Akamai

WHITE PAPER

Anatomía del ataque a una API

Comprensión de BOLA y de las vulnerabilidades de gestión de inventario

Introducción

La mayoría de los equipos de seguridad ya saben que la búsqueda proactiva de amenazas resulta un elemento esencial de un programa de seguridad empresarial eficaz, en especial en lo que respecta a las API. Las API suelen proporcionar acceso directo a los datos, las funciones y los flujos de trabajo. Y, aunque generalmente se usan medidas básicas de seguridad perimetral para proteger las aplicaciones, el abuso de API y otros tipos de ataques no paran de crecer. De hecho, algunos de los incidentes de seguridad de mayor repercusión que han aparecido en los titulares en los últimos años están relacionados con las API. Para entender mejor estos perfiles de ataque, como la vulnerabilidad Autorización a nivel de objeto comprometida (BOLA) y las vulnerabilidades por una gestión de inventario inadecuada, en este documento:

- Se repasarán los conceptos básicos de las API.
- Se analizará el motivo por el que la seguridad de las API es un tema cada vez más importante.
- Se utilizarán algunos incidentes de seguridad de las API de gran repercusión para resaltar las áreas clave en este contexto.
- Se ilustrarán las funciones necesarias para realizar una búsqueda efectiva de amenazas de API.

Conceptos básicos de las API y los terminales

Para empezar, repasemos algunos términos básicos. Las API se utilizan para muchos fines, desde la funcionalidad de empresa a cliente (B2C), la colaboración e integración de empresa a empresa (B2B) y las funciones de desarrollo e integración internas. Las API web, que se comunican a través del mismo protocolo HTTP que usan los navegadores web, son el modelo de implementación más habitual. La funcionalidad específica que proporcionan estas API también se conoce a veces como servicios o productos de API.

Al pensar en la seguridad de las API, también es importante entender el concepto de terminal. Aunque este término a veces se utiliza para referirse a los dispositivos informáticos de usuario final, su significado varía en el contexto de las API. Se puede considerar que un terminal de API es un único recurso accesible que forma parte de la API y también incluye la operación que se puede realizar en este.

> Este es un ejemplo sencillo. Un terminal de API que devuelve información de pedidos para una empresa específica puede representase de la siguiente forma: GET /orders/{orderID}. En este caso, GET es un método HTTP específico, mientras que orders y orderID representan el recurso concreto que se solicita a través de la API.

¿Por qué las API representan el próximo gran desafío de seguridad?

Anteriormente, un atacante podía haber puesto su mira en un centro de datos empresarial con la intención de acceder a un servidor específico para obtener los datos de una organización. O bien, tal vez hubiera intentado inspeccionar el tráfico de la red de la empresa para obtener datos confidenciales. En estas situaciones, la búsqueda proactiva de amenazas podría centrarse en actividades como las pruebas de penetración con el fin de bloquear los puntos por los que los atacantes podrían entrar.

En el mundo actual, donde el uso de las API está muy extendido, la dinámica es otra. Cualquier persona externa puede acceder, intrínsecamente, a muchas API mediante credenciales y claves que, en ocasiones, son la única línea de defensa. Además, los atacantes tienen una gran habilidad a la hora de poner en peligro estos elementos. Es más, algunos de los tipos más perjudiciales de abuso de API pueden producirse porque partes a las que se les ha otorgado acceso a estas deciden utilizarlas de formas no autorizadas.

Ataques reales a las API

En Akamai, el 31 % de todo el tráfico que protegemos es tráfico de API. Este aumento del tráfico de las interfaces de programación de aplicaciones tiene efectos secundarios, como un aumento de los ataques y abusos. Gartner prevé que en 2024 se duplicarán los abusos a las API y las filtraciones de datos. Mientras tanto, muchos equipos de seguridad lo único que pueden hacer es intentar mantenerse al día. Las API siguen multiplicándose, mientras que las herramientas de seguridad de aplicaciones existentes ofrecen una protección de API muy limitada.

