Container Vulnerability Scanning in GitLab CI/CD Pipeline

Automating Defect Detection with Trivy

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Submitted by:

Pranay Bhatia (17935037)

Agenda

- Setting the ground
- Introduction to topic
- Comparison
- Experiment
- Conclusion
- Lessons learnt and Q/A

https://www.101qs.com/2518-a-lot-of-doors

Motivation

- **House anology**
  - House with many doors and windows
  - Buglurs and thiefs with bad intentions always be outside.
- **What wont help much:**
  - Assuming all buglur might look buglury.
  - Tracking each individual Buglur.
- **Challenge:**
  - !!! House must be protected at any cost. !!!

https://www.101qs.com/2518-a-lot-of-doors

Motivation

- **What will you do to protect the house?**
  - **First:** Count all windows and doors you have. **(Attack Surface)**
  - **Second:** Check security of each entrance**. (vulnerability intelligence)**
  - **Third:** Check if there some defects in locks in doors. **(threat intelligence)**
  - **Fourth:** Update to new and advanced protection time to time.
- **Observations:**
  - Resources are limited.
  - Time is limited.

https://www.101qs.com/2518-a-lot-of-doors

Topic

- **Topic:**

Home Security Automation

https://www.linkedin.com/pulse/key-security-layers-docker-containers-mohit-vaish/

Topic

- ~~Topic:~~

~~**Home Security Automation**~~

- **Topic:**

Container Security Automation

https://www.linkedin.com/pulse/key-security-layers-docker-containers-mohit-vaish/

Topic

- ~~Topic:~~

~~Home Security Automation~~

- **Topic:**

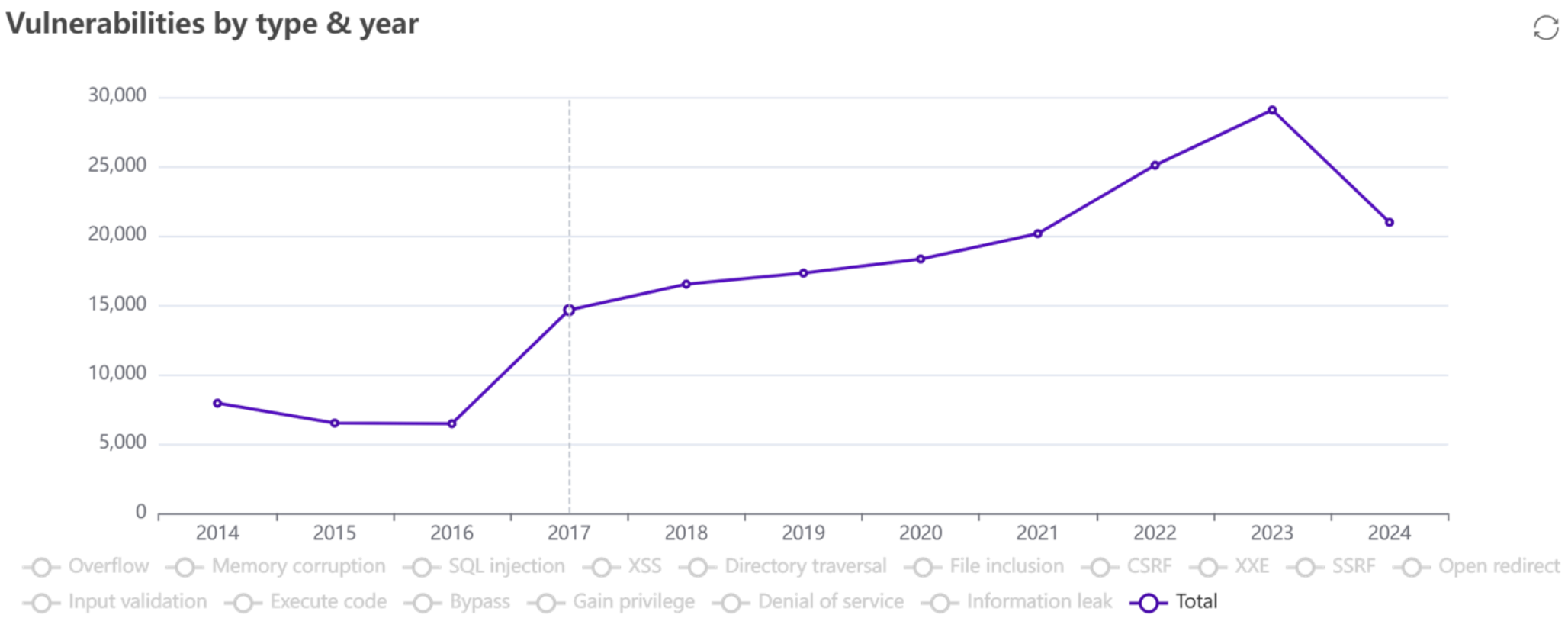
automatic static analysis workflows

Container Security Automation

https://www.linkedin.com/pulse/key-security-layers-docker-containers-mohit-vaish/

