KubeCon | CloudNativeCon

Europe 2022

WELCOME TO VALENCIA

KubeCon CloudNativeCon Europe 2022

Digging Into Your App's Container Image Layers for Sneaky Vulnerabilities

Pablo Galego, VMware

Digging Into Your App's Container Image Layers for Sneaky Vulnerabilities

KubeCon CloudNativeCon Europe 2022

This session walks you through mitigating critical vulnerabilities in popular container images like Java-based ones, from the obvious to the sneaky ones, and how to leverage layer explorer tools to narrow the search field for the latter.

Pablo Galego

Software Engineer, VMware

The story behind this

We wanted to establish a **vulnerability baseline** for the Java microservices under development.

We needed a way to tell if under a new circumstance we were potentially worsening our security posture:

A baseline is what allows to make that judgement.

The story behind this

Good starting point…

- Limited number of micro-services, built with up-to-date technologies.
- Already in place a vulnerability scanning phase in CI running Aqua's Trivy.

…to achieve a sound goal:

- ✓ No fixable, critical vulnerabilities in production

Why is it relevant?

- Containers are an excellent tool for easing deployment and development, but bring their unique challenges to securing them.
- Containers may add to the developer's responsibility patching the OS vulnerabilities.
- This shift happened while containers are marketed as boxes that "just work" (why would you even want to look into them?)
- Many of the OSS vulnerability scanning tools are noisy by default and require some tweaking.

Objective of this talk

See the steps to mitigate the reported vulnerabilities of a Java container:

- The basics are very easy!
- As always, once we remove the magic, going that extra mile isn't that difficult also!

Let's quickly remove *part* of the magic

Keep in mind

Container images are sets of *layers*.

Effectively, only RUN, COPY and ADD instructions in a Dockerfile **create a new layer**.

```
FROM docker.io/bitnami/minideb:buster
LABEL maintainer "Pablo Galego"
ENV OS="amd64-linux-debian10"
COPY empty_file /
RUN install_packages ca-certificates curl
USER 1001
ENTRYPOINT [ "curl" ]
CMD [ "--help" ]
```

Keep in mind

Container images are sets of *layers*.

Effectively, only RUN, COPY and ADD instructions in a Dockerfile **create a new layer**.

```
❯ docker build . -t pablo/curl
[+] Building 7.8s (8/8) FINISHED
 => [internal] load build definition from Dockerfile                0.0s
 => => transferring dockerfile: 253B                                0.0s
 => [internal] load .dockerignore                                   0.0s
 => => transferring context: 2B                                     0.0s
 => [internal] load metadata for docker.io/bitnami/minideb:buster   1.4s
 => [internal] load build context                                   0.0s
 => => transferring context: 31B                                    0.0s
 => CACHED [1/3] FROM docker.io/bitnami/minideb:buster              0.0s
 => [2/3] COPY empty_file /                                         0.0s
 => [3/3] RUN install_packages ca-certificates curl                 6.2s
 => exporting to image                                              0.1s
 => => exporting layers                                             0.1s
 => => writing image                                                0.0s
 => => naming to docker.io/pablo/curl                               0.0s
```

Keep in mind

Layers are… just folders and configuration files

```
❯ docker save -o curl.tar pablo/curl && tar tvf curl.tar
drwxr-xr-x  0 0      0           0 Mar 31 15:40 8f3c172c/
-rw-r--r--  0 0      0           3 Mar 31 15:40 8f3c172c/VERSION
-rw-r--r--  0 0      0         931 Mar 31 15:40 8f3c172c/json
-rw-r--r--  0 0      0    13505536 Mar 31 15:40 8f3c172c/layer.tar
-rw-r--r--  0 0      0        1633 Mar 31 15:40 cf1d166b.json
drwxr-xr-x  0 0      0           0 Mar 31 15:40 ef0f194d/
-rw-r--r--  0 0      0           3 Mar 31 15:40 ef0f194d/VERSION
-rw-r--r--  0 0      0         401 Mar 31 15:40 ef0f194d/json
-rw-r--r--  0 0      0    70799360 Mar 31 15:40 ef0f194d/layer.tar
drwxr-xr-x  0 0      0           0 Mar 31 15:40 f3c0722b/
-rw-r--r--  0 0      0           3 Mar 31 15:40 f3c0722b/VERSION
-rw-r--r--  0 0      0         477 Mar 31 15:40 f3c0722b/json
-rw-r--r--  0 0      0        1536 Mar 31 15:40 f3c0722b/layer.tar
-rw-r--r--  0 0      0         360 Jan  1  1970 manifest.json
-rw-r--r--  0 0      0          93 Jan  1  1970 repositories
```