Las soluciones de seguridad de aplicaciones existentes no están diseñadas para las API

En 2024, los abusos de las API y las filtraciones de datos relacionadas casi se duplicarán[1]

31 %
Tráfico web que corresponde a las API [2]

1 Los 10 aspectos más importantes que los responsables de ingeniería de software deben saber sobre las API

2 Los investigadores de amenazas de Akamai han identificado que el 31 % del tráfico total protegido por Akamai es tráfico de API

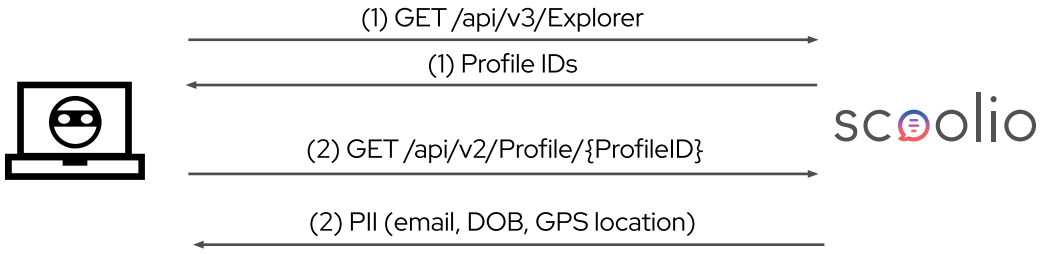
Para explicar esto en la práctica, vamos a revisar un caso auténtico en el que se muestra el efecto real que los ataques a las API pueden tener en las empresas y en sus clientes.

Caso real

Robo de cuentas | Scoolio

Un incidente que tuvo una gran repercusión ocurrió en 2021 y afectó a la aplicación alemana de educación Scoolio. En ella se recopila una gran cantidad de información de los estudiantes que la usan. Por ejemplo, se realizan tests de personalidad, se proporcionan funciones de chat y redes sociales, además de gestionarse actividades como planificación de estudios y tutorías. Estos componentes incluyen mucha información de identificación personal (PII). La investigadora de seguridad Lilith Wittmann detectó una vulnerabilidad BOLA en las API de la aplicación de educación, que permitía utilizar dos llamadas a API para acceder a PII y a otros tipos de información de cualquier otro usuario de la aplicación.

Así es como funcionaba:



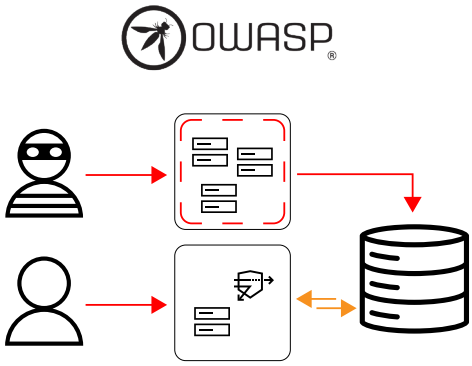
Paso 1
Se envía una llamada a la API GET /api/v3/Explorer.
Esta llamada devuelve varios identificadores únicos universales (UUID), a los que se hace referencia como ProfileID en esta implementación.

Paso 2
Se envía una llamada a la API GET /api/v2/Profile/{ProfileID}.
Esta solicitud devuelve mucha PII sobre el usuario en cuestión, incluida la dirección de correo electrónico, la fecha de nacimiento, la ubicación GPS, entre otros datos.

Importancia del uso de los UUID

Aunque ambos escenarios se centran en el uso de los UUID, el uso de estos identificadores es, de hecho, una práctica muy útil. El hecho de que se utilicen números generados aleatoriamente en lugar de una secuencia predecible de identificadores de usuario dificulta al atacante el acceder a la información de los usuarios en masa. El problema surge cuando la información de los UUID queda muy expuesta y se combina con vulnerabilidades BOLA.

