json -o sbom.json ./package-lock.json
          trivy fs --format spdx-json -o sbom.json .

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

Workflow steps:



Example for artifacts location: