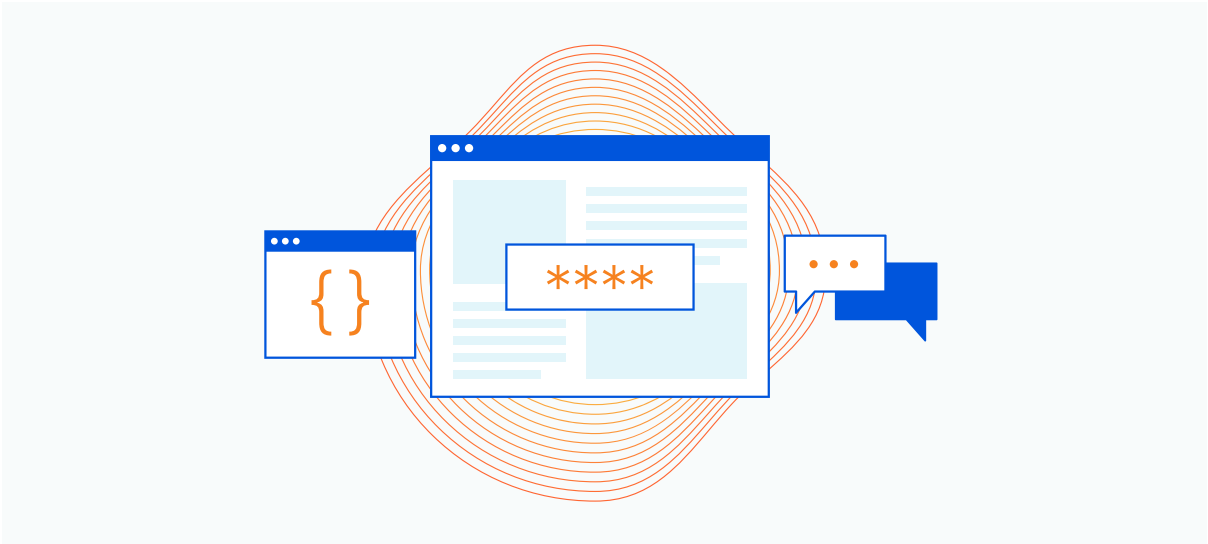
LIVRE BLANC

CLOUDFLARE

Guide de la sécurité des API

+33 7 57 90 52 73 | enterprise@cloudflare.com | www.cloudflare.com/fr-fr/

L'univers des applications tourne autour des API



Aujourd'hui, nous savons tous que ce sont les API, les interfaces de programmation d'applications, qui font tourner le monde. Plus concrètement, elles permettent aux différentes applications modernes de communiquer entre elles. Les applications web ou mobiles ont accès à un backend où sont stockées et traitées les données. Les API peuvent être publiques, elles permettent alors à des applications de différentes entreprises de communiquer, ou privées, ce qui est courant et permet de regrouper des applications internes au service d'objectifs commerciaux.

Résultat ? Des applications, sites Web et applications mobiles plus robustes et complets avec des fonctionnalités plus vastes et des données plus diversifiées.

Par exemple, au lieu de créer intégralement leur propre service de paiement, les entreprises de covoiturage peuvent en ajouter un à l'aide des API des entreprises de paiement. Les sites comparateurs de vols sont également un bon exemple. Afin de nous présenter les heures de vols, prix, destinations et tout ce que nous avons besoin de savoir pour acheter un billet d'avion, ils se connectent aux bases de données des compagnies aériennes à l'aide d'appels d'API pour obtenir les données souhaitées et les afficher dans la page de résultats de notre recherche dans le comparateur.

Les API deviennent de plus en plus importantes et de plus en plus d'entreprises se décrivent comme une « API avant tout ». Dans certains cas, le véritable produit proposé par l'entreprise est une API avec un modèle économique organisé autour de sa propre consommation. Par exemple, si une entreprise spécialisée dans les données météorologiques a fait de son API son produit, d'autres entreprises qui ont besoin d'informations météorologiques lui paieront un abonnement mensuel pour pouvoir accéder à l'API. Dans de nombreux autres cas, étant donné qu'il est prévu qu'une application interagisse avec d'autres applications, les API sont conçues parallèlement, voire avant le code exécutant la fonctionnalité réelle du produit (voire, avant celui-ci), au lieu d'être ajoutées à la fin du développement.

