Conteneurs et sécurité

Timothée Ravier
siosm@floss.social - tim.siosm.fr/cours - github.com/travier

5e année cycle ingénieur, filière STI
Option Sécurité des Systèmes Embarqués et du Cloud (2SEC)

2024 - 2025

Points d'attention

- Sécurité de l'hôte hébergeant les conteneurs :
  - Matériel
  - Noyau et système de base
  - Gestionnaire de conteneurs
  - Méchanismes d'isolation entre les conteneurs
- Sécurité de l'image de base d'un conteneur :
  - Distribution et intégrité
  - Distribution des secrets
- Sécurité de l'application dans le conteneur :
  - Système de base et mises à jour
  - Langage de programmation & Framework

Sécurité de l'hôte

Matériel : Nombreux éléments partagés

Attaques par canaux cachés :

- Pas d'isolation physique : matériel partagé :
  - Rowhammer
  - Reverse Engineering Intel DRAM Addressing and Exploitation
  - Cross Processor Cache Attacks
- Attaques utilisant des canaux cachés (ou détournées) :
  - temps & cache : Meltdown and Spectre

Noyau Linux

- Cible privilégiée : point commun entre tous les conteneurs
- L'isolation fournie par les namespaces et les cgroups a encore des limites
- Cas non prévus lorsque l'on combine plusieurs namespaces
- Certaines parties du noyau n'ont pas encore été adaptées aux namespaces

Noyau Linux

- Surface d'attaque large (appels systèmes) :
  - Gestion de la mémoire : Dirty CoW
  - Gestion des caches : Dirty Pipe Linux Vulnerability: Overwriting Files in Container Images

Noyau Linux : namespaces & overlayfs

- Beaucoup de vulnérabilités trouvées par le passé :
  - CLONE_NEWUSER|CLONE_FS root exploit
  - Linux namespaces: It is possible to escape from bind mounts (CVE-2015-2925)
  - User namespaces + overlayfs = root privileges
  - CVE-2015-4176, CVE-2015-4177, CVE-2015-4178
  - CVE-2015-1328 : incorrect permission checks in overlayfs
  - CVE-2015-8660, etc. (liste non-exhaustive)
- Failles noyau donc très souvent critiques (« local root »)
- Vulnérabilité récente :
  - CVE-2023-0386: Datadog Security Labs

Noyau Linux : user namespaces

- Surface d'attaque étendue avec les user namespace
- Important de limiter les capabilities même avec les user namespace
- Exemples de vulnérabilités :
  - Anatomy of a user namespaces vulnerability (CVE-2013-1858)
  - The Discovery and Exploitation of CVE-2022-25636
  - How The Tables Have Turned: An analysis of two new Linux vulnerabilities in nf_tables
  - CVE-2022-0185, CVE-2022-0185 in Linux Kernel Can Allow Container Escape in Kubernetes

Protection du noyau Linux

Réduction de la surface d'attaque :

- Retirer des appels systèmes du noyau (dur)
- Limiter les appels systèmes disponibles : seccomp-bpf (plus facile)

Outils pour générer des filtres seccomps fins :

- Improving Linux container security with seccomp
- go2seccomp

Durcissement du noyau Linux

- « Historiquement » : PaX & grsecurity
- Les patchs ne sont plus disponibles publiquement
- Travail en cours par le *Kernel Self Protection Project* pour améliorer progressivement la situation *upstream*
- Arrivée progressive de Rust dans le noyau Linux :
  - NVMe Driver
  - Android Binder Driver
  - Apple AGX GPU driver

Durcissement distributions Linux

- **Mises à jour régulières**
- Exemple de vulnérabilité récente :
  - glibc : CVE-2023-6779, Qualys Security Advisory
- Durcissement « classique » (c.f. guides et recommandations)

Outils de gestion de conteneurs

- Les outils d'infrastructure pour gérer les conteneurs font parti de la surface d'attaque
- Vulnérabilité récente dans runc :
  - CVE-2024-21626 (Docker Security Advisory, runc upstream, Snyk)
- Historique des vulnérabilités :
  - Container CVE List
  - Container Security Site

Docker et sécurité

> « Docker is about running random code downloaded from the Internet and running it as root. »

Spécificités Docker

- Accès au démon docker ⇔ `root`
- Restreindre impérativement l'accès au démon docker : Why we don't let non-root users run Docker in CentOS, Fedora, or RHEL
- Guides pour utiliser Docker correctement : Docker security
- Applicable aussi à Podman si le mode démon est configuré

