OpenAM 14 SAML 設定ガイド

OSSTech **株式会社**

更新日	2023 **年** 4 **月** 21 **日**
リビジョン	2

目次

1 はじめに

1.1 本書の目的

本書では、OpenAM を SAML IdP として設定し、対象とする SAML SP との間で SAML を利用したシングルサインオン環境を構築するための手順を説明します。既に OpenAM がインストールされ、初期設定が完了していることを前提とします。

1.2 前提条件

本書では、以下のサーバー構成を前提とします。

- OpenAM 管理コンソールへアクセスする URL
  - https://openam01.example.co.jp/openam
- 一般ユーザーが OpenAM へアクセスする URL
  - https://sso.example.co.jp/openam

1.3 対象とする SAML SP

本書で対象とする SAML SP は以下のサービス/アプリケーションです。

- Google Workspace(旧: G Suite)
- Salesforce

2 SAML IdP の設定

本章では、OpenAM を SAML IdP として動作させるための設定手順について説明します。

2.1 SAML 用署名鍵の作成と OpenAM での利用設定

SAML メッセージに対して署名を行なうための署名鍵を作成し、OpenAM で利用できるように設定します。

OpenAM にはデフォルトで「test」という署名鍵が登録されていますが、これは全ての OpenAM に含まれる共通の鍵であるため、IdP のなりすましなどの脆弱性につながります。そのため、署名鍵を新規に作成し OpenAM にインポートします。

2.1.1 キーストアと鍵ペアの生成

JDK の keytool コマンドを利用して、鍵ペアを作成します。

```
$ keytool -genkeypair \
-keyalg rsa \
-alias openam-idp \
-dname "CN=sso.example.co.jp,OU=development,O=EXAMPLE,L=Shinagawa-ku,ST=Tokyo,
C=JP" \
-keypass xxxxxxxx \           # 秘密鍵のパスワードを指定
-keystore mykeystore.jceks \  # キーストアファイル名を指定
-storetype JCEKS \            # キーストアタイプを指定（例：JCEKS）
-storepass changeit \         # キーストアのパスワードを指定
                                （Java のキーストアのパスワード初期値は
                                 changeit です）
-validity 3650 \              # 鍵の有効期限を指定（例：10 年）
-keysize 2048                 # 鍵の長さを指定
```

「\(バックスラッシュ)」はコマンドラインの途中で改行を行うために入力しています。「\」を入れずに、全てのオプションを一行で指定することも可能です。「#」以降の文字列はコメントであるため、実際にはコマンドラインに入力する必要はありません。

各オプションについて説明します。

- -genkeypair
  - 鍵ペアを新規に作成するオプションです。
- -keyalg アルゴリズム名
  - 鍵のペアを生成するのに使うアルゴリズムを指定します。

- -alias エイリアス名
  - 証明書の別名を指定します。任意の名前を指定可能です。
- -dname 識別名
  - 識別名を指定します。
- -keypass パスワード
  - 秘密鍵のパスワードを指定します。
- -keystore キーストアファイル名
  - キーストアファイル名を指定します。
- -storetype キーストアタイプ
  - キーストアタイプを指定します。
- -storepass パスワード
  - キーストアのパスワードを指定します。
- -validity 日数
  - 鍵の有効期限を日数で指定します。
- -keysize ビット数
  - 鍵の長さをビットで指定します。

2.1.2 キーストアと鍵ペアの配置

作成した鍵ペアを OpenAM で利用できるように設定します。

- 鍵ペアを任意のパスに配置します。
  - 本ドキュメントでは、例として「/opt/osstech/var/lib/data/openam/private」に配置し、鍵ペアファイル名を「mykeystore.jceks」と仮定して説明します。
  - このとき、鍵ペアファイルが Tomcat プロセスの実行ユーザー “tomcat” が読み取れるようにパーミッションを設定します。

```
# mkdir -p /opt/osstech/var/lib/data/openam/private
# cp  mykeystore.jceks /opt/osstech/var/lib/data/openam/private
# chown -R root:tomcat /opt/osstech/var/lib/data/openam/private
# chmod 750 /opt/osstech/var/lib/data/openam/private
# chmod 640 /opt/osstech/var/lib/data/openam/private/mykeystore.jceks
```

2.1.3 キーストアと鍵ペアのパスワードファイルを作成

キーストアと鍵ペアファイルに設定されているパスワードをそれぞれ符号化して、テキストファイルに保存します。このテキストファイルは OpenAM がキーストアと鍵ペアを読み込む際に使用します。パスワードの符号化は OpenAM の管理コンソール (Web インタフェース) から、以下の手順で行ないます。

なお、符号したテキストファイルはキーストア用ファイルと鍵ペア用ファイルに分けて保存します。ここでは、キーストア用の符号化パスワードファイルの作成方法のみ説明します。鍵ペアにおいても、同様の手順で符号化パスワードをファイルに保存してください。ファイル名は任意です。

1. OpenAM に管理者ユーザーでログインします。
2. ブラウザのアドレスバーに以下の URL を入力し、Enter を押下します。
   - https://openam01.example.co.jp/openam/encode.jsp
3. 「符号化するパスワードを入力してください」のテキストエリアにキーストアのパスワードを入力し、「符号化」ボタンを押下します。キーストアのパスワードとは、「キーストアと鍵ペアの生成」で、keytool コマンドの「-storepass」オプションで指定したパスワードです。
4. ブラウザ画面に表示された符号化されたパスワードを、以下のコマンドを実行して任意のファイルに保存します。ここでは、/opt/osstech/var/lib/data/openam/private/.storepass に保存したと仮定します。符号化パスワードの末尾に改行コードが入ると OpenAM からの読み込みに失敗するため、tr コマンドを利用してファイル末尾の改行コードを削除しています。

```
# cd /opt/osstech/var/lib/data/openam/private/
# vim .storepass
 エンコードされたキーストアのパスワードを入力して保存
# tr -d '\n' < .storepass  > tmp && mv tmp .storepass
```

5. ファイルパーミッションを設定します。

```
# chown root:tomcat /opt/osstech/var/lib/data/openam/private/.storepass
# chmod 640 /opt/osstech/var/lib/data/openam/private/.storepass
```

以上で完了です。

鍵ペアについても、同様の手順で符号化後のパスワードをファイルに保存します (/op-

t/osstech/var/lib/data/openam/private/.keypass として保存したと仮定します)。

OpenAM を冗長化構成 (サイト構成) で構築している場合は、1 号機で作成したキーストアやパスワードファイルを 2 号機にコピーしてください。(1 号機と 2 号機は同じファイルを使用します。)

2.1.4 OpenAM のキーストア設定を変更

OpenAM の管理コンソールから、新規に作成したキーストアと鍵ペアを使用するように設定を変更します。

1. OpenAM に管理者ユーザーでログインします。
2. 「設定」タブ→「デフォルトサーバー」→「セキュリティ」→「キーストア」を開きます。
3. 「キーストア」セクションの各項目に以下の値を入力します。
   - 「キーストアファイル」: /opt/osstech/var/lib/data/openam/private/mykeystore.jceks
   - 「Keystore Type」: JCEKS
   - 「キーストアパスワードファイル」: /opt/osstech/var/lib/data/openam/private/.storepass
   - 「非公開鍵パスワードファイル」: /opt/osstech/var/lib/data/openam/private/.keypass
   - 「証明書エイリアス」: openam-idp
4. 画面右下の「変更の保存」ボタンを押下します。
5. OpenAM を再起動します。

以上で完了です。

OpenAM を冗長化構成 (サイト構成) で構築している場合、この設定を変更すると全ての OpenAM サーバーのキーストア設定が変更されます。そのため、全ての OpenAM サーバーに新しい SAML 用署名鍵を配置してから、設定を変更してください。

2.1.5 証明書のエクスポート

鍵ペアから証明書をエクスポートします。この証明書は SAML SP 側に登録する必要があります。以下のコマンドを実行します。

```
# cd /opt/osstech/var/lib/data/openam/private/
# keytool -keystore ./mykeystore.jceks \
-storetype JCEKS -export -alias openam-idp -file idp.der
   Enter keystore password:  キーストアのパスワードを入力します
```

```
   Certificate stored in file <idp.der>
```

これで、署名鍵から DER 形式の証明書が作成されました。続いて、DER 形式の証明書を PEM 形式に変換します。

```
$ openssl x509 -in idp.der -inform DER -out idp.pem -outform PEM
```

以上で完了です。生成された idp.pem を後述の手順で SAML SP に登録します。

2.2 SAML IdP の作成

OpenAM が IdP として動作するように設定します。

2.2.1 IdP の作成

設定は OpenAM の「共通タスク」から行います。

1. OpenAM に管理者ユーザーでログインします。
2. OpenAM 管理コンソールで対象のレルムをクリックします。
3.「共通タスク」で「SAMLv2 プロバイダを作成」をクリックします。
4.「SAMLv2 プロバイダを作成」のメニューから「ホストアイデンティティープロバイダの作成」をクリックします。
5.「このプロバイダのメタデータがありますか?」は「いいえ」をチェックします。
6.「メタデータ」の「名前」の URL を https://sso.example.co.jp/openam に変更します。
   - デフォルトでは管理コンソールへのアクセスした FQDN(openam01.example.co.jp) の URL になっています。
   - 443 番はデフォルトポートであるため、ポート番号も省略します。
7.「署名鍵」はプルダウンメニューから選択します。「キーストアと鍵ペアの生成」で指定したエイリアス名を選択してください。
8.「新しいトラストサークル」にトラストサークルの名前を入力します。トラストサークルとは SAML の信頼関係を結ぶ SAML IdP と SAML SP のグループです。ここでは”usrcot” と入力します。
9.「ベース URL」の URL を https://sso.example.co.jp/openam に変更します。
   - デフォルトでは管理コンソールへアクセスする FQDN(openam01.example.co.jp) の URL になっています。
   -「管理コンソールへアクセスする FQDN」と「一般ユーザーがアクセスする FQDN」が異なる場合、「一般ユーザーがアクセスする URL」を入力します。
   - ここに設定された値をベースにエンドポイント URL が生成されます。
   - この機能は osstech-openam14-14.2.0-16 以降で利用可能です。これより前のバージョンでエンドポイント URL を変更するには IdP の作成後「エンドポイント URL の調整」を行う必要があります。
10. 画面右上の「設定」ボタンをクリックして設定を保存します。属性マッピングは特に設定しません。
11. 設定完了の画面が表示されます。次のアクションを求められますが、ここでは「終

了」ボタンをクリックします。

デフォルトで登録されている署名鍵「test」は、全ての OpenAM に含まれる共通の鍵であるため、本番運用では使用しないでください。評価環境の場合でも、長期の運用は避けてください。SAML IdP のなりすましの危険性があります。
本番環境などでは、必ず独自の署名鍵を作成してください。

2.2.2 エンドポイント URL の調整

「管理コンソールへのアクセスする FQDN」と「一般ユーザーがアクセスする FQDN」が異なる場合に本作業が必要となります。「IdP の作成」の「9」でベース URL を一般ユーザーがアクセスする FQDN に変更して作成した場合、この作業を行う必要はありません。メタデータ内の SAML のエンドポイントの URL は管理コンソールでアクセスした際の FQDN (openam01.example.co.jp) の URL になります。これを一般ユーザーがアクセスする FQDN(sso.example.co.jp) に変更を行います。作業は ssoadm コマンドを使用します。(ssoadm コマンドについてはコマンドライン利用手順書を参照してください)

1. SAML 設定のエクスポート

```
# /opt/osstech/bin/ssoadm  export-entity \
    --adminid amAdmin \
    --password-file [パスワードファイル] \
    --entityid "https://sso.example.co.jp/openam" \
    --realm "[レルム名]" \
    --meta-data-file saml_meta.xml \
    --extended-data-file saml_data.xml
```

2. URL の変更

```
# sed -i -e "s@https://openam01.example.co.jp:443/openam@
                    https://sso.example.co.jp/openam@g" ./saml_meta.xml
# sed -i -e "s@https://openam01.example.co.jp:443/openam@
                    https://sso.example.co.jp/openam@g" ./saml_data.xml
```

ファイル内の<SingleSignOnService>の Location で指定されている URL の FQDN が一般ユーザーでアクセスする FQDN に変わっていることを確認します。

```
# view saml_meta.xml
```

3. トラストサークルからの除外と設定の削除

設定をインポートする前に、IdP の設定をトラストサークルから除外して削除を行います。

- トラストサークルから除外

```
# /opt/osstech/bin/ssoadm  remove-cot-member \
     --adminid amAdmin \
     --password-file [パスワードファイル] \
     --entityid "https://sso.example.co.jp/openam" \
     --realm "[レルム名]" \
     --cot usrcot
```

- 設定の削除

```
# /opt/osstech/bin/ssoadm  delete-entity \
     --adminid amAdmin \
     --password-file [パスワードファイル] \
     --entityid "https://sso.example.co.jp/openam" \
     --realm "[レルム名]"
```

4. 変更した設定のインポート

```
# /opt/osstech/bin/ssoadm import-entity \
     --adminid amAdmin \
     --password-file [パスワードファイル] \
     --realm "[レルム名]" \
     --cot usrcot \
     --meta-data-file saml_meta.xml \
     --extended-data-file saml_data.xml
```

以上で SAML IdP の作成は完了です。

以下の URL にアクセスすると IdP のメタデータをダウンロード可能です。

```
https://openam01.example.co.jp/openam/saml2/jsp/exportmetadata.jsp?
entityid=https://sso.example.co.jp/openam&realm=[レルム名]
```

3 SAML SP の設定

本章では、OpenAM への SAML SP の登録と SAML SP への OpenAM の登録の手順について説明します。

3.1 Google Workspace シングルサインオン設定

本章では、OpenAM と Google Workspace を SAML により連携し、シングルサインオンを実現するための設定手順について説明します。

3.1.1 OpenAM へ Google Workspace(SAML SP) を登録

Google Workspace を SAML SP として OpenAM に登録します。

1. OpenAM に管理者ユーザーでログインします。
2. OpenAM 管理コンソールで対象のレルムをクリックします。
3.「共通タスク」で「G Suite の設定」をクリックします。
4. 表示された画面で再度「G Suite の設定」をクリックします。
5. 以下の値を選択・入力します。
   - 「トラストサークル」を選択します。ここでは ”usrcot” を選択します。
   - 「ドメイン名」の「新しい値」に Google Workspace のドメイン名を入力し、「追加」をクリックします。Google Workspace のマルチドメイン機能を利用している場合は、プライマリドメインのドメイン名を入力してください。（ドメインとして”yourdomain.co.jp” を設定したものとします。）
6.「現在の値」にドメイン名が追加されたことを確認し、右上の「作成」をクリックします。
7.「メタデータは正常に設定されました。「了解」をクリックして、サービスプロバイダを設定するためのパラメーターを取得します。」と表示されたら「了解」をクリックします。
8.「G Suite のシングルサインオンの設定画面」が表示されるため、以下の 3 つの URL を控えます。
   - サインインページの URL
   - サインアウトページの URL
   - パスワード変更の URL
9. 検証証明書の「ダウンロードするには、ここをクリックします。」をクリックし、証明書ファイルをダウンロードしておきます。デフォルトでは「OpenSSOCert.txt」と

いうファイル名で保存されます。後述の手順でこの証明書を Google Workspace に登録します。

10. 画面右下の「終了」ボタンをクリックします。
11. 上部メニューの「連携」タブを開きます。
12.「トラストサークル設定」の「エンティティープロバイダ」に Google Workspace が登録されていることを確認します。

以上で完了です。

3.1.2 Google Workspace へ OpenAM(SAML IdP) を登録

Google Workspace のシングルサインオン機能を有効化し、OpenAM(SAML IdP) の URL などを設定します。

1. Google Workspace の管理コンソールにアクセスしログインします。

```
https://www.google.com/a/yourdomain.co.jp/
```

2.「セキュリティ」 →「シングルサインオン（SSO）の設定」を開きます。
3.「サードパーティの ID プロバイダで SSO を設定する」をチェックし、以下の 3 つの URL を入力します。
   - ログイン ページの URL
     –「OpenAM へ Google Workspace(SAML SP) を登録」で控えた「サインインページの URL」を入力します。以下のような URL となります。

```
https://sso.example.co.jp/openam/SSORedirect/metaAlias
/レルム名/idp
```

   - ログアウト ページ URL
     –「OpenAM へ Google Workspace(SAML SP) を登録」で控えた「サインアウトページの URL」を入力します。以下のような URL となります。

```
https://sso.example.co.jp/openam/UI/Logout?
goto=https://mail.google.com/a/yourdomain.co.jp/
```

   - パスワード変更 URL
     –「OpenAM へ Google Workspace(SAML SP) を登録」で控えた「パスワード

変更の URL」を入力します。この URL は任意の URL を指定可能であるため、OpenAM のパスワード変更画面を利用しない場合は別の URL を設定します。

4. 一旦「保存」をクリックします。
5.「認証の確認」の「証明書の更新」をクリックし、「OpenAM へ Google Workspace(SAML SP) を登録」で作成しておいた OpenAM 側の署名鍵の証明書ファイル（OpenSSOCert.txt）をアップロードします。
6.「ドメイン固有の発行元を使用」をチェックします。
7.「ネットワークマスク」 は入力しません。
8.「保存」をクリックします。

以上で完了です。

3.1.3 Google Workspace へのユーザー登録

Google Workspace へ SAML によるシングルサインオンを利用してログインする場合でも、Google Workspace 側にアカウントの登録が必要です。Google Workspace におけるユーザー登録方法は次の URL で説明されています。

- https://support.google.com/a/answer/33310?hl=ja

3.1.4 シングルサインオンの動作確認

ここまでの設定を行うことで、Google Workspace へシングルサインオンするための準備が整いました。実際に Google Workspace へシングルサインオンして動作を確認します。ここでは、Gmail にアクセスしてみます。Gmail は以下の URL からアクセスします。

- https://mail.google.com/a/yourdomain.co.jp/

OpenAM のログイン画面が表示されるはずです。OpenAM のユーザ名とパスワードを入力してログインします。認証に成功すると、自動的にリダイレクトされて Gmail の画面が表示されます。

3.2 Salesforce シングルサインオン設定

3.2.1 前提条件

Salesforce の SAML 連携を設定するにあたり、事前に以下の設定が完了しているものとします。

1. 「SAML IdP の設定」の手順を実施して、SAML IdP の作成が完了している。
2. Salesforce の「私のドメイン」機能が有効化されている (「私のドメイン」として”domainname” を設定したものとします)。
3. Salesforce に一般ユーザーが登録され、ユーザーの統合 ID に OpenAM の ID と同じ文字列が設定されている。
4. OpenAM に一般ユーザーが登録され、Salesforce の該当ユーザーの統合 ID と同じ ID が割り当てられている。

3.2.2 「私のドメイン」機能の有効化

Salesforce で SAML の SP-Initiated SSO を利用するためには Salesforce 側で「私のドメイン」機能を有効にする必要があります。設定手順については Salesforce のドキュメントをご参照ください。

3.2.3 Salesforce に OpenAM(SAML IdP) を登録

管理権限を持つユーザーで Salesforce へログインし、シングルサインオンの設定を行います。 まずは Salesforce で SAML を有効化します。設定場所は [設定] → [ID] → [シングルサインオン設定] と遷移し、表示された画面で「編集」ボタンをクリックします。「SAML を有効化」をチェックし「保存」ボタンをクリックします。

次に SAML IdP を登録します。「SAML シングルサインオン構成」の「新規」ボタンをクリックし、次の表の情報を入力していきます。詳細な設定手順については Salesforce のドキュメントをご参照ください。

項目名	設定値
名前	[任意の認証サービス名]
	https://sso.example.co.jp/openam
発行者	https://sso.example.co.jp/openam
ID プロバイダの証明書	「証明書のエクスポート」で作成した証明書をアップロード
証明書の署名要求	自己署名証明書を生成
署名要求メソッド	RSA-SHA256
アサーション復号化証明書	アサーション暗号化なし
SAML ID 種別	アサーションには、ユーザオブジェクトの統合 ID が含まれます
SAML ID の場所	ID は、Subject ステートメントの NameIdentifier 要素にあります

項目名	設定値
サービスプロバイダの起動要求バインド	HTTP リダイレクト
ID プロバイダのログイン URL	[SAML IdP をトップレルムに設定した場合] https://sso.example.co.jp/openam/SSORedirect/metaAlias/idp [SAML IdP をサブレルムに設定した場合] https://sso.example.co.jp/openam/SSORedirect/metaAlias/レルム名/idp
カスタムログアウト URL	Salesforce 側でログアウトを実行したあとに遷移する URL を指定します。 Salesforce のログアウトと同時に OpenAM からもログアウトしたい場合は OpenAM のログアウト URL を指定します。 例 : https://sso.example.co.jp/openam/UI/Logout?goto=https://domainname.my.salesforce.com/
カスタムエラー URL	無し (空のままとします)
シングルログアウトを有効にする	チェックしない
API 参照名	自動で入力されます
エンティティ ID	https://domainname.my.salesforce.com
ユーザプロビジョニングの有効化	チェックしない

設定が完了したら「保存」ボタンをクリックします。

「SAML シングルサインオン設定」の一覧に作成した設定が表示されます。その名前をクリックし、「メタデータのダウンロード」をクリックしてメタデータ (XML ファイル) を保存します。このメタデータは後述の手順で使用します。

使用する認証サービスの変更を行います。[設定] → [会社の設定] → [私のドメイン] と遷移し、表示された画面の下部にある [認証設定] の「編集」ボタンをクリックします。[認証サービス] 欄の「ログインフォーム」のチェックを外し、「(作成した認証サービス名)」にチェックを入れ、「保存」ボタンをクリックします。

3.2.4 OpenAM へ Salesforce(SAML SP) を登録

OpenAM に Salesforce を SAML SP として登録します。

1. OpenAM に管理者ユーザーでログインします。
2. OpenAM 管理コンソールで対象のレルムをクリックします。
3.「共通タスク」で「SAMLv2 プロバイダを作成」をクリックします。
4.「リモートサービスプロバイダを登録」をクリックします。
5.「メタデータファイルはどこに存在しますか?」は「ファイル」を選択します。
6.「アップロード」をクリックして、Salesforce からダウンロードしたメタデータをアップロードします。
7.「トラストサークル」で「既存のトラストサークルに追加します」をチェックし、「既存のトラストサークル」として”usrcot” を選択します。
8. 画面右上の「設定」ボタンをクリックして設定を保存します。
9. 確認画面が表示されるため「了解」をクリックします。
10. 上部メニューの「連携」タブをクリックします。
11.「トラストサークル設定」の「エンティティープロバイダ」に Salesforce が登録されていることを確認します。
12.「エンティティープロバイダ」の SAML IdP(OpenAM サーバーの URL) をクリックします。
13.「NameID 値マップ」を以下のように変更します。
    - 削除：「urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified=」
    - 追加：「urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified=uid」
    - 既に上記の「追加」の値が追加されている場合はこの手順を実施する必要はありません。
14. 画面右上の「保存」をクリックして設定を保存します。

以上で完了です。

3.2.5 シングルサインオンの動作確認

ここまでの設定を行うことで、Salesforce へシングルサインオンするための準備が整いました。実際に Salesforce へシングルサインオンして動作を確認します。以下の URL にアクセスします。

- https://domainname.my.salesforce.com/

OpenAM のログイン画面が表示されるはずです。OpenAM のユーザ名とパスワードを入力してログインします。認証に成功すると、自動的にリダイレクトされて Salesforce の画面が表示されます。

4 NameID の変更

OpenAM（SAML IdP）のアカウントと SAML SP のアカウントを紐付ける識別子を NameID と言います。本章では、NameID を変更する方法について説明します。

4.1 NameID として利用する属性を変更する

これまでの手順では、NameID として OpenAM のログインユーザー名（LDAP の uid 属性）を設定していましたが、これをメールアドレス（LDAP の mail 属性）などに変更することも可能です。

たとえば、Google Workspace のマルチドメイン機能を利用している場合は、NameID には Google Workspace のメールアドレスを指定する必要があります。ここでは、NameID をメールアドレスに変更する手順を説明します。なお、例では SAML SP が要求する NameID フォーマットが「urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified」であることを前提としています。

1. OpenAM に管理者ユーザーでログインします。
2. 上部メニューの「連携」タブを開きます。
3. 「トラストサークル設定」の「エンティティープロバイダ」の SAML IdP(OpenAM サーバーの URL) をクリックします。
4. 「NameID の書式」の「NameID 値マップ」を変更します。
   - 「現在の値」から「urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified=uid」を選択し「削除」ボタンをクリックします。
   - 「新しい値」に「urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified=mail」と入力し「追加」ボタンをクリックします。“mail” の部分には、OpenAM のユーザーデータストアでメールアドレス（Google Workspace のメールアドレス）が保存されている属性を指定します。
5. 画面右上部の「保存」ボタンをクリックします。

以上で完了です。

4.2 SAML SP 毎に異なる NameID を設定する

osstech-openam14-14.2.0-16 以降では SAML SP の設定にも NameID 値マップを設定することができるため、同一 NameID フォーマットで SAML SP 毎に異なる NameID を設定することが可能です。 SAML IdP と SP の両方の NameID 値マップの設定に同一 NameID

フォーマットが存在する場合、SAML SP の設定が利用されます。

ここでは、複数の SP (「SP1」と「SP2」と「その他の SP」) が同一の NameID フォーマット「urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified」を要求し、NameID として別々の LDAP 属性 (それぞれ「cn」「mail」「uid」) を利用する場合の設定方法を説明します。 前提として SAML SP の「SP1」「SP2」「SP3(=その他の SP)」が既に OpenAM に登録されているものとします。

1. OpenAM に管理者ユーザーでログインします。
2. 上部メニューの「連携」タブを開きます。
3. 「トラストサークル設定」の「エンティティープロバイダ」にある「SP1」のエンティティ ID をクリックします。
4. 「表明コンテンツ」タブ内の「NameID の書式」の「NameID 値マップ」に「urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified=cn」を追加し、「保存」を押下します。
5. 「戻る」を押下し、「3」と同様の手順で「SP2」の設定画面を開きます。
6. 「4」と同様にして「urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified=mail」を追加し、保存します。
7. 「戻る」を押下し、「3」と同様の手順で SAML IdP の設定画面を開きます。
8. 「表明コンテンツ」タブ内の「NameID の書式」の「NameID 値マップ」に「urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified」が存在する場合は値を選択し、「削除」を押下します。「urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified=uid」を追加し、保存します。

以上で完了です。

上記の設定を行った場合、生成される NameID の値は下記のようになります。ただし、LDAP 属性「cn」「mail」「uid」には値が格納されているものとします。

NameID 送付先	NameID の値
SP1	LDAP 属性の cn
SP2	LDAP 属性の mail
SP3(その他の SP)	LDAP 属性の uid

4.3 注意 1

「NameID 値マップ」を変更する前に既に SAML SP にシングルサインオンしたことがあるユーザーの場合、この「NameID 値マップ」の変更は反映されない場合があります。

たとえばユーザーが Google Workspace にシングルサインオンすると、ユーザーデータストアのユーザーエントリに以下の 2 つの属性が保存されます。

- sun-fm-saml2-nameid-info
- sun-fm-saml2-nameid-infokey

これらの属性には NameID に関する情報が保存されます。ユーザーエントリにこれらの属性が保存されていると、OpenAM は「NameID 値マップ」の設定を無視して、ユーザーエントリに保存されている値を NameID として利用します。そのため、既に Google Workspace にシングルサインオンしたことがあるユーザーの場合、この「NameID 値マップ」の変更は反映されません。

新しい「NameID 値マップ」の設定を反映させるためには、ユーザーエントリから上記の属性を削除し、OpenAM を再起動してください。

ユーザーエントリに属性を保存しない設定も可能です。詳細は「連携の持続性を無効にする」を参照してください。

4.4 注意 2

この設定を行った後に、Google Workspace を「OpenAM へ Google Workspace(SAML SP) を登録」の手順で登録すると、上記の手順で設定した NameID の設定が元に戻ってしまいます。そのため、「OpenAM へ Google Workspace(SAML SP) を登録」の手順を実施したあとには必ず上記の NameID の変更も実施してください。

osstech-openam14-14.0.0-54 より NameID の設定が元に戻らないよう修正されました。

5 属性情報の連携

SP に送られる SAML 認証応答メッセージには、NameID に加えてユーザーの属性情報を付加することも可能です。本章では、属性情報を付加する方法について説明します。

5.1 SAML 応答メッセージに属性情報を付加する

ここではユーザーのメールアドレス (LDAP の mail 属性) を「EmailAddress」という名前で付加する場合の設定手順について説明します。属性情報は SP ごとに設定することができます。