Les entreprises consacrent du temps et des efforts de réflexion pour aboutir à une API permettant d'exposer les données avec le plus de pertinence possible, ce qui est fondamental pour générer des revenus et des modèles commerciaux.

Cependant, il est difficile d'élaborer des API parfaites. En effet, à l'instar de tout composant logiciel, elles peuvent comporter des vulnérabilités, conduisant à des risques pour la sécurité que nous allons évoquer dans ce document.

Sans entrer dans les détails, les API sont partout, elles auront le vent en poupe dans les années à venir, et doivent donc être protégées. C'est pourquoi nous allons examiner les attaques contre les API et les aspects de défense en profondeur lorsqu'il s'agira de penser la sécurité des API organisationnelles.

Ampleur des API : quelques chiffres

50 %

du trafic circulant dans Cloudflare est du trafic d'API

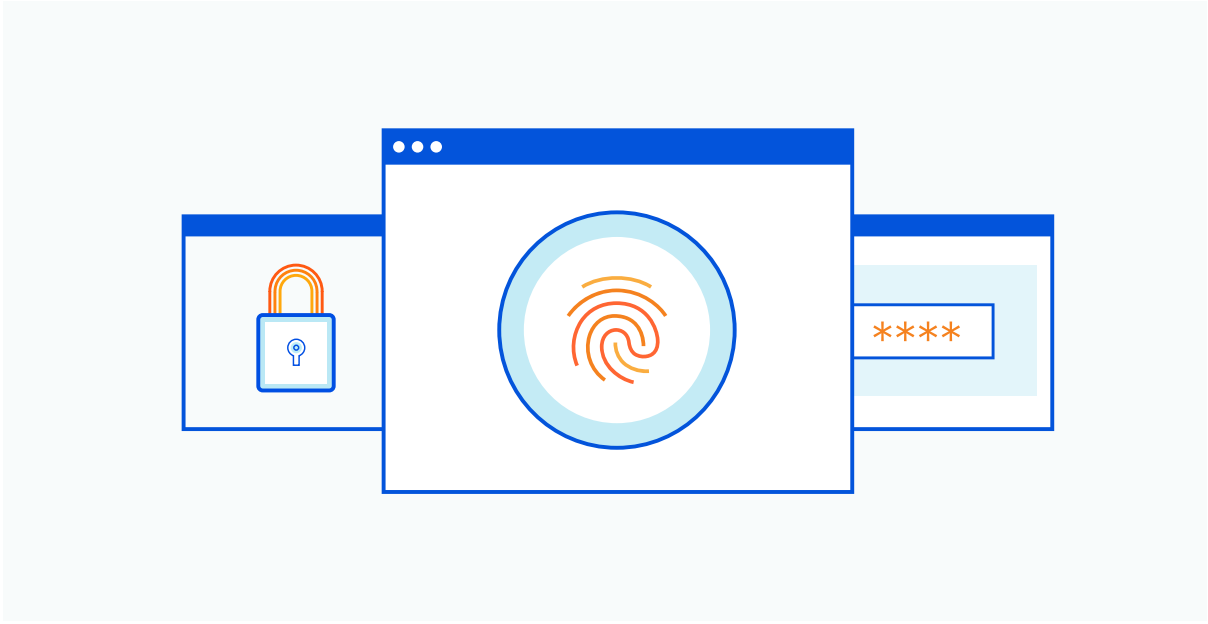
61 %

d'augmentation du trafic d'API d'une année sur l'autre

Le site Programmable Web[1] fait état de plus de 24 000 API publiées et reconnues. Il s'avère toutefois que la plupart des API sont privées et relient des applications internes entre elles. Il estime que le nombre d'API privées se compterait en millions.

Et lorsque nous disons que les API gagnent du terrain, Cloudflare est aux premières loges pour observer leur croissance. D'après les données de Cloudflare Radar pour la première moitié de 2021, environ la moitié du trafic du réseau Cloudflare concernait des API. Qui plus est, ce chiffre a augmenté de 61 pour cent entre 2020 et 2021.

Compte tenu de l'importance des données qu'elles exposent, nous voyons déjà à quel point elles représentent une énorme surface d'attaque que nous devons protéger. Comment le savons-nous ? Au cours des dernières années, de nombreuses attaques de premier plan ont ciblé des API.



[1]https://www.programmableweb.com/apis/directory

Une surface d'attaque en pleine expansion

Nous savons que les API sont partout et qu'elles sont fondamentales pour la réussite d'une entreprise moderne. Elles exposent la logique d'application et peuvent transmettre des données sensibles à d'autres applications.

Il s'avère toutefois, et cela ne surprendra personne, que les attaquants sont au courant de cet état de fait et comptent bien exploiter cette surface d'attaque en pleine expansion au sein de l'entreprise.

Gartner avait peut-être raison lorsqu'il annonçait[2] que d'ici 2022, les utilisations abusives des API « cesseraient d'être rares et deviendraient le vecteur d'attaque le plus fréquent, donnant lieu à des violations de données pour les applications web des entreprises. »

Il reste à prouver que les API sont en train de devenir le vecteur d'attaque le plus fréquemment ciblé, mais il est déjà évident que les API vont rester dans le viseur des attaquants.

Nous disons « rester » car le monde a déjà été témoin de certaines violations très médiatisées imputables à une sécurité insuffisante des API ou à un développement des API qui ne tenait pas compte de la sécurité.

T-Mobile a été victime d'une attaque API en 2015 lorsque 15 millions de clients ayant acheté de nouveaux appareils ou ouvert des comptes T-mobile ont vu leurs informations exposées. Il s'agissait de leurs nom, adresse, numéro de sécurité sociale, de permis de conduire et de passeport. Vice précise que l'attaque a été perpétrée par simple modification du paramètre de numéro de téléphone dans l'appel d'API. Le numéro de téléphone de n'importe quel utilisateur pouvait être interrogé et l'API T-Mobile envoyait des réponses avec des données sensibles du compte de la personne dont le numéro de téléphone était interrogé.

Une autre violation liée à une API a touché le service postal des États-Unis (USPS) lorsqu'il a été découvert qu'il manquait. Une autorisation de base à l'API prenant en charge le suivi en temps réel des colis. Lorsqu'un utilisateur se connectait, il pouvait demander les informations de compte de n'importe quel utilisateur en utilisant les paramètres de recherche par caractère générique, qui pouvaient renvoyer l'ensemble des enregistrements d'un jeu de données. C'est ainsi que 60 millions de détenteurs de comptes USPS se sont retrouvés en danger.

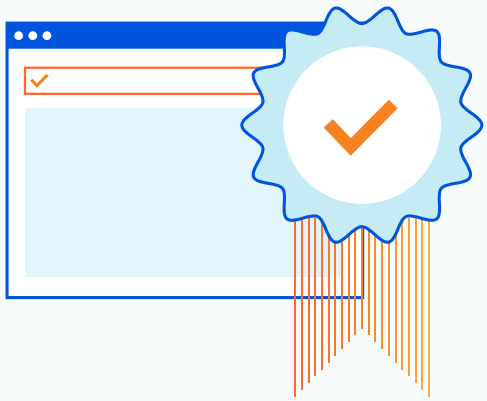
En 2019, JustDial, un grand moteur de recherche local en Inde, a en quelque sorte divulgué toutes les données client lorsqu'il a laissé accessible à tous les anciennes versions des API, qui avaient été depuis remplacées par de nouvelles versions. Plus grave encore, aucune authentification n'avait réellement été prévue, ce qui signifie que n'importe qui pouvait appeler les API et éliminer les données du serveur de production. Autrement dit, il était possible d'accéder aux données utilisateur sans procédés techniques avancés.