Playground

- **demo-service**: A dummy artifact representing a microservice with a plugin system, built with Maven.
- **demo-service-plugins/hello-plugin**: A plugin that says hello.
- **container-library**: A set of container images for building the demo-service.

Find it at

pablogalegoc/kubecon2022-playground

Let’s see it live

The following slides will not be part of the final presentation.

They are just a sneak peak on what we will address in the demo.

Demo

Building the demo-service image: first, the builder image


```
~/KubeCon2022/container-library/java-16/maven on  main
❯ docker build . -t local/maven:3.8.5-java16
[+] Building 2.6s (7/7) FINISHED
 [ ... ]
 ⇒ CACHED [1/2] FROM docker.io/bitnami/java:16.0.2-debian-10-r82                0.0s
 ⇒ [2/2] RUN --mount=source=install-maven.sh,target=/provision/install-maven.sh
   DEST_DIR=/opt/maven MAVEN_VERSION=3.8.5     /provision/install-maven.sh       1.8s
 [ ... ]
 ⇒ ⇒ naming to docker.io/local/maven:3.8.5-java17                               0.0s
```

Demo

Building the demo-service image: then, the app image

Using dockerfile-maven plugin

```
~/KubeCon2022/demo-service on ᚠ main
❯ ./mvnw install
[INFO] ...
[INFO] Successfully built demo-service-boot:0.1.0-SNAPSHOT
[INFO] ...
[INFO] Reactor Summary for KubeCon: Demo Service 0.1.0-SNAPSHOT:
[INFO]
[INFO] KubeCon: Demo Service ........... SUCCESS [  1.251 s]
[INFO] Demo Service Core ............... SUCCESS [  0.671 s]
[INFO] Demo Service Infrastructure ..... SUCCESS [  0.632 s]
[INFO] Demo Service API ................ SUCCESS [  0.262 s]
[INFO] Demo Service Boot ............... SUCCESS [01:10 min]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
```

Demo

Refining the Trivy command for a convenient output


```
~/KubeCon2022 on  main
❯ trivy image demo-service-boot:0.1.0-SNAPSHOT
❌  ~200 vulnerabilities, we have to prioritize.

❯ trivy image --severity CRITICAL demo-service-boot:0.1.0-SNAPSHOT
❌  26 total, 15 from the base image. We can filter better because there's nothing we can do here:
    +---------------+--------------------+----------+-------------------------+-------------------+
    |    LIBRARY    | VULNERABILITY ID | SEVERITY |    INSTALLED VERSION    |  FIXED VERSION    |
    +---------------+--------------------+----------+-------------------------+-------------------+
    | libc-bin      | CVE-2021-33574   | CRITICAL |          2.28-10        |                   |
    +---------------+--------------------+----------+-------------------------+-------------------+

❯ trivy -q image --severity CRITICAL --ignore-unfixed demo-service-boot:0.1.0-SNAPSHOT
✅  13 critical vulnerabilities that can be addressed right now
```