1. OpenAM に管理者ユーザーでログインします。
2. 上部メニューの「連携」タブを開きます。
3. 「トラストサークル設定」の「エンティティープロバイダ」から、認証応答メッセージに属性情報を付加する SP を選択してクリックします。
4. 「表明処理」タブを開きます。
5. 「属性マッパー」の「属性マップ」に付加する属性を追加します。形式は「SAML 認証応答メッセージ内のパラメーター名=属性名」です。
   - 「EmailAddress=mail」
6. 画面右上の「保存」ボタンを押下します。

以上で完了です。この設定を行うと、SAML 認証応答メッセージに以下のような形式で属性情報が付加されます。

```
<samlp:Response (省略)>
(省略)
  <saml:Assertion (省略)>
    (省略)
    <saml:NameID (省略)>xxxxx</saml:NameID>
    (省略)
    <saml:AttributeStatement>
      <saml:Attribute Name="EmailAddress">
        <saml:AttributeValue (省略)>taro@osstech.co.jp</saml:AttributeValue>
      </saml:Attribute>
    </saml:AttributeStatement>
  </saml:Assertion>
</samlp:Response>
```

6 ポリシーベースアクセス制御の設定

OpenAM には認証済みユーザーがどの SP にアクセス可能かを制御する認可の機能があります。この機能を使用することで、権限の無いユーザーに対して SP へのアクセスを拒否したり、一部の SP へのアクセスに対して追加の認証を求めるといった動作が可能です。OpenAM でポリシーを定義することで認可を行うため、「ポリシーベースアクセス制御」と呼びます。

本章では、ポリシーベースアクセス制御を利用する方法について説明します。

6.1 ポリシーベースアクセス制御の有効化

ポリシーベースアクセス制御を利用するためには、まず機能を有効化する必要があります。OpenAM 14 のバージョンにより設定方法が異なります。

6.1.1 osstech-openam14-14.5.0-0 以降の手順

osstech-openam14-14.5.0-0 以降のバージョンでは SP 毎にポリシーベースアクセス制御を有効化することができます。ここでは特定の SP に対してポリシーベースアクセス制御を有効にします。

全ての SP にポリシーベースアクセス制御を有効にする場合は IdP 側の設定を変更します。 手順 3 の SP を IdP に読み替えて設定を変更してください。

1. OpenAM に管理者ユーザーでログインします。
2. 上部メニューの「連携」タブを開きます。
3. 「トラストサークル設定」の「エンティティープロバイダ」から、アクセス制御の対象となる SP を選択してクリックします。
4. 「高度」タブを開きます。
5. 「ポリシー」の「ポリシーに基づくエンドポイントの保護」をチェックします。
6. 画面右上の「保存」ボタンを押下します。

以上で完了です。

6.1.2 osstech-openam14-14.2.0-16 以前の手順

osstech-openam14-14.2.0-16 以前のバージョンでは IdP の設定を追加します。全ての SP に対してポリシーベースアクセス制御が有効になります。

1. OpenAM に管理者ユーザーでログインします。
2. 上部メニューの「連携」タブを開きます。
3.「トラストサークル設定」の「エンティティープロバイダ」から、IdP を選択してクリックします。
4.「高度」タブを開きます。
5.「IDP Adapter」の「IDP アダプタクラス」に「jp.co.osstech.oam.saml2.plugins.PolicyCheckIDPAdapter」を入力します。
6. 画面右上の「保存」ボタンを押下します。

以上で完了です。

6.2 ポリシーの設定

ポリシーベースアクセス制御を利用するためには、IdP と SP の組み合わせに対する認可（ポリシー）の設定が必要です。 ここでは SAML 用ポリシーの作成手順を記載します。

ポリシーの各設定の詳細については、別紙「Policy Agent リファレンスマニュアル」をご参照ください。

6.2.1 リソースタイプの作成

ポリシーの作成にあたり、SAML 用のリソースタイプが必要となります。リソースタイプは以下の手順で作成することが可能です。

1. OpenAM に管理者ユーザーでログインします。
2. OpenAM 管理コンソールで対象のレルムをクリックします。
3. メニューの「認可」→「リソースタイプ」を開きます。
4.「新規リソースタイプ」または「リソースタイプの追加」をクリックします。
5. 必要事項を入力します。
   - 「名前」には任意の名称 (リソース名) を入力します。(SAMLv2 Endpoint 等)
   - 「パターンの指定」に以下の内容を入力し、「パターンの追加」をクリックします。
     - idpEntityID=*&spEntityID=*
   - 「アクションの指定」に以下の内容を入力し、「アクションの追加」をクリックします。
     - IssueAssertion
     - デフォルトの状態：許可
6.「作成」を押します。

以上で完了です。

6.2.2 ポリシーの作成

リソースタイプを作成した後はポリシーを作成します。ポリシーの作成にはポリシーセットも用意する必要があります。ポリシーセット及びポリシーは以下の手順で作成することが可能です。

