TrackMe - Cybersecurity project 2019

Vision

TrackMe is a simple and **secure web application** for position **tracking** from a social point of view; it tracks your position and shows the positions of your friends.

Components/Architecture

TrackMe is implemented in nodeJs and based on a three tier architecture (front-end, back-end and a database).

Front End

- The client implemented is **responsive** (follows bootstrap best practices)
- interacts with the server through a RESTFul API

Back End

- The server is the main core of the application, takes inspiration from this article in order to avoid the most commons security mistakes in node programming
  - Routing is supported by express middlewares (i.e. `authenticationMiddleware` for session validation)
  - SQL injection is avoided using a no-SQL database (mongoDB) instance hosted on mLab servers and an orm
  - Server API/static files are availble only using a HTTPS calls (for this mockup we create a self-signed SSL certificate with openSSL
  - Server is sand-boxed into a Docker container which allows on the one hand to protect the server process from the outside hostile environment and, on the other hand to scale (largely used for distributed systems) and be cross-platform

Main features

- Log-in/sign-in sends credentials using a basic auth Base64 encoding to an HTTPS end point
- Sign-in uses a third party mailing service (mailgun) in order to send a second factor authentication code
- Passwords are stored using a SHA-256 hash function (provided by crypto libarary)
- Sessions are stored in the local storage of the browser and dynamically generated (session lifetime can be increased using a refresh token)
- We are currently using google maps API in order to show the current position of the user

Get Started

1. Create ssl certificate and private key using openssl

```
cd TrackMe.BackEnd
mkdir credentials
cd credentials

openssl req -newkey rsa:2048 -new -nodes -x509 -days 3650 -keyout key.pem -out cert.pem
```

2. Create a file with your API KEYs for the third party services used (Google API KEY, MailGun API KEY, MailGun Domain, DB connection string)

`projectCode/credentials/environment.js` :

```
var maps = "YOUR_GOOGLE_MAPS_API_KEY";
var mail = "edoardolenzi9@gmail.com";
var db = "mongodb://db_user:db_password@host:port/db_name";
var mailgun_key = "78fdXXXXXXXXXXXXXXXXXXXXXXXXXXXX7b3f";
var mailgun_domain = "sandboxXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.mailgun.org";


module.exports = {
    maps,
    db,
    mail,
    mailgun_key,
    mailgun_domain
}
```

3. Download node dependencies and start server:

```
cd projectCode
npm install
node index.js
```

Disclaimer

> The material contained in this repository is restricted to students/professors of the Cybersecurity course of the Master degree in Multimedia Communication and Information Technologies and of the Master degree in Computer Science at University of Udine.
>
> > It prohibited any use other than that inherent to the course, and in particular is expressly prohibited its use for any commercial purposes and/or for profit.

- License: GPL-3