Demo

Address base image vulnerabilities

```
~/KubeCon2022/demo-service on  main
❯ trivy -q image --severity CRITICAL --ignore-unfixed demo-service-boot:0.1.0-SNAPSHOT

demo-service-boot:0.1.0-SNAPSHOT (debian 10.10)
═══════════════════════════════════════════════
Total: 2 (CRITICAL: 2)

+─────────────+──────────────────+──────────+──────────────────+──────────────────+
|   LIBRARY   | VULNERABILITY ID | SEVERITY | INSTALLED VERSION |  FIXED VERSION   |
+─────────────+──────────────────+──────────+──────────────────+──────────────────+
| libssl1.1   | CVE-2021-3711    | CRITICAL | 1.1.1d-0+deb10u6  | 1.1.1d-0+deb10u7 |
+─────────────+                  +          +                  +                  +
| openssl     |                  |          |                  |                  |
+─────────────+──────────────────+──────────+──────────────────+──────────────────+

~/KubeCon2022/demo-service on  main
❯ git diff src/main/dist/Dockerfile

 18 : 18 |RUN java -Djarmode=layertools -jar application.jar extract
 19 : 19 |
 20 :    |FROM openjdk:16.0.1-slim-buster
    : 20 |FROM openjdk:16.0.2-slim-buster
 21 : 21 |
 22 : 22 |ARG AUTH
```

Demo

Address runtime dependencies' vulnerabilities


```
~/KubeCon2022/demo-service on ᚠ main
❯ trivy -q image --severity CRITICAL --ignore-unfixed demo-service-boot:0.1.0-SNAPSHOT

usr/local/bin/helm (gobinary)
═══════════════════════════════
Total: 1 (CRITICAL: 1)

+──────────────────────────────────+──────────────────+──────────+──────────────────+──────────────────+
|             LIBRARY              | VULNERABILITY ID | SEVERITY | INSTALLED VERSION |  FIXED VERSION   |
+──────────────────────────────────+──────────────────+──────────+──────────────────+──────────────────+
| github.com/containerd/containerd | CVE-2021-43816   | CRITICAL | v1.5.7            | 1.5.9            |
+──────────────────────────────────+──────────────────+──────────+──────────────────+──────────────────+

~/KubeCon2022/demo-service on ᚠ main
❯ git diff src/main/dist/Dockerfile

  8 ⋮  8 │SHELL ["/bin/bash", "-o", "pipefail", "-c"]
  9 ⋮  9 │RUN tmp_file=$(mktemp)  \
 10 ⋮    │    && wget --progress=dot:giga -O "${tmp_file}" https://get.helm.sh/helm-v3.7.1-linux-amd64.tar.gz \
    ⋮ 10 │    && wget --progress=dot:giga -O "${tmp_file}" https://get.helm.sh/helm-v3.8.1-linux-amd64.tar.gz \
 11 ⋮ 11 │    && tar zxf "${tmp_file}" -C "/tmp" linux-amd64/helm
 12 ⋮ 12 │
```

Demo

Address Spring Boot dependencies' vulnerabilities

```
~/KubeCon2022/demo-service on  main
❯ trivy -q image --severity CRITICAL --ignore-unfixed demo-service-boot:0.1.0-SNAPSHOT

Java (jar)
==========
Total: 10 (CRITICAL: 10)

+-----------------------------------------+------------------+----------+-------------------+-----------------+
|                LIBRARY                  | VULNERABILITY ID | SEVERITY | INSTALLED VERSION |  FIXED VERSION  |
+-----------------------------------------+------------------+----------+-------------------+-----------------+
| ...                                     | ...              | ...      | ...               | ...             |
+-----------------------------------------+------------------+----------+-------------------+-----------------+
| org.springframework.boot:spring-boot    | CVE-2022-22965   | CRITICAL | 2.6.2             | 2.5.12, 2.6.6   |
+-----------------------------------------+                  +          +-------------------+-----------------+
| org.springframework:spring-beans        |                  |          | 5.3.14            | 5.3.18, 5.2.20  |
+-----------------------------------------+                  +          +                   +                 +
| org.springframework:spring-webmvc       |                  |          |                   |                 |
+-----------------------------------------+------------------+----------+-------------------+-----------------+

~/KubeCon2022/demo-service on  main
❯ git diff src/main/dist/pom.xml

 305: 305|    <resilience4j.version>1.6.1</resilience4j.version>
 306: 306|    <testcontainers.version>1.16.2</testcontainers.version>
 307:     |    <spring-boot.version>2.6.2</spring-boot.version>
     : 307|    <spring-boot.version>2.6.6</spring-boot.version>
 308: 308|  </properties>
 309: 309|
```

