Node.js Validator

Overview

MobileFirst Foundation provides a Node.js framework to enforce security capabilities on external resources.
The Node.js framework is provided as an npm module (**passport-mfp-token-validation**).

This tutorial shows how to protect a simple Node.js resource, `GetBalance`, by using a scope (`accessRestricted`).

Prerequsites:

- Read the Using the MobileFirst Server to authenticate external resources (../) tutorial.
- Understanding of the MobileFirst Foundation security framework (../../).

The passport-mfp-token-validation module

The passport-mfp-token-validation module provides an authentication mechanism to verify access tokens that are issued by the MobileFirst Server.

To install the module, run:

```
npm install passport-mfp-token-validation@8.0.X
```

Usage

- The sample uses the `express` and `passport-mfp-token-validation` modules:

  ```
  var express = require('express');
  var passport = require('passport-mfp-token-validation').Passport;
  var mfpStrategy = require('passport-mfp-token-validation').Strategy;
  ```

- Set up the `Strategy` as follows:

  ```
  passport.use(new mfpStrategy({
     authServerUrl: 'http://localhost:9080/mfp/api',
     confClientID: 'testclient',
     confClientPass: 'testclient',
     analytics: {
        onpremise: {
           url: 'http://localhost:9080/analytics-service/rest/v3',
           username: 'admin',
           password: 'admin'
        }
     }
  }));
  ```

  - `authServerUrl`: Replace `localhost:9080` with your MobileFirst Server IP address and port number.
  - `confClientID`, `confClientPass`: Replace the confidential client ID and password with the

ones that you defined in the MobileFirst Operations Console.
    - `analytics`: The analytics item is optional, and required only if you wish to log analytics events to MobileFirst Foundation.
      Replace `localhost:9080`, `username`, and `password` with your Analytics Server IP address, port number, user name, and password.
- Authenticate requests by calling `passport.authenticate`:

```
var app = express();
app.use(passport.initialize());

app.get('/getBalance', passport.authenticate('mobilefirst-strategy', {
      session: false,
      scope: 'accessRestricted'
   }),
   function(req, res) {
      res.send('17364.9');
   });

var server = app.listen(3000, function() {
   var port = server.address().port
   console.log("Sample app listening at http://localhost:%s", port)
});
```

    - The `Strategy` to employ should be `mobilefirst-strategy`.
    - Set `session` to `false`.
    - Specify the `scope` name.

Sample

Download the Node.js sample (https://github.com/MobileFirst-Platform-Developer-Center/NodeJSValidator/tree/release80).

Sample usage

1. Navigate to the sample's root folder and run the command: `npm install` followed by: `npm start`.
2. Make sure to update the confidential client (../#confidential-client) and secret values in the MobileFirst Operations Console.
3. Deploy either of the security checks: **UserLogin (../../user-authentication/security-check/)** or **PinCodeAttempts (../../credentials-validation/security-check/)**.
4. Register the matching application.
5. Map the `accessRestricted` scope to the security check.
6. Update the client application to make the `WLResourceRequest` to your servlet URL.