DevOps Shack

Trivy Real-time Usecase With Examples

Click Here To Enrol To Batch-5 | DevOps & Cloud DevOps

Trivy is a versatile and powerful open-source vulnerability scanner from Aqua Security. It can scan container images, file systems, and Git repositories for vulnerabilities and misconfigurations. Here are some detailed use cases of Trivy, complete with examples and commands:

1. Scanning Container Images

Use Case: Identify vulnerabilities in container images.

Example:

```
trivy image nginx:latest
```

Explanation: This command scans the latest version of the official Nginx Docker image for known vulnerabilities.

2. Scanning Local Filesystems

Use Case: Scan local directories or files for vulnerabilities, including configuration files.

Example:

```
trivy fs /path/to/directory
```

Explanation: This command scans all files and subdirectories within the specified directory for vulnerabilities.

3. Scanning Git Repositories

Use Case: Check Git repositories for vulnerabilities in the code and dependencies.

Example:

```
trivy repo https://github.com/aquasecurity/trivy
```

Explanation: This command scans the specified GitHub repository for vulnerabilities.

4. Scanning Dockerfile

Use Case: Detect vulnerabilities and misconfigurations in Dockerfile.

Example:

```
trivy config --config-policy /path/to/policy.rego Dockerfile
```

Explanation: This command scans a Dockerfile for vulnerabilities and checks it against specified Rego policies for misconfigurations.

5. Integration with CI/CD Pipelines

Use Case: Automate vulnerability scanning in CI/CD pipelines.

Example (GitHub Actions):

```
name: Trivy Scan

on: [push]

jobs:
  scan:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout code
        uses: actions/checkout@v2
      - name: Run Trivy vulnerability scanner
        uses: aquasecurity/trivy-action@v0.0.14
        with:
          image-ref: 'nginx:latest'
```

Explanation: This GitHub Actions workflow runs a Trivy scan on the latest Nginx image whenever code is pushed to the repository.

6. Scanning Kubernetes Manifests

Use Case: Identify vulnerabilities and misconfigurations in Kubernetes manifests.

Example:

```
trivy k8s --report all --namespace default
```
Explanation: This command scans all Kubernetes resources in the default namespace for vulnerabilities and misconfigurations.

7. Generating Reports

Use Case: Create detailed vulnerability reports in various formats.

Example:

```
trivy image --format json --output report.json nginx:latest
```
Explanation: This command scans the latest Nginx image and outputs the results in JSON format to a file named `report.json`.

8. Scanning Files with Custom Configuration

Use Case: Use custom configuration files to specify scan parameters.

Example:

```
trivy fs --config /path/to/config.yaml /path/to/directory
```
Explanation: This command scans a specified directory using the parameters defined in the custom configuration file `config.yaml`.

9. Ignore Specific Vulnerabilities

Use Case: Ignore specific vulnerabilities during scanning.

Example:

```
trivy image --ignore-unfixed nginx:latest
```
Explanation: This command scans the latest Nginx image but ignores vulnerabilities that have not yet been fixed.

10. Scanning for Misconfigurations in IaC (Infrastructure as Code)

Use Case: Detect misconfigurations in Terraform, CloudFormation, and other IaC files.

Example:

```
trivy config /path/to/terraform/files
```

Explanation: This command scans Terraform files for potential misconfigurations and security risks.

Detailed Example Commands

1. **Scanning a Docker Image and Saving the Output in JSON:**

   ```
   trivy image --format json --output output.json python:3.9
   ```

2. **Scanning a Local Filesystem for Vulnerabilities:**

   ```
   trivy fs /var/www/html
   ```

3. **Scanning a Remote GitHub Repository:**

   ```
   trivy repo https://github.com/docker-library/python
   ```

4. **Running Trivy in a CI/CD Pipeline (GitLab CI Example):**

```
stages:
  - security

trivy-scan:
  stage: security
   image: aquasec/trivy:latest
   script:
    - trivy image --exit-code 1 --severity HIGH,CRITICAL nginx:latest
```

12. **Ignoring Unfixed Vulnerabilities and Low Severity Issues:**

    ```
    trivy image --ignore-unfixed --severity HIGH,CRITICAL ubuntu:20.04
    ```

These use cases and examples demonstrate the versatility of Trivy in enhancing the security of various components in the DevOps pipeline.