WAFaaS: la seule fois où « as-a-Service » signifie « mauvais-Service ».

Avec le lancement du modèle T Ford en 1908, le premier véhicule produit en série au monde, Henry Ford avait la volonté de mobiliser les Américains. À l'époque, il déclarait que la voiture était disponible pour ses clients « dans toutes les couleurs qu'ils souhaitaient, à condition qu'elles soient noires ». Le manque de variété ou d'options de personnalisation n'a certainement pas empêché les acquéreurs potentiels d'acheter leur première automobile.

L'approche sans complaisance d'Henry Ford n'aurait jamais fonctionné aujourd'hui, car les consommateurs s'attendent désormais à ce que la personnalisation soit présente dans tous les domaines, des vêtements aux expériences numériques.

Pour les nombreuses équipes chargées de l'expérience utilisateur (UX) qui repoussent sans cesse les limites de la personnalisation pour satisfaire les consommateurs, il existe de nombreuses équipes DevOps qui testent et déploient en permanence de nouveaux codes pour créer des applications innovantes et solides. Il est donc important que ces applications soient sécurisées.

Pourtant, la sécurité des applications dynamiques natives du cloud reste un défi qui a été exacerbé par les outils existants tels que les pare-feux d'applications web (WAF) traditionnels.

La sécurité des applications est le cauchemar de la communauté de la sécurité depuis des années. Les applications des années 90 étaient simples, monolithiques et étaient accompagnées de mises à jour mensuelles si l'équipe de développement parvenait à faire évoluer le développement assez rapidement. Ce n'était qu'une réflexion après coup et, pour permettre aux développeurs et aux équipes d'assurance qualité de continuer à travailler, cette sécurité était généralement mise en œuvre sur le périmètre.

Avec une liste de signatures d'attaques connues, un pare-feu d'application web (WAF) serait capable de prendre une décision binaire sur la base de chaque requête web. Cela signifie que lorsqu'une requête arrive, le WAF essaie de trouver une correspondance dans la base de données des signatures d'attaques et, si c'est le cas, la requête est refusée.

Cette approche unique a relativement bien fonctionné jusqu'à l'arrivée du cloud computing, lorsque le DevOps a rendu le WAF inutile. La vitesse à laquelle les applications étaient mises à jour s'était accélérée et le WAF, avec son approche manuelle, ne pouvait tout simplement plus suivre. Alors que les fournisseurs d'AppSec tentaient d'automatiser la sécurité de la même manière que le DevOps l'a fait pour le développement, il est apparu clairement que les anciens WAF étaient incapables de gérer le rythme.

C'est ainsi qu'a commencé le passage au WAF « As-A-Service » (WAFaaS).
Sous couvert d'automatisation, les fournisseurs de sécurité ont commencé à proposer des WAF « sans casse-tête », c'est-à-dire des WAF accompagnés de modèles ou d'assistants censés offrir une sécurité globale pour toutes les applications sans que les équipes chargées de la sécurité des applications n'aient à consacrer trop de temps à leur gestion.

Le WAFaaS bloque sur les mises à jour

Mais il est devenu évident que ces offres WAFaaS sont tout aussi inefficaces que les WAF traditionnels lorsqu'il s'agit de protéger les applications natives du cloud. L'utilisation d'un WAFaaS peut donner aux équipes de sécurité des applications un faux sentiment de sécurité, laissant leurs applications et microservices ouverts aux attaques.

En réalité, les offres de WAFaaS ne sont pas adaptées aux besoins de l'application qu'elles protègent. Cette réincarnation d'un WAF traditionnel n'offre aucune possibilité de s'adapter automatiquement aux mises à jour des applications.

Sans une compréhension approfondie de l'application, de son contenu et de ses utilisateurs, le WAFaaS fournit inévitablement un niveau de sécurité inférieur pour éviter les faux positifs.

De plus, le WAFaaS n'est pas intégré nativement dans les environnements cloud. Cela signifie qu'il n'est pas possible de déployer la sécurité avec le code (que ce soit dans les K8, pour les fonctions sans serveur ou à l'aide d'un agent), et les offres traditionnelles de WAF pour ces environnements fournissent une sécurité sous-optimale.

Il existe également le problème mineur de l'automatisation effective. En l'absence de périmètre réseau dans l'environnement cloud, chaque micro service finit par devenir un périmètre en soi.
Et si le WAFaaS peut sembler une solution décente pour une application en cloud, chaque nouveau microservice ne sera pas automatiquement protégé dès qu'il sera déployé, ce qui induit un risque.

Le plus inquiétant pour les équipes de sécurité des applications qui utilisent un WAFaaS pour protéger une application « cloud native » est le fait qu'il ne peut protéger l'application que contre le trafic web direct.

Dans un environnement de microservices distribués, le WAFaaS ne voit pas une grande partie du trafic de l'application, et peut encore moins protéger l'application contre les attaques.
Avec la prolifération des API, la surface d'attaque des applications s'est élargie, et les WAFaaS ne peuvent pas se protéger contre tout le trafic entrant par l'intégration d'une API tierce.
Il ne peut pas non plus assurer une protection en cas de communications entre microservices.
Pour les applications natives du cloud, il n'est pas adapté à la situation.

Ces applications ne peuvent être protégées que par une sécurité native du cloud qui s'adapte automatiquement aux changements d'application. Les équipes AppSec doivent comprendre que, dans l'environnement disparate des microservices, l'application est la somme de toutes ses parties et qu'à ce titre, les solutions de sécurité doivent protéger chaque actif. Comme il n'existe pas deux applications identiques, les applications natives du cloud computing ont besoin de solutions de sécurité qui ne reposent pas sur une approche unique.

Le WAF traditionnel était un outil important pour protéger les applications naissantes des années 1990. À l'instar du modèle T Ford, qui a rendu le transport privé accessible à tous, le WAF a introduit la sécurité des applications à la portée du plus grand nombre grâce à une solution configurable à l'époque où personne ne s'inquiétait des mises à jour des applications et des déploiements automatisés.