OpenAM 14 持続 Cookie 認証モジュール
利用手順書

OSSTech 株式会社

更新日 2022 年 7 月 14 日
リビジョン 1.2

目次

1 はじめに

本文書は、OSSTech 版 OpenAM14 に含まれる持続 Cookie 認証モジュールの利用手順書です。

1.1 機能概要

持続 Cookie 認証モジュールの機能について説明します。

本モジュールは、認証成功時にユーザー情報を JWT として有効期限付きの Cookie に保存し、次回以降の認証時にその Cookie を利用して認証を行うモジュールです。

- 一定期間、認証を省略する
- 一定期間、多要素認証を省略する

の 2 通りの使い方があります。

1.1.1 一定期間、認証を省略する

通常 OpenAM が発行する Cookie はセッション Cookie であり、ブラウザを閉じると認証状態が破棄されます。 それに対し持続 Cookie 認証モジュールは、認証成功時に期限付きの Cookie を発行することでブラウザを閉じても認証状態を維持させることができ、Cookie が有効な間は認証を省略することができます。

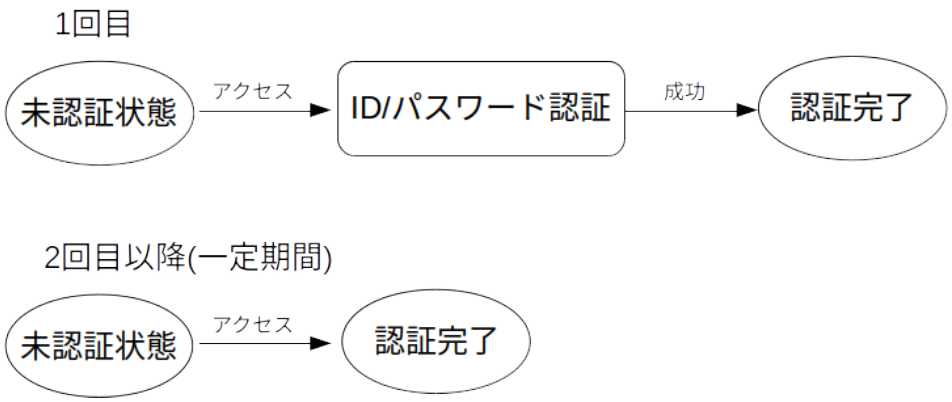

図 1　認証を省略した利用イメージ

1.1.2 一定期間、多要素認証を省略する

持続 Cookie 認証モジュールの後に多要素認証のモジュールを設定することによって、Cookie の有無で認証時に多要素認証を行うかどうかを分岐させることができます。

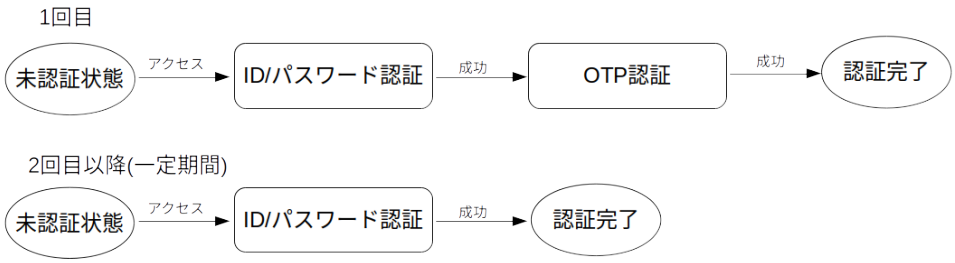

図 2　多要素認証を省略した利用イメージ

さらに持続 Cookie の発行確認認証モジュールを併せて利用することで、持続 Cookie を発行するかどうかをユーザーに選択させることができ、次回アクセス時の動作をユーザーが変更することもできます。持続 Cookie を発行する場合は次回アクセス時に多要素認証をすることなくログインでき、発行しない場合は多要素認証を求められます。

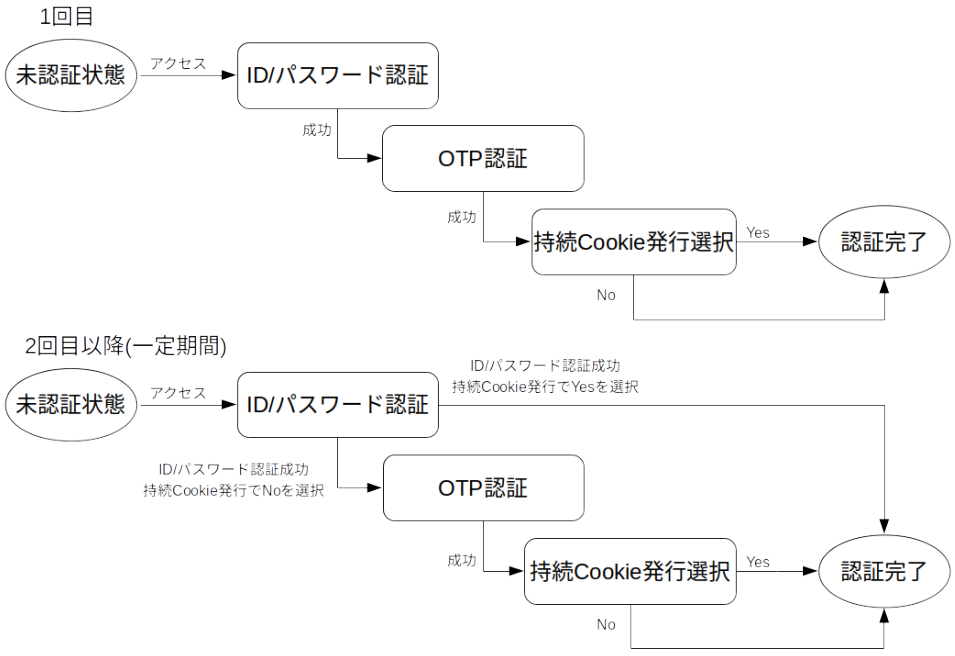

図 3　多要素認証を省略し Cookie 発行確認の利用イメージ

2 事前準備

持続 Cookie 認証モジュールを利用するには以下の事前準備が必要です。

- OpenAM の初期設定
- 持続 Cookie 認証モジュールと組み合わせて利用する認証モジュールの設定
- 組織認証用鍵ペアの作成
- 組織認証用の証明書エイリアスの変更
- 認証ポストプロセスクラスの設定

2.1 組織認証用鍵ペアの作成

持続 Cookie 認証モジュールは公開鍵暗号方式で暗号化した情報を持つ Cookie を作成します。そのため、公開鍵と秘密鍵の鍵ペアを作成する必要があります。下記の手順で作成します。ここでは OpenAM が利用するデフォルトのキーストアに追加することを前提としています。

```
$ keytool -genkeypair \
-keyalg rsa \
-alias top-realm \
-dname "CN=sso.example.co.jp,OU=development,O=EXAMPLE,L=Shinagawa,ST=Tokyo,C=JP" \
-keypass changeit \
-keystore /opt/osstech/var/lib/tomcat/data/openam/openam/keystore.jceks \
-storetype jceks \
-storepass changeit \
-validity 3650 \
-keysize 2048
```

キーストアのパスワード (-storepass) と秘密鍵のパスワード (-keypass) は、OpenAM のサーバー設定のキーストアの設定に合わせる必要があります。changeit は OpenAM が利用しているデフォルトのキーストアのパスワードです。