Introduction

Introduction



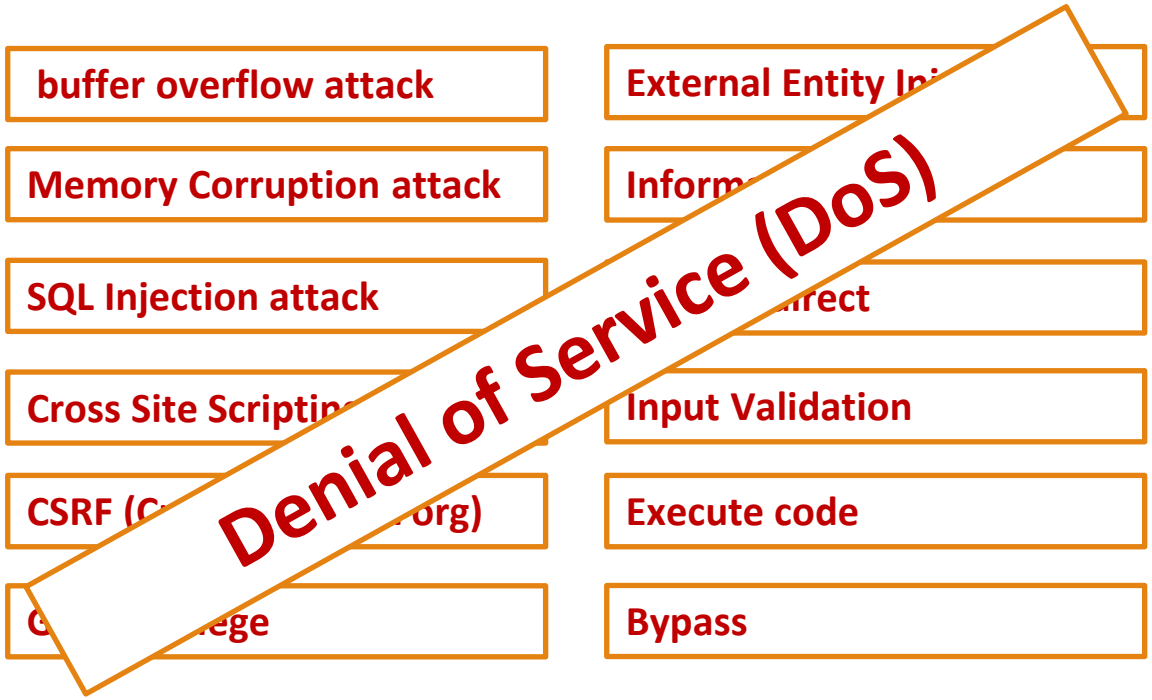
https://www.cvedetails.com/

Vulnerabilities

- buffer overflow attack
- Memory Corruption attack
- SQL Injection attack
- Cross Site Scripting (XSS)
- CSRF (Cross site reg. Forg)
- Gain privilege
- External Entity Injection
- Information leak
- Open redirect
- Input Validation
- Execute code
- Bypass