Facebook, bien que très avancée dans l'utilisation de GraphQL, a également connu plusieurs violations à cause de ses API. Par exemple, à la fin de l'année 2019, une base de données de Facebook a été vidée, exposant à de grands risques les nom, numéro de téléphone et ID utilisateur de plus de 260 millions d'utilisateurs.

La liste des organisations qui ont connu des problèmes de sécurité concernant les API est longue. Les raisons en sont nombreuses. D'abord, les vulnérabilités d'API sous-jacentes dues à une conception non sécurisée ouvrent la voie aux attaques. Qui plus est, jusqu'à présent, les organisations ne disposaient pas d'outils de sécurité consacrés aux architectures donnant la priorité aux API. Peut-être utilisaient-elles des outils de sécurité web tels que des pare-feu WAF ou la limitation de débit, mais ceux-ci ont été conçus pour protéger les applications, et non les API. Cette situation conduit à des menaces telles que les faux positifs et fait naître de manière évidente la nécessité d'une sécurité des API conçue pour un trafic largement automatisé.

[2]Source : Gartner : « API Security: What You Need to Do to Protect Your APIs », mars 2021

Une nouvelle version du top 10 OWASP des risques liés aux API



L'aspect positif dans tout cela est que la Fondation OWASP, une organisation qui se consacre depuis longtemps à l'amélioration de la sécurité des applications, s'intéresse désormais à la question. OWASP est connue pour son Top 10 des risques pour la sécurité des applications Web et elle vient de publier le Top 10 des risques liés à la sécurité des API, une liste des risques et vulnérabilités concernant les API.

En réalité, tout ce qui nous inquiète depuis longtemps au sujet de la sécurité des applications est valable également pour la conception et la sécurisation des API.

À commencer par le fait que n'importe quelle organisation pour laquelle les API sont une priorité doit envisager la sécurité dès le début, au moment même de la conception des API. Examinons certaines des attaques que nous venons d'évoquer, et le risque pour la sécurité selon OWASP qu'elles ont exploité.

Top 10 OWASP des risques liés aux API

1. **Broken object level authorization (Violation de l'autorisation au niveau de l'objet)**
2. **Broken User Authentication (Défaillance de l'authentification des utilisateurs)**
3. **Excessive Data Exposure (Exposition excessive de données)**
4. **Lack of Resources & Rate Limiting (Manque de ressources et limitation du débit)**
5. **Broken Function Level Authorization (Violation de l'autorisation au niveau de la fonction)**
6. **Mass Assignment (Attribution de masse)**
7. **Security Misconfiguration (Mauvaise configuration de sécurité)**
8. **Injection**
9. **Improper Assets Management (Mauvaise gestion des ressources)**
10. **Insufficient Logging & Monitoring (Connexion et surveillance insuffisantes)**

Principales difficultés concernant la sécurité des API

1. Violation de l'authentification et de l'autorisation

Examinons plus en détail quelques-uns des principaux risques pour la sécurité des API relevés par OWASP que les attaques précédemment mentionnées ont exploités, en commençant par l'authentification et l'autorisation.

JustDial a succombé à une violation de l'authentification au niveau des points de terminaison, ce qui permettait à n'importe qui de les appeler. Lorsque l'authentification est en place, seuls les appels d'API disposant des bons certificats TLS, clés d'API, jetons web, etc. sont autorisés à effectuer des requêtes, ce qui réduit de manière drastique les risques pour la sécurité des API.

Si l'on regarde le numéro 1 de la liste OWASP, de nombreuses attaques d'API profitent d'une autorisation faible, corrompue ou inexistante, c'est ce que nous avons pu observer dans le cas d'USPS et de T-Mobile. La violation de l'autorisation au niveau de l'objet est courante : il s'agit pour un attaquant d'exploiter des points de terminaison API en remplaçant l'ID d'un objet pour lequel il n'a pas de droit d'accès par le numéro d'ID d'un objet pour lequel il dispose des autorisations d'accès. Il suffit souvent de changer l'ID d'objet dans une requête pour obtenir l'autorisation d'accéder aux données.

