Open Opened 5 days ago by **Rubén Montero**

Hacer login... ¡con cookies HttpOnly!

Resumen

- Entenderemos que el *login* de la tarea anterior sigue siendo inseguro
- Explicaremos qué es HttpOnly en una cookie
- Implementaremos una nueva versión de *login*

Descripción

En nuestro `BetterUserLogin.js` de la tarea anterior, permitimos que el usuario se *loguee* y el servidor enviará una *cookie* de sesión:

```
Set-Cookie: SESSION_TOKEN=7392b38de92a93bc973b0affe
```

Esto, por desgracia... ¡sigue siendo algo inseguro!

De la misma forma que `sessionStorage` es vulnerable, un atacante que mediante XSS haya conseguido ejecutar JavaScript malicioso en nuestro navegador podrá robar los datos que están almacenados en `document.cookie` .

HttpOnly

`HttpOnly` es un atributo que el servidor puede mandar junto con la *cookie*, así:

```
Set-Cookie: SESSION_TOKEN=7392b38de92a93bc973b0affe; HttpOnly
```

Si lo hace, dicha *cookie* no será *accesible* desde código JavaScript. Es decir, aunque un atacante malicioso (o nosotros mismos) indague en `document.cookie` ... ¡No verá nada!

La *cookie* sigue siendo almacenada por nuestro navegador, pero de forma *todavía más segura*.

La tarea

Crea una nueva pantalla `src/screens/user/best_login/BestUserLogin.js` .

`BestUserLogin.js` **será idéntico** a `BetterUserLogin.js` (cópialo y pégalo).

Atacar el *login* `/v4/` en vez de `/v3/`

Modifica tu función `onSubmit` en `BestUserLogin.js` **así**:

```
-   axios.post('http://raspi:8082/api/v3/sessions', { username: username, password: password }).then(response => {
+   axios.post('http://raspi:8082/api/v4/sessions', { username: username, password: password }).then(response => {
      console.log("Las cookies actuales son las siguientes:");
      console.log(document.cookie);
    });
```

Conectándolo en `App.js`

Añade en `App.js` :

```
      <Route path='/best_login' element={<BestUserLogin/>}/>
```

Ahora ya estará visible la página http://localhost:3000/best_login

¿Qué tal se ve? ¿Es igual de bonito que el de la tarea anterior?

Bien.

Una prueba

Borra tus datos de navegación (CTRL+SHIFT+DELETE). Todas tus *cookies*, historial,... deben haber desaparecido.

Ahora, en http://localhost:3000/best_login **haz** Click derecho > Inspeccionar y **verifica** lo que aparece por consola.

Si tecleas tu usuario y contraseña y te *logueas*... ¿Qué aparece reflejado en la consola?

¡No hay cookie! ¡El servidor no la está mandando!

Sí la está mandando, pero no es accesible desde `document.cookie`. Por eso es más segura.

Si viajas a tu página http://localhost:3000 y haces click en Inspeccionar > Red... ¿Cómo son las peticiones que tu navegador manda a http://raspi:8082/api/v2/dashboards? ¿Verificas que se está mandando la *cookie* de sesión en las cabeceras, igual que en la tarea anterior?

🏆 Por último

Sube tus cambios al repositorio en un nuevo *commit*.

Rubén Montero @ruben.montero changed milestone to %Sprint 5 5 days ago