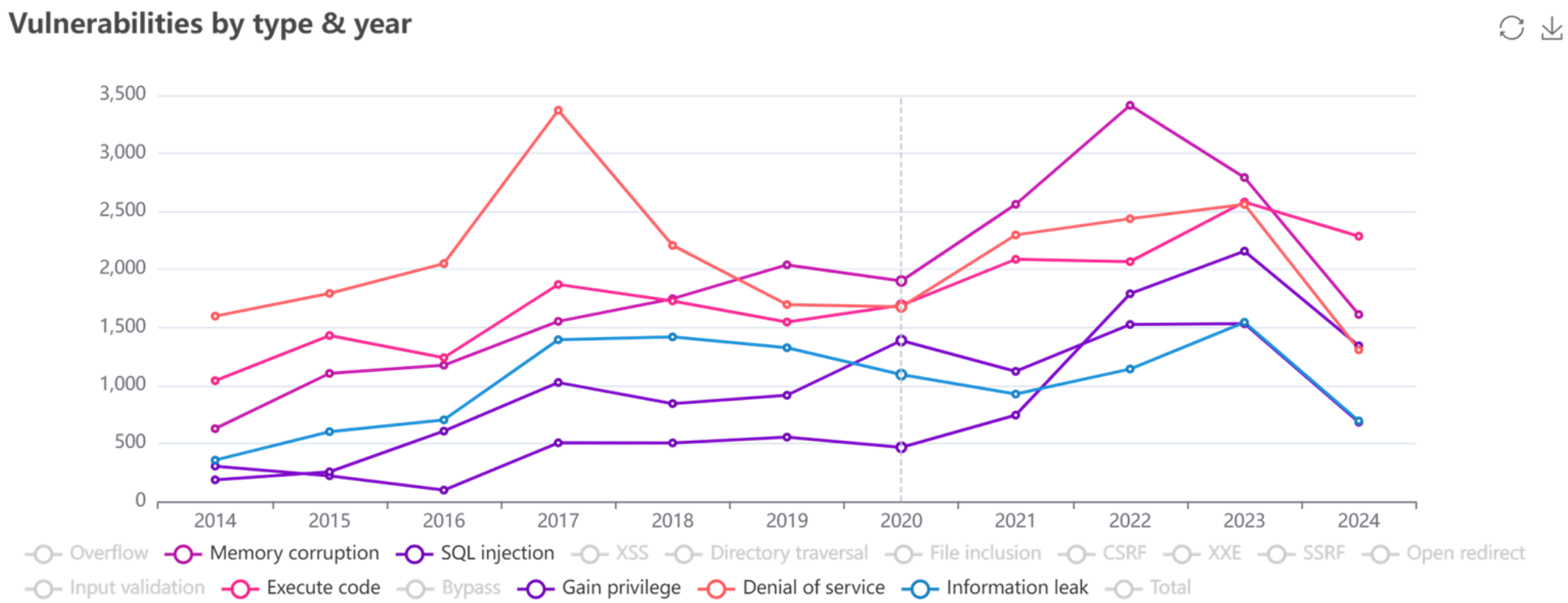
https://www.cvedetails.com/

Vulnerabilities



https://www.cvedetails.com/

Number of attacks per year

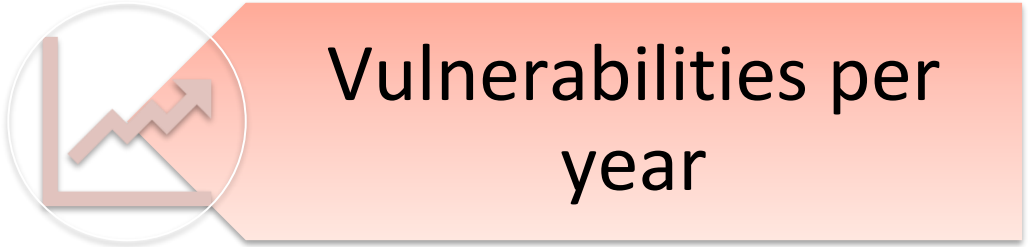


https://www.cvedetails.com/

Problem Statement

Problem statement

- **Increasing Complexity** of Modern Applications
- **Security Concerns** with Containerization
- Need for **Automated Security Measures**
- Leveraging **GitLab for Automated Validation**



Objective

Deliverables

- **Technical**
  - **Dockerfile** for the container in which **trivy runs on the gitlab runner**
  - **Automated static analysis** of yaml files for defect
  - **Implement admission control work flow**, which rejects PR on critical errors
- **Theoretical**
  - **Existing meaningful additions to trivy implementation**
  - Addressing **Alternative Scientific research** on the topic

Background

YAML File Validation in GitLab **CI/CD** Pipeline

- **Yaml File**
  - human-readable data serialization language [1]
  - often used for writing configuration files
  - Better readibility than json
- **CI/CD**
  - Automatically checks new code
  - deploys code to target after passing tests
  - Saves time by automating repetitive tasks.
  - Reduces human error in the deployment process.

```yaml
database:
  user: admin
  password: supersecretpassword
  host: localhost
  port: 3306

services:
  web:
    image: webapp:latest
    ports:
      - "80:80"
  db:
    image: mysql:latest
    environment:
      MYSQL_ROOT_PASSWORD: password123
    ports:
      - "3306:3306"
logging:
  level: DEBUG
  output: /var/log/app.log
```