2.2 組織認証用の証明書エイリアスの変更

「組織認証用鍵ペアの作成」で作成した証明書を利用するように OpenAM の設定を変更します。

1. OpenAM 管理コンソール上部の「設定」タブ→「認証」→「コア属性」を開きます。
2. 「セキュリティ」タブの「組織認証の証明書のエイリアス」にエイリアス (`-alias`) の `top-realm` を設定します。

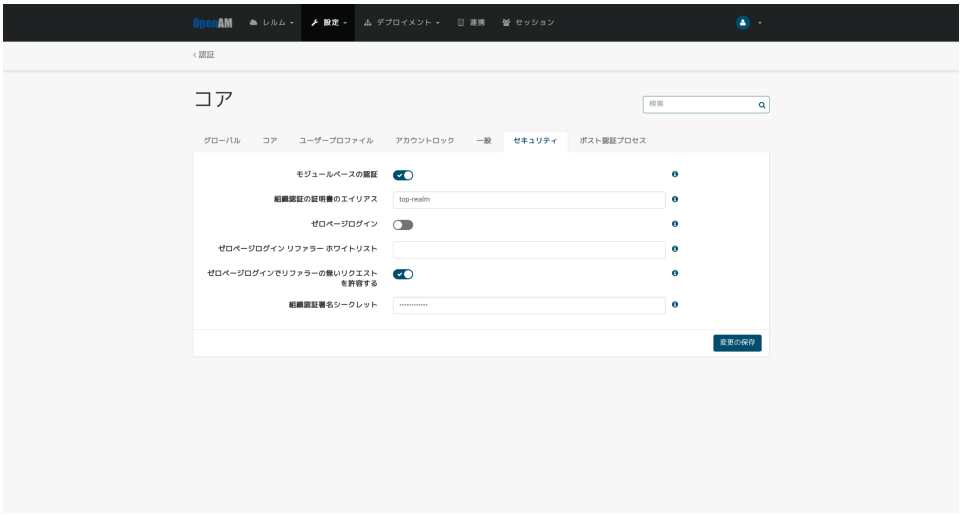

図 4　組織認証の証明書のエイリアスの変更

3. 「変更の保存」を押下します。

2.3 認証ポストプロセスクラスの設定

持続 Cookie 認証モジュールは、前回のログイン成功時に設定された Cookie 値を利用して認証します。認証ポストプロセスクラスを設定することによって、認証成功時に Cookie のセットが行われるようになります。

1. OpenAM 管理コンソールの対象のレルム→「認証」→「設定」を開きます。
2. 「ポスト認証プロセス」タブの「認証ポストプロセスクラス」に以下の文字列を追加します。

```
org.forgerock.openam.authentication.modules.persistentcookie.Persistent
CookieAuthModule
```

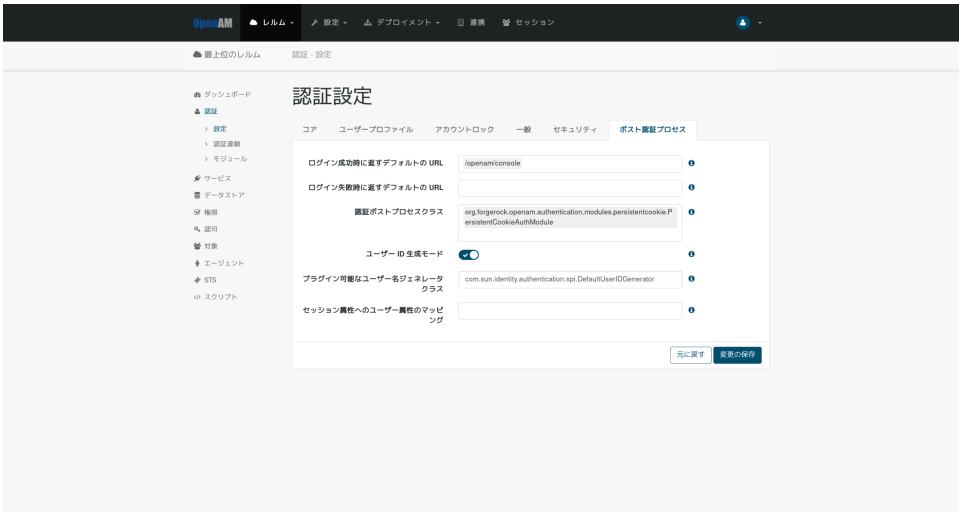

図 5　認証ポストプロセスクラスへの追加

3. 「変更の保存」を押下します。

3 認証モジュールの追加

3.1 持続 Cookie 認証モジュールの追加

ここでは、持続 Cookie 認証モジュールを利用するための設定方法を説明します。

1. OpenAM 管理コンソールにログイン後、対象のレルムを選択します。
2.「認証」→「モジュール」に移動し、「モジュールの追加」を押下します。
3.「名前」に認証モジュール名 (ここでは `persistentCookie`) を入力し、「種類」のドロップダウンリストから **持続** `Cookie` を選択します。

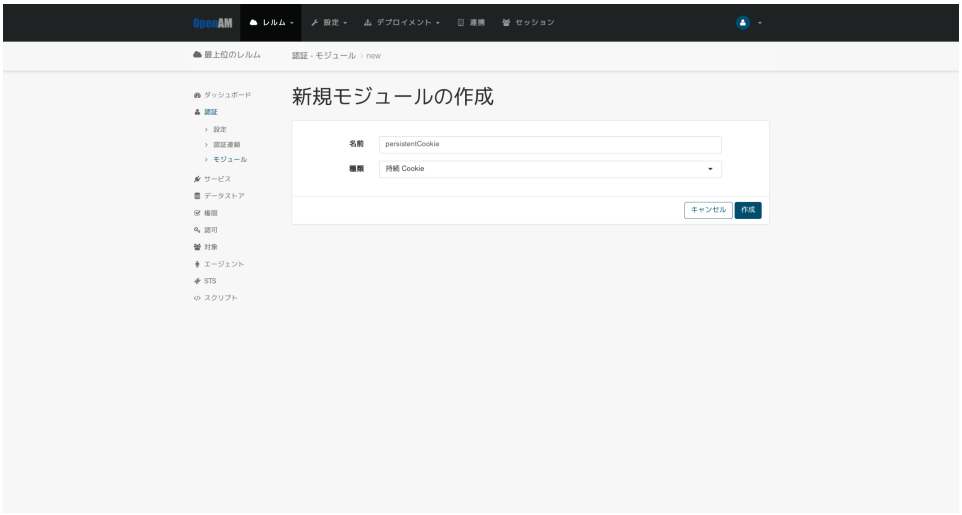

図 6　持続 Cookie 認証モジュールの作成

4.「作成」を押下すると、認証モジュールの設定画面に遷移します。

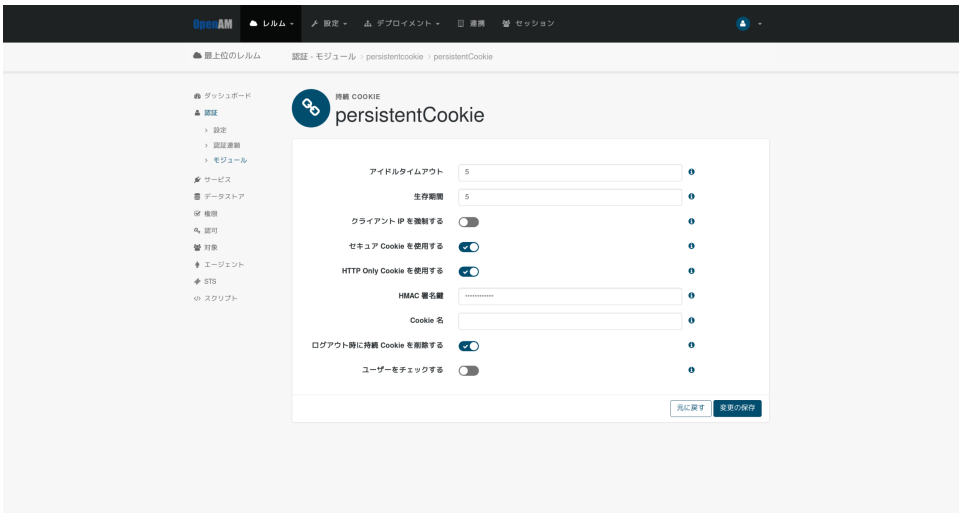

図 7　持続 Cookie 認証モジュールの設定

5. 各項目の設定を行い、「変更の保存」を押下します。
   各項目の詳細は下記を参照してください。

項目名	設定内容
アイドルタイムアウト	Cookie が無効になるリクエスト間の最大アイドル時間 (時間単位)
生存期間	Cookie の最大有効期間 (時間単位)
クライアント IP を強制する	持続 Cookie を発行したクライアント IP と同じ IP アドレスからのみ使用可能にするかどうか
セキュア Cookie を使用する	持続 Cookie に `Secure` 属性を設定するかどうか
HTTP Only Cookie を使用する	持続 Cookie に `HttpOnly` 属性を設定するかどうか
HMAC 署名鍵	Cookie の HMAC 署名に使用する Base64 でエンコードされた 256 ビットの鍵 作成方法は「HMAC 署名鍵の作成」を参照して下さい
Cookie 名	持続 Cookie の名前
ログアウト時に持続 Cookie を削除する	ユーザーがログアウトしたときに持続 Cookie を削除するかどうか

