¿Qué es una clave de API?

Las claves de API se utilizan para identificar y autorizar la aplicación o servicio que realiza la llamada. Generalmente tienen una larga duración y son estáticas hasta que se rotan y a menudo tienen un conjunto fijo de permisos. Se utilizan principalmente para comunicaciones de servidor a servidor o para acceder a datos públicos, estos tokens generalmente no representan a un usuario específico.

¿Cómo funciona una clave de API?

Una clave de API es una larga cadena de caracteres generada por el proveedor de la API y compartida con usuarios autorizados. Esta clave debe incluirse en el encabezado de la solicitud al acceder a la API. Las claves de API son simples y efectivas para necesidades básicas de seguridad. Por ejemplo, servicios populares como Google Maps API y AWS proporcionan claves de API para controlar el acceso y monitorear el uso.

```
curl -GET https://api.example.com/endpoint -H "Authorization: api-key TU_API_KEY"
```

Las claves de API no son tan efectivas como otras formas de autenticación de API, como OAuth 2.0 y JSON Web Token (JWT) , pero aún juegan un papel importante al ayudar a los productores de API a monitorear el uso, ya que es el método más sencillo y ampliamente utilizado para asegurar las APIs.

¿Cuáles son sus pros y contras?

Pros

- Simple de implementar: Las claves de API son fáciles de implementar y usar. Implican adjuntar una clave al encabezado de la solicitud, lo que lo convierte en un método sencillo para que los desarrolladores y clientes lo entiendan y utilicen.
- Fácil de monitorear: Las claves de API son fáciles de monitorear. Puedes rastrear el uso de cada clave y revocarlas si es necesario.
- Limitación de tasa efectiva: Las claves de API son efectivas para la limitación de tasa. Puedes establecer un límite en el número de solicitudes por clave para prevenir abusos.
- Adecuado para datos no sensibles: Las claves de API son adecuadas para datos no sensibles o APIs disponibles públicamente, donde los requisitos de seguridad son

menores.

Contras

- Seguridad limitada: Las claves de API no son lo suficientemente seguras para datos sensibles, especialmente para aplicaciones del lado del cliente. A menudo se utilizan en comunicaciones de máquina a máquina.
- No adecuado para la autenticación de usuarios: Las claves de API están vinculadas a aplicaciones o sistemas, no a usuarios individuales, lo que dificulta identificar usuarios específicos o rastrear sus acciones.
- Sin expiración de token: Las claves de API son típicamente estáticas y no expiran. Si una clave se compromete, podría ser mal utilizada indefinidamente a menos que se regenere manualmente.

¿Cuáles son los casos de uso para las claves de API?

- Comunicación de servicio a servicio: Las claves de API son adecuadas para escenarios donde las aplicaciones necesitan comunicarse con APIs directamente a través de CLIs. Por ejemplo, llamar a las APIs de OpenAI.
- APIs públicas: Al exponer APIs al público, las claves de API proporcionan un método sencillo de control de acceso.
- Configuración simplificada: Para necesidades de autenticación rápidas y simples, especialmente en la fase de desarrollo. A diferencia de la autenticación de máquina a máquina, las claves de API no requieren registro de cliente previo, y tampoco necesitan intercambiarse por un access token. Simplemente pasas tu clave de API como un parámetro en tu solicitud y simplemente funciona.

¿Cuál es la diferencia entre los Tokens de Acceso Personal (PAT) y la comunicación de Máquina a Máquina (M2M)?

Al hablar de claves de API, los tokens de acceso personal y Máquina a máquina (Machine-to-machine) también pueden mencionarse juntos ya que todos pueden acceder programáticamente a recursos de API a través de comandos CLI, o establecer comunicación entre servicios de backend.

Tokens de Acceso Personal (PATs)

Un token de acceso personal también es una cadena pero representa ***la identidad y permisos de un usuario específico***, se genera dinámicamente tras una autenticación o inicio de sesión exitoso, y típicamente tiene una duración limitada pero puede ser renovado. Proporciona un control de acceso detallado a datos y capacidades específicas del usuario y se utilizan comúnmente para herramientas CLI, scripts o acceso personal a API. La principal diferencia es que es más específico y se utiliza para acciones específicas del usuario.

Máquina a Máquina (M2M)

La comunicación M2M es cuando los dispositivos intercambian datos automáticamente sin intervención humana en un sentido más amplio.

En el contexto de OpenID Connect (OIDC) (u OAuth 2.0), las aplicaciones M2M usan el Flujo de credenciales del cliente (Client credentials flow) , como se define en el protocolo OAuth 2.0 RFC 6749 , que soporta protocolos estándar similares. Generalmente involucra a una aplicación cliente (una máquina o servicio) accediendo a recursos ya sea por sí misma o en nombre de un usuario. Es ideal para situaciones donde solo los clientes de confianza pueden acceder a servicios de backend.

Revisión #2
Creado 2024-11-13 13:45:33 UTC por Marluan Espiritusanto
Actualizado 2024-11-13 14:12:47 UTC por Marluan Espiritusanto