[1] https://www.redhat.com/en/topics/automation/what-is-yaml

YAML File Validation in GitLab **CI/CD** Pipeline

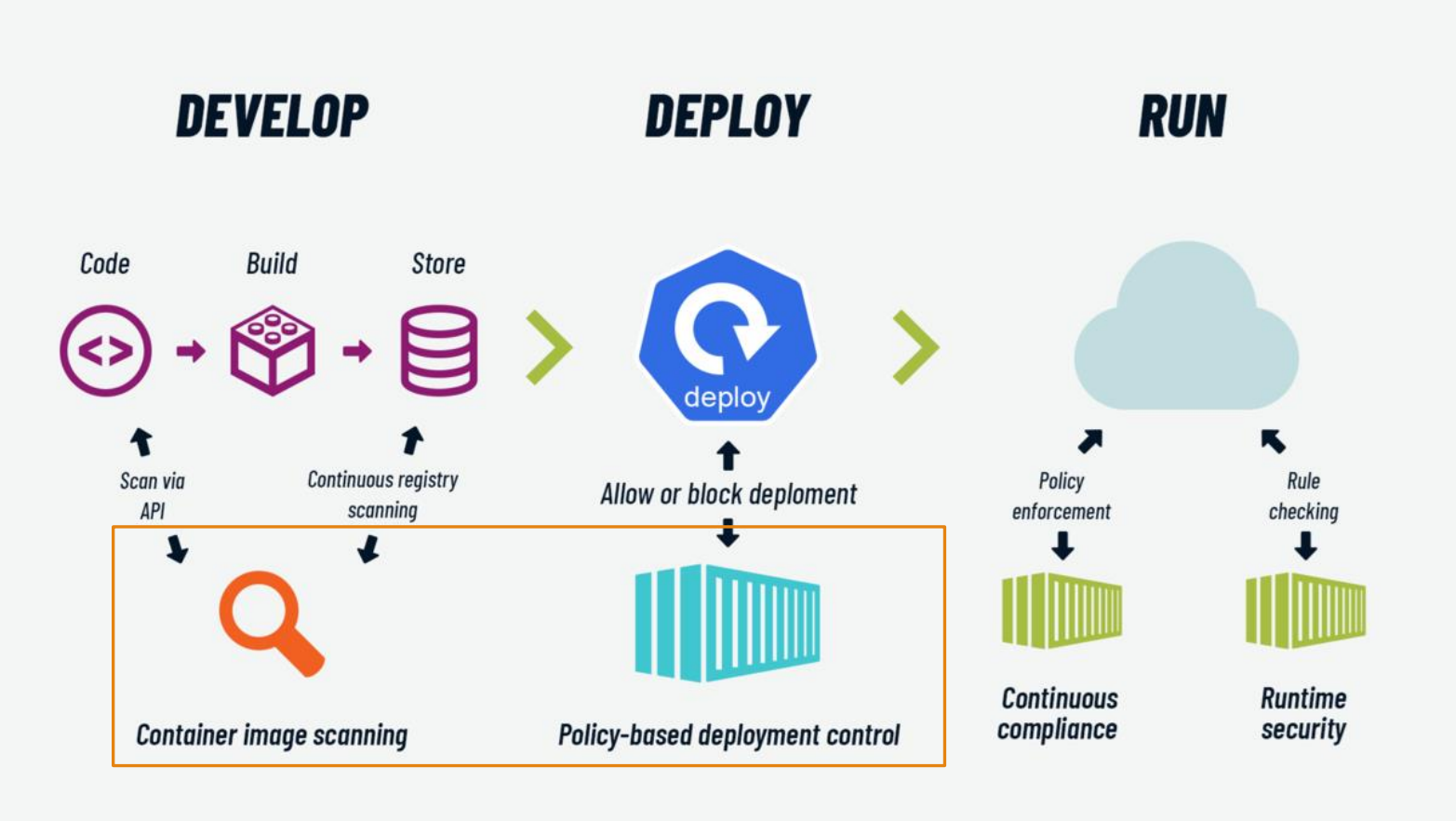
- **Validation**
  - Storing sensitive data in plain text
  - Weak password
  - Debug level logging in production
- **Problems:**
  - Access Controls
  - Storing Secrets
  - Logging Levels
  - Environment-Specific Configurations

```yaml
database:
  user: admin
  password: supersecretpassword # Storing
sensitive data in plain text
  host: localhost
  port: 3306

services:
  web:
    image: webapp:latest
    ports:
      - "80:80"
  db:
    image: mysql:latest
    environment:
      MYSQL_ROOT_PASSWORD: password123
# Weak password
    ports:
      - "3306:3306"
logging:
  level: DEBUG
# Debug level logging in production
  output: /var/log/app.log
```

[1] https://www.redhat.com/en/topics/automation/what-is-yaml

Dev Sec Ops workflow

- **Container image scanning**
  - Scanning project
  - Scanning docker images
  - Scanning vulnerabilitis
- **Policy-based deployment control**
  - RBAC
  - Admissoin control workflow
  - Report generation



https://blog.sparkfabrik.com/en/container-security-how-to

Container Image scanning

Security Tools

https://hub.docker.com/r/aquasec/trivy

www.okyaytechald.com

https://www.projectquay.io/

Anchore Engine:
https://catalog.redhat.com/software/container-stacks/detail/5e9872c4d4ae96b7493f08e1

improves-container-security-and-compliance/

https://www.linkedin.com/pulse/semi-automating-blackduck-brit-glazer/

Security Tools Analysis

Tool	Type	CI/CD	Compliance	Features
Anchor-Engine	Open-source	Yes	Yes	vulnerability scanning security and compliance
Black Duck	Commercial	No	Extensive	detailed open-source management and compliance capabilities
Clair	Open-source	Yes	Limited	vulnerability scanning
Quay Security scan	Open-source	No	No	basic vulnerability scanning with own container registery
Trivy	Open-source	Yes	No (Yes, in Trivy 2.0)	Scans container images, file systems, and Git repositories for vulnerabilities. Works in Kubernetes environment.
Twistlock	Commercial	Yes	Extensive	extensive compliance