項目名	設定内容
ユーザーをチェックする	認証中のユーザーと持続 Cookie のユーザーが同一かを確認するかどうか

ここでは「Cookie 名」に任意の Cookie 名 (`persistent-cookie`) を指定し、HMAC 署名鍵を設定したものとします。

一定期間、多要素認証を省略する場合は、ログアウト時に Cookie が削除されないように「ログアウト時に持続 Cookie を削除する」を無効にします。加えて、共有 PC を使用している場合などに JWT のユーザーと認証中のユーザーが異なってユーザーがログインできなくならないように「ユーザーをチェックする」を有効にします。

3.1.1 HMAC 署名鍵の作成

HMAC 署名鍵の作成手順について説明します。「HMAC 署名鍵」には Base64 方式で変換した 256bit 以上の乱数を設定します。

下記のコマンドを実行して作成します。ここでは、乱数の長さに 32Byte (=256bit) を指定しています。

```
$ openssl rand -base64 32
```

3.2 持続 Cookie の発行確認認証モジュールの追加

「一定期間、多要素認証を省略するかどうか」をユーザーに選択させるときには以下の設定を行ってください。

1. OpenAM 管理コンソールにログイン後、対象のレルムを選択します。
2.「認証」→「モジュール」に移動し、「モジュールの追加」を押下します。
3.「名前」に認証モジュール名 (ここでは `persistentCookieApprove`) を入力し、「種類」のドロップダウンリストから**持続 Cookie の発行確認** を選択します。

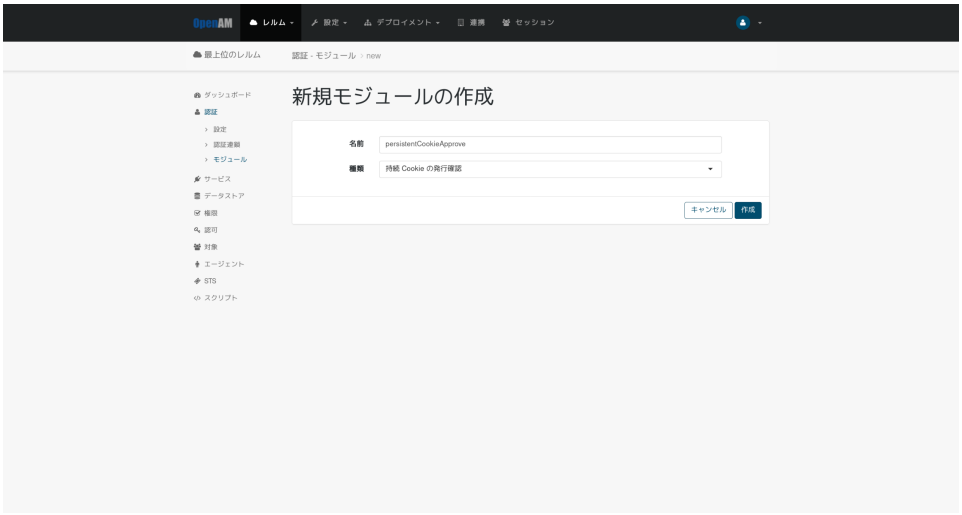

図 8　持続 Cookie の発行確認認証モジュールの作成

4.「作成」を押下し、認証モジュールの設定画面に移動します。

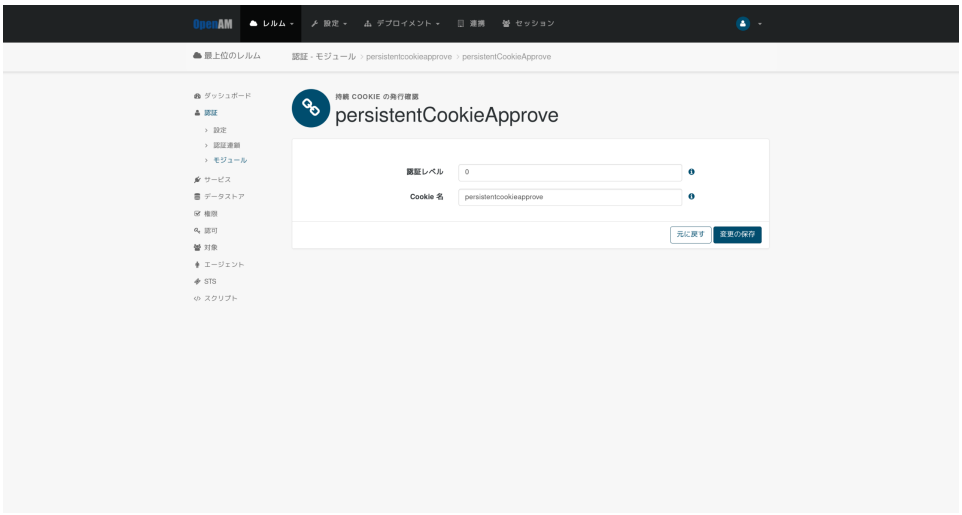

図 9　持続 Cookie の発行確認認証モジュールの設定

5. 各項目の設定を行い、「変更の保存」を押下します。
   通常、変更する必要はありませんが、「Cookie 名」を変更したい場合には下記の手順に沿って変更してください。
   1. 持続 Cookie の発行確認認証モジュールの「Cookie 名」に任意の名前 (ここでは `pCookieApprove`) を入力して「変更の保存」を押下します。

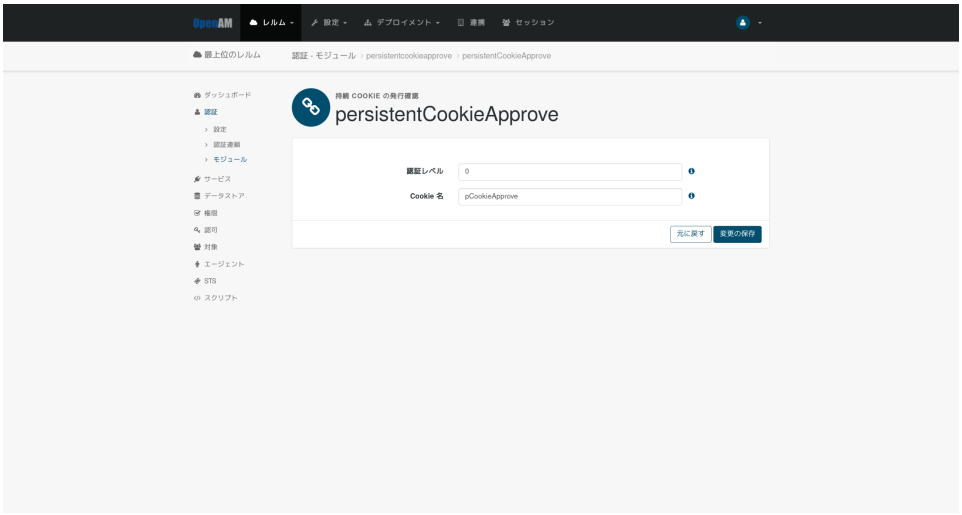

図 10　持続 Cookie の発行確認認証モジュールの「Cookie 名」の変更

   2. `/opt/osstech/share/tomcat/webapps/openam/XUI/templates/openam/authn/PersistentCookieApprove2.html` の以下の部分を変更して保存します。

```
$('#idToken2_1').on('click', function() {
    if ($('input[name="persistentcookieapprove"]').prop('checked')) {
        cookieHelper.setCookie("persistentcookieapprove", "No", expire,
cookie_path);
    }
});
```

上記の setCookie() の引数の persistentcookieapprove を 1. で設定した Cookie 名 (ここでは pCookieApprove) に変更します。

```
$('#idToken2_1').on('click', function() {
    if ($('input[name="persistentcookieapprove"]').prop('checked')) {
        cookieHelper.setCookie("pCookieApprove", "No", expire,
cookie_path);
    }
});
```