Demo

Only xstream left... but where is it?


```
~/KubeCon2022/demo-service on ᚠ main
❯ trivy -q image --severity CRITICAL --ignore-unfixed demo-service-boot:0.1.0-SNAPSHOT

Java (jar)
══════════
Total: 10 (CRITICAL: 10)

+-----------------------------------+-----------------+----------+-------------------+---------------+
|              LIBRARY              | VULNERABILITY ID | SEVERITY | INSTALLED VERSION | FIXED VERSION |
+-----------------------------------+-----------------+----------+-------------------+---------------+
| com.thoughtworks.xstream:xstream  | CVE-2021-21342  | CRITICAL | 1.4.15            | 1.4.16        |
+                                   +-----------------+          +                   +               +
|                                   | CVE-2021-21344  |          |                   |               |
+                                   +-----------------+          +                   +               +
|                                   | CVE-2021-21345  |          |                   |               |
+                                   +-----------------+          +                   +               +
|                                   | CVE-2021-21346  |          |                   |               |
+                                   +-----------------+          +                   +               +
|                                   | CVE-2021-21347  |          |                   |               |
+                                   +-----------------+          +                   +               +
|                                   | CVE-2021-21350  |          |                   |               |
+                                   +-----------------+          +                   +               +
|                                   | CVE-2021-21351  |          |                   |               |
+-----------------------------------+-----------------+----------+-------------------+---------------+
```

Demo

Only xstream left… but where is it?

Demo

We'll have to dive in to see what we find...


```
~/KubeCon2022/demo-service on  main
❯ dive demo-service-boot:0.1.0-SNAPSHOT
Image Source: docker://demo-service-boot:0.1.0-SNAPSHOT
Fetching image... (this can take a while for large images)
Analyzing image...
Building cache...

● Layers
Cmp   Size  Command
     69 MB  FROM 970b29bb2c78938
    8.8 MB  set -eux; apt-get update; apt-get install -y --no-install-recommends ca-certificates p11-kit;
    318 MB  set -eux; arch="$(dpkg --print-architecture)"; case "$arch" in 'amd64') downloadUrl...
    334 kB  |4 AUTH=ups-this-is-a-secret EMAIL=pgalego@kube.con vs_gid=1000 vs_uid=1000 /bin/sh -c groupadd...
      0 B   |4 AUTH=ups-this-is-a-secret EMAIL=pgalego@kube.con vs_gid=1000 vs_uid=1000 /bin/sh -c mkdir  ...
      0 B   |4 AUTH=ups-this-is-a-secret EMAIL=pgalego@kube.con vs_gid=1000 vs_uid=1000 /bin/sh -c mkdir /app...
     45 MB  #(nop) COPY file:8d3fc8de879373a45d2bc4beae8833ed8bdc12fcb6347fd6f3044b1e6f5c66d7 in /usr/local...
     60 MB  #(nop) COPY dir:a290b91078ef3a11604216c43b5e73783d9ea3004fa5c77db9ac7fa9d316772c in /app/
    252 kB  #(nop) COPY dir:6b7582a0ba151e59e2fd936fa4cd9b41f20011948bd6b85c01a2d3426927b20c in /app/
      0 B   #(nop) COPY dir:3875f37b8a0ed74947e5279c54167139802f07290f41d7ffa0e4a72b52d6168f in /app/
     27 kB  #(nop) COPY dir:b7b59093daeec62b51f4b66266a1aeb96cc3d701c620cbeb5751bb619ace2f4c in /app/
     46 MB  #(nop) COPY dir:b208b78e0c636a7d1387f7227ec59b038e5c62ce84d14c5f1f92dab36443a312 in /plugins/
```

End of the demo

Conclusions

- Container images are just collections of files, don't be afraid of unpacking them and looking inside!
- Many vulnerability scanning tools need fine-tuning to provide a useful output
- Mitigating the reported vulnerabilities is often an easy task: Tune the scan, set sensible expectations and get the job done

Thank you!