**Step 1 - Running container scan on base image requested by prospect (Jenkins/agent:jdk17)**

Dockerfile:

### ==== Base Image ====

FROM jenkins/agent:jdk17

LABEL image="jenkins/inbound-agent:latest-jdk17" distro="debian" version="0.2.2-secured"

### ==== Configuración Inicial ====

USER root

RUN echo $PATH

### ==== Verificar Java ====

RUN java -version

WORKDIR /scan

RUN useradd -m -d /home/luser -s /bin/bash luser

USER luser

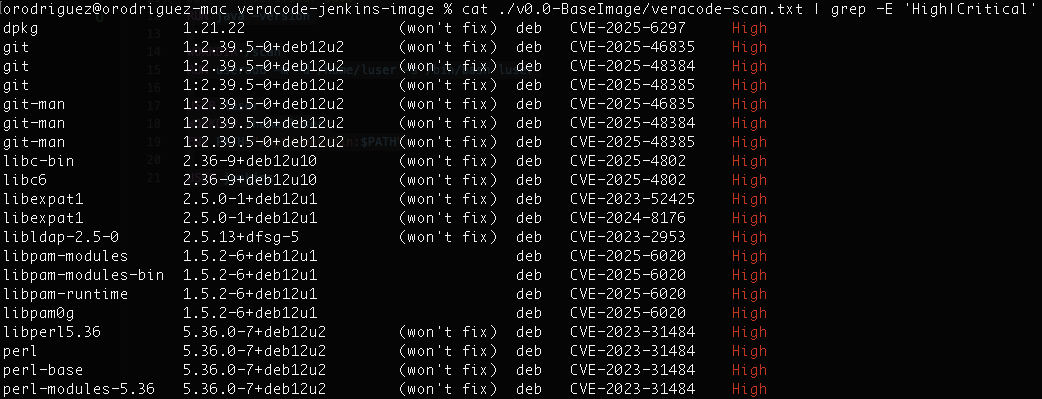
WORKDIR /home/luser

ENV PATH="/opt/venv/bin:$PATH"

USER jenkins

After build, it generated image veracode-agente:v0.0-BaseImage.

Container scan results with Veracode Container Security (available in /v0.0-BaseImage/Veracode-scan.txt)



These are vulnerabilities brought from base image so the prospect will remediate or mitigate them to proceed with POV. Now, next steps of these tests will run on top of this base image.

**Step 2 - Running container scan on new image after installing basic tools (Jenkins/agent:jdk17 + Basic Tools)**

Dockerfile:

### ==== Base Image ====

FROM jenkins/agent:jdk17

LABEL image="jenkins/inbound-agent:latest-jdk17" distro="debian" version="0.2.2-secured"

### ==== Configuración Inicial ====

USER root

# Update and install all packages in a single layer to reduce image size

RUN apt-get update && \

apt-get upgrade -y && \

apt-get install -y --no-install-recommends \

curl \

gnupg \

unzip \

git \

git-lfs \

perl \

openssl \

xz-utils \

libfreetype6 \

gnutls-bin \

zip \

bash \

gradle \

maven && \

# Remove vulnerable packages

apt-get purge -y \

libbcel-java \

libjgit-java && \

rm -rf /usr/share/java/{bcel,jgit}\* && \

# Attempt sqlite3 upgrade (with fallback)

(apt-get purge -y libsqlite3-0 && \

apt-get install -y --no-install-recommends libsqlite3-0=3.40.1-2+deb12u2 || \

echo "Warning: Could not install updated sqlite3 package, proceeding anyway") && \

# Clean up

apt-get autoremove -y && \

apt-get clean && \

rm -rf /var/lib/apt/lists/\*

RUN echo $PATH

### ==== Verificar Java ====

RUN java -version

WORKDIR /scan

RUN useradd -m -d /home/luser -s /bin/bash luser

USER luser

WORKDIR /home/luser

ENV PATH="/opt/venv/bin:$PATH"

USER jenkins

After build, it generated image veracode-agente:v0.1-BasicTools

Container scan results with Veracode Container Security (available in /v0.1-BasicTools/veracode-scan.txt)

A screenshot of a computer program

AI-generated content may be incorrect.

