



Image Scanning using Clair

DUSTIN VAN BUSKIRK

Dustin Van Buskirk

Senior Solutions Architect

 **codefresh**



Agenda

- Installing Clair
- Configure cfstep-paclair OOTB Step
- Setup Storage Integration
- Upload Clair Reports

Security is Critically Important



Anything you can do that helps with your security story which has a low cost should be a given.



The cost to remediate vulnerability is exponentially lower the sooner you address in Software Development Lifecycle.



Let's setup a simple step today together with minimal cost to operate to help us save on the price of vulnerabilities.



Where does Clair fit into CI/CD?

- Open Source project (low cost)
- Simple vulnerability scanning
- Prevent vulnerabilities from being introduced
- Reports to remediate vulnerabilities



Clair Reports

HTML report is created



Clair Scans in CI Pipeline

- Scan Docker image
- Fail build based on predetermined thresholds
- Report results back to Docker image




Clair Report Storage

Store Clair report for you build

Clair Helm Chart

Download/Configuration/Installation of Clair via Helm Chart
<https://github.com/coreos/clair/tree/master/contrib/helm>

 clair	clair	CLUSTER sales-prod@FirstKubernetes	REVISION 1	MODIFIED 6 days ago	CHART clair-0.1.4	DEPLOYED
Install complete						
<p>Clair is an open source project for the static analysis of vulnerabilities in application containers.</p> <p>https://github.com/coreos/clair</p>						
						▶ RUN TEST ✕ DELETE ↔ BADGE

YAML Step for Clair Scan

```
CheckClair:
  image: codefresh/cfstep-paclair:3.0.0
  environment:
    - IMAGE=example-voting-app/worker
    - TAG=${{CF_BRANCH_TAG_NORMALIZED}}-${{CF_SHORT_REVISION}}
  on_success:
    metadata:
      set:
        - ${{BuildingDockerImage.imageId}}:
            - CLAIR_REPORT: "https://g.codefresh.io/api/testReporting/s3/amazon/example-voting-app/${{CF_BUILD_ID}}/clair-scan-example-voting-app-
worker-${{CF_BRANCH_TAG_NORMALIZED}}-${{CF_SHORT_REVISION}}.html"
```

Annotating the Docker image with vulnerability information

Images

[Summary](#) [Dockerfile](#) [Logs](#) [Layers](#)

Image Name: example-voting-app/worker

Code

Repo: dustinvanbuskirk/example-voting-app
SHA: [662a528](#)
Branch: master
Last Commit: Merge pull request #71 from dustinvanbuskirk/new-branch-for-changes commit By: dustinvanbuskirk

Build Info



Image Size: 1.60 GB



Annotations

CLAIR_REPORT <https://eporotting-5509ting-i>

CLAIR_VULNS_LOW	2
CLAIR_VULNS_UNKNOWN	8
CLAIR_VULNS_NEGLIGIBLE	6
CLAIR_VULNS_HIGH	13
CLAIR_VULNS_MEDIUM	41

CLAIR_VULNS_LOW	2
CLAIR_VULNS_UNKNOWN	8
CLAIR_VULNS_NEGLIGIBLE	6
CLAIR_VULNS_HIGH	13
CLAIR_VULNS_MEDIUM	41

04/12/18 12:23 pm   dustinvanbuskirk: r.cfc.io/dustinvanbuskirk/example-voting-app/worker:master-662a528 [Copy](#)

04/12/18 12:22 pm   dustinvanbuskirk: Image created

Setup Storage Integration

Add Storage Integration to Codefresh

<https://codefresh.io/docs/docs/configure-ci-cd-pipeline/test-reports/#connecting-your-storage-account>

Upload Clair Reports



Add Report Archiving Step to Codefresh Pipeline









YAML to Upload Clair Report to Storage via Integration

```
ArchiveReport:
  title: Upload Clair Report
  image: codefresh/cf-docker-test-reporting
  working_directory: ./reports
  environment:
    - REPORT_INDEX_FILE=clair-scan-example-voting-app-worker--${CF_BRANCH_TAG_NORMALIZED}--${CF_SHORT_REVISION}.html
```

Accessing your stored reports from Codefresh



 **dustinvanbuskirk/example-voting-app: Merge pull request #71 from dustinvanbuskirk/new-branch-for-changes com...** [VIEW LOG](#) [TEST REPORT](#) 

dustinvanbuskirk  662a528  master  worker  4 minutes 0 second  12 minutes ago  Images  View YAML 

PAclair Security Reporting

NUMBER OF VULNERABILITIES BY RISK



ASSET VULNERABILITIES

Only show fixable

CVE	SEVERITY	PACKAGE	CURRENT VERSION	FIXED IN VERSION	INTRODUCED IN
> CVE-2011-3374	Negligible	apt	1.4.8		sha256:3e17c6eae66cd23c59751c8d8f5eaf7044e0611dc5cebb12b1273be07cdac242_912447e6cf6d
< CVE-2018-15473	Medium	openssh	1:7.4p1-10+deb9u1	1:7.4p1-10+deb9u4	sha256:4a1ed13b6faa4be7117a973f02c46398e98adfb4a2af34cb279fc5908e37ccba_912447e6cf6d

DESCRIPTION:

OpenSSH through 7.7 is prone to a user enumeration vulnerability due to not delaying bailout for an invalid authenticating user until after the packet containing the request has been fully parsed, related to auth2-gss.c, auth2-hostbased.c, and auth2-pubkey.c.

TECHNICAL IMPACT:

Access Vector	Access Complexity	Authentication	Confidentiality Impact	Integrity Impact	Availability Impact
Network	Low	None	Complete	Complete	Complete
Adjacent Network	Medium	Single	Partial	Partial	Partial
Local	High	Multiple	None	None	None

ADDITIONAL INFORMATION:

<https://security-tracker.debian.org/tracker/CVE-2018-15473>

> CVE-2018-15919	Medium	openssh	1:7.4p1-10+deb9u1		sha256:4a1ed13b6faa4be7117a973f02c46398e98adfb4a2af34cb279fc5908e37ccba_912447e6cf6d
> CVE-2008-3234	Negligible	openssh	1:7.4p1-10+deb9u1		sha256:4a1ed13b6faa4be7117a973f02c46398e98adfb4a2af34cb279fc5908e37ccba_912447e6cf6d
> CVE-2017-15906	Medium	openssh	1:7.4p1-10+deb9u1	1:7.4p1-10+deb9u3	sha256:4a1ed13b6faa4be7117a973f02c46398e98adfb4a2af34cb279fc5908e37ccba_912447e6cf6d
> CVE-2007-2243	Negligible	openssh	1:7.4p1-10+deb9u1		sha256:4a1ed13b6faa4be7117a973f02c46398e98adfb4a2af34cb279fc5908e37ccba_912447e6cf6d
> CVE-2007-2768	Negligible	openssh	1:7.4p1-10+deb9u1		sha256:4a1ed13b6faa4be7117a973f02c46398e98adfb4a2af34cb279fc5908e37ccba_912447e6cf6d
> CVE-2018-6003	Medium	libtasn1-6	4.10-1.1	4.10-1.1+deb9u1	sha256:74d44b20f851c8ef0b042070ba8eb018b386f50fdae5c37871d3fe7b4cfb4956_912447e6cf6d



Demo time

Questions?

Schedule a 1:1 with our
DevOps Experts

Sign up for FREE! 120
builds/month

[Codefresh.io](https://codefresh.io)



Thank You

See our upcoming Codefresh Live events at:

codefresh.io/events

