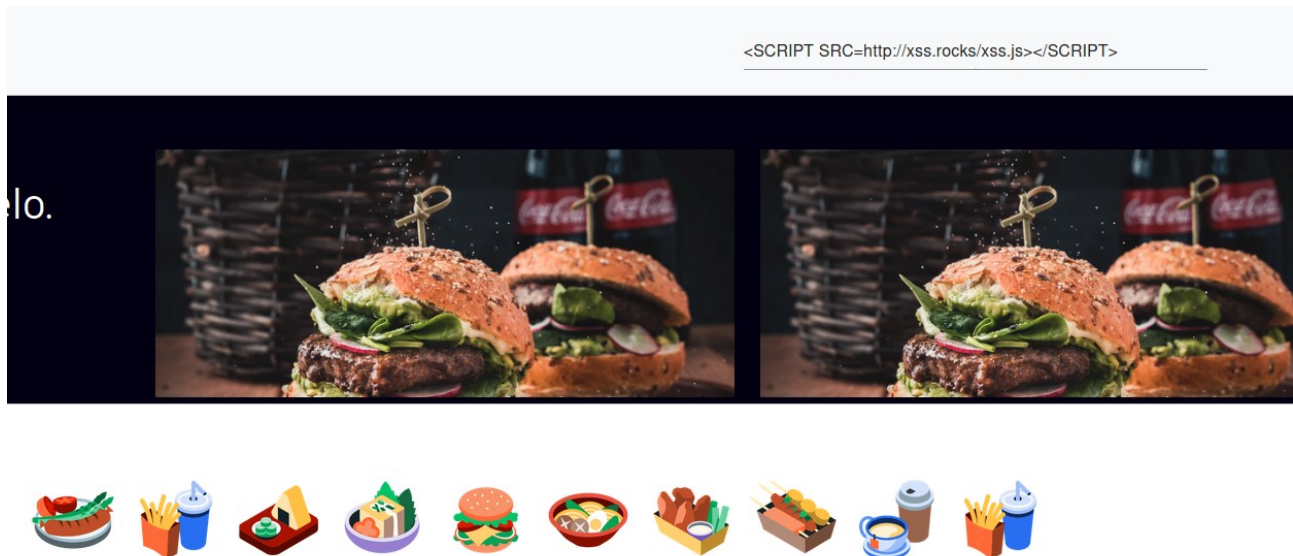


Security tests for UBEats

XSS

SearchBar

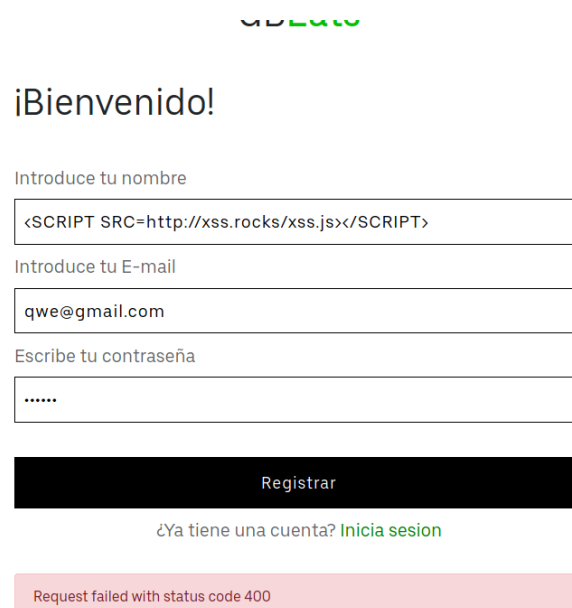
The injection did not work.



New user

I tried to insert a javascript snippet on the user name so when the user account was displayed the injection would trigger the alert.

It did not work because the backend checks for XSS on its input, therefore the request was denied with a 400.



New restaurant

The same idea but on the restaurant registration page. Again the request was denied because it contained invalid characters, used on the XSS snippet.

¡Añada su restaurante!

Introduzca el nombre de su restaurante

Introduzca su E-mail

Escriba su contraseña

Escriba la calle

Introduzca su telefono

Introduzca su IBAN

Registrar

Request failed with status code 400

New deliveryman

The same test on the deliveryman registration portal. The backend blocks the request.

Deliveryman Name

E-mail


Password

Registrar

Request failed with status code 400

Update user

Change user attributes once the registration is done

| | |
|---|--------------------------------|
|  | Name: qwerty |
| | Phone: |
| Invitation code | k4tb6g |
| Address | <IMG SRC=/ onerror="alert(Stri |
| Email | qwe@gmail.com |
| <button>Save changes</button> | |
| <button>Log out</button> | |

The attribute with the code snippet is saved on DDBB but it is not executed.

SQL Injection

New user

SQL injection on the user registration portal. The request it is not accepted because the checks on the backend blocked it.

¡Bienvenido!

Introduce tu nombre

qwe; DROP DATABASE ubereats;

Introduce tu E-mail

qwe@gmail.com

Escribe tu contraseña

.....

Registrar

¿Ya tiene una cuenta? [Inicia sesión](#)

Request failed with status code 400

New user 2

This time the script is hidden inside the password. The test had the same effect. No execution and no acceptance of the request by the backend.

UBEREATS

¡Bienvenido!

Introduce tu nombre

Introduce tu E-mail

Escribe tu contraseña

Registrar

¿Ya tiene una cuenta? [Inicia sesion](#)

Request failed with status code 400

```
ubereats=# \d
```

| Schena | Name | Type | Owner |
|--------|------------------------------|----------|-------|
| public | categories | table | qwe |
| public | categories_cat_id_seq | sequence | qwe |
| public | customers | table | qwe |
| public | deliverymans | table | qwe |
| public | extra_items | table | qwe |
| public | extra_items_extraitem_id_seq | sequence | qwe |
| public | favourites | table | qwe |
| public | feedbacks | table | qwe |
| public | items | table | qwe |
| public | items_item_id_seq | sequence | qwe |
| public | order_extraltens | table | qwe |
| public | order_items | table | qwe |
| public | orders | table | qwe |
| public | orders_order_id_seq | sequence | qwe |
| public | reports | table | qwe |
| public | reports_rep_id_seq | sequence | qwe |
| public | restaurants | table | qwe |
| public | type_items | table | qwe |
| public | type_restaurants | table | qwe |
| public | types | table | qwe |
| public | types_type_id_seq | sequence | qwe |
| public | users | table | qwe |

(22 rows)

```
ubereats=# \q
```

Update user

SQL injection while updating the user to test the PUT endpoint. The code was saved on the DDBB but was not executed.

UBEREATS

Name: qwe

Phone:

Invitation code: k4tb6g

Address:

Email:

Save changes









Your changes have been saved

Log out

Static analysis

An static analysis was made using OWASP ZAP. The following warnings were discovered.

The Cross-domain misconfiguration is a product on the local execution of the app. It is not expected to occur on the Heroku deployment.

-   Alerts (6)
- ▶  Cross-Domain Misconfiguration (15)
 - ▶  X-Frame-Options Header Not Set (8)
 - ▶  Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) (15)
 - ▶  X-Content-Type-Options Header Missing (15)
 - ▶  Information Disclosure - Suspicious Comments (2)
 - ▶  Timestamp Disclosure - Unix (23)