Lab 3.5: Per User Authentication Matching

Purpose
The purpose of this module is to enable a simple discovery service on the service provider based on the user's email address

Requirements
• Access to SSO configuration files • A federation already configured, in this case will be Azure AD • SmartPlan application

Overview

On IAM Academy 3 you have seen the pros and cons of federation discovery. Some of the challenges were:

- The Nascar problem: how to avoid a login page full of logos.
- How can we make discovery both user friendly and secure, without exposing too much information?
- What information do we leak to the discovery provider?

Among the most recent solutions for federation discovery, Ubisecure has developed a JS plugin called "per user authentication matching."

Per user authentication matching is a UI extension that delivers smoother user experience for end users. The JavaScript permits Administrators to configure groups of users or whole organisations towards a specific authentication method or methods. This limits the authentication options presented to specific users and makes sure that organisations that require higher level of assurance are given those options.

Typical use case

If a website offers multiple authentication methods to sign in, automatically selecting the preferred method based on the email address entered by the user streamlines the user experience and reduces training and support costs.

For example, the software could be configured so that:

- Users from email domain @example.IDP.com will be redirected to their standard SAML identity provider login
- Users from email domain @SMS.customer.com will be require an SMS code to login. Their login will move directly to a login page from their identity provider in which they need to fill in their SMS one-time password.
- Users from email domain @gmail.com will be redirected to OAuth 2.0-based Google login page.
- All other users will be prompted to log in with password (the most basic authentication method)

How does it work in practice?

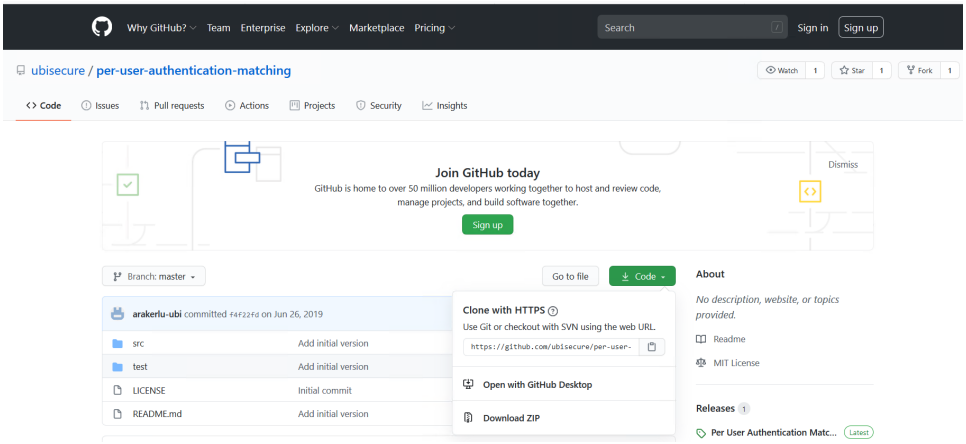
For this example use case, when you utilise Per user authentication matching in your IDS environment, the login screen will show only a field requesting the users email address as the initial login screen. Based on email address domain entered, the user is redirected to their appropriate authentication provider. The login password field will only be shown for users from organisations without an external authentication provider or two-factor authentication method activated.

Instructions

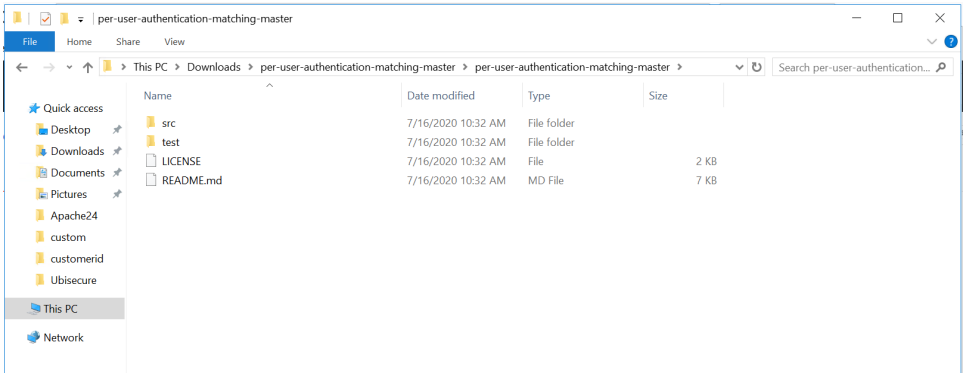
For this exercise, we will use Azure AD (which we configured in Lab 3.1) as the federated authentication method. Azure AD usernames are email addresses, so we'll use the email domains to match the users with their home identity provider.

Part 1: Installation

1. Download the per-user-authentication-matching package from Ubisecure Github repository. Open the page https://github.com/ubisecure/per-user-authentication-matching
2. Click the green button "Code", and then "Download ZIP." Save to any local folder such as Downloads.



