Verslag opdracht 3 - IAM

Robbe Willeme

# Introductie

Dit verslag beschrijft de opzet van een IAM-oplossing voor een softwareproject bestaande uit een API als backend en een SPA als frontend. Initieel was het de bedoeling om een bestaand project te gebruiken bestaande uit een Angular applicatie en een API geschreven in ASP.net Core, maar omwille van redelijk wat verouderde versies gaf dit problemen. Om hier verder geen tijd aan te verliezen heb ik een dummy spa (Angular) en api (.NET) opgezet, het geheel moet een takeaway applicatie voorstellen.

Als OpenID Connect provider maak ik gebruik van Auth0.

# Threat analyse

Afbeelding met schermopname, tekst, Lettertype, nummer

Automatisch gegenereerde beschrijving

Bovenstaand schema geeft de algemene architectuur weer van de applicatie met hieronder een opsomming van de mogelijke threats.

* T01: Deze threat bevindt zich tussen de frontend en auth0. De gebruiker logt in via de login van auth en krijgt acces en id token terug. Het is belangrijk dat deze tokens te vertrouwen en onderweg niet aangepast zijn, zo niet kunnen onbevoegden mogelijk toegang krijgen tot de api resources.
* T02: Deze threat bevindt zich tussen frontend en de api. De acces token dient op api gevalideerd te worden op geldigheid en signatures.
* T03: Bij de authenticatie krijgt de gebruiker een login scherm te zien van auth0, hier kan mogelijk een man in the middle attack plaatsvinden waarbij een fake login pagina van auth0 is opgezet.
* T04: De user claims die door auth0 teruggeven worden kunnen mogelijk incorrect zijn.

# Auth0

Auth0 wordt gebruikt als OpenID Connect provider in mijn applicatie. In het dashboard heb ik een applicatie aangemaakt genaamd ‘cybersec’, deze applicatie zal gebruikt worden voor de authenticatie op de client. het enige wat daarin geconfigureerd diende te worden is de callback uri:

Afbeelding met tekst, schermopname, Lettertype, algebra

Automatisch gegenereerde beschrijving

Daarnaast heb ik ook een API aangemaakt in auth0 om later de autorisatie te kunnen implementeren:

Afbeelding met tekst, schermopname, Lettertype, nummer

Automatisch gegenereerde beschrijving

# Werking van de applicatie

Als frontend heb een dummy Angular applicatie opgezet waar mensen eten kunnen bestellen via takeaway.

Afbeelding met tekst, Lettertype, Graphics, logo

Automatisch gegenereerde beschrijvingDe applicatie is heel simpel en toont 2 verschillende schermen op basis van de loginstatus van de gebruiker. De homepagina ziet er als volgt uit:

Als de gebruiker op deze knop drukt wordt hij doorverwezen naar de inlogpagina van auth0 waar hij zich moet inloggen en toestemming moet geven:

Afbeelding met tekst, schermopname, Menselijk gezicht, Website

Automatisch gegenereerde beschrijvingAfbeelding met tekst, schermopname, Lettertype, software

Automatisch gegenereerde beschrijving

Na een succesvolle login wordt de gebruiker terug doorverwezen naar de applicatie, maar nu krijgt hij een ander scherm te zien aangezien de loginstatus verandert is:

Afbeelding met tekst, schermopname, Lettertype, ontwerp

Automatisch gegenereerde beschrijving

Hier heeft de gebruiker de optie om de menukaart weer te geven, dit zal gebeuren via een call naar API, of om uit te loggen. Als de gebruiker op “bekijk menukaart” klikt krijgt hij de verschillende opties te zien die hij kan bestellen:

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

# Authenticatie en autorisatie

De globale werking van de applicatie is in de vorige sectie al besproken, hier ga ik verder ingaan op de verschillende componenten en code die is toegevoegd om authenticatie en autorisatie te implementeren.

## OIDC-client-ts

Op de client heb ik gebruik gemaakt van de OIDC-client-ts library om auth0 te implementeren. Deze library heeft een object genaamd ‘UserManager’ die alle functies bevat om een gebruiker te authentiseren via de openid-connect flow.

In mijn Angular applicatie heb ik een auth service aangemaakt die alle nodige functies bevat. Deze service maakt bij aanroep een usermanager object aan, de configuratie ziet er als volgt uit:

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

* Authority: De link naar de auth0 server
* Client\_id: een unieke id die terug te vinden is in de applicatie die is aangemaakt in het auth0 dashboard.
* Redirect\_uri: De link naar de callback.
* Response\_type: het type antwoord
* Scope: Dit bepaalt welke info er over de gebruiker wordt teruggeven in de token.

## Flow

Wanneer de gebruikers op de login knop drukt zal volgende functie uitgevoerd worden:

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

De signinRedirect functie van UserManager zal de gebruiker doorverwijzen naar de login pagina van auth0, vervolgens zal de callback worden aangeroepen:

Afbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijving

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

De Callback component zal de completeLogin functie aanroepen en vervolgens teruggaan naar de home pagina. Hierin wordt de signInRedirectCallback functie van userManager gebruikt om de gebruiker verder te authentiseren.

Vervolgens is de gebruiker aangemeld en kan hij op de knop “bekijk menukaart” klikken, dit zal volgende functie aanroepen om de data te gaan ophalen van de API:

Afbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijving

Deze functie gaat eerst Accestoken ophalen van de gebruiker en vervolgens een request doen naar de API waar deze token wordt meegegeven.

## API

De API is geschreven in ASP.net core en bevat 1 route die de menukaart geeft. Nadat een gebruiker succesvol in ingelogd kan hij gebruik maken van de acces token om resources op te vragen van de API.

Hiervoor heb ik volgende zaken moeten configureren in API code:

Afbeelding met tekst, schermopname, software, Lettertype

Automatisch gegenereerde beschrijving

* Authority: De link naar de auth0 server.
* Audience: Deze verwijst naar de api die is aangemaakt via het auth0 dashboard.

Vervolgens moest er nog enkele middleware toegevoegd worden aan de pipeline:

Afbeelding met tekst, Lettertype, schermopname, lijn

Automatisch gegenereerde beschrijving

De UseAccesTokenValidation is een validation die ik zelf geschreven om te valideren op de aanwezigheid van een acces token bij het sturen van een request. Als de request niet door deze validatie geraakt zal niet verder worden gezet.

Als laatste moest er nog het “Authorize” attribuut worden toegevoegd aan de route:

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

## OPA

Helaas moet ik bekennen dat ik door tijdsgebrek en de problemen die ik ben tegengekomen bij het implementeren van auh0 niet meer voldoende tijd had om dit deel volledig te implementeren.

# Opstarten van de applicatie

Ik heb gebruik gemaakt van Docker en Docker compose om te applicatie te starten, de docker-compose file bevindt zich in de root van het project en hier mee kan de applicatie gestart worden.

Vervolgens is de frontend beschikbaar op: <http://localhost:4200>.

# Conclusie