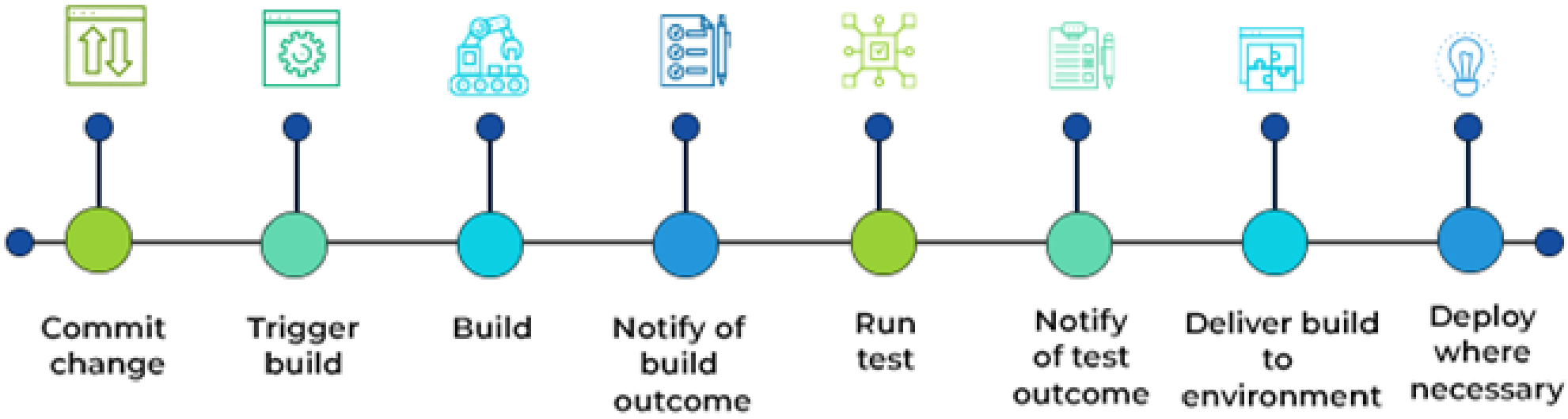
Security Tools Analysis (Trivy)

Scanner	OS Packages	Application Dependencies	Easy to use	Accuracy	Suitable for CI
Trivy	✅	✅ (8 languages)	★★★	★★★	★★★
Clair	✅	×	★	★★	★★
Anchore Engine	✅	✅ (4 languages)	★★	★★	★★★
Quay	✅	×	★★★	★★	×
Docker Hub	✅	×	★★★	★	×
GCR	✅	×	★★★	★★	×

https://aquasecurity.github.io/trivy/v0.17.2/comparison/

Policy based deployments controls

CI/CD Pipeline



Img. Source. https://www.spiceworks.com/tech/devops/articles/what-is-ci-cd/

Admission control workflow in CI/CD

- **Validation:** Validate Automatically checks new YAML files for errors and security issues before they are merged.
- **Decision:** Accepts or rejects changes based on the validation results to ensure only secure, error-free configurations are deployed.
- **Feedback:** Provides immediate feedback to developers, helping them fix issues quickly and maintain high-quality code.

User story

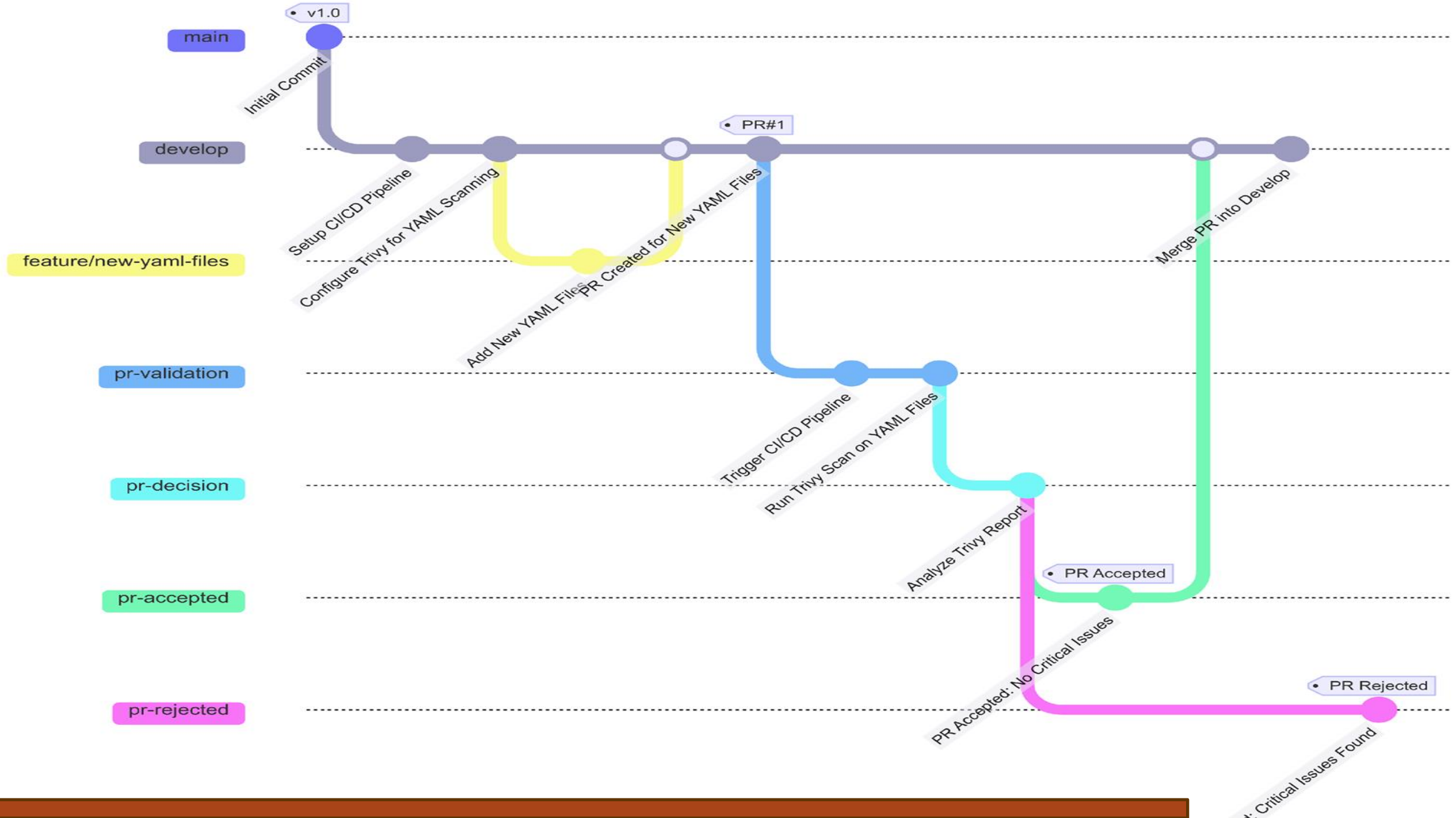
When A user creates a PR which includes new YAML files.