3. Extract the contents of the ZIP file.



4. Go to C:\Program Files\Ubisecure\ubilogin-sso\ubilogin\custom\resources and create a new subdirectory "script"
5. Copy file C:\Users\Administrator\Downloads\per-user-authentication-matching-master\per-user-authentication-matching-master\src*per-user-authentication-matching.js* to the folder C:\Program Files\Ubisecure\ubilogin-sso\ubilogin\custom\resources\script on your Ubisecure SSO installation.
6. Define the scripts in the resource index (*"C:\Program Files\Ubisecure\ubilogin-sso\ubilogin\custom\resource.index"*) by adding the following line.

resource.index

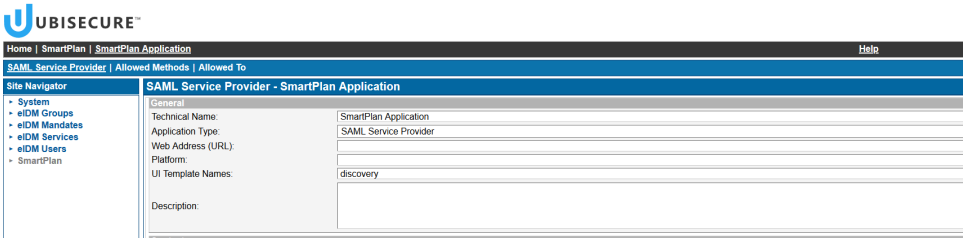
```
script/per-user-authentication-matching.js = resources/script/per-user-authentication-matching.js
```

7. Create a new template called discovery. Create a file discovery.properties on directory C:\Program Files\Ubisecure\ubilogin-sso\ubilogin\custom\templates
8. Add discovery to the template index by adding the line below.

template.index

```
discovery = templates/discovery.properties
```

9. Next, open SSO Management Console and set the new template "discovery" on SmartPlan Application. See below field "UI Template Names."



10. Finally include the script *per-user-authentication-matching.js* in the `javascript` property in the template properties as shown below.

discovery.properties

```
javascript = /resource/script/jquery.js, /resource/script/per-user-authentication-matching.js
```

11. Now the script is installed.

Part 2: Configuration

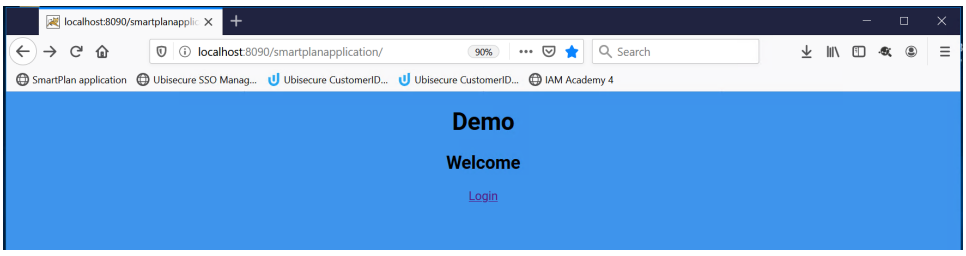
The main configuration occurs on the configuration object `perUserAuthenticationMatchingParams`:

1. Open the file C:\Program Files\Ubisecure\ubilogin-sso\ubilogin\custom\resources\script\per-user-authentication-matching.js and edit it as show below.

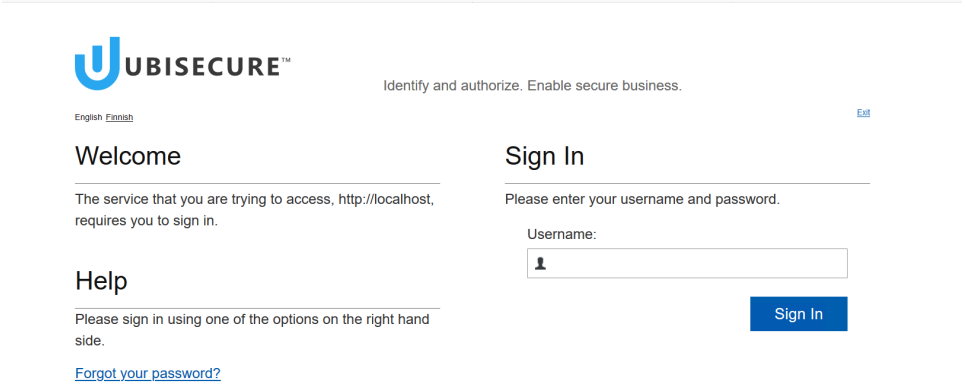
per-user-authentication-matching.js

```
const perUserAuthenticationMatchingParams = {
  enabled: true,
  autoRedirect: true,
  validByDefault: true,
  validationRules: {
    "azure.saml.yourname":  [ /@iamacademy.ubisecure.com$/g,
                        /@ubisecure.com$/g ],
    "password.2":  [ /@smartplan.com$/g ],
  },
  errorMessages: {
    "INVALID_USERNAME": {
      ""  : "Invalid username",
      "fi": "Virheellinen käyttäjätunnus",
      "sv": "Ogiltigt användarnamn"
    }
  }
}
```

2. Obs: Replace "azure.saml.yourname" with the exact method name you defined on Lab 3.1, Part 1. Then save the file.
3. Restart Ubilogin Server
4. Now open SmartPlan Application.



5. When you click "Login" button you will redirected to a login page that looks like this:



6. The login page doesn't show any authentication method, but it asks you to enter your username.
7. Try to login separately with the users:

    - scott.long@smartplan.com
    - jeremy.mills@iamacademy.ubisecure.com

    Verify that the authentication method is correctly matched based on user's email domain.

8. Finally, verify that you can log in successfully in each case.

***** END OF EXERCISES *****