Les paramètres de requête et de chemin d'API comprennent l'ID de la ressource en cours d'accès :

Appel autorisé :

`GET api.greatsampleapis.com/v1/users/235` où 235 est l'ID utilisateur.

Les appels d'API manipulés peuvent obtenir un accès sans autorisation en remplaçant 235 par 236, c'est-à-dire en corrigeant l'identifiant d'objet, dans le cas présent l'ID utilisateur, pour accéder aux données de l'utilisateur 236.

```
GET api.greatsampleapis.com/v1/users/236
```

En l'absence de contrôles de l'autorisation, cet appel peut aboutir. Les développeurs doivent modéliser les menaces pesant sur leurs points de terminaison pour empêcher les attaquants de corriger ou de modifier la valeur d'un ID d'objet dans le but d'accéder aux autres données. Il peut être judicieux également d'utiliser des valeurs non prévisibles pour les ID d'objet, afin d'éviter qu'elles soient séquentielles et faciles à deviner.

2. Attribution de masse, exposition des données et attaques par injection

D'autres attaques, d'une autre catégorie, provoquent une exposition excessive des données dans les réponses ou permettent de modifier des objets internes par des entrées.

On parle d'exposition excessive des données lorsqu'une API cherche à exposer largement les propriétés d'un objet et renvoie des données en excès dans une réponse, en partant du principe que les clients à l'origine de la requête filtreront les données.

Les attaquants peuvent utiliser des détails supplémentaires fournis par la réponse pour organiser une attaque encore plus puissante ou des e-mails d'hameçonnage. Par exemple, si une réponse renvoie l'ensemble des données ci-dessous, elles peuvent être utilisées pour produire des e-mails d'hameçonnage très convaincants.

```
{
“Id”: 213,
“FirstName”:”Sanjay”,
“LastName”:”Smythe”,
“EmailAddress”:”ssmythe@hacketyhack.com”,
“Occupation”: “Assistant to the Deputy Associate Vice Sub-undersecretary”,
“DOB”:”1986-05-21”,
“Bank”:”Easygo Financial”,
“AccountNumber”: 1362886306,
“PetName”: “Aloysius”,
 }
```

Les attaques par attribution de masse permettent aux appels d'API de modifier ou d'exploiter des valeurs internes lorsque les API exposent des objets internes et des variables.

Voici comment OWASP présente les choses :

> « Afin de faciliter la tâche des développeurs, les infrastructures logicielles permettent parfois aux développeurs de lier automatiquement les paramètres des requêtes HTTP à des variables ou des objets du code de programme…Il arrive que des attaquants utilisent cette méthodologie pour créer de nouveaux paramètres que le développeur n'avait pas prévus et qui, à leur tour, créent ou remplacent dans le code du programme de nouvelles variables ou de nouveaux objets qui n'étaient pas dans les intentions des développeurs. »

Quelle doit être l'attitude des développeurs ? Ils doivent être conscients des risques qu'ils font naître lorsqu'ils font appel à l'attribution de masse dans le développement et doivent éviter d'exposer des objets ou des variables internes susceptibles d'être exploités. Il peut également être judicieux de dresser une liste d'autorisations des propriétés qui peuvent être mises à jour par les clients.

Les applications web sont depuis longtemps à la merci des attaques par injection et il en va de même pour les API. Les attaques par injection sont bien connues, nous ne nous y attarderons donc pas, nous nous contenterons de préciser que les entrées doivent être validées et assainies avant d'être transmises. Il convient de tout faire pour favoriser le recours à des mesures de prévention des fuites de données dans les réponses API et de limiter le nombre d'enregistrements qui peuvent être renvoyés afin d'empêcher un incident de divulgation massive.