Then A GitLab validation pipeline is triggered.

And Scans new YAML Files for defects using Trivy.

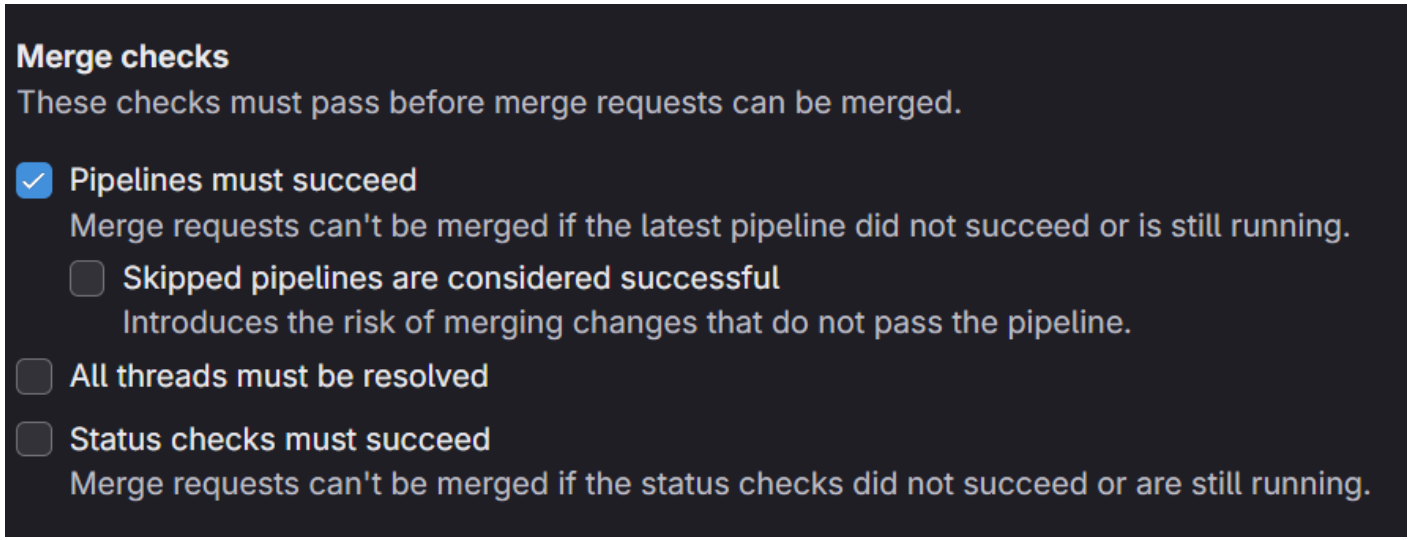
Then PR is either accepted or rejected based on reports generated.

Workflow


v1.0
main
Initial Commit
develop
Setup CI/CD Pipeline
Configure Trivy for YAML Scanning
PR#1
feature/new-yaml-files
Add New YAML Files
PR Created for New YAML Files
Merge PR into Develop
pr-validation
Trigger CI/CD Pipeline
Run Trivy Scan on YAML Files
pr-decision
Analyze Trivy Report
PR Accepted
pr-accepted
PR Accepted: No Critical Issues
PR Rejected
pr-rejected

Gitlab configurations

- Make main protected branch. (To make sure no direct commits are allowed to main).
- Enable Merge checks in Gitlab. Settings > Merge Requests.