Isolation : SELinux

- Contenir les conteneurs à l'aide des catégories (MCS)
- Tous les conteneurs utilisent le même type (même accès)
- Associe à l'exécution un ensemble de catégories à chaque conteneur
- SELinux garantie qu'il ne peut pas y avoir d'intéraction entre les conteneurs si leur ensemble de catégories sont distincts
- Supporté par Docker et Podman
- Limite la portée d'une grande partie des vulnérabilités (ex. CVE-2016-9962)

Isolation : AppArmor

- AppArmor supporté par Docker et Podman :
  - https://docs.docker.com/engine/security/apparmor/
  - https://cloud.google.com/container-optimized-os/docs/how-to/secure-apparmor

Isolation : Virtualisation matérielle

- Isolation supplémentaire à l'aide de la virtulisation matérielle
- Kata Containers
  - Hyperviseurs : QEMU, Cloud-Hypervisor and Firecracker
  - Learn about Red Hat OpenShift sandboxed containers
- Aussi gVisor et Nabla Containers
  - gVisor Networking Security
- Confidential Containers
  - Combine virtualisation, attestation et chiffrement (mémoire et disques)
  - Intel TDX, AMD SEV and IBM Z Secure Execution
  - What is the Confidential Containers project?

Sécurité du conteneur et de son image de base

Sécurité du conteneur

- **Restriction des permissions** :
  - Utiliser un utilisateur non privilégié (i.e. différent de `root`)
  - Limiter les *capabilities*
  - Ne pas autoriser l'élévation des priviléges (*NoNewPrivileges*)
  - Appliquer un filtre *seccomp-bpf* strict
- **Image du conteneur en lecture seule (RO)**
- **Stockage persistent externe (volumes)** pour :
  - la configuration en lecture seule
  - les données en lecture / écriture

Sécurité de l'image de conteneur

- Contrôle de l'intégrité du conteneur :
  - stockage, transport sur le réseau, vérification avant exécution
- Signer les images de conteneurs :
  - sigstore
  - Signer les images de conteneurs avec cosign
  - Fonctionnement avec gestion de clés ou avec clés éphémères (*keyless*)
- Typo-squatting et multi-registre :
  - Container Image Squatting in a Multi-Registry World

Image de conteneur : Distribution et gestion des secrets ?

- Considérer une image de conteneur comme publique
- Ne pas inclure de secrets dans une image de conteneur
  - Sinon, établir un control d'accès fort
- Utilisation de variables d'environnement ou de volumes
- Récupération dynamique des secrets :
  - HashiCorp Vault / OpenBao

Sécurité de l'application

Système de base du conteneur

- Conteneurs sont souvent créés à partir d'un système de base
- La taille n'est pas toujours un facteur de décision (même si l'on cherche souvent à la réduire le plus possible)
- Choisir un système de base avec des bonnes propriétés de sécurité et correctement supporté
- Les distributions standard (Debian, Ubuntu, Fedora, Red Hat & CentOS Stream) restent une valeur sûre.

Système de base du conteneur : Alpine ?

- Attention aux images basées sur Alpine
- Compatibilité avec musl libc (vs glibc)
- Gestionnaire de paquet spécifique (apk) :
  - Remote Code Execution in Alpine Linux

Mises à jour du conteneur

- Conteneur ⇔ énorme binaire statique
- Absolument prévoir un cycle de mise à jour
- Intégration et déploiement continue (CI/CD)
- Beaucoup d'images proposés sur le Registry ne sont pas à jour ou malveillantes
  - Analysis on Docker Hub malicious images: Attacks through public container images

Scanneurs de vulnérabilités

- Outils pour vérifier qu'une image ne contient pas de vulnérabilité déjà connue :
  - Clair (intégré dans Quay.io)
  - Aqua Security Trivy

Supply Chain et licences

- Attention au respect des licences open source
- Nombreux outils de gestion de la *Supply Chain*
- Suivit des licences et des vulnérabilités trouvées dans les bibliothèques utilisées par les applications
- Software Supply Chain, Software Bill of Materials (SBOM)

Langages de programmation

- Une application vulnérable sera toujours vulnérable dans un conteneur
- Utiliser des langages de programmation et des « frameworks » avec des propriétés intéressantes en terme de sécurité :
  - Rust
  - Go, Kotlin
  - Haskell, OCaml, Elixir, Erlang
  - Java, Python, Javascript, Typescript

Suite

Sécurité de la virtualisation