3. html の変更が反映されない可能性があるため、ブラウザのキャッシュを削除します。

4 認証連鎖の追加

4.1 一定期間、認証を省略する

ここでは、一定期間、認証を省略する場合の認証連鎖の設定方法を説明します。

1. OpenAM 管理コンソールにログイン後、対象のレルムを選択します。
2. 「認証」→「認証連鎖」に移動し、「認証連鎖の追加」を押下します。
3. 「名前」に任意の認証連鎖名 (ここでは `persistentCookieService`) を入力し、「作成」を押下します。

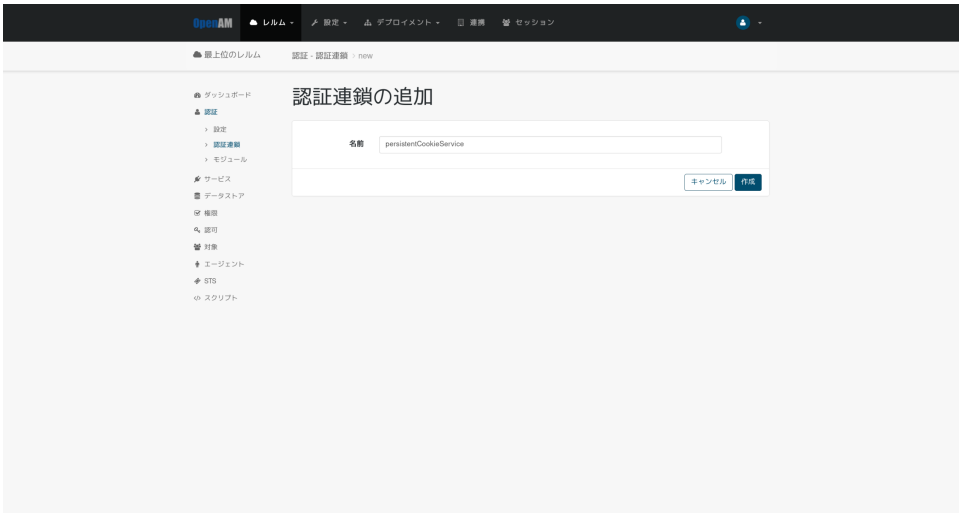

図 11　認証連鎖の追加

4. 「認証モジュールの追加」を押下し、「モジュールの選択」のドロップダウンリストから持続 Cookie 認証モジュール (ここでは `persistentCookie`) を選択し、「基準の選択」のドロップダウンリストから `Sufficient` を選択して「OK」を押下します。

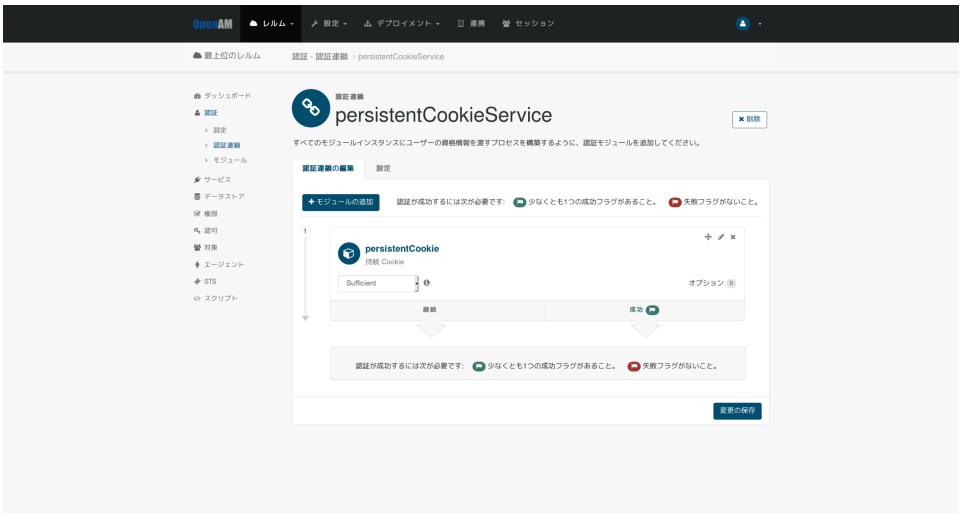

図 12　持続 Cookie 認証モジュールの追加

5. 4. と同様にして、設定済みの持続 Cookie 認証モジュールと組み合わせて利用する認証モジュール (ここでは DataStore) を選択し、「基準の選択」から Required を選択して「OK」を押下します。

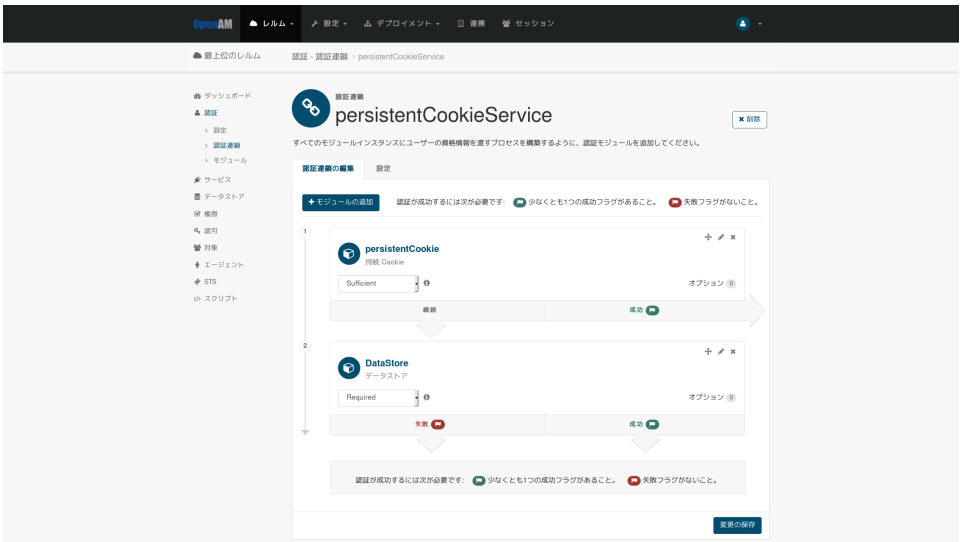

図 13　データストア認証モジュールの追加

6.「変更の保存」を押下します。
7.「認証」→「設定」に移動し、「組織認証設定」のドロップダウンリストから作成した認証連鎖 (ここでは persistentCookieService) を選択し、「変更の保存」を押下します。

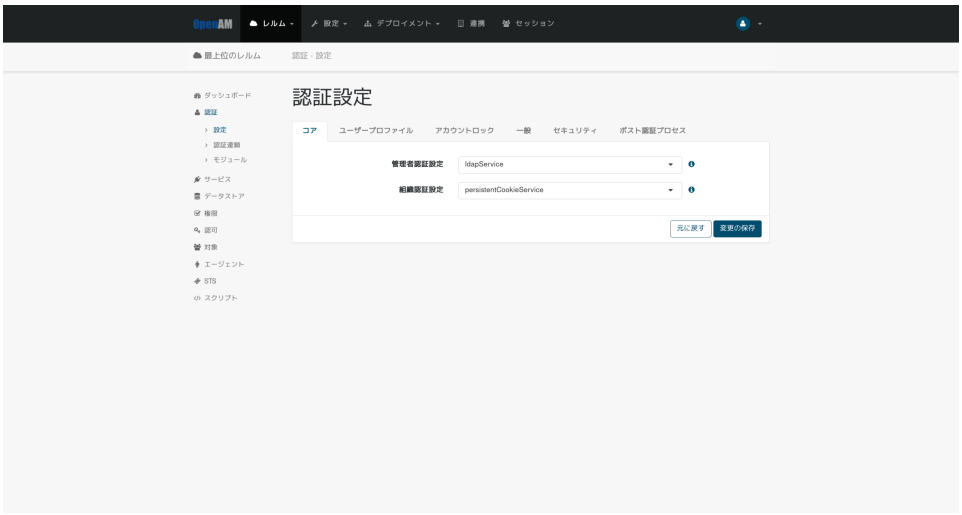

図 14　組織認証設定の変更

4.2 一定期間、多要素認証を省略する

ここでは、一定期間、多要素認証を省略する場合の認証連鎖の設定方法を説明します。

1. OpenAM 管理コンソールにログイン後、対象のレルムを選択します。
2.「認証」→「認証連鎖」に移動し、「認証連鎖の追加」を押下します。
3.「名前」に任意の認証連鎖名 (ここでは `persistentCookieBranchService`) を入力し、「作成」を押下します。

