Labo 1 – Jentse Vander Hulst

Opdracht : Account registeren op juiceshop met een reguliere username die geen email is.

Graphical user interface, text

Description automatically generatedTools : Burp suite  
  
Testen van het registeren met een gewone naam zonder email:

Dit blijkt niet te werken want er is effectief validatie op het form dat ons verplicht een soort van email in de username te zetten. We weten dus al dat er client validatie is.

Als we wel een geldig email invoeren dan kunnen we ons gaan registeren.

Graphical user interface, application

Description automatically generated

Als we op de registratie knop klikken en dit pakketje gaan intercepten met Burp suite zien we onze POST request naar de website. Hierin zien we ook de gegevens die we hebben ingevoerd in plain tekst vanonder.

Text

Description automatically generated

A screenshot of a computer

Description automatically generatedWe zouden nu de waarde van onze “email” kunnen aanpassen naar een naam die we willen en dit doorsturen naar de server en kijken of dit effectief registreert. Ik pas voor mijn voorbeeld nu de email aan naar JentseVanderHulst en forward dit pakketje naar de server om te kijken of het lukt.

Ik probeer vervolgens dus aan te melden met de username JentseVanderHulst om te testen of het effectief gelukt is de user te registeren met een “foute” naam.

Graphical user interface, application

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidenceEn dit is ook gelukt, onze username is JentseVanderHulst zonder iets van email in de naam, er is dus blijkbaar geen server side validatie op de usernames die binnenkomen. Enkel client side is er een validatie die checkt of de username op een email lijkt.

Labo 2 :

Code staat ook op github : https://github.com/JentseV/SoftwareSecurity.git

* 1. SOP during fetch

Build an API and host it locally (e.g. using nodeJS on localhost:5000). Take a screenshot of the code.

// simple api server that listens on localhost:5000 and replies with "Hello from api!"

const { json } = require('express');

const express = require('express');

const server = express();

const HOST = "localhost";

const PORT = "5000";

const reply = "Hello from api!";

server.get("",(request,response) => {

    response.status(200).json(reply);

})

server.listen(PORT,HOST,() => {

    console.log("Server started listening...");

})

Build a simple application (only HTML and Javascript) and host it on a different origin (e.g. localhost:5001)

// simple web application running on localhost:5001

const express = require("express");

const path = require('path');

const app = express();

const HOST  = "localhost";

const PORT = "5001";

app.use(express.static(\_\_dirname));

app.get("", (request,response) => {

    response.status(200).sendFile(path.join(\_\_dirname,'/index.html'));

})

app.listen(PORT,HOST,()=> {

    console.log("Running web app");

})

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Simple website</title>

</head>

<body>

    <h1 id="test" name="test">Fetch api test</h1>

    <button style="height: 100px; width: 100px;" onclick="fetchApi()">Api call</button>

    <script>

        function fetchApi(){

            fetch('http://localhost:5000').then(res => res.json()).then(data => console.log(data)).catch(error => console.error(error));

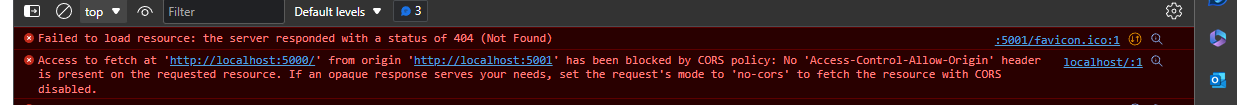
        }

    </script>

</body>

</html>

Have the simple application call the API. Does this work? Take a screenshot of the error.



Now fix the problem by setting the most restrictive CORS policy. Take a screenshot of the fix.

We voegen extra header informatie toe waardoor we de applicatie toelaten onze api te lezen met deze code.

server.use((request,response,next) => {

    response.header("Access-Control-Allow-Origin", "http://localhost:5001"); // allow our web app to access our api

    next();

})

Zo wordt onze API code als volgend :

// simple api server that listens on localhost:5000 and replies with "Hello from api!"

const { json } = require('express');

const express = require('express');

const server = express();

const HOST = "localhost";

const PORT = "5000";

const reply = "Hello from api!";

server.use((request,response,next) => {

    response.header("Access-Control-Allow-Origin", "http://localhost:5001"); // allow our web app to access our api

    next();

})

server.get("",(request,response) => {

    response.status(200).json(reply);

})

server.listen(PORT,HOST,() => {

    console.log("Server started listening...");

})

Text

Description automatically generatedEn dan werkt het wel

* 1. SOP in the DOM:

Build an application that opens a new tab in the browser (window.open). Make sure to allow pop-ups in your browser.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>1.2</title>

</head>

<body>

    <h1> Test window.open </h1>

    <button onclick="openNewWindow()">Open new window</button>

    <button onclick="readNewWindow()">get html from other window</button>

    <button onclick="readCrossWindow()"> Get html from other window that isnt in our SOP</button>

    <script>

        let windowx;

        let windowL;

        function openNewWindow(){

            windowx = window.open("http://localhost:5000/newWindow.html");

        }

        function readNewWindow(){

            let p = document.createElement("p");

            console.log("Working cause same source");

            console.log(windowx.document);

            p.textContent = windowx.document.getElementById("ppp").textContent;

            document.body.appendChild(p);

        }

        function readCrossWindow(){

            let ps = document.createElement("p");

            windowL = window.open("https://juice-shop.herokuapp.com/#/");

            console.log("Doesnt work cause different source");

            console.log(windowL.document);

            ps.textContent = windowL.document.getElementsByClassName("ng-star-inserted")[0].textContent;

            document.body.appendChild(ps);

        }

    </script>

</body>

</html>

Ensure the source document (the one calling ‘window.open’) and the target document (the one called by window.open) are the same origin. Try to access the html content of the target document from the xource document. Does it work? Take a screenshot.

Text

Description automatically generated

Now alter the origin of the target document. What happens? Take a screenshot.

A screenshot of a computer

Description automatically generated