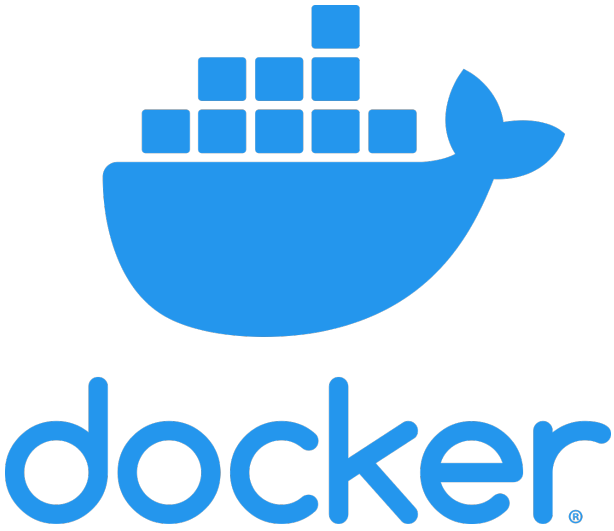
Dockerfile Audit Report

Security Team

March 19, 2020



Confidential

Contents

Chapter 1

Summary

The auditing was completed on March 19, 2020. An overview of the results is presented in the table below.

Total Dockerfile tested	2	100%
Audit Passed	0	0.0%
Audit Failed	2	100.0%

Table 1.1: Summary of the results of the Dockerfile audit.

Given the results obtained, the compliance level is Poor

Chapter 2

Policy Rules

The present audit was done with the following rules enabled:

Rule Name	Rule Description	Rule Details
ENFORCE_REGISTRY	Allow images to be based (using the FROM command) only on images belonging to approved repositories.	The following registries are allowed: sudneo.registry.com:5000.
FORBID_TAGS	Restrict the use of certain tags for the images the build is sourced from (using FROM command)	The following tags are forbidden: latest, stable, prod, stage.
FORBID_INSECURE_REGISTRIES	Forbid the use of HTTP protocol for the registries from which source images are stored.	
FORBID_ROOT	Forbid the container to run as a privileged (root) user.	
FORBID_PRIVILEGED_PORTS	Forbid the image to expose privileged ports that require administrative permissions.	
FORBID_PACKAGES	Forbid the installation/use of dangerous packages.	The following packages are forbidden: sudo, vim, netcat, nc, curl, wget.
FORBID_SECRETS	Forbid the inclusion of secrets in the image.	The following patterns are forbidden: id_rsa, private_key, password, key, secret. The following patterns are whitelisted: id_rsa.pub

Table 2.1: Policy rules enforced for this audit.

The rules violation breakdown is as follows:

Rule Name	**Total Violation**	**Violation %**
ENFORCE_REGISTRY	2	28.57%
FORBID_TAGS	0	0.0%
FORBID_INSECURE_REGISTRIES	0	0.0%
FORBID_ROOT	2	28.57%
FORBID_PRIVILEGED_PORTS	2	28.57%
FORBID_PACKAGES	1	14.29%
FORBID_SECRETS	0	0.0%

Table 2.2: Policy rules violation breakdown.

Chapter 3

Audit Details

In this section, the audit results for each analyzed file will be reported.

3.1 nginx

Audit outcome: FAIL

The details for the violated rules are in the following table:

Rule Name	Description	Mitigation	Statement
ENFORCE_REGISTRY	Registry Docker Hub is not an allowed registry to pull images from.	The FROM statement shouldbe changed using images from one of the allowed registries: sudneo.registry.com:5000	`FROM debian:buster-slim`

FORBID_ROOT	No USER statements found.By default, if privileges are not dropped, the container will run as root.	Create a user and add a USER statement before the entrypoint of the image to run the application as a non-privileged user.	`N/A`
FORBID_PRIVILEGED_PORTS	The container exposes a p rivileged port: 'port': '80', 'protocol': 'tcp'.Privileged ports requirethe application which uses it to run as root.	Change the configuration for the application to bind on a port greater than1024, and change the Dockerfile to reflect this m odification.	`EXPOSE 80`

Table 3.1: Failed policy tests for *nginx*.

3.2 wallabag

Audit outcome: FAIL

The details for the violated rules are in the following table:

Rule Name	Description	Mitigation	Statement
ENFORCE_REGISTRY	Registry Docker Hub is not an allowed registry to pull images from.	The FROM statement shouldbe changed using images from one of the allowed registries: sudneo.registry.com:5000	`FROM alpine:latest`
FORBID_ROOT	No USER statements found.By default, if privileges are not dropped, the container will run as root.	Create a user and add a USER statement before the entrypoint of the image to run the application as a non-privileged user.	`N/A`

FORBID_PRIVILEGED_PORTS	The container exposes a p rivileged port: 'port': '80', 'protocol': 'tcp'.Privileged ports requirethe application which uses it to run as root.	Change the configuration for the application to bind on a port greater than1024, and change the Dockerfile to reflect this m odification.	`EXPOSE 80`
FORBID_PACKAGES	Forbidden package "curl" is installed or used.	The RUN/CMD/ENTRYPOINT statement should be reviewed and package "curl" shou ld be removed unless absolutely necessary.	`set -ex && apk update && apk upgrade --available && apk add ansible curl git libwebp mariadb-client nginx pcre php7 php7-amqp php7-bcmath php7-ctype php7-curl php7-dom php7-fpm php7-gd php7-gettext php7-iconv php7-json php7-mbstring php7-openssl php7-pdo_mysql php7-pdo_pgsql php7-pdo_sqlite php7-phar php7-session php7-simplexml php7-tokenizer php7-xml php7-zlib php7-sockets php7-xmlreader php7-tidy py-mysqldb py-psycopg2 py-simplejson rabbitmq-c s6 tar tzdata && rm -rf /var/cache/apk/* && ln -sf /dev/stdout /var/log/nginx/access.log && ln -sf /dev/stderr /var/log/nginx/error.log && curl -s https://getcomposer.org/installer | php && mv composer.phar /usr/local/bin/composer && git clone --branch $WALLABAG_VERSION --depth 1 https://github.com/wallabag/wallabag.git /var/www/wallabag`

Table 3.2: Failed policy tests for *wallabag*.