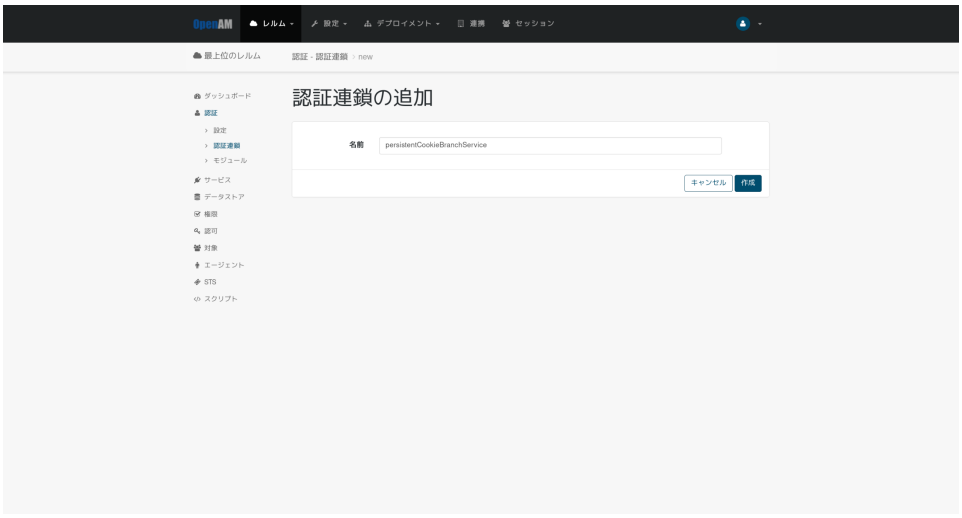

図 15　認証連鎖の追加

4.「認証モジュールの追加」を押下し、「モジュールの選択」のドロップダウンリストから設定済みのユーザーを特定する認証モジュール (ここでは `DataStore`) を選択し、「基準の選択」のドロップダウンリストから `Requisite` を選択して「OK」を押下します。

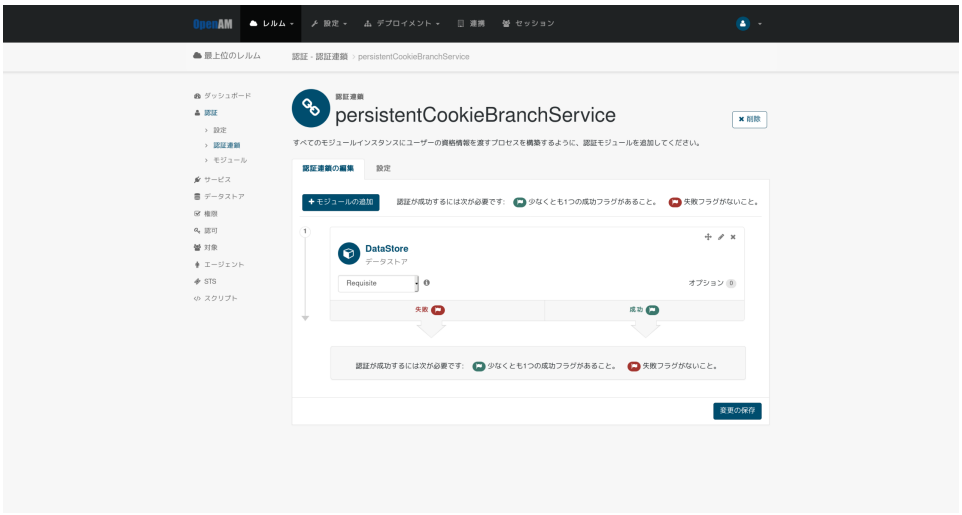

図 16　データストア認証モジュールの追加

5. 4. と同様にして、持続 Cookie 認証モジュール (ここでは `persistentCookie`) を選択し、「基準の選択」から `Sufficient` を選択して「OK」を押下します。

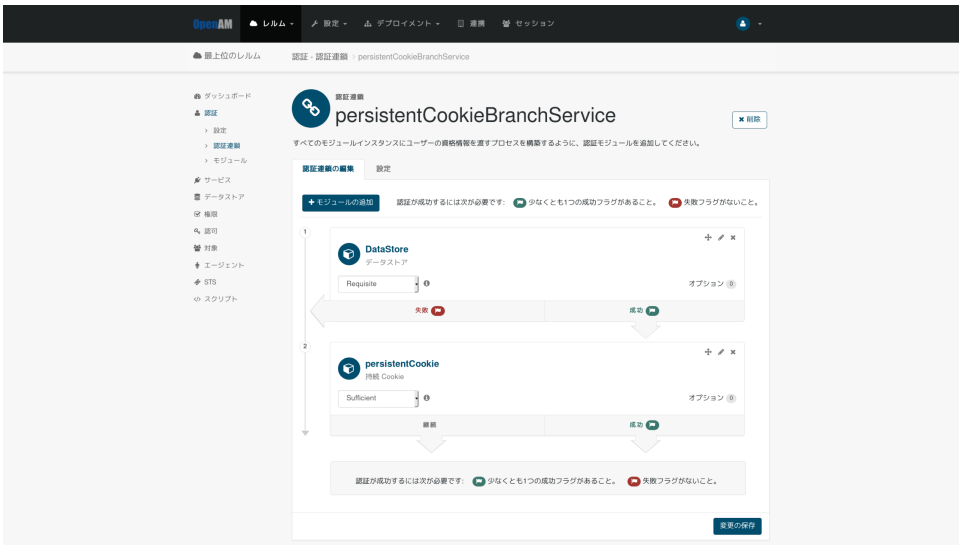

図 17　持続 Cookie 認証モジュールの追加

6. 4. と同様にして、追加の認証を行う認証モジュール (ここでは `ForgeRockAuthenticator`) を選択し、「基準の選択」から `Required` を選択して「OK」を押下します。

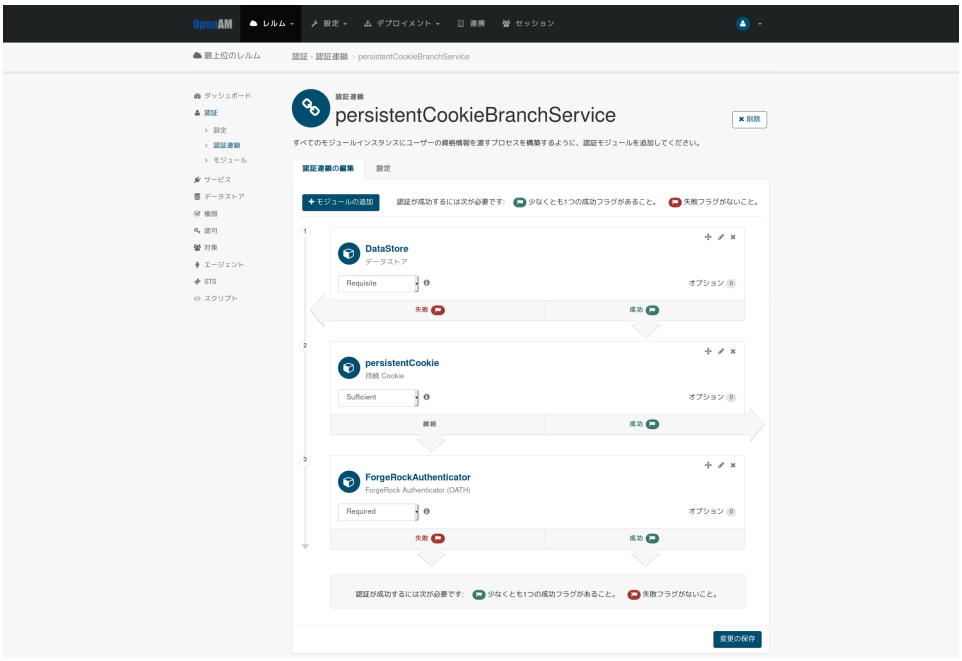

図 18　ForgeRock Authenticator (OATH) 認証モジュールの追加

7.「変更の保存」を押下します。
8.「認証」→「設定」に移動し、「組織認証設定」のドロップダウンリストから作成した認証連鎖 (ここでは `persistentCookieBranchService`) を選択し、「変更の保存」を押下します。