Comparing results (High and Critical vulns) between “veracode-agente:v0.0-BaseImage” and “Veracode-agente:v0.1-BasicTools”, there are no difference in reported vulnerabilities:

A screenshot of a computer

AI-generated content may be incorrect.

**Step 3 - Running container scan on new image after installing basic tools and Pipeline Scan (Jenkins/agent:jdk17 + Basic Tools + Pipeline Scan)**

Dockerfile:

### ==== Base Image ====

FROM jenkins/agent:jdk17

LABEL image="jenkins/inbound-agent:latest-jdk17" distro="debian" version="0.2.2-secured"

### ==== Configuración Inicial ====

USER root

# Update and install all packages in a single layer to reduce image size

RUN apt-get update && \

apt-get upgrade -y && \

apt-get install -y --no-install-recommends \

curl \

gnupg \

unzip \

git \

git-lfs \

perl \

openssl \

xz-utils \

libfreetype6 \

gnutls-bin \

zip \

bash \

gradle \

maven && \

# Remove vulnerable packages

apt-get purge -y \

libbcel-java \

libjgit-java && \

rm -rf /usr/share/java/{bcel,jgit}\* && \

# Attempt sqlite3 upgrade (with fallback)

(apt-get purge -y libsqlite3-0 && \

apt-get install -y --no-install-recommends libsqlite3-0=3.40.1-2+deb12u2 || \

echo "Warning: Could not install updated sqlite3 package, proceeding anyway") && \

# Clean up

apt-get autoremove -y && \

apt-get clean && \

rm -rf /var/lib/apt/lists/\*

RUN echo $PATH

### ==== Verificar Java ====

RUN java -version

### ==== Veracode Pipeline Scan (Versión Segura) ====

ENV VERACODE\_DIR=/opt/veracode \

PIPELINE\_SCAN\_JAR=/opt/veracode/pipeline-scan.jar \

PIPELINE\_SCAN\_URL="https://downloads.veracode.com/securityscan/pipeline-scan-LATEST.zip"

RUN mkdir -p ${VERACODE\_DIR} && \

useradd -m veracode && \

curl -sSL ${PIPELINE\_SCAN\_URL} -o /tmp/pipeline-scan.zip && \

unzip -p /tmp/pipeline-scan.zip pipeline-scan.jar > ${PIPELINE\_SCAN\_JAR} && \

rm /tmp/pipeline-scan.zip && \

chmod 755 ${PIPELINE\_SCAN\_JAR}

WORKDIR /scan

RUN useradd -m -d /home/luser -s /bin/bash luser

USER luser

WORKDIR /home/luser

ENV PATH="/opt/venv/bin:$PATH"

USER jenkins

After build, it generated image veracode-agente:v0.2-PS

Container scan results with Veracode Container Security (available in /v0.2-PS/veracode-scan.txt)

A screenshot of a computer program

AI-generated content may be incorrect.

**Comparing results (High and Critical vulns) between “veracode-agente:v0.1-BasicTools” and “veracode-agente:v0.2-PS”, there are 5 new vulns after installing Pipeline Scan:**

A screenshot of a computer screen

AI-generated content may be incorrect.

**Step 4 - Running container scan on new image after installing basic tools and Java API Wrapper (Jenkins/agent:jdk17 + Basic Tools + Java API Wrapper)**

Dockerfile:

### ==== Base Image ====

FROM jenkins/agent:jdk17

LABEL image="jenkins/inbound-agent:latest-jdk17" distro="debian" version="0.2.2-secured"

### ==== Configuración Inicial ====

USER root

# Update and install all packages in a single layer to reduce image size

RUN apt-get update && \

apt-get upgrade -y && \

apt-get install -y --no-install-recommends \

curl \

gnupg \

unzip \

git \

git-lfs \

perl \

openssl \

xz-utils \

libfreetype6 \

gnutls-bin \

zip \

bash \

gradle \

maven && \

# Remove vulnerable packages

apt-get purge -y \

libbcel-java \

libjgit-java && \

rm -rf /usr/share/java/{bcel,jgit}\* && \

# Attempt sqlite3 upgrade (with fallback)

(apt-get purge -y libsqlite3-0 && \

apt-get install -y --no-install-recommends libsqlite3-0=3.40.1-2+deb12u2 || \

echo "Warning: Could not install updated sqlite3 package, proceeding anyway") && \

