



# Anatomie d'une attaque API

Comprendre BOLA et les exploitations de  
gestion des inventaires

## Introduction

---

La plupart des équipes de sécurité comprennent désormais que la recherche proactive des menaces est un élément essentiel d'un programme de sécurité d'entreprise efficace, en particulier en ce qui concerne les interfaces de programmation d'applications (API). Les API fournissent souvent un accès direct aux données, aux fonctionnalités et aux flux de travail. Et alors que les mesures de sécurité du périmètre de base sont largement utilisées pour protéger les applications, les abus d'API et d'autres types d'attaques sont en hausse. En fait, certains des incidents de sécurité les plus médiatisés de ces dernières années étaient liés aux API. Afin de mieux comprendre ces profils d'attaque, tels que l'autorisation brisée au niveau de l'objet (BOLA) et la mauvaise gestion des inventaires, ce document vise à :

- examiner les notions de base des API ;
- examiner pourquoi la sécurité des API est un sujet de plus en plus important ;
- utiliser les incidents de sécurité des API très médiatisés pour mettre en évidence les domaines clés de sécurité des API ;
- illustrer les fonctionnalités nécessaires pour rechercher efficacement les menaces API.

## Notions de base sur les API et les points de terminaison

---

Pour commencer, examinons la terminologie de base. Les API sont utilisées à de nombreuses fins, de la fonctionnalité B2C (business-to-consumer) à la collaboration et à l'intégration B2B (business-to-business) en passant par les fonctions de développement et d'intégration internes. Les API Web, qui communiquent sur le même protocole HTTP que celui utilisé par les navigateurs Web, sont le modèle de mise en œuvre le plus courant. Les fonctionnalités spécifiques fournies par ces API peuvent également parfois être appelées services ou produits API.

Lorsque l'on pense à la sécurité des API, il est également important de comprendre le concept de point de terminaison. Bien que ce terme soit parfois utilisé pour désigner les terminaux informatiques des utilisateurs finaux, il a une signification différente dans le contexte des API. Le point de terminaison API peut être considéré comme une ressource accessible unique faisant partie de l'API, ainsi que comme l'opération qui peut être effectuée sur elle.

**Voici un exemple simple : Un point de terminaison API qui renvoie des informations sur les commandes d'une entreprise spécifique peut être représenté comme suit : GET /orders/{orderID}. Dans ce cas, GET est une méthode HTTP spécifique, tandis que orders et orderID représentent la ressource particulière demandée via l'API.**

## Pourquoi les API constituent-elles le prochain défi majeur en matière de sécurité ?

Par le passé, un attaquant pouvait chercher à pénétrer dans le centre de données d'une entreprise pour accéder aux données d'une organisation et les exfiltrer d'un serveur spécifique. Il pouvait également tenter d'inspecter le trafic réseau de l'entreprise pour collecter des données sensibles. Dans ces scénarios, la recherche proactive des menaces pouvait se concentrer sur des activités telles que les tests de pénétration pour bloquer les points d'entrée possibles des acteurs malveillants.

Dans le monde des API, la dynamique est différente. De nombreuses API sont intrinsèquement accessibles à n'importe qui dans le monde extérieur, les informations d'identification et les clés constituant parfois la seule ligne de défense. Et les acteurs malveillants sont de plus en plus habiles à compromettre ces éléments. En outre, certains des types d'abus d'API les plus dévastateurs peuvent être le fait d'acteurs auxquels l'accès aux API a été accordé mais qui choisissent de les utiliser de manière non autorisée.

## Attaques d'API dans le monde réel

Chez Akamai, 31 % du trafic que nous protégeons correspond à du trafic d'API. Cette augmentation du trafic des API entraîne des effets en aval, tels que l'augmentation des attaques et des abus. [Gartner prévoit qu'en 2024](#), les abus d'API et les violations de données doubleront. Pendant ce temps, de nombreuses équipes de sécurité sont bloquées en mode rattrapage. Les API ne cessent de se multiplier, tandis que les outils de sécurité des applications existants offrent une protection des API très limitée.