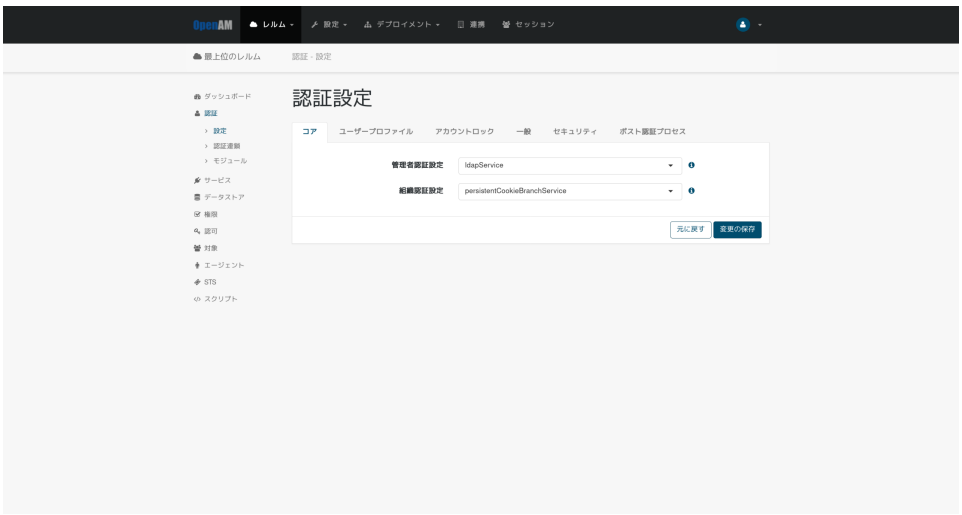

図 19　組織認証設定の変更

4.2.1 「一定期間、多要素認証を省略するかどうか」をユーザーに選択させる

ここでは、「一定期間、多要素認証を省略するかどうか」をユーザーに選択させる場合の認証連鎖の設定方法を説明します。

1. 「一定期間、多要素認証を省略する」と同様の設定を行います。
2. 作成した認証連鎖を編集するため、認証連鎖名 (ここでは `persistentCookieBranchService`) を押下して認証連鎖編集画面に移動します。

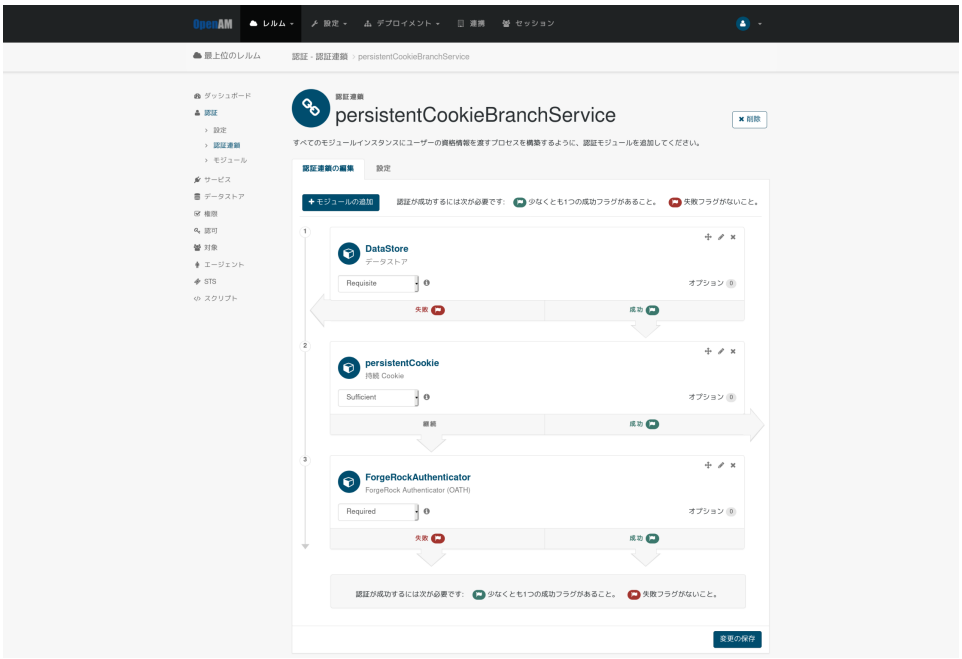

図 20　認証連鎖の編集

3. 「一定期間、多要素認証を省略する」の 6. で設定した認証モジュール (ここでは `ForgeRockAuthenticator`) の「基準の選択」を `Required` から `Requisite` に変更します。
4. 「モジュールの追加」を押下し、「モジュールの選択」のドロップダウンリストから持続 Cookie の発行確認認証モジュール (ここでは `persistentCookieApprove`) を選択し、「基準の選択」のドロップダウンリストから `Required` を選択して「OK」を押下します。

