**. Update Base Images:**

* **Action:** Regularly update your base container images to the latest secure versions. Vulnerabilities are often found and fixed in the underlying operating systems or core libraries of base images.
* **Action:** Choose minimal base images that only contain necessary components to reduce the attack surface.

**2. Rebuild Images with Patched Software Components:**

* **Action:** For each identified vulnerable component (Curl, Zlib, Golang Crypto, Apache MINA, AsyncHttpClient, Apache Tomcat), update them to the recommended patched versions *within your Dockerfile or container build process*.
* **Action:** After updating the components, rebuild your container images to include these fixes.
* **Example (Conceptual Dockerfile snippet):**

Dockerfile

# For a component updated via package manager

RUN apt-get update && apt-get install --only-upgrade -y curl libz1

# For a language-specific library (e.g., Golang)

# Ensure your build process fetches the updated Go module

# RUN go get golang.org/x/crypto@v0.31.0

# For applications like Tomcat, ensure you are deploying the fixed version

# COPY --from=tomcat:9.0.99 /usr/local/tomcat /usr/local/tomcat

**3. Implement Container Image Scanning:**

* **Action:** Integrate static container image vulnerability scanning tools (e.g., Trivy, Clair, Aqua Security, Snyk) into your CI/CD pipeline.
* **Action:** Configure scanners to check for known vulnerabilities in the OS packages and application libraries included in your images.
* **Action:** Set up policies to break the build or alert developers if high or critical severity vulnerabilities are detected in container images.

**4. Regularly Rebuild and Redeploy Images:**

* **Action:** Establish a schedule for regularly rebuilding your application images, even if your application code hasn't changed. This ensures you pick up the latest security patches for the operating system and all included software components.
* **Action:** Redeploy your applications using these updated and patched images.

**5. Review and Harden Image Configurations:**

* **Action:** For services like Apache Tomcat running within containers, ensure their configurations are hardened according to security best practices. For example, keep writes for the default servlet disabled, review partial PUT usage, and secure session persistence if used.
* **Action:** For components like Apache MINA, after upgrading, ensure you explicitly configure whitelists for deserialization.

**6. Principle of Least Privilege:**

* **Action:** Ensure your Dockerfiles are constructed to include only the necessary software and libraries required for the application to run.
* **Action:** Run your containers with the least possible privileges. Avoid running containers as the root user unless absolutely necessary.