# Clean up

apt-get autoremove -y && \

apt-get clean && \

rm -rf /var/lib/apt/lists/\*

RUN echo $PATH

### ==== Verificar Java ====

RUN java -version

### ==== Veracode API Wrapper ====

ENV VERSION="25.8.16.1" \

ARTIFACT\_NAME="vosp-api-wrappers-java" \

INSTALL\_DIR="/opt/veracode"

RUN mkdir -p ${INSTALL\_DIR} && \

curl -L -o /tmp/wrapper.zip \

"https://repo1.maven.org/maven2/com/veracode/vosp/api/wrappers/${ARTIFACT\_NAME}/${VERSION}/${ARTIFACT\_NAME}-${VERSION}-dist.zip" && \

unzip /tmp/wrapper.zip -d ${INSTALL\_DIR} && \

rm /tmp/wrapper.zip

WORKDIR /scan

RUN useradd -m -d /home/luser -s /bin/bash luser

USER luser

WORKDIR /home/luser

ENV PATH="/opt/venv/bin:$PATH"

USER jenkins

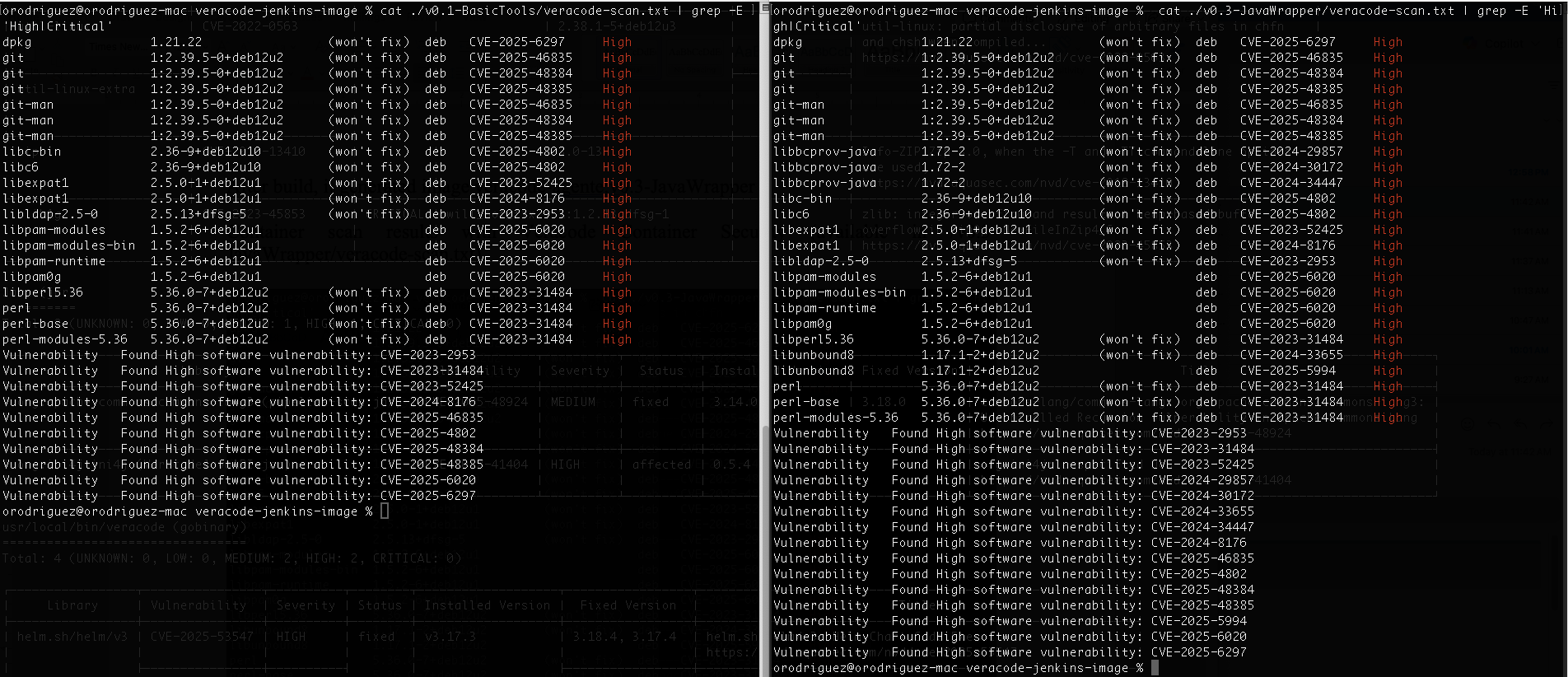
After build, it generated image veracode-agente:v0.3-JavaWrapper