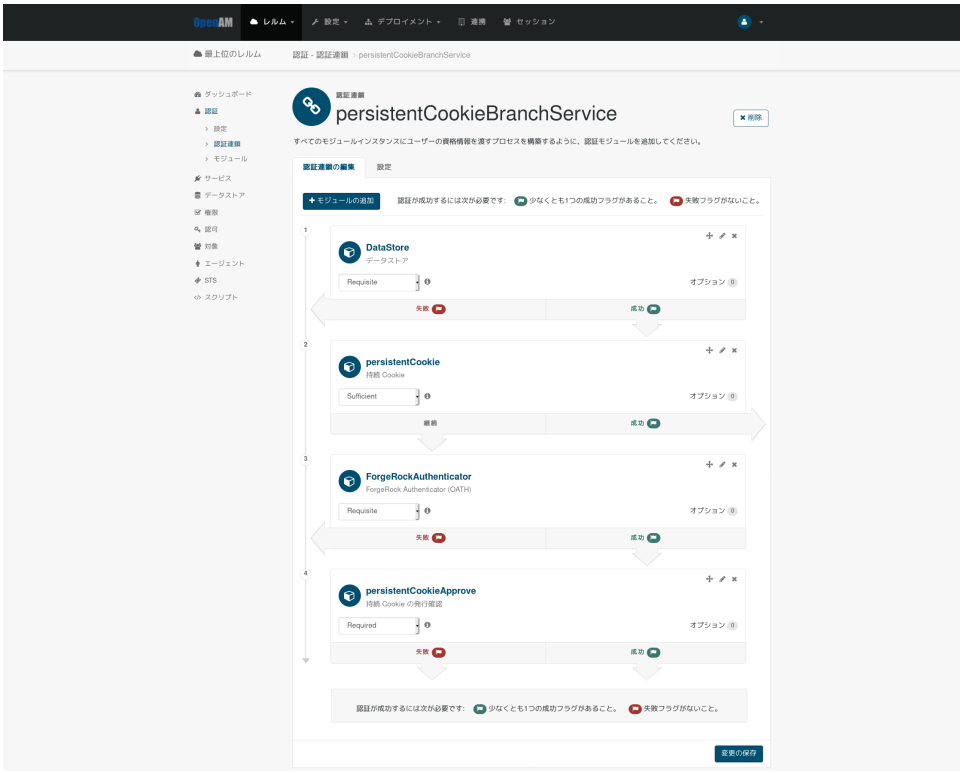

図 21　持続 Cookie の発行確認認証モジュールの追加

4.「変更の保存」を押下します。

5 認証時の動作

5.1 一定期間、認証を省略する

ここでは、一定期間、認証を省略するのように設定した場合の認証時の動作について説明します。

1. OpenAM にアクセスします。
2. データストア認証モジュールの画面でユーザーの ID とパスワードを入力し、「ログイン」を押下します。

図 22　データストア認証

3. ログインに成功するとユーザープロファイル画面に遷移します。

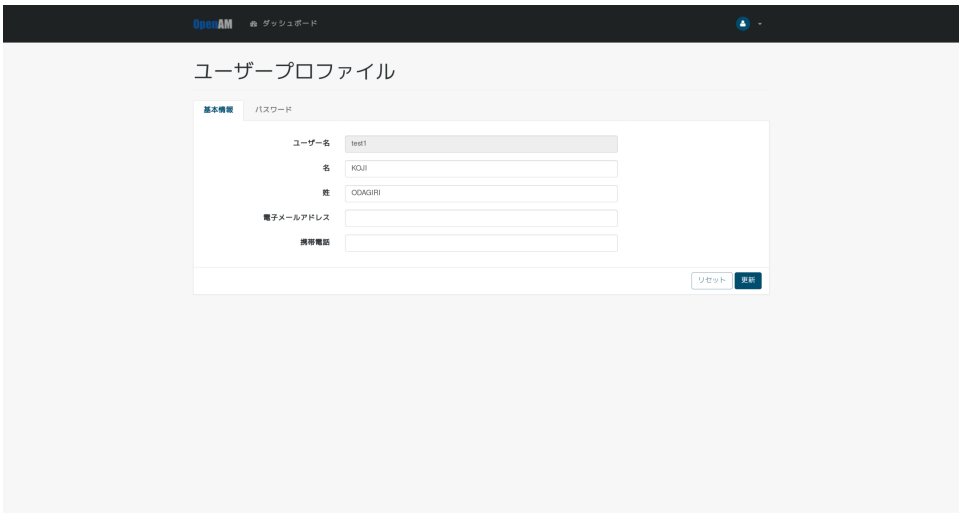

図 23　ユーザープロファイル画面

4. ブラウザを閉じた後、再度 OpenAM にアクセスします。
5. 3. の初回認証時に Cookie がセットされるため、持続 Cookie 認証に成功してユーザープロファイル画面に遷移します。

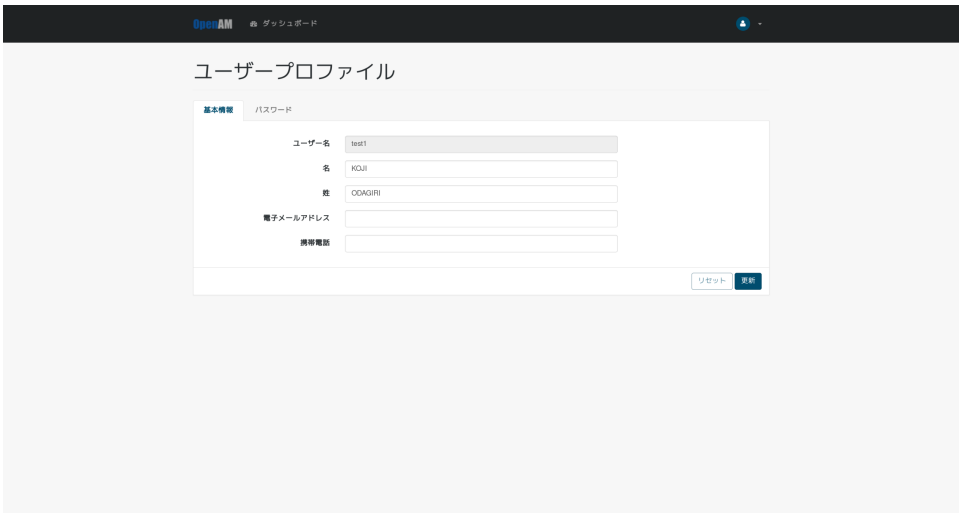

図 24　ユーザープロファイル画面

5.2 一定期間、多要素認証を省略する

ここでは、一定期間、多要素認証を省略するのように設定した場合の認証時の動作について説明します。

1. OpenAM にアクセスします。
2. データストア認証モジュールの画面でユーザーの ID とパスワードを入力し、「ログイン」を押下して ForgeRock Authenticator (OATH) モジュールの画面に移動します。

図 25　データストア認証

3. Authenticator のワンタイムパスワードを入力して「送信」を押下します。

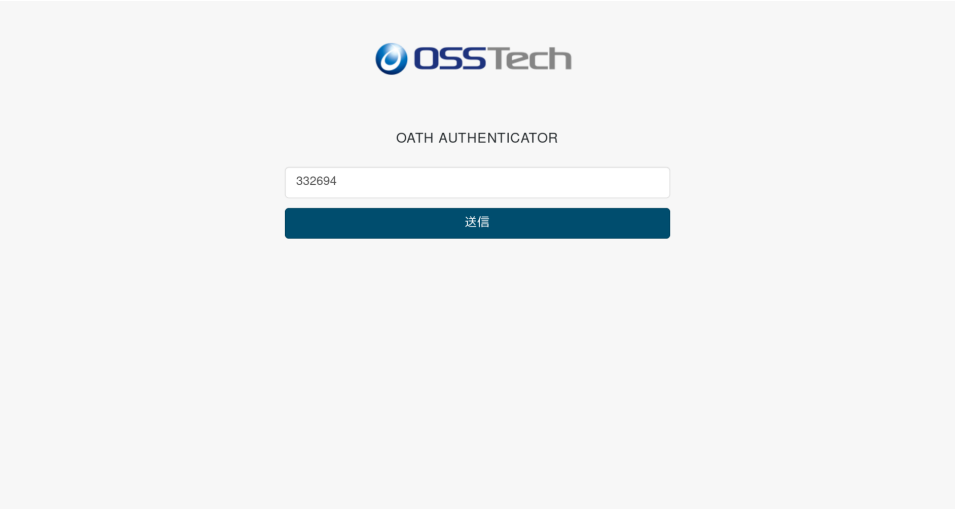

図 26　ForgeRock Authenticator (OATH) 認証

4. ログインに成功するとユーザープロファイル画面に遷移します。

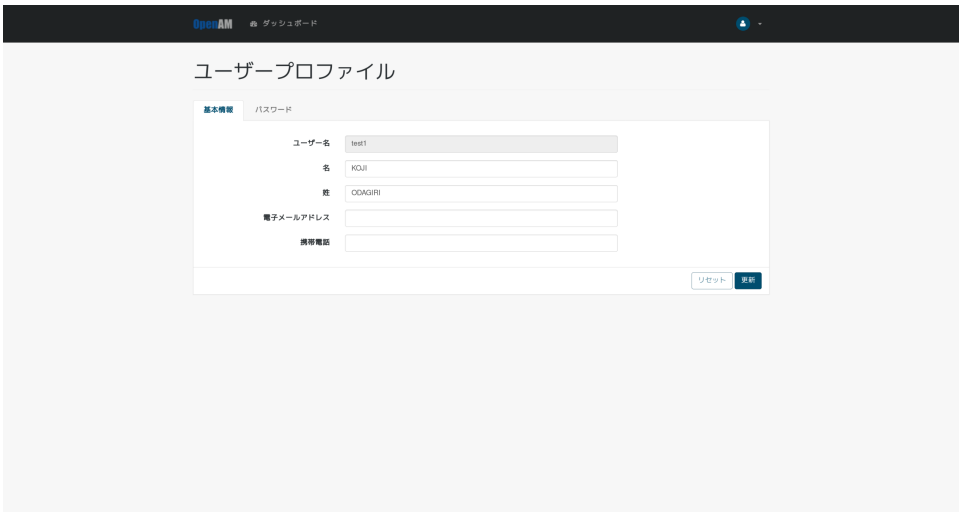

図 27　ユーザープロファイル画面

5. ブラウザを閉じた後 (またはログアウト後)、再度 OpenAM にアクセスします。
6. データストア認証モジュールの画面でユーザーの ID とパスワードを入力し、「ログイン」を押下します。

図 28　データストア認証

7. 4. の初回認証時に Cookie がセットされるため、持続 Cookie 認証に成功してその後の多要素認証を行うことなくユーザープロファイル画面に遷移します。