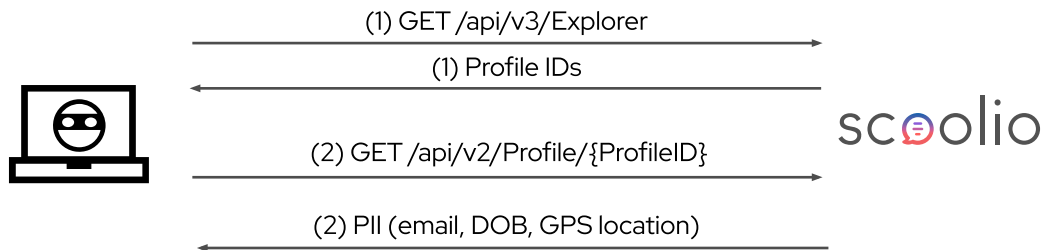
Pour mieux comprendre ce problème, examinons une étude de cas qui illustre l'impact réel que les attaques d'API peuvent avoir sur les entreprises et leurs clients.

## Étude de cas

### Piratage de comptes | Scoolio

L'incident survenu en 2021, qui a affecté l'application éducative allemande Scoolio, est un exemple qui a été très médiatisé. L'application collecte des informations détaillées auprès des utilisateurs étudiants. Elle permet par exemple d'effectuer des tests de personnalité, il fournit des fonctionnalités de réseaux sociaux et de chat, et gère des activités comme la planification des études et le tutorat. Ces fonctions permettent d'accumuler une foule d'informations personnelles. La chercheuse en sécurité Lilith Wittmann a découvert une vulnérabilité BOLA dans les API de l'application éducative qui permettait d'utiliser deux appels d'API pour accéder aux données personnelles et à d'autres données de n'importe quel autre utilisateur de l'application éducative.

Voici le fonctionnement :



#### Étape 1

Envoyer un appel d'API GET `/api/v3/Explorer`.

Cet appel renvoie des UUID, appelés ID de profil dans cette implémentation.

#### Étape 2

Envoyer un appel d'API GET `/api/v2/Profile/{profileID}`.

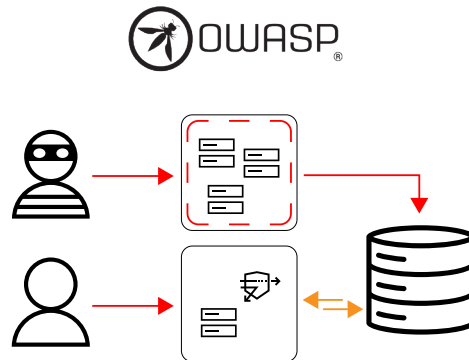
Cette demande renvoie des informations personnelles détaillées pour l'utilisateur concerné, notamment son e-mail, sa date de naissance, sa position GPS, etc.

#### Intérêt de l'utilisation de l'UUID

Bien que les deux scénarios se concentrent sur l'utilisation d'UUID, l'utilisation d'UUID est en fait une très bonne pratique. Le fait d'utiliser des chiffres générés de façon aléatoire au lieu d'une séquence prévisible d'identifiants d'utilisateurs complique l'accès d'un acteur malveillant à des informations massives d'utilisateurs. Le problème survient lorsque les informations UUID sont exposées de manière trop permissive et combinées à des vulnérabilités BOLA.

## Mauvaise gestion des inventaires

L'un des autres aspects de cette vulnérabilité API est qu'elle a bénéficié d'une **mauvaise gestion des inventaires**, risque qui figure à la 9e place sur la liste des 10 principaux risques pour la sécurité des API selon l'OWASP. Si l'on observe attentivement la séquence d'attaque, on remarque que la première étape est appliquée à la version 3 de l'API, tandis que la deuxième étape a été effectuée contre la version 2. Des améliorations ont été apportées dans la version 3 afin de mieux encadrer l'accès aux informations personnelles. Cependant, ces améliorations ont été compromises par le fait que la version 2, plus vulnérable, restait accessible à tous. En fin de compte, les versions 2 et 3 ont été affectées par la vulnérabilité BOLA. Mais la présence inutile de la version 2 a rendu l'impact de la vulnérabilité plus grave.