Gestión del inventario inadecuada



Otro aspecto de esta vulnerabilidad de API es que se aprovechaba de la gestión incorrecta del inventario, que aparece como número 9 en la lista de las 10 principales vulnerabilidades de las API, según el Proyecto Abierto de Seguridad de Aplicaciones Web (OWASP). Si observa con atención la secuencia de ataque, se dará cuenta de que el primer paso se aplica a la versión 3 de la API, mientras que el segundo paso se realiza con la versión 2. Gracias a las mejoras realizadas en la versión 3 se proporcionaba un control de acceso más estricto a la información de identificación personal, si bien la importancia de esta se vio mitigada por el hecho de que a la versión 2, más vulnerable, podía seguir accediendo cualquiera. En última instancia, tanto la versión 2 como la versión 3 se vieron afectadas por la vulnerabilidad BOLA. Pero la presencia innecesaria de la versión 2 hizo que las consecuencias de la vulnerabilidad fueran más graves.

¿Qué medidas toman actualmente las organizaciones para proteger sus API?

Muchas organizaciones se centran en estos tres pilares para abordar la seguridad de las API:

1. Autorización centralizada: en primer lugar, la implementación de un motor de autorización centralizado para todos los puertos de acceso a las API reducirá el riesgo de vulnerabilidad en estas, al evitar errores en el desarrollo que dan lugar a mecanismos de autorización defectuosos.

2. Pruebas de API: una segunda práctica importante son las pruebas de API. Las pruebas de todas las vulnerabilidades, especialmente de las autorizaciones comprometidas, mediante análisis de código estático y pruebas dinámicas, detectarán los posibles problemas en las primeras fases del proceso de desarrollo.

3. Protección del tiempo de ejecución: el tercer pilar fundamental es un conjunto de protecciones del tiempo de ejecución para el entorno de producción. Incluso los equipos más proactivos no podrán detectar todas las vulnerabilidades antes de la fase de implementación. Por ese motivo, es fundamental inspeccionar el acceso de los usuarios a los datos de producción y evitar, en la medida de lo posible, la explotación de categorías conocidas de vulnerabilidades.

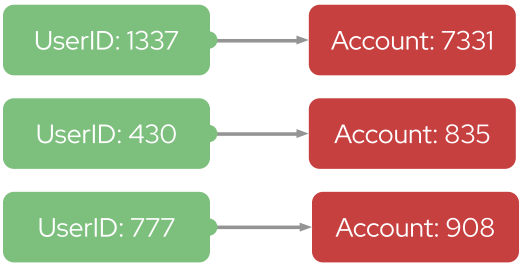
Estas tres prácticas proporcionan una excelente base para su estrategia de seguridad de las API. No obstante, también resulta importante recordar que ni son exhaustivas ni con ellas se consiguen resultados definitivos. Por ejemplo, incluso en aquellas organizaciones en las que se usa la autorización centralizada no existe la garantía de que los desarrolladores sigan siempre las prácticas recomendadas. Por último, las herramientas de protección de aplicaciones existentes suelen ser útiles a la hora de detectar patrones de ataque conocidos, pero menos a la hora de detectar amenazas más sutiles como BOLA.

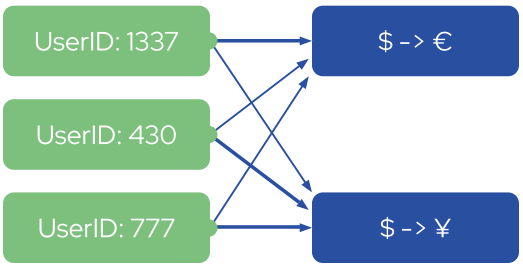
¿Cómo puede mejorar esta base con técnicas de detección de BOLA más avanzadas?