3. Utilisation abusive des ressources et API fantômes/indésirables

D'autres types d'attaques peuvent exercer une utilisation abusive des API, consommant ainsi des quantités excessives de ressources informatiques qui saturent et succombent devant ce qui ressemble à une attaque DoS. Si aucune limite n'est imposée sur des aspects tels que le nombre de requêtes par client/ressource, les enregistrements renvoyés au sein d'une seule réponse ou le volume de la charge utile de la demande, la voie est libre pour ces attaques.

Comme nous l'avons vu dans le cas des attaques qui ont frappé JustDial, il peut arriver que des API de production soient oubliées et deviennent fantômes ou indésirables dans la mesure où elles sont rarement protégées et sont donc exposées à des exploitations. Comme pour tout ce qui a trait à la sécurité, il est impératif que nous ayons une bonne visibilité sur notre parc informatique ou notre surface d'attaque pour ensuite appliquer les contrôles de sécurité qui s'imposent. La visibilité sur l'ensemble de nos points de terminaison API est tout aussi importante.

Pendant le développement des API, les équipes doivent pouvoir compter sur une solide méthode de suivi des versions d'API afin de connaître avec précision les API qui sont en production et celles qui sont obsolètes.

Éléments à prendre en compte pour la sécurisation des API.

Nous avons évoqué ce que sont les API, expliqué en quoi elles sont importantes et présenté les attaques générales qui ciblent les API. Examinons maintenant la manière dont Cloudflare a organisé la sécurité des API pour sécuriser celles-ci contre les attaques les plus courantes. Pour être efficace, la sécurité des API doit englober tous les éléments allant de la visibilité aux modèles de sécurité positive en passant par l'arrêt des utilisations abusives et la protection des données.

Cloudflare API Shield

Pierre angulaire de la visibilité

VISIBILITÉ

Comme c'est le cas dans tous les aspects de la sécurité, avant de pouvoir protéger quelque chose, il importe de le voir. Il en va de même pour les API, en particulier lorsque les entreprises comptent leurs points de terminaison API par centaines, voire par milliers.

La découverte et la visibilité des API sont un aspect fondamental de la gestion des API. C'est ce qui permet aux organisations de toujours avoir un aperçu précis de leur parc de points de terminaison API et d'éviter les difficultés que poseraient des API fantômes ou indésirables.

Comme nous l'avons vu dans le cas de JustDial, si des organisations perdent la trace de leurs API, des violations de données risquent de se produire.

Défense en profondeur des API

Protections de la couche 7 des API

Nous avons depuis longtemps déployé des pare-feu applicatifs web pour protéger les applications contre les attaques DDoS sur la couche 7. La protection des API doit commencer par un grand nombre de ces contrôles fiables, tels que la limitation du débit et la protection contre les attaques DDoS pour parer les attaques en déni de service et les tentatives de connexion par force brute ainsi que les utilisations abusives en général perpétrées par des adresses IP spécifiques. C'est ce qui mettra en place des limites à l'utilisation des API et garantira la disponibilité en protégeant du 4e point du top OWASP des risques liés aux API, à savoir le manque de ressources et la limitation du débit.

Les règles du WAF peuvent être utilisées pour identifier et bloquer des attaques courantes ciblant les API.