Container scan results with Veracode Container Security (available in /v0.3-JavaWrapper/veracode-scan.txt)

A screenshot of a computer

AI-generated content may be incorrect.

**Comparing results (High and Critical vulns) between “veracode-agente:v0.1-BasicTools” and “veracode-agente:v0.3-JavaWrapper”, there are 5 new vulns after installing Java API Wrapper; these are the same 5 new vulns are being injected when installing Pipeline Scan in Step-3 of this document:**



**Step 5 - Running container scan on new image after installing basic tools and HMAC API Signing (Jenkins/agent:jdk17 + Basic Tools + api-signing)**

Dockerfile  
### ==== Base Image ====

FROM jenkins/agent:jdk17

LABEL image="jenkins/inbound-agent:latest-jdk17" distro="debian" version="0.2.2-secured"

### ==== Configuración Inicial ====

USER root

# Update and install all packages in a single layer to reduce image size

RUN apt-get update && \

apt-get upgrade -y && \

apt-get install -y --no-install-recommends \

curl \

gnupg \

unzip \

git \

git-lfs \

perl \

openssl \

xz-utils \

libfreetype6 \

gnutls-bin \

zip \

bash \

gradle \

maven && \

# Remove vulnerable packages

apt-get purge -y \

libbcel-java \

libjgit-java && \

rm -rf /usr/share/java/{bcel,jgit}\* && \

# Attempt sqlite3 upgrade (with fallback)

(apt-get purge -y libsqlite3-0 && \

apt-get install -y --no-install-recommends libsqlite3-0=3.40.1-2+deb12u2 || \

echo "Warning: Could not install updated sqlite3 package, proceeding anyway") && \