Una de las claves para detectar y mitigar BOLA y otro tipo de vulnerabilidades más sutiles de las API es conformar las relaciones entre las entidades involucradas en la actividad de las API, entre las que se incluyen actores, como los usuarios, que intentan acceder a los recursos, además de los propios recursos. Si puede asignar estas conexiones entre las entidades de actores y las entidades de procesos de negocio que interactúan con una API, podrá diferenciar entre actividades legítimas e ilegítimas a la hora de analizar eventos de API que, de lo contrario, serían idénticos.

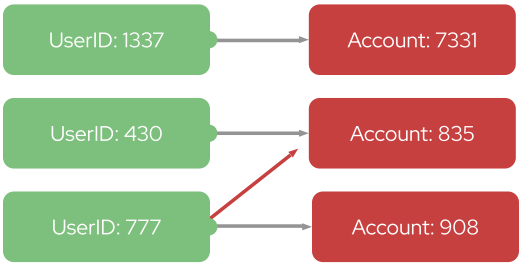
Diagrama de la asignación de relaciones

Para comprender mejor la asignación de relaciones, observe este ejemplo básico. Una aplicación de banca admite dos acciones. Una acción permite leer los datos de su cuenta, incluida información como el saldo de la misma, las transacciones recientes, etc. La segunda acción permite ver los tipos de cambio de las divisas. La relación entre los usuarios y recursos de estos ejemplos es muy distinta. El acceso a la información de la cuenta debe estar limitado a un solo usuario. Por su parte, la función de tipo de cambio debería estar disponible de forma habitual para todos los usuarios.





Aunque se trata de un ejemplo muy básico, la creación de un modelo más sofisticado de asignación de relaciones entre entidades hace que sea mucho más práctico evitar o detectar BOLA.



Aquí podemos ver a un usuario que intenta acceder a una cuenta que no es suya. La llamada a la API específica puede ser idéntica, pero el contexto adicional que se obtiene gracias a la asignación de entidades deja claro que no se debe permitir.

Detección avanzada de ataques BOLA en la práctica

Tras esto, apliquemos el concepto a ejemplos más complejos, como las vulnerabilidades del caso real. A continuación se muestran fragmentos de las entidades incluidas en el escenario:

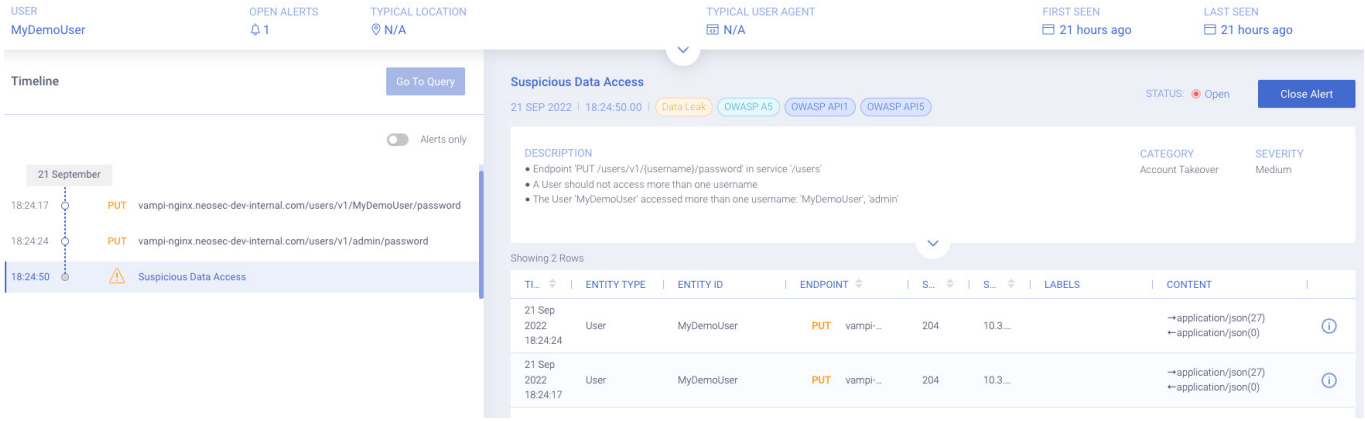
scoolio

GET/api/v3/Profile/{ProfileID}

Encabezados:

- Authorization:<MyAccessToken>

La entidad del actor aparece resaltada en color verde y el recurso solicitado (el ID del perfil) aparece resaltado en color rojo. Una vez que se comprenden estas relaciones, se pueden tomar medidas para aplicar la lógica general, como limitar el acceso de un actor a un solo recurso cuando sea necesario. El proceso puede llegar a ser bastante complicado, ya que las relaciones pueden ser más complejas de lo mencionado e incluir dimensiones de uno a muchos. Son técnicas como el aprendizaje automático y el análisis del comportamiento las que lo hacen posible. Por ejemplo, una detección correcta de una vulnerabilidad BOLA en uno de nuestros clientes tendría este aspecto:



En este ejemplo, se ha simulado una vulnerabilidad BOLA en un entorno de laboratorio. A través de la asignación de entidades y del análisis del comportamiento, nuestra plataforma detectó la vulnerabilidad BOLA y generó una alerta con información detallada. Un analista de seguridad o investigador de amenazas que vea la alerta observará que MyDemoUser ha accedido a su propio perfil de usuario para cambiar su contraseña, acción que está autorizada. Sin embargo, poco después en la línea temporal, podemos ver que ha realizado otra llamada a la API para cambiar la contraseña del administrador. Como, obviamente, se trata de un acto no autorizado por la relación existente entre el actor y el recurso, se genera una alerta.

Por dónde empezar en su iniciativa de seguridad de API

Mantener la seguridad de las API es una tarea permanente para la mayoría de las organizaciones, lo que hace que pueda resultar difícil el saber por dónde empezar. Aunque los tres pilares fundamentales anteriores proporcionan un punto de partida útil, la eficacia de su enfoque mejorará considerablemente si sigue estas tres recomendaciones con su implementación:

1. Garantizar que el inventario de API esté siempre actualizado.
2. Supervisar los entornos de API de producción y no de producción.
3. Aplicar las relaciones entre las entidades.

No se pueden proteger las API que no se sabe que existen. Por lo tanto, una protección eficaz de las API comienza con un inventario de API actualizado y una evaluación de la estrategia de seguridad. Del mismo modo, a medida que aplique sus funciones de supervisión de seguridad de API, es importante ampliarlas a implementaciones de API de producción y no de producción. Y, lo que es más importante, la forma en que supervise y aplique las API debe ir más allá de las acciones en sí y tener en cuenta las relaciones existentes entre las entidades implicadas en la actividad de API. Esto le permitirá detectar vulnerabilidades y brechas en materia de protección, así como garantizar la conformidad con los modelos de uso de API previstos. Conocer el comportamiento en sus API le permitirá detectar cualquier tipo de abuso.

¿Desea obtener más información sobre los ataques a las API y cómo puede protegerse contra ellos? Consulte la información desglosada de las 10 principales vulnerabilidades de las API según OWASP.

Akamai

Akamai protege la experiencia de sus clientes, su personal, sus sistemas y sus datos, ayudándole a integrar la seguridad en todo lo que crea, dondequiera que lo cree o distribuya. La visibilidad de las amenazas globales que ofrece nuestra plataforma nos permite adaptar y desarrollar su estrategia de seguridad para integrar el enfoque Zero Trust, detener el ransomware, proteger las aplicaciones y las API o combatir los ataques DDoS, y le proporciona la confianza necesaria para innovar, crecer y transformar todo su entorno. Para obtener más información acerca de las soluciones de cloud computing, seguridad y distribución de contenido de Akamai, visite akamai.com y akamai.com/blog, o siga a Akamai Technologies en X, antes conocido como Twitter, y LinkedIn.