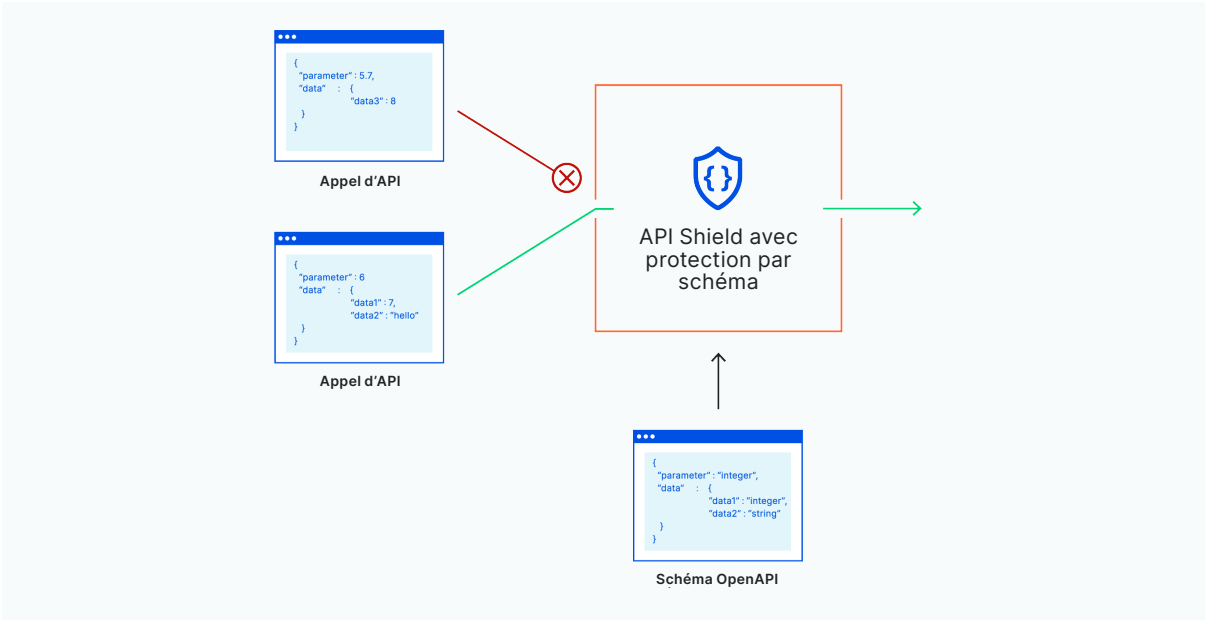
Authentification et autorisation

Authentification mTLS

Dans les attaques API que nous avons présentées, nous avons pu constater à quel point l'absence d'authentification peut être dévastatrice. L'authentification doit être intégrée dès le départ et elle doit être étayée par un protocole TLS à authentification réciproque pour imposer une identité basée sur des certificats dans des cas d'utilisation impliquant par exemple des mobiles ou l'IdO. Cette démarche est un modèle plus positif, à liste d'autorisations, qui n'autorise que les requêtes des clients légitimes possédant des certificats valides à se connecter.

Vérifications des identifiants compromis

Les API ne sont pas à l'abri des attaques par infiltration de comptes, qui consistent à tenter de se connecter de manière répétée avec des identifiants volés. Ces identifiants de compte peuvent être compromis par des violations chez des tiers sur lesquels l'organisation n'exerce aucun contrôle. Dans le cadre des contrôles d'authentification, la sécurité des API doit être en mesure d'analyser les identifiants d'authentification dès la connexion, et la comparer à une base de données d'identifiants dont on sait qu'ils ont fuité. Si les identifiants semblent compromis, la sécurité des API doit déclencher des mesures de sécurité supplémentaires telles que la réinitialisation du mot de passe ou l'authentification multifacteur et, bien sûr, bloquer la tentative.



La validation de schéma évalue chaque requête par rapport à un schéma d'API qui consigne ou bloque celles qui ne sont pas conformes à ce schéma.

Protection positive des API

Validation de schéma des API

Les développeurs se donnent beaucoup de mal pour créer un schéma d'API, qui figure dans la documentation, ou des règles de base, correspondant à l'interaction avec l'API telle qu'elle est attendue de la part des autres. C'est ce qui établit les éléments tels que les méthodes de requête et les opérations sur chaque point de terminaison (GET /utilisateurs, POST /utilisateurs) ou les paramètres d'entrée et de sortie pour chaque opération. OpenAPI v3, également connu comme la norme Swagger, est le schéma le plus connu pour définir les API.

Pour être fiable, la sécurité des API doit être un modèle Zero Trust positif qui met en application ce schéma.

Une fois que le schéma est en place, les requêtes doivent être automatiquement validées par rapport à celui-ci. Toutes les opérations d'API sont bloquées, à l'exception de celles dont la conformité à ce schéma a été validée.