.gitlab-ci.yaml

.gitlab-ci.yaml

```yaml
.trivy-scan-template:
  image: aquasec/trivy:latest
  stage: scan
  allow_failure: true
  rules:
    - if: '$CI_PIPELINE_SOURCE == "merge_request_event"'

00 scan trivy misconfig:
   extends: .trivy-scan-template
   script:
     - trivy fs --scanners misconfig .

00 scan trivy secret:
   extends: .trivy-scan-template
   script:
     - trivy fs --scanners secret .
```

.gitlab-ci.yaml

```yaml
00 scan trivy license:
   extends: .trivy-scan-template
   script:
     - trivy fs --scanners license .

00 scan trivy file system:
   extends: .trivy-scan-template
   script:
     - trivy fs --scanners vuln .

00 scan trivy IAAC:
   extends: .trivy-scan-template
   script:
     - trivy conf --severity HIGH,CRITICAL .
```

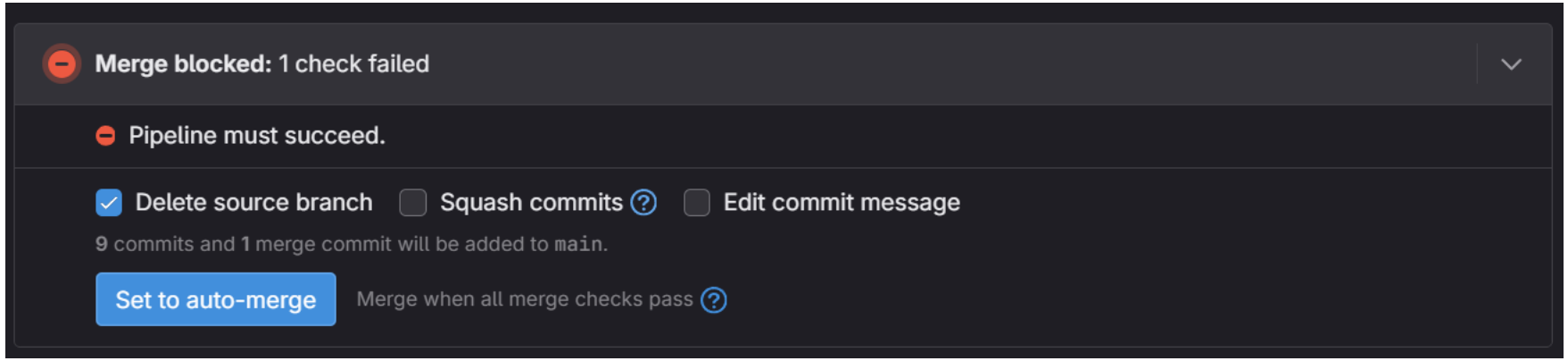
Trivy based docker image for gitlab runner

