AGO
19

SPRINT 3 - MEETING 44

Seguridad

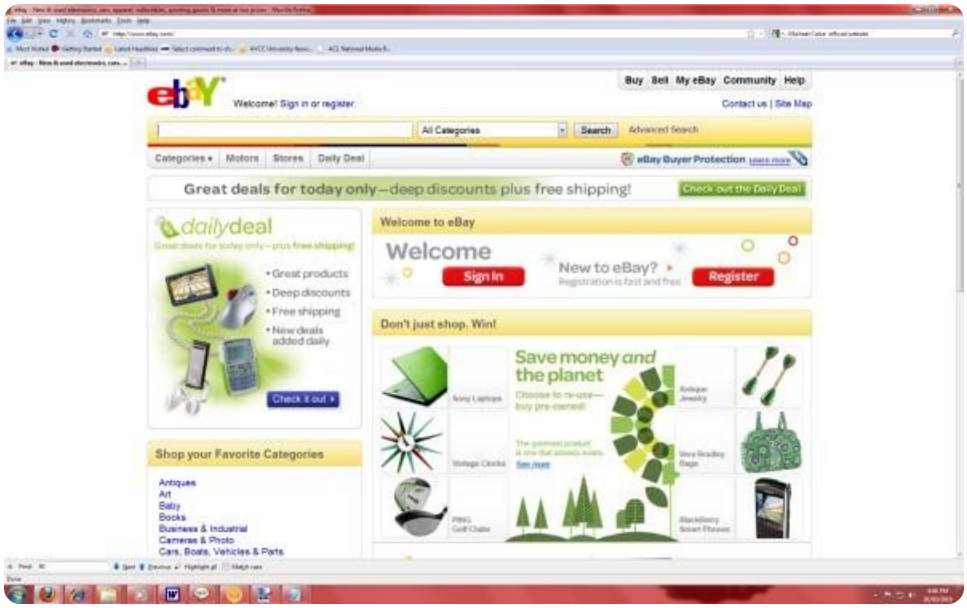
Grabación

Presentación

La prioridad es la protección de datos

> *"Afortunadamente hay muchas cosas que pueden hacerse para disminuir la cantidad de metadata disponible al público. La más sencilla es revisar la configuración de tu teléfono para asegurar mayores niveles de seguridad, y elegir cuáles aplicaciones te gustarían utilizar o no relacionadas a la data de ubicación." — Graham Cluley**. Escritor sobre seguridad informática.***

Hasta los gigantes caen



En el año 2000, Michael Calce, un estudiante de secundaria canadiense de 14 años decidió desatar un ataque de Denegación de Servicios en varios sitios web comerciales de alto perfil, incluidos Amazon, CNN, eBay y Yahoo. Un experto de la industria estimó que los ataques resultaron un daño

Photo by Erik Mclean on Unsplash

Según el artículo de Ericka Chickowski publicado en *Bitdefender*, la seguridad de las APIs es una de las mayores preocupaciones para la ciberseguridad. Esto se debe a que, según el informe de Akamai del año 2019, un 83% del tráfico web proviene de las APIs. Este número toma relevancia cuando tenemos en cuenta que las APIs están conectadas con diferentes sistemas que involucran información personal. Está claro que al momento de desarrollar una aplicación es fundamental no solo tener en cuenta sino protegerla.

Contar con seguridad en las APIs garantiza que las aplicaciones -que desarrollarás y utilizarás- funcionen adecuadamente, como así también la protección de los datos de los/as usuarios/as.

David Ruiz en su artículo menciona que *"si una API tiene problemas, está desprotegida o hackeada, lo más probable es que se infrinjan sus datos. El resultado es la exposición pública de datos financieros, médicos o personales. Una verdadera invasión de la privacidad."*

Por ejemplo, Facebook en el 2019 sufrió una filtración de 267 millones de usuarios/as que fueron expuestos/as en internet. Es posible que los hayan obtenido de la API de Facebook antes de que la compañía restrinja el acceso a los números de teléfono en el 2018.

La seguridad es primordial para evitar correr riesgos de vulneración de información a futuro y que el daño sea mayor.

En el encuentro anterior aprendimos a crear nuestras APIs con Express JS. En esta bitácora, encontrarás qué conceptos necesitas comprender sobre la seguridad en las APIs.

Conceptos clave sobre seguridad

Para comenzar, diferenciaremos dos conceptos: el de autenticación con el de autorización:

Fuente: (Parte 3) Segurança em APIs RESTful de Thiago Lima

1. **Autenticación.** Es identificar al/a usuario/a para después darle los permisos que requiera autorización.

2. **Autorización.** Es el permiso para acceder al recurso que estamos buscando. Por ejemplo, al ingresar a una red de wifi, te pide contraseña y una vez ingresada nos autoriza.

