Azure Functions : sans serveur, mais pas sans faille ?

Danger pour la sécurité d'Azure Functions ? Microsoft a estimé que non, malgré la démonstration que lui ont faite des chercheurs.

Ces derniers sont parvenus à « s'échapper » du conteneur hébergeant une fonction… et à remonter vers l'hôte Docker. La vidéo ci-dessous illustre la phase terminale de la démarche : le listage des processus en cours sur ledit hôte.

Les chercheurs ont exploité une fonction sur laquelle ils avaient la main. Ils y ont d'abord injecté un *shell* inversé pour permettre la connexion à un serveur sous leur contrôle. Deuxième étape : l'ajout d'outils parmi lesquels nmap. Celui-ci a permis de détecter plusieurs ports ouverts. Et trois processus associés : NGINX, MSI et Mesh.

Le premier de ces composants, *open source* et largement documenté, ne présentait pas de failles connues. Les deux autres, propriétaires, avaient plus de chances d'être vulnérables. Mais ils se trouvaient dans des dossiers nécessitant un accès root.

Azure Functions : des conteneurs à haut privilège

MSI (Managed Service Identity) assure une gestion automatisée des identités. Il s'est avéré plus difficile de déterminer à quoi servait Mesh. Les chercheurs ont pu trouver des informations dans les journaux de *build* d'une image Docker stockée dans un registre appartenant à un employé de Microsoft.

Les chercheurs ont téléchargé cette image et en ont fait un conteneur dont ils ont extrait Mesh. En explorant le binaire, ils ont repéré des capacités nécessitant des privilèges. Entre autres, une destinée à monter un système de fichiers. Ils ont réussi à l'invoquer par le biais d'un serveur HTTP local. Et ainsi à monter un système de fichiers de leur cru. Sa cible : le répertoire /etc/sudoers.d. Son contenu : de quoi obtenir les privilèges de niveau root.

Cela fait, les chercheurs ont analysé les capacités des différents processus exécutés dans le conteneur. Ils en ont conclu que celui-ci disposait du drapeau –privileged. En y associant leurs droits root et une technique d'« échappatoire » connue de longue date, ils ont pu exécuter la commande ps (listage des processus) sur l'hôte Docker.

> *Quick and dirty way to get out of a privileged k8s pod or docker container by using cgroups release_agent feature. pic.twitter.com/q8BI8ASBO8*
>
> *— Felix Wilhelm (@_fel1x) July 17, 2019*

Microsoft considère que cette démonstration ne met pas Azure Functions en danger. La raison :

elle n’atteint pas le « bout de la chaîne ». L’hôte Docker est, en l’occurrence, lui-même un invité, exécuté dans un environnement Hyper-V.

Illustration principale © GKSD – Fotolia