Dockerfile

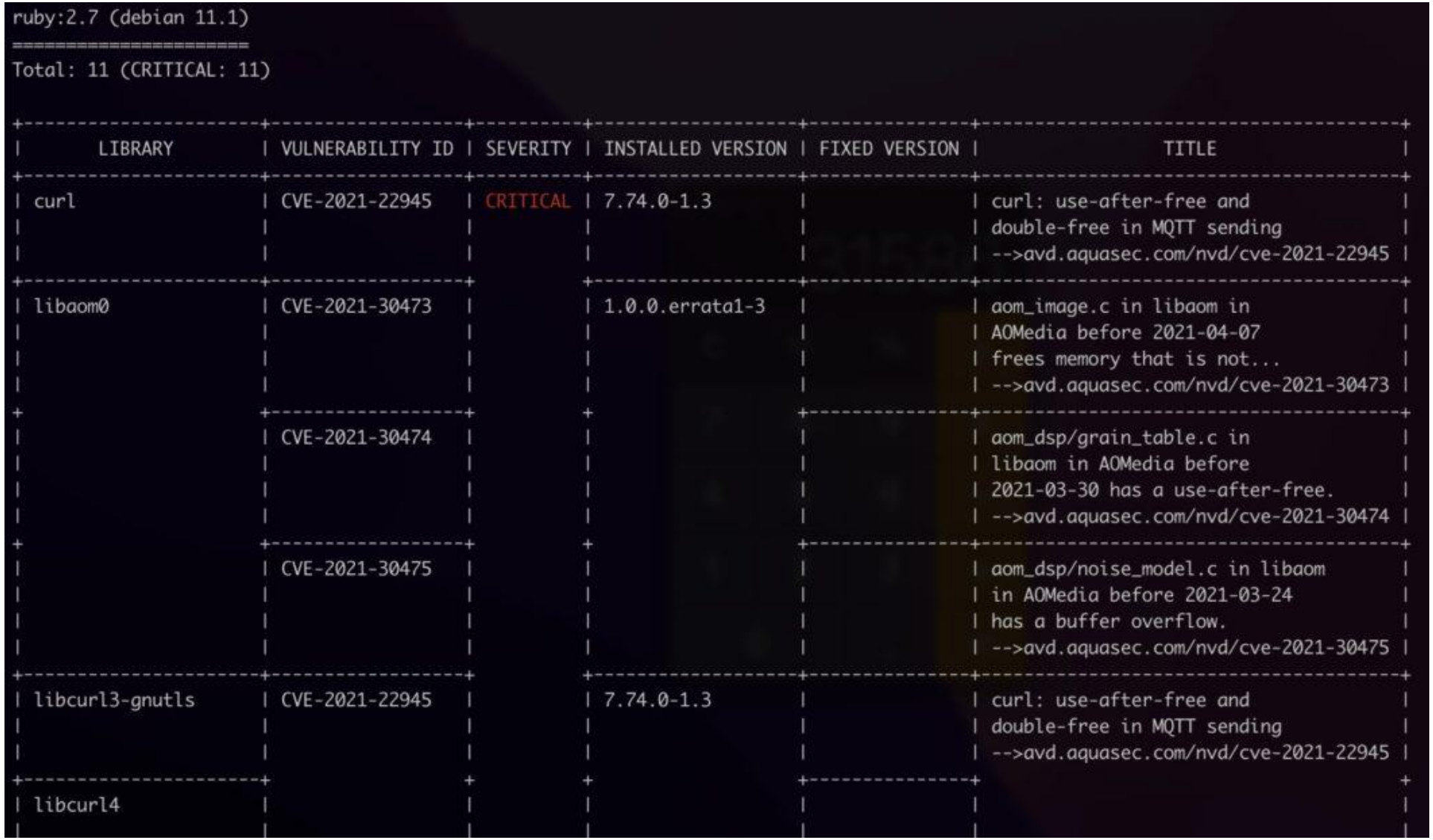
```
FROM alpine:3.18
ENV TRIVY_VERSION=v0.18.3
RUN apk add --no-cache \
    curl \
    && curl -sfL https://raw.githubusercontent.com/aquasecurity
    /trivy/main/contrib/install.sh | sh -s -- -b /usr/local/bin
    ${TRIVY_VERSION} \
    && apk del curl
WORKDIR /app
 # COPY script.sh /app/

 ENTRYPOINT ["trivy"]
 CMD ["--help"]
```

Trivy Vulnurabity Reports

Trivy Report



Trivy Report



ruby:2.7 (debian 11.1)

Total: 11 (CRITICAL: 11)

LIBRARY	VULNERABILITY ID	SEVERITY	INSTALLED VERSION	FIXED VERSION	TITLE
curl	CVE-2021-22945	CRITICAL	7.74.0-1.3		curl: use-after-free and double-free in MQTT sending -->avd.aquasec.com/nvd/cve-2021-22945
libaom0	CVE-2021-30473		1.0.0.errata1-3		aom_image.c in libaom in AOMedia before 2021-04-07 frees memory that is not... -->avd.aquasec.com/nvd/cve-2021-30473
	CVE-2021-30474				aom_dsp/grain_table.c in libaom in AOMedia before 2021-03-30 has a use-after-free. -->avd.aquasec.com/nvd/cve-2021-30474
	CVE-2021-30475				aom_dsp/noise_model.c in libaom in AOMedia before 2021-03-24 has a buffer overflow. -->avd.aquasec.com/nvd/cve-2021-30475
libcurl3-gnutls	CVE-2021-22945		7.74.0-1.3		curl: use-after-free and double-free in MQTT sending -->avd.aquasec.com/nvd/cve-2021-22945
libcurl4					

Conclusion

Conclusion

- **Enhanced Security:**
  - early detection of potential security risks.
  - Hide sensitive data.
- **Efficiency and Consistency:**
  - Automation saves time and reducing manual errors.
  - Maintains consistency.
- **Quality Assurance:**
  - Ensures high-quality, secure code.
  - Admission control workflows prevent ensures only defect-free configurations are deployed.

References

- *Bhardwaj, P. (2023). Detecting Container vulnerabilities leveraging the CICD pipeline MSc Research Project Cybersecurity. https://norma.ncirl.ie/6512/1/preetibhardwaj.pdf*
- *Sultan, S., Ahmad, I., & Dimitriou, T. (2019). Container Security: Issues, Challenges, and the Road Ahead. IEEE Access, 7, 52976–52996. https://doi.org/10.1109/access.2019.2911732*
- *Tiwari, H. (2023, October 10). Enhancing Container Security Through Automated Vulnerability Scanning and Remediation with Trivy. Insights2Techinfo. https://insights2techinfo.com/enhancing-container-security-through-automated-vulnerability-scanning-and-remediation-with-trivy/*