Como te imaginarás, ¡ambos conceptos están relacionados! La autenticación se puede pensar como un conjunto de credenciales para que el servidor pueda verificar tu identidad y, a la vez que esa información es válida. La autorización se basa del proceso de autenticación dado que este garantiza, una vez que valida las credenciales, el acceso para determinar a qué recursos se puede ingresar en el sistema. Ahora que conocemos la diferencia entre autenticación y autorización,

Diversidad de formas para proteger los datos

Los/as usuarios/as se enfrentan a la necesidad de tener que recordar cada vez más contraseñas por la cantidad de aplicaciones que usan. ¡En la mayoría de los casos esto puede resultar un problema! Para simplificar, los sitios o aplicaciones suelen contar con **sistemas de autenticación y autorización** que guardan los datos sobre la identidad de sus usuarios/as. Para ello existen distintas formas para asegurar una API, entre ellas:

- http authentication 1999. La cabecera de petición Authorization contiene las credenciales para autenticar a un usuario en un servidor, usualmente luego de que el servidor haya respondido con un estado 401 Unauthorized y la cabecera WWW-Authenticate. Es uno de los primeros métodos que se utilizaba y puede haber quedado desactualizado.
- Cookies. Son archivos pequeños que permiten guardar y transportar datos según nuestras preferencias a través del navegador, y los envía al servidor. En un inicio, cuando se crearon, se usaban para guardar únicamente información puntual con el objetivo de mejorar la experiencia del usuario. Luego pasaron a ser sesiones y guardar el estado en el servidor.
- Claves API. Es simple y rápido de implementar. El servidor genera una clave de acceso única para el cliente. El cliente es quien envía esta clave única en cada solicitud. Fue el método más utilizado por los desarrolladores ya que resolvía algunos problemas del modelo de autenticación básica.
- JSON Web Tokens (JWT). Actualmente, es el estándar abierto de seguridad más utilizado para autenticar usuarios. En este caso, con la primera petición de autentificación, el servidor generará un token basado en esas credenciales.

Es recomendable encriptar los datos sensibles de los/as usuarios/as, por ejemplo las contraseñas nunca se deben guardar en texto plano. El cifrado) es uno de los métodos más seguros que se utiliza para proteger o transmitir archivos de forma confidencial. Para decodificar un texto encriptado es necesario que el destinatario conozca la regla por la cual se ha cifrado el texto.

Helmet al rescate

Helmet es una librería que nos previene de los ataques más comunes. Su instalación es sencilla (como con todo con NPM :-)) y su implementación —a través de un middleware global— más aún.

```
#instalación por consola
npm install helmet --save
```

```
#implementación
const express = require('express')
const helmet = require('helmet')
const app = express()
app.use(helmet())
```

¿De qué nos previene específicamente?

- dnsPrefetchControl. Deshabilita el header X-DNS-Prefetch-Control de los navegadores.
- clickjacking. Es una técnica para engañar a usuarios con el objeto de que revelen información confidencial o tomar control de su ordenador cuando hacen clic en páginas web aparentemente inocentes.
- hidePoweredBy. Deshabilita el header X-Powered-By de los navegadores.
- hsts. ▢(https://helmetjs.github.io/docs/hsts/)Setea el header Strict-Transport-Security en todos los requests.
- ieNoOpen. Setea el header X-Download-Options para prevenir la ejecución de descargas en Internet Explorer.
- xssFilter. Agrega una protección para XSS.

Algunos de los conceptos pueden sonar extraños o difíciles de comprender, ¡no te preocupes! El desarrollo tiene tantas ramificaciones que un tema te va llevando al otro. Lo más importante en este momento es que comprendas que la seguridad es vital a la hora de realizar una aplicación, y cuáles son algunas de las formas de implementarla. En el próximo encuentro veremos cuales son los ataques más comunes y cómo prevenirlos.

Método de fuerza bruta

El método de fuerza bruta es uno de los más antiguos, pero aún sigue estando vigente. Consiste en probar millones de contraseñas -una tras otra- mediante una aplicación hasta dar con la contraseña correcta.

La NCSC (National Cyber Secure Centre) publicó un artículo con las 100.000 contraseñas más comunes. Aquí está la lista del top 10:

- 123456
- 123456789
- qwerty
- password
- 111111
- 12345678
- abc123
- 1234567
- password1
- 12345

Aquí encontrarás el listado de las 100.000 contraseñas más comunes. Puede que encuentres una

Cierre

Llegamos a un momento clave de las APIs ¡su seguridad! La protección de la información de los/as usuarios/as es una de nuestras responsabilidades y tenemos que llevarla a cabo de la mejor manera. Conocimos cuáles son los métodos de autenticación que podemos elegir, para una correcta seguridad de las APIs deberás analizar siempre cual es mejor de acuerdo a tus necesidades al momento de desarrollar una aplicación.

¡Prepárate para el próximo encuentro!

Profundiza

Te invitamos a conocer más sobre el tema de esta bitácora.

Herramientas

Programas necesarios para facilitar tu experiencia.

Comunidad

Sumérgete en la red donde los profesionales de DWFS interactúan.

Challenge

Te proponemos el siguiente desafío, ¿te animas?