## Quelles mesures les entreprises prennent-elles aujourd'hui pour protéger leurs API ?

De nombreuses entreprises abordent la sécurité des API en se concentrant sur les trois piliers suivants :

1. **Autorisation centralisée** : tout d'abord, la mise en œuvre d'un moteur d'autorisation centralisé pour tous les ports d'accès API réduira le risque de vulnérabilité des API en évitant les erreurs de développement qui entraînent des mécanismes d'autorisation défectueux.
2. **Test des API** : le test des API constitue une deuxième pratique importante. Le fait de tester toutes les vulnérabilités, en particulier les autorisations brisées, au moyen de l'analyse statique de code et des tests dynamiques, permet de faire apparaître les problèmes tôt dans le processus de développement.
3. **Protection à l'exécution** : le troisième pilier fondamental est constitué par un ensemble de protections à l'exécution pour l'environnement de production. Même les équipes les plus proactives ne détectent pas toutes les vulnérabilités avant le déploiement. Il est donc essentiel d'inspecter l'accès des utilisateurs aux données de production et d'empêcher l'exploitation de catégories de vulnérabilités connues, dans la mesure du possible.

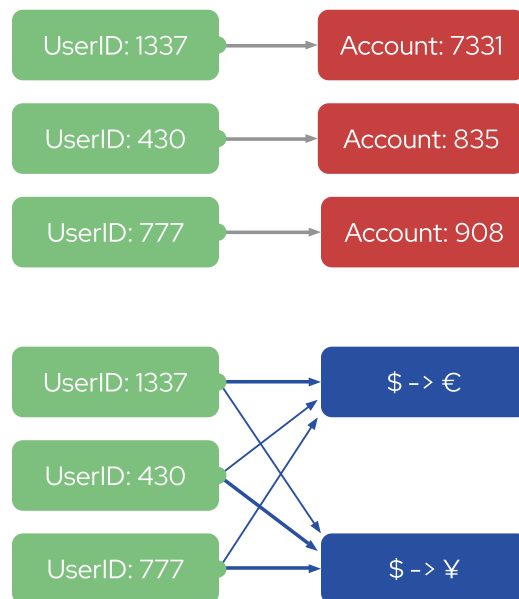
Ces trois pratiques constituent une excellente base pour votre stratégie de sécurité des API. Mais il est également important de se rappeler qu'elles ne sont ni parfaites ni complètes. Par exemple, même les entreprises disposant d'une autorisation centralisée n'ont pas la garantie que les développeurs suivront toujours les meilleures pratiques. Enfin, les outils de protection des applications existants sont souvent efficaces pour détecter des schémas d'attaque connus, mais ils le sont moins pour détecter des menaces plus nuancées comme BOLA.

## Comment utiliser ces fondements avec des techniques de détection BOLA plus avancées ?

L'une des clés pour détecter et atténuer les vulnérabilités BOLA et autres vulnérabilités API nuancées consiste à modéliser les relations entre les entités impliquées dans l'activité des API. Outre les ressources elles-mêmes, cela inclut les acteurs, tels que les utilisateurs, qui tentent d'accéder aux ressources. Si vous cartographiez ces connexions entre les entités d'acteur et les entités de processus métier interagissant avec une API, cela vous permet de différencier les activités légitimes des activités illégitimes lors de l'analyse d'événements API par ailleurs identiques.

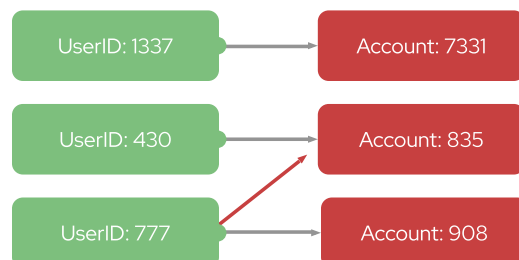
### Illustration de la cartographie des relations

Pour mieux comprendre la cartographie des relations, voici un exemple élémentaire. Une application bancaire prend en charge deux actions. L'une des actions consiste à lire les données de votre compte, y compris les informations comme le solde du compte, les transactions récentes, etc. La deuxième action consiste à afficher les taux de change. La relation entre les utilisateurs et les ressources dans ces exemples est très différente. L'accès aux informations du compte doit être limité à un seul utilisateur. En revanche, la fonction de taux de change doit généralement être accessible à tous les utilisateurs.



Bien qu'il s'agisse d'un exemple élémentaire, la création d'un modèle plus sophistiqué de cartographie des relations entre entités permet de prévenir ou de détecter les BOLA de façon beaucoup plus pratique.

Ici, un utilisateur tente d'accéder à un compte qu'il ne possède pas. L'appel API spécifique peut être identique, mais le contexte ajouté fourni par la cartographie d'entités indique clairement que l'accès ne doit pas lui être accordé.