# Clean up

apt-get autoremove -y && \

apt-get clean && \

rm -rf /var/lib/apt/lists/\*

RUN echo $PATH

### ==== Verificar Java ====

RUN java -version

### ==== Veracode API Signing ====

RUN apt-get update && apt-get install -y --no-install-recommends \

python3.11 \

python3.11-venv \

python3-pip && \

rm -rf /var/lib/apt/lists/\* && \

update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.11 1 && \

python3 -m venv /opt/venv && \

/opt/venv/bin/pip install --upgrade pip==24.0 && \

/opt/venv/bin/pip install --upgrade setuptools==78.1.1 && \

/opt/venv/bin/pip install --upgrade urllib3==2.5.0 && \

/opt/venv/bin/pip install veracode-api-signing

WORKDIR /scan

RUN useradd -m -d /home/luser -s /bin/bash luser

USER luser

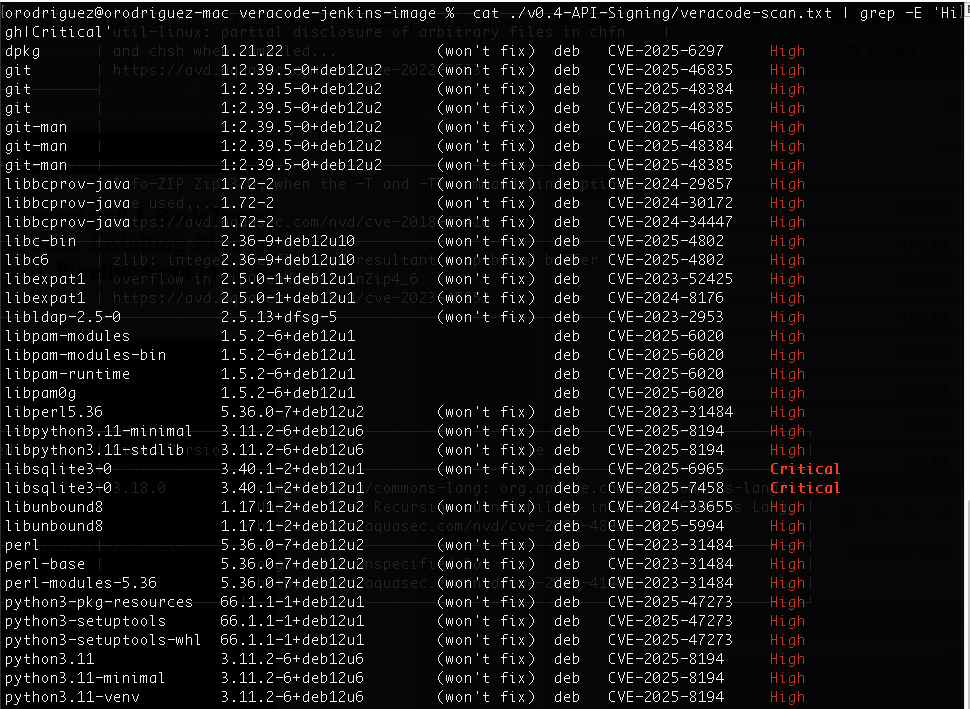
WORKDIR /home/luser

ENV PATH="/opt/venv/bin:$PATH"

USER jenkins

After build, it generated image veracode-agente:v0.4-API-Signing

Container scan results with Veracode Container Security (available in /v0.4-API-Signing/veracode-scan.txt)



**Comparing results (High and Critical vulns) between “veracode-agente:v0.1-BasicTools” and “veracode-agente:v0.4-API-Signing”, there are 15 new vulns after installing api--signing:**

A screenshot of a computer

AI-generated content may be incorrect.

**Step 6 - Running container scan on new image after installing basic tools and Veracode tools (Jenkins/agent:jdk17 + Basic Tools + Veracode CLI)**

**Dockerfile**

### ==== Base Image ====

FROM jenkins/agent:jdk17

LABEL image="jenkins/inbound-agent:latest-jdk17" distro="debian" version="0.2.2-secured"

### ==== Configuración Inicial ====

USER root

# Update and install all packages in a single layer to reduce image size

RUN apt-get update && \

apt-get upgrade -y && \

apt-get install -y --no-install-recommends \

curl \

gnupg \

unzip \

git \

git-lfs \

perl \

openssl \

xz-utils \

libfreetype6 \

gnutls-bin \

zip \

bash \

gradle \

maven && \

# Remove vulnerable packages

apt-get purge -y \

libbcel-java \

libjgit-java && \

rm -rf /usr/share/java/{bcel,jgit}\* && \

# Attempt sqlite3 upgrade (with fallback)

(apt-get purge -y libsqlite3-0 && \

apt-get install -y --no-install-recommends libsqlite3-0=3.40.1-2+deb12u2 || \

echo "Warning: Could not install updated sqlite3 package, proceeding anyway") && \

# Clean up

apt-get autoremove -y && \

apt-get clean && \

rm -rf /var/lib/apt/lists/\*

RUN echo $PATH

### ==== Veracode CLI ====

RUN mkdir -p /app && \

cd /app && \

curl -fsS https://tools.veracode.com/veracode-cli/install | sh && \

find / -name veracode -type f -exec mv {} /usr/local/bin/ \; && \

chmod +x /usr/local/bin/veracode && \

rm -rf /app

### ==== Verificar Java ====

RUN java -version

WORKDIR /scan

RUN useradd -m -d /home/luser -s /bin/bash luser

USER luser

WORKDIR /home/luser

ENV PATH="/opt/venv/bin:$PATH"

USER jenkins

After build, it generated image veracode-agente:v0.5-CLI

Container scan results with Veracode Container Security (available in /v0.5-CLI/veracode-scan.txt)

