Práctica: Construcción de Imágenes Seguras en Docker

En esta práctica, aprenderás cómo construir imágenes de Docker seguras siguiendo las mejores prácticas de seguridad y empleando herramientas de análisis. Utilizaremos dos herramientas específicas:

1. **Dockle**: Esta herramienta analiza las configuraciones de seguridad y las prácticas recomendadas en los `Dockerfile`. Nos ayuda a identificar configuraciones inseguras en las imágenes, como el uso de usuarios root o etiquetas de imagen inadecuadas.
    - Repositorio oficial: Dockle en GitHub
2. **Trivy**: Es una herramienta de escaneo de vulnerabilidades que identifica paquetes vulnerables en las imágenes de Docker. Trivy busca vulnerabilidades conocidas en el software dentro de la imagen.
    - Repositorio oficial: Trivy en GitHub

Paso 1 (en el taller ya estan instlados): Configuración de Docker y las Herramientas

Asegúrate de tener Docker instalado en tu sistema. Luego, instala **Dockle** y **Trivy** siguiendo las instrucciones en sus repositorios oficiales:

- **Instalación de Dockle**:

```
curl -Lo dockle https://github.com/goodwithtech/dockle/releases/latest/download/
tar zxvf dockle_Linux-64bit.tar.gz
sudo mv dockle /usr/local/bin/
```

- **Instalación de Trivy**:

```
sudo apt install -y wget
wget https://github.com/aquasecurity/trivy/releases/latest/download/trivy_Linux-
tar zxvf trivy_Linux-64bit.tar.gz
sudo mv trivy /usr/local/bin/
```

Paso 2: Descargar una Imagen de Docker

Para esta práctica, descargaremos una imagen de PHP utilizando el comando `docker pull` o

`docker build` si prefieres construir una imagen personalizada.

```
docker pull php:latest
```

Paso 3: Escanear la Imagen con Dockle

Ahora que tienes una imagen de PHP descargada, puedes usar Dockle para realizar un análisis de seguridad. Esto mostrará advertencias y recomendaciones de seguridad basadas en las prácticas recomendadas.

Ejecuta Dockle en modo `debug` para ver un análisis detallado:

```
dockle -debug php:latest
```

Resultado Esperado

Es probable que Dockle genere advertencias como las siguientes:

- **CIS-DI-0001: Create a user for the container**
  - **Descripción**: No se debe usar el usuario `root` en los contenedores. Crea un usuario no root para mejorar la seguridad.
- **DKL-DI-0006: Avoid latest tag**
  - **Descripción**: Evita usar la etiqueta `latest` para la imagen, ya que puede causar inconsistencias y problemas de seguridad cuando se actualizan las versiones.

Paso 4: Crear un `Dockerfile` Seguro

Con base en las advertencias anteriores, crearemos un `Dockerfile` mejorado para evitar el uso de `root` y la etiqueta `latest`.

Ejemplo de un `Dockerfile` Seguro para PHP

```
# Usa una versión específica de la imagen
FROM php:7.4

# Crea un usuario sin privilegios
RUN useradd -m phpuser

# Establece el usuario creado como el usuario predeterminado para el contenedor
USER phpuser

# Configura el trabajo principal (puedes agregar más instrucciones según tu caso)
WORKDIR /var/www/html
COPY . /var/www/html

# Exposición del puerto 80
EXPOSE 80

# Comando de inicio
CMD ["php", "-S", "0.0.0.0:80"]
```

Este `Dockerfile` evita el uso del usuario `root` y selecciona una versión específica de PHP, en lugar de `latest`.

Paso 5: Volver a Escanear la Imagen con Dockle

Construye la nueva imagen con el `Dockerfile` seguro:

```
docker build -t php:secure .
```

Luego, ejecuta Dockle nuevamente en la nueva imagen para verificar si las advertencias han disminuido:

```
dockle -debug php:secure
```

Deberías ver menos advertencias relacionadas con los usuarios y las etiquetas de imagen.

Paso 6: Escaneo de Vulnerabilidades con Trivy

Ahora que hemos creado una imagen más segura en cuanto a configuración, usaremos Trivy para verificar vulnerabilidades en los paquetes de software de la imagen.

Escaneo de una Imagen Segura (Alpine)

Ejecuta Trivy en la imagen `alpine` para observar cómo una imagen mínima puede contener menos vulnerabilidades:

```
trivy image alpine
```

Deberías ver pocos o ningún problema de vulnerabilidades, ya que Alpine es una imagen de Linux ligera y segura.

Escaneo de una Imagen PHP

Ejecuta Trivy en la imagen PHP para identificar vulnerabilidades presentes en ella:

```
trivy image php:7.4
```

Este comando te mostrará una lista de vulnerabilidades y su severidad (alta, media o baja). Trivy proporciona detalles de las vulnerabilidades encontradas, lo que permite evaluar si necesitas aplicar parches o tomar medidas adicionales de seguridad.

Conclusión

En esta práctica, aprendiste a usar Dockle y Trivy para mejorar la seguridad de las imágenes de Docker. Las herramientas de seguridad y análisis como estas son esenciales para mantener un entorno seguro en contenedores, y deben ser parte de un flujo de trabajo de desarrollo seguro.