## Détection avancée des attaques BOLA en pratique

Appliquons maintenant ce concept à des exemples plus complexes, tels que les vulnérabilités de l'étude de cas. Vous trouverez ci-dessous des extraits des entités impliquées dans le scénario :

# scoolio

GET/api/v3/Profile/{profileID}

En-tête :

- Authorization : <MyAccessToken>




L'entité d'acteur est mise en évidence en vert et la ressource demandée (l'ID de profil) est mise en évidence en rouge. Une fois que ces relations sont comprises, des mesures peuvent être prises pour appliquer une logique générale telle que la limitation de l'accès d'un acteur à une seule ressource lorsque cela est approprié. Ceci est loin d'être trivial, car les relations peuvent être plus complexes que cela et inclure des dimensions « un à plusieurs ». Mais certaines techniques comme l'apprentissage automatique et l'analyse comportementale rendent cela possible. Par exemple, la détection réussie d'une vulnérabilité BOLA de l'un de nos clients se présente comme suit :

The screenshot displays the Akamai Security Center interface. At the top, it shows the user 'MyDemoUser' with 1 open alert and a typical location of 'N/A'. The main section is titled 'Suspicious Data Access' and includes a 'Go To Query' button. The alert is dated '21 SEP 2022' at '18:24:50.00' and is categorized as 'Data Leak' with a severity of 'Medium'. The description lists several findings: 'Endpoint "/>

Dans cet exemple, une vulnérabilité BOLA a été simulée dans un environnement de laboratoire. Grâce à la cartographie des entités et à l'analyse comportementale, notre plateforme a détecté la BOLA et généré une alerte riche en informations. Un analyste en sécurité ou un détecteur de menaces qui consulte l'alerte verra que MyDemoUser a accédé à son propre profil utilisateur pour modifier son mot de passe, action qui est sanctionnée. Mais on observe sur le calendrier qu'il a effectué peu de temps après un autre appel API pour modifier le mot de passe administrateur. S'agissant clairement d'un acte non autorisé sur la base de la relation entre l'acteur et la ressource, l'alerte a été générée.

## Par où commencer en ce qui concerne votre initiative de sécurité des API

La sécurité des API est un processus continu pour la plupart des entreprises. Il peut donc être difficile de savoir par où commencer. Bien que les trois piliers fondamentaux ci-dessus constituent un point de départ utile, l'efficacité de votre approche sera grandement améliorée si vous suivez les trois recommandations suivantes lors de la mise en œuvre :

-  1. Veillez à ce que votre inventaire d'API soit toujours à jour
-  2. Contrôlez les environnements d'API hors production et de production
-  3. Appliquez les relations entre les entités

Vous ne pouvez pas protéger les API que vous ne connaissez pas. Ainsi, une protection efficace des API commence par un inventaire des API à jour et une évaluation de la stratégie de sécurité. De même, au fur et à mesure que vous développez vos capacités de surveillance de la sécurité des API, il est important de les étendre aux implémentations API de production et hors production. Et surtout, la surveillance et l'application de vos API doivent aller au-delà des seules actions et prendre en compte les relations entre les entités impliquées dans votre activité des API. Cela vous permettra de détecter les vulnérabilités et les lacunes en matière de protection et d'assurer la conformité avec les modèles d'utilisation prévus des API. La compréhension des comportements au sein de vos API vous permettra de détecter tout abus.

**Vous souhaitez en savoir plus sur les attaques API et sur la manière dont vous pouvez vous en protéger ? Consultez notre aperçu des 10 principaux risques pour la sécurité des API selon l'OWASP.**



Akamai protège votre expérience client, votre personnel, vos systèmes et vos données en vous aidant à intégrer la sécurité dans tout ce que vous créez, dans toutes vos conceptions et diffusions. La visibilité de notre plateforme sur les menaces mondiales nous aide à adapter et à faire évoluer votre stratégie de sécurité (activer Zero Trust, bloquer les ransomwares, sécuriser les applications et les API, ou lutter contre les attaques DDoS), vous donnant ainsi la confiance nécessaire pour innover, prospérer et ouvrir le champ des possibles. Pour en savoir plus sur les solutions de Cloud Computing, de sécurité et de diffusion de contenu, rendez-vous sur [akamai.com](https://akamai.com) et [akamai.com/blog](https://akamai.com/blog), ou abonnez-vous à Akamai Technologies sur [X](#) (anciennement Twitter) et [LinkedIn](#).