**A screenshot of a computer

AI-generated content may be incorrect.**

**Comparing results (High and Critical vulns) between “veracode-agente:v0.1-BasicTools” and “veracode-agente:v0.5-CLI”, there are 6 new vulns after installing Veracode CLI (GHSA-557j-xg8c-q2mm it’s the same vuln CVE-2025-53547 -** [**https://github.com/advisories/GHSA-557j-xg8c-q2mm**](https://github.com/advisories/GHSA-557j-xg8c-q2mm) **):**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**Step 7 - Running container scan on new image after installing basic tools and Veracode tools (Jenkins/agent:jdk17 + Basic Tools + Pipeline Scan + Java API Wrapper + api-signing + CLI)**

**Dockerfile:**

### ==== Base Image ====

FROM jenkins/agent:jdk17

LABEL image="jenkins/inbound-agent:latest-jdk17" distro="debian" version="0.2.2-secured"

### ==== Configuración Inicial ====

USER root

# Update and install all packages in a single layer to reduce image size

RUN apt-get update && \

apt-get upgrade -y && \

apt-get install -y --no-install-recommends \

curl \

gnupg \

unzip \

git \

git-lfs \

perl \

openssl \

xz-utils \

libfreetype6 \

gnutls-bin \

zip \

bash \

gradle \

maven && \

# Remove vulnerable packages

apt-get purge -y \

libbcel-java \

libjgit-java && \

rm -rf /usr/share/java/{bcel,jgit}\* && \

# Attempt sqlite3 upgrade (with fallback)

(apt-get purge -y libsqlite3-0 && \

apt-get install -y --no-install-recommends libsqlite3-0=3.40.1-2+deb12u2 || \

echo "Warning: Could not install updated sqlite3 package, proceeding anyway") && \

# Clean up

apt-get autoremove -y && \

apt-get clean && \

rm -rf /var/lib/apt/lists/\*

RUN echo $PATH

### ==== Veracode CLI ====

RUN mkdir -p /app && \

cd /app && \

curl -fsS https://tools.veracode.com/veracode-cli/install | sh && \

find / -name veracode -type f -exec mv {} /usr/local/bin/ \; && \

chmod +x /usr/local/bin/veracode && \

rm -rf /app

### ==== Verificar Java ====

RUN java -version

### ==== Veracode Pipeline Scan (Versión Segura) ====

ENV VERACODE\_DIR=/opt/veracode \

PIPELINE\_SCAN\_JAR=/opt/veracode/pipeline-scan.jar \

PIPELINE\_SCAN\_URL="https://downloads.veracode.com/securityscan/pipeline-scan-LATEST.zip"

RUN mkdir -p ${VERACODE\_DIR} && \

useradd -m veracode && \

curl -sSL ${PIPELINE\_SCAN\_URL} -o /tmp/pipeline-scan.zip && \

unzip -p /tmp/pipeline-scan.zip pipeline-scan.jar > ${PIPELINE\_SCAN\_JAR} && \

rm /tmp/pipeline-scan.zip && \

chmod 755 ${PIPELINE\_SCAN\_JAR}

RUN apt-get update && apt-get install -y --no-install-recommends \

python3.11 \

python3.11-venv \

python3-pip && \

rm -rf /var/lib/apt/lists/\* && \

update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.11 1 && \

python3 -m venv /opt/venv && \

/opt/venv/bin/pip install --upgrade pip==24.0 && \

/opt/venv/bin/pip install --upgrade setuptools==78.1.1 && \

/opt/venv/bin/pip install --upgrade urllib3==2.5.0 && \

/opt/venv/bin/pip install veracode-api-signing

### ==== Veracode API Wrapper ====

ENV VERSION="25.8.16.1" \

ARTIFACT\_NAME="vosp-api-wrappers-java" \

INSTALL\_DIR="/opt/veracode"

