

# Supply Chain Attack : Le cas du registre privé Docker

Geoffrey Sauvageot-Berland – Call for paper - *leHACK* 2024

## Résumé :

Docker est aujourd'hui un outil de conteneurisation incontournable, largement adopté dans le développement logiciel. Cette expansion a souligné l'importance de sécuriser tous les composants associés, comme le registre privé Docker. Alternative au Docker Hub public, c'est une plateforme open-source où les développeurs peuvent stocker, gérer et distribuer leurs applications localement. Hélas, la documentation officielle et de nombreux articles en ligne ne sensibilisent pas suffisamment les utilisateurs à la nécessité de sécuriser le registre dès sa mise en place. En effet, sa configuration par défaut est vulnérable, car elle permet notamment un accès anonyme sans contrôle d'accès. Actuellement, il semblerait qu'aucune organisation n'ait encore subi les conséquences d'une attaque ciblant un registre privé, du fait qu'il est moins utilisé que le Docker Hub et que ce vecteur d'attaque reste méconnu. Cette étude expose donc une *Supply Chain Attack* qui vise une application hébergée sur un registre privé Docker, dans le but de compromettre son cycle de développement. Enfin, des contre-mesures seront présentées pour atténuer cette menace.

# Sommaire :

## I. Introduction

### 1. Qu'est-ce que Docker ?

- - Brefs rappels
- - Statistiques clés sur les attaques impactant Docker

### 2. Qu'est-ce qu'un registre privé Docker ?

- Exposition des registres privés Docker non sécurisés sur Internet
- Quelles sont les problématiques de sécurité ? (Pas d'authentification par défaut, HTTP, Absence de système de gestion de permissions granulaires (qui peut push/pull/delete, etc.)
- Que peut-on faire ? Voler, supprimer ou encore falsifier les images qui y sont stockées.

## II. Scénario d'exploitation – Attaque du cycle de développement d'une application

*Cette partie déroule une démonstration avec un scénario d'exploitation qui affecte un registre privé Docker n'imposant pas d'authentification. Le registre est accompagné d'un conteneur watchtower, qui se charge de la surveillance et du déploiement automatique des images (applications) dès lors que ce conteneur détecte des changements sur le registre.*

- Phase de reconnaissance : Comment identifier la présence d'un registre privé Docker sur un réseau ?
- Exécution locale de l'application : Identifier les technologies utilisées et détecter la présence de données sensibles.
- *Backdooring* de l'image : Création d'une backdoor en utilisant les langages de programmation pris en charge par l'application.
- Refonte de l'image initiale : Création d'un fichier *Dockerfile* en se basant sur l'image originale, et insertion de la *backdoor*.
- Assemblage de l'image : Construction de la nouvelle image en veillant à conserver les mêmes caractéristiques que l'image initiale. (nom, version)
- *Upload* de l'image sur le registre : Empilement de l'image originale avec les nouvelles couches atomiques (Cela signifie que dès lors qu'une couche est créée, elle ne peut pas être modifiée).

- Déploiement de l'image : Instanciation de l'application par l'intermédiaire du conteneur *watchtower*. (l'approche conceptuelle de *watchtower* « DooD » sera abordée rapidement afin de faciliter la compréhension). *L'approche DooD (Docker outside of Docker) consiste à exécuter et gérer des conteneurs Docker à partir d'un autre conteneur Docker « Maître » par l'intermédiaire du démon Docker)*
- RCE (Remote Code Execution) : Exécution de code à distance depuis la machine de l'attaquant.
- Élévation de privilège au sein de l'image : Par défaut pour le service web de ladite application, l'utilisateur est `www-data`. Suite à la refonte du *Dockerfile*, cet utilisateur disposera de droit `root` au sein du conteneur.

### III. Solutions de remédiation

- Imposer un contrôle d'accès utilisateur : Authentification simple (HTTP Basic Access authentication) ou bien opter pour une authentification forte (OAuth).
- Chiffrer les communications (TLS)
- Signer numériquement les images stockées au sein du registre.
- Cloisonnement réseau

### IV. Conclusion

- Take aways

## informations demandées

**Difficulté :** Moyenne

**Déjà traité par *lehack* ?** NON (Après avoir vérifié les sujets traités depuis 2014 sur le site officiel)

**Déjà traité par une autre conférence ?** NON (D'après mes recherches...)

**Langue de la présentation :** Français

**Durée :** 45 minutes (avec questions)

**Type d'intervention :** « Talk »

**Nom et biographie :** Geoffrey Sauvageot-Berland, Ingénieur diplômé d'État en informatique, Auditeur en sécurité offensive chez Orange Cyberdefense, Chargé d'enseignement à CPE LYON. Fondateur du site « Le Guide Du Secops », Auteur chez IT-connect.

**Lieu de départ et moyen de transport prévu :** Lyon, en train

**Votre compte Twitter :** AUCUN

**Prévoyez-vous de faire la démonstration d'un outil/exploit/technique ?** OUI

**Prévoyez-vous de publier un outil/exploit ?** NON

**Prévoyez-vous de diffuser du son/musique ?** NON

**Votre entreprise est-elle sponsorisée ou souhaiterait-elle l'être ?** OUI

**Autre information :** Ce sujet va faire l'objet d'un article dans la revue scientifique MISC (Mai/Juin 2024 N°133)