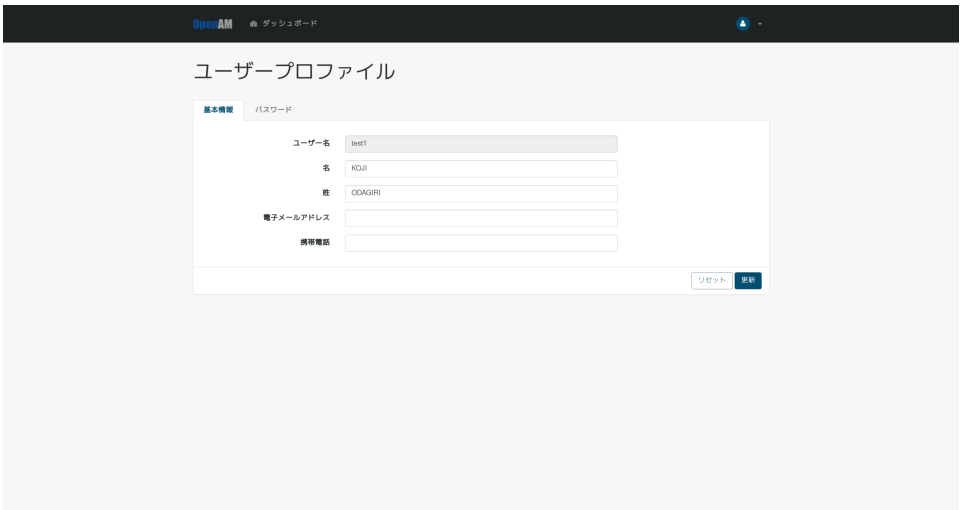

図 29　ユーザープロファイル画面

5.2.1 「一定期間、多要素認証を省略するかどうか」をユーザーに選択させる

ここでは、「一定期間、多要素認証を省略するかどうか」をユーザーに選択させるのように設定した場合の認証時の動作について説明します。

1. OpenAM にアクセスします。

2. データストア認証モジュールの画面でユーザーの ID とパスワードを正しく入力し、「ログイン」を押下すると ForgeRock Authenticator (OATH) モジュールの画面に遷移します。

図 30　データストア認証

3. Authenticator のワンタイムパスワードを正しく入力し、「送信」を押下すると持続 Cookie の発行確認認証モジュールの画面に遷移します。

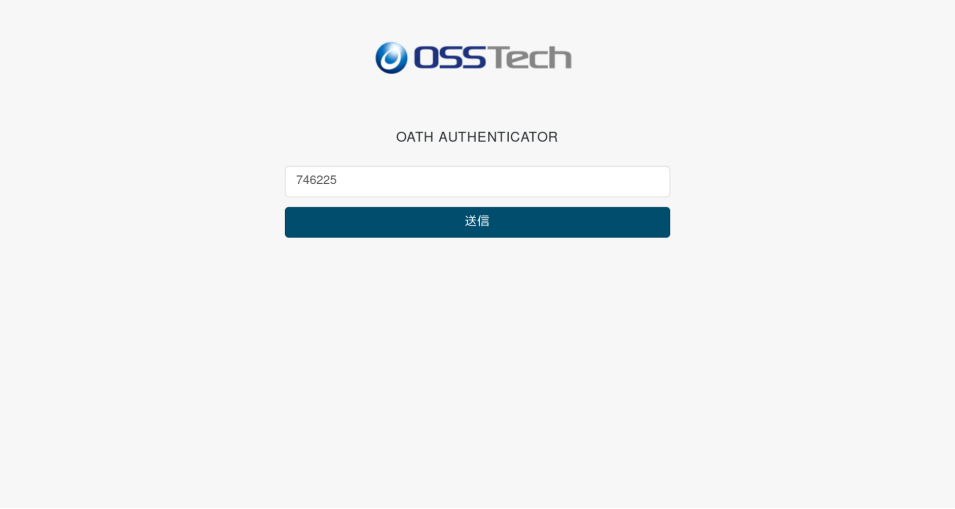

図 31　ForgeRock Authenticator (OATH) 認証

4.「多要素認証について確認」画面の「次回以降このブラウザからのログインで多要素認証をしない」で「はい」か「いいえ」を選択します。

図 32　持続 Cookie の発行確認認証

- 「はい」を選択した場合、「次回以降表示しない」にチェックを入れるかどうかに関わらず、上記の「一定期間、多要素認証を省略する」と同様の動作をします。
- 「いいえ」を選択した場合、再度 OpenAM にアクセスした際にも多要素認証が要求され、その後持続 Cookie の発行確認認証モジュールの画面が表示されます。
- 「次回以降表示しない」にチェックを入れて「いいえ」を選択した場合、上記の「いいえ」を選択した場合と同様に多要素認証が要求されますが、持続 Cookie の発行確認認証モジュールの画面は表示されなくなります。

6 Cookie 削除インターフェース

本章ではログイン画面に Cookie を削除するチェックボックスを準備する手順を説明します。

6.1 持続 Cookie 削除のチェックボックス

一定期間、多要素認証を省略すると設定した場合、持続 Cookie が有効な期間は多要素認証が不要となります。ログイン画面に Cookie を削除するチェックボックスを表示し、ユーザーがチェックを入れてログインすることで再度多要素認証を行うことが可能です。

図 33　チェックボックスを導入したログイン画面

デフォルトのログイン画面にはチェックボックスは表示されません。チェックボックスを表示するためには設定が必要です。

6.2 設定手順

チェックボックスを表示するために次の 2 ファイルの変更が必要です。

1. /opt/osstech/share/tomcat/webapps/openam/XUI/config/AppConfiguration.js
2. 1 段階目の認証のテンプレート HTML ファイル
   - 1 段階目の認証がデータストア認証の場合は/opt/osstech/share/tomcat/webapps/openam/XUI/ templates/openam/authn/DataStore1.html です。

それぞれのファイルの変更方法について説明します。

6.2.1 AppConfiguration.js の変更内容

partialUrls に _CookieCheckBox.html ファイルを読み込む設定を追加します。 下記のように 1 行追加してください。

```
   partialUrls: [
       〜
       "partials/login/_TextOutput.html",
       "partials/login/_CookieCheckBox.html", <- 追加
       "partials/login/_PollingWait.html"
   ]
```

6.2.2 DataStore1.html の変更内容

「持続 Cookie が存在したらチェックボックスを表示する定義」と「チェックボックスをチェックした状態でログインすると Cookie を削除する JavaScript」を追加します。

- チェックボックス表示定義

```
{{#equals type "ConfirmationCallback"}}
  {{#if ../showRememberLogin}}
    {{> login/_RememberLogin }}
  {{/if}}
  <!-- 追加ここから -->
  {{cookieCheckbox "persistent-cookie"
      "templates.user.LoginTemplate.cookieCheckbox"}}
  <!-- 追加ここまで -->
{{/equals}}
```

追加設定の第二引数に持続 Cookie 名を指定します。上記の場合は持続 Cookie 名は

persistent-cookie と指定しています。認証モジュールで設定した Cookie 名に読み替えてください。

第三引数の値は画面に表示されるチェックボックスの文言のキー値です。日本語の場合は/opt/osstech/share/tomcat/webapps/openam/XUI/locales/ja/translation.json で定義された内容が表示されます。“templates.user.LoginTemplate.cookieCheckbox” は、「多要素認証を行う」となります。

- Cookie 削除の JavaScript

```
        </fieldset>
    </form>
</div>
<!-- 追加ここから -->
<script language="javascript" type="text/javascript">
  $(document).ready(function(){
    const helper = require("org/forgerock/commons/ui/common/util/CookieHelper");
    const cookie_path = "/";
    const cookie_domain = "【Cookie ドメイン】";
    $('#loginButton_0').on('click', function() {
      const cookie_name = "【持続 Cookie 名】";
      if ($('input[name="【持続 Cookie 名】"]').prop('checked')) {
        helper.deleteCookie(cookie_name, cookie_path, cookie_domain);
      }
    });
</script>
<!-- 追加ここまで -->
```

ログインを実行した際に持続 Cookie を削除する JavaScript です。【持続 Cookie 名】に実際の持続 Cookie 名、【Cookie ドメイン】は OpenAM のドメインを設定します。

以上で作業は完了です。正しく設定されれば持続 Cookie が発行された状態でログイン画面にアクセスするとチェックボックスが表示されます。

6.2.3 留意事項

チェックボックスは Cookie が存在した場合に表示されます。Cookie が存在しない初回アクセスでは表示されません。Cookie 値の精査までは行っていないため、持続 Cookie の JWT の有効期限 (項目名: アイドルタイムアウト) が切れている場合でもチェックボックスは表示されます。

7 改版履歴

- 2020 年 8 月 18 日 リビジョン 1.0
  - 初版作成
- 2020 年 9 月 1 日 リビジョン 1.1
  - Cookie 削除インターフェースを追加
- 2022 年 7 月 14 日 リビジョン 1.2
  - 表紙の社名を OSSTech 株式会社に変更