RUN mkdir -p ${INSTALL\_DIR} && \

curl -L -o /tmp/wrapper.zip \

"https://repo1.maven.org/maven2/com/veracode/vosp/api/wrappers/${ARTIFACT\_NAME}/${VERSION}/${ARTIFACT\_NAME}-${VERSION}-dist.zip" && \

unzip /tmp/wrapper.zip -d ${INSTALL\_DIR} && \

rm /tmp/wrapper.zip

### ==== validacion.sh y validación de jackson ====

#COPY validacion.sh /usr/local/bin/validacion.sh

#RUN chmod +x /usr/local/bin/validacion.sh && \

# sed -i 's/\r$//' /usr/local/bin/validacion.sh

WORKDIR /scan

RUN useradd -m -d /home/luser -s /bin/bash luser

USER luser

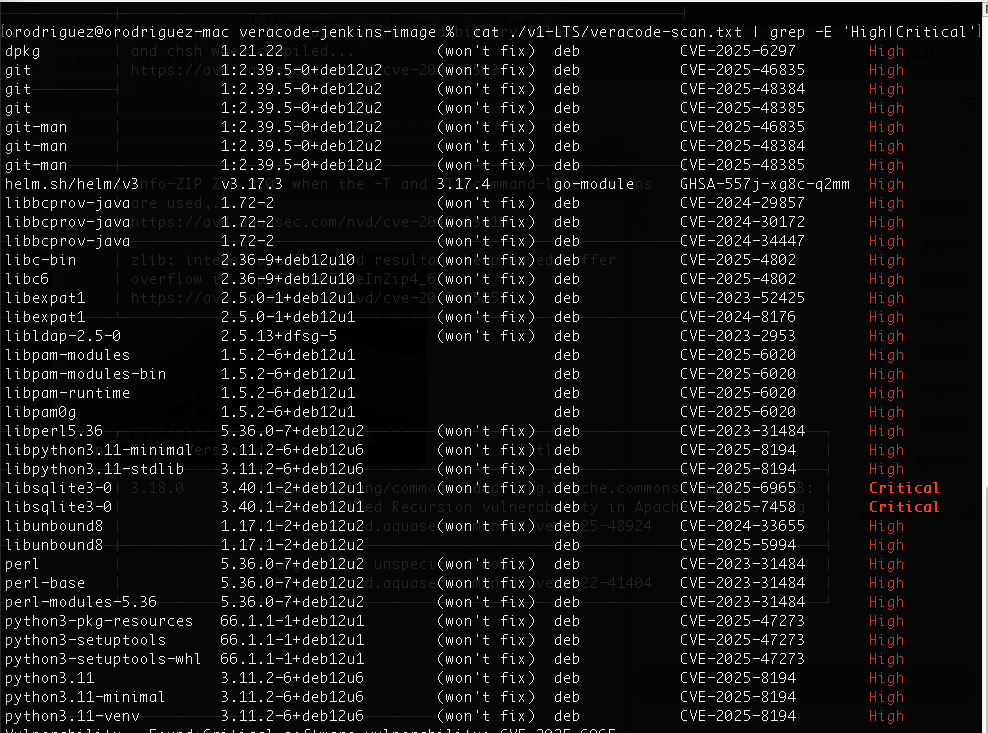
WORKDIR /home/luser

ENV PATH="/opt/venv/bin:$PATH"

USER jenkins

After build, it generated image veracode-agente:latest

Container scan results with Veracode Container Security (available in /v1-LTS/veracode-scan.txt)

****

**Comparing results (High and Critical vulns) between “veracode-agente:v0.1-BasicTools” and “veracode-agente:latest”, there are 16 new vulns after installing Veracode tools (Pipeline Scan, Java API Wrapper, api—signing and CLI):**

**A screenshot of a computer

AI-generated content may be incorrect.**

**IN SUMMARY:**

* When installing Pipeline Scan, Java API Wrapper, api-signing and Veracode CLI, all of them are injecting the following High vulns:
  + CVE-2024-29857
  + CVE-2024-30172
  + CVE-2024-34447
  + CVE-2024-33655
  + CVE-2024-5994
* When installing api-signing, these other High vulns are arising:
  + CVE-2025-8194 (in libraries libpython3.11-minimal and libpython3.11-stdlib)
  + CVE-2025-6965 (Critical)
  + CVE-2025-7458 (Critical)
  + CVE-2025-47273 (in libraries python3-pkg-resources, python3-setuptools and python3-setuptools-whl)
  + CVE-2025-8194 (in libraries python3.11, python3.11-minimal and python3.11-venv)
* When installing Veracode CLI, this additional High vuln arises:
  + GHSA-557j-xg8c-q2mm (It’s the same CVE-2025-53547 - <https://github.com/advisories/GHSA-557j-xg8c-q2mm> )