API, architecture et analyse de la situation

Il est important de souligner que les API ne sont par nature ni plus ni moins risquées, et qu'elles offrent une foule d'avantages dans l'environnement professionnel actuel. Toutefois, avec les API, le risque sort du cadre traditionnel de la sécurité et investit d'autres sphères. Tant que nous n'appréhenderons pas mieux ce transfert du risque, les API demeureront une source de failles de sécurité pour les entreprises.

F5 Labs a récemment analysé des rapports Open Source sur les incidents liés aux API, englobant à la fois les violations de données confirmées commises par des pirates et les vulnérabilités identifiées par des chercheurs en sécurité. Neuf incidents ont ainsi été dénombrés en 2018. Ce chiffre est passé à 35 en 2019 et à 25 au premier semestre 2020.
À ce rythme, environ 50 incidents liés aux API devraient être rapportés d'ici fin 2020. Lorsque nous classons ces incidents par catégorie, le problème le plus courant observé est l'absence totale d'authentification auprès des endpoints des API (31,34 % des incidents), suivie par un mauvais processus d'authentification (19,4 %) et un mauvais processus d'autorisation (17,9 %).

Autrement dit, la cause la plus fréquente des incidents liés aux API au cours des deux dernières années tient vraisemblablement à un faible niveau de maturité de la sécurité.

Cartographie de la nouvelle surface d'attaque

Le seul but d'une API est de faciliter le transfert d'informations vers et depuis un réseau selon des modalités nécessairement et délibérément invisibles pour l'utilisateur lambda.

Les API REST occupant aujourd'hui une place prépondérante, ce transfert d'informations s'effectue via le Web, à l'aide de méthodes HTTP. Même en supposant que chaque partenaire ait complètement protégé ses arrières pour chacune de ses API privées, ce qui va à l'encontre du principe de « Zero Trust » et de l'inéluctabilité d'au moins une violation, chaque endpoint d'API représente une extension de la surface d'attaque et exige donc la mise en place de contrôles, à l'instar de tout autre endpoint.

Le problème est encore plus délicat avec les API publiques.
Il ne s'agit pas simplement de nouvelles brèches dans le périmètre, mais d'ouvertures rendues publiques à une communauté dotée des compétences requises pour trouver et exploiter des vulnérabilités ou mauvaises configurations.

À l'extrémité du spectre, bon nombre des applications Web les plus grandes et les plus populaires qui dépendent fortement de l'intégration de services tiers contiennent désormais des centaines d'API. Chaque API constitue une opportunité distincte pour les attaquants, un transfert du trafic (sinon une augmentation du trafic) vers un autre contexte, et une dépendance de l'activité et de la sécurité vis-à-vis d'une entité en dehors du contrôle du propriétaire du système.

Ce type d'architecture représente un modèle de risque suffisamment différent sur le plan de la visibilité, de la complexité, de l'inventaire et des partenariats commerciaux pour que les anciens postulats de référence en termes de modélisation des risques et des menaces perdent tout leur

sens. Quels que soient le secteur d'activité de l'entreprise et la valeur de ses données, son API lui permet d'atteindre quelque chose d'autre, qu'il s'agisse d'un partenaire, d'un client ou d'un élément d'infrastructure renforcé inaccessible par d'autres moyens. La nature conjonctive des API signifie que les attributs cibles importent moins que les contrôles de base.

Vus sous cet angle, les incidents liés aux API tels que ceux constatés (failles d'authentification, échecs d'authentification, échecs d'autorisation et données en entrée non nettoyées) résultent de la collision entre un ancien mode de pensée et une nouvelle réalité des risques résultant de l'évolution des pratiques métiers.

Alors que faire ?

Bien entendu, toutes les entreprises qui publient des API ne s'exposent à un désastre sécuritaire. Nombre d'entre elles ont parfaitement réussi à mettre en œuvre ces technologies et principes de conception. Ci-dessous figurent quelques grandes lignes directrices pour gérer les risques liés aux API REST, quel que soit le contexte :

· **Inventaire** : il est impossible de sécuriser ce que l'on ne connaît pas. Dans un contexte de standardisation vers le modèle DevOps (propice à l'informatique cachée), et d'intégrations de plus en plus granulaires limitées à certaines branches d'activité, maintenir une visibilité sur les endpoints des API n'est pas un exercice anodin.

· **Authentification** : toutes les API nécessitent une authentification. Le consensus émergeant est qu'OpenID Connect, qui repose sur le protocole d'autorisation OAuth 2.0, est la méthode privilégiée (et éprouvée) d'authentification des API.

· **Autorisation** : en raison de la nature obscure et décentralisée du trafic des API, le maintien d'un contrôle strict sur les autorisations des agents est essentiel pour prévenir les altérations, les attaques par énumération ou les déplacements latéraux. OAuth 2.0 est la norme recommandée pour gérer les autorisations des API. Dans le cadre de cette norme, le format JSON Web Token (JWT) devient la méthode privilégiée d'autorisation basée sur des jetons.

· **Chiffrement** : il est judicieux de forcer l'utilisation du protocole HTTPS pour les connexions des API, mais comme OAuth 2.0 requiert TLS afin d'assurer la confidentialité des clés secrètes, HTTPS est de toute façon de plus en plus obligatoire pour les API.

· **Médiation/passerelles d'API** : les passerelles d'API sont indispensables aux architectes d'entreprise qui doivent gérer un large éventail d'API et leur trafic. Elles conviennent tout particulièrement à la gestion du trafic d'API nord-sud (c'est-à-dire provenant de connexions externes). De nombreuses passerelles intègrent également des fonctions d'authentification et d'autorisation qui limitent l'impact d'une intrusion lorsqu'un endpoint d'API public est compromis par d'autres moyens.

Les API s'imposent déjà comme nouvelle norme de facto pour l'intégration des entreprises.
La question n'est pas de savoir si « elles sont sûres », mais «comment les rendre suffisamment sûres». Dans la pratique, les API constituent une avancée révolutionnaire qui exige un changement fondamental d'approche.

Malheureusement, bon nombre d’entreprises utilisant des API n’ont pas saisi l’enjeu. Cette prise de conscience est d’autant plus importante si nous considérons les API comme l’exemple le plus évident d’une évolution plus profonde et vaste de la conception des systèmes.