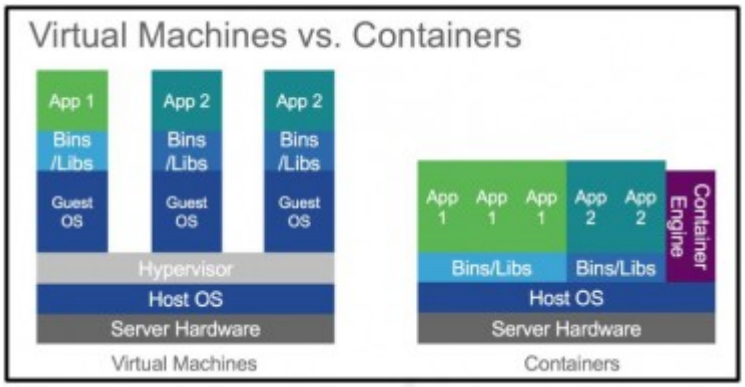
31 bonnes pratiques pour sécuriser les conteneurs Docker

Les entreprises s'intéressent de plus en plus aux conteneurs – et en particulier à Docker (même si des alternatives existent). « *Le placement en conteneurs permet aux développeurs de packager une application et toutes ses dépendances dans un format standardisé, sans besoin ni de recompilation, ni de recherche ou d'installation des environnements adaptés* », explique le Sans Institute, pour résumer l'attrait des conteneurs. Une flexibilité qui fait mouche alors que se multiplient les environnements Cloud. D'où la volonté de nombre d'entreprises d'amener Docker jusque sur les environnements de production, alors que la technologie a d'abord gagné ses galons auprès des développeurs et sur les environnements de tests.



Comme le précise le Sans Institute, une organisation américaine à but non lucratif regroupant 165 000 professionnels de la sécurité (Sans signifiant Sysadmin, Audit, Network, Security), l'arrivée de Docker en production soulève toutefois de nouvelles questions, en particulier pour les auditeurs de sécurité. D'où la publication par le Sans d'une checklist permettant à ces spécialistes d'aborder la sécurisation de ces environnements. Une réelle difficulté pour les entreprises, du fait tant de l'architecture sur laquelle reposent les conteneurs (différente de celle, familière, des VM, voir schéma ci-contre), des concepts nouveaux amenés par la technologie (devops, microservices), que du manque de standardisation des outils de sécurité, comme le souligne le Sans.

Vulnérabilités dispersées

« *Auditer des déploiements Docker peut s'avérer difficile tant du fait de l'architecture Docker elle-même que des modèles de microservices et de déploiement continu qui sont si bien alignés avec Docker* », résume l'organisation. Selon les chiffres du spécialiste du monitoring New Relic, 46 % des conteneurs ont une durée de vie inférieure à une heure, et 11 % d'entre eux 'vivent' même moins d'une minute. Cette caractéristique, associée à la densité des architectures en conteneurs, peut aboutir à « *un cauchemar* » en matière de gestion des environnements. « *Suivre exactement ce qui a été déployé, où cela l'a été et quand* » devient en soi un défi, souligne le Sans.

Autre difficulté relevée par l'organisation : la dispersion des vulnérabilités dans les conteneurs et dans les OS hôtes. Des environnements qui requièrent généralement des outils d'audit différents, qui plus est. Qui plus est, si une organisation va chercher ses images Docker en externe, le risque de ramener une faille est grand. Un ingénieur de Docker a estimé que 30 % des images présentes sur le Docker Hub, la place de marché publique de l'éditeur, contenaient des vulnérabilités.

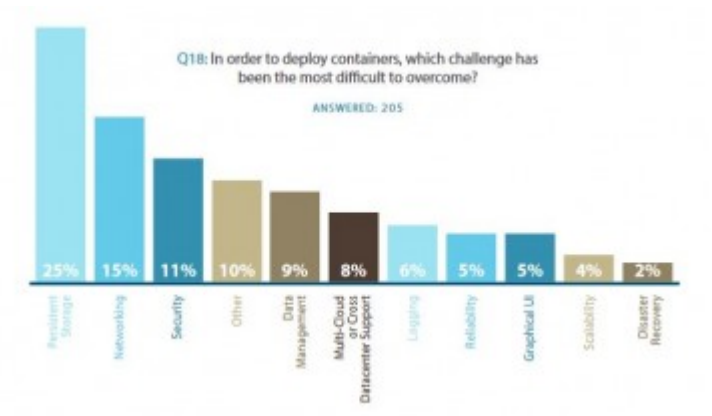
Les secrets mal gardés dans Docker

Sur la base de ces constats, le Sans Institute livre 31 bonnes pratiques de sécurité dans les environnements Docker. Des classiques (comme l'installation des patchs au niveau des hôtes mais aussi de Docker lui-même, l'installation limitée aux seuls composants utiles, la ségrégation des conteneurs en fonction de la confidentialité des données, la centralisation des logs…), mais aussi des conseils plus spécifiques (comme le durcissement de l'OS hôte, l'obligation d'utiliser des images signées, l'interdiction de faire tourner un processus conteneurisé en root, la limitation des ressources pour chaque conteneur afin d'éviter les dénis de service).

En particulier, le Sans souligne que « *de nombreuses techniques courantes employées pour stocker des secrets sur des hôtes dédiés ou des machines virtuelles – comme les dotfiles (fichiers cachés dans les systèmes Unix, NDLR) avec des accès limités ou le stockage dans des fichiers chiffrés – ne sont plus viables dans des environnements avec des conteneurs.* » Dans ces derniers, si le secret est stocké dans l'image Docker, il sera accessible à quiconque récupérant l'image depuis un magasin de conteneurs (qu'il s'agisse de celui public fourni par l'éditeur ou de sa version privée). Stocker ledit secret dans les variables n'est guère préférable, puisqu'il pourra fuiter via les logs, via des commandes de Docker ou être partagé avec des conteneurs associés. Cette particularité doit pousser les entreprises à déployer des solutions différentes comme Vault, Keywhiz, ou Crypt, insiste le Sans. Même si ces options souffrent encore d'un manque de maturité, selon l'organisation.

Communications entre conteneurs

Autre spécificité de la technologie, offrant une piste intéressante à des assaillants : par défaut, les conteneurs peuvent envoyer du trafic via le réseau interne de Docker que les ports soient ouverts ou pas. « *Les utilisateurs Docker devraient désactiver ces communications inter-conteneurs et demander des associations explicites entre conteneurs ou l'ouverture de ports publics pour ces échanges* », écrit l'organisation spécialisée dans la sécurité.



Selon une étude menée par Devops.com et publiée en juin dernier, les difficultés que rencontrent les entreprises avec les conteneurs se concentrent sur l'exploitation : la sécurité notamment, mais surtout le réseau et les problématiques de stockage persistant. Le signe d'un manque certain de maturité de la technologie quand elle aborde les environnements de production. Le Sans ne manque pas, de son côté, de relever la jeunesse des outils d'audit à disposition des spécialistes de la sécurité. Même si, pour l'organisation, Docker Bench for Security (ici sur Github) est aujourd'hui une solution des plus prometteuse.

A lire aussi :

Les conteneurs de Docker s’imposent, y compris dans la prod

BlaBlaCar généralise les conteneurs et préfère Rocket à Docker

Open Source : Docker est-il menacé d’implosion ?