1. OpenAM に管理者ユーザーでログインします。
2. OpenAM 管理コンソールで対象のレルムをクリックします。
3. メニューの「認可」→「ポリシーセット」を開きます。
4. 「新規プロバイダポリシーセット」または「プロバイダポリシーセットの追加」をクリックします。[*1]
5. 新規プロバイダポリシーセットの作成画面で必要事項を入力します。
   - 「id」には「SAML2ProviderService」を入力します。
   - 「名前」には任意の名称（ポリシーセット名）を入力します。（例：SAML2 Endpoint Protection Policy Set）
   - 「リソースタイプ」は、リソースタイプの作成で作成したリソースタイプ名を指定します。（例の場合、SAMLv2 Endpoint）
6. 「作成」をクリックしてポリシーセットを作成します。
7. 作成されたポリシーセットの画面が表示されたら「新しいポリシーの追加」をクリックします。
8. 必要事項を入力します。
   - 「名前」には任意の名称 (ポリシー名) を入力します。
   - 「リソースタイプ」は、リソースタイプの作成で作成したリソースタイプ名を指定します。（例の場合、SAMLv2 Endpoint）
   - 「リソース」ではまず「リソースパターンの選択」で「idpEntityID=*&spEntityID=*」を選択します。入力欄が表示されたら保護対象とする IdP と SP のエンティティ ID を入力し「追加」を押します。複数登録する場合は「Add Resource」から同様の手順で追加します。
     - 「idpEntityID」に OpenAM の SAML IdP のエンティティ ID を指定してください。
     - 「spEntityID」に OpenAM に登録した SAML SP のエンティティ ID を指定

[*1] 「新規ポリシーセット」や「ポリシーセットの追加」をクリックしないように注意してください。

してください。

- OpenAM 上に登録のある、各 SAML IdP / SP のエンティティ ID の値は、OpenAM 管理コンソールの上部「連携」メニューから確認可能です。

9. 「作成」を押します。
10. 作成されたポリシーの画面で「アクション」「対象」「条件」を設定します。これらの設定方法については、別紙「Policy Agent リファレンスマニュアル」を参照してください。

以上で完了です。

6.3 制限事項

ポリシーベースアクセス制御機能には以下の制限があります。制限に違反した場合、想定とは異なるアクセス制御となる可能性がありますのでご注意下さい。

- ポリシーの「リソース」にはワイルドカードを指定できません
- ポリシーの「条件」は単一の条件としてください

7 送信属性同意機能の設定

送信属性同意機能は IdP が SP にユーザーの属性情報を連携する前にユーザーに同意を求める機能です。この機能は osstech-openam14-14.5.0-0 以降で利用可能です。

7.1 設定手順

7.1.1 組織認証用鍵ペアの作成

送信属性同意機能では公開鍵暗号方式で暗号化した情報をローカルストレージに保存します。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「キーストアと鍵ペアの作成」で作成したキーストアファイルに鍵ペアを追加することを前提としています。

```
$ keytool -genkeypair \
-keyalg rsa \
-alias attribute-consent \
-dname "CN=sso.example.co.jp,OU=development,O=EXAMPLE,L=Shinagawa-ku,ST=Tokyo,
C=JP" \
-keypass xxxxxxxx \
-keystore mykeystore.jceks \
-storetype JCEKS \
-storepass changeit \
-validity 3650 \
-keysize 2048
```

OpenAM を冗長化構成 (サイト構成) で構築している場合は、1 号機で作成したキーストアファイルを 2 号機にコピーしてください。 (1 号機と 2 号機は同じファイルを使用します。)

7.1.2 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 に管理者ユーザーでログインします。
2. OpenAM 管理コンソールで対象のレルムをクリックします。
3. メニューの「認証」→「設定」を開きます。
4. 「セキュリティ」タブの「組織認証の証明書のエイリアス」に `attribute-consent` を設定します。

5.「変更の保存」をクリックします。

以上で完了です。

7.1.3 送信属性同意機能の有効化

送信属性同意機能はリモート SP 毎に有効・無効を設定することができます（デフォルトは無効）。対象の SP に対して以下の手順で有効化します。

1. OpenAM に管理者ユーザーでログインします。
2. 上部メニューの「連携」タブを開きます。
3.「トラストサークル設定」の「エンティティープロバイダ」から、対象の SP を選択してクリックします。
4.「表明処理」タブを開きます。
5.「送信する属性の同意」の「属性送信の同意画面を表示する」にチェックし、保存します。

以上で完了です。

7.1.4 同意が必要な属性の設定

送信属性同意機能で同意の対象とする属性（アサーション内の属性）を設定します。この設定は同意が必要かどうかの判定だけではなく、同意画面の表示にも利用されるため、表示内容を含む以下のフォーマットで指定します。

- アサーション内の属性名
- アサーション内の属性名|表示名
- アサーション内の属性名|言語|表示名

ここでは「saml-uid」と「saml-mail」を同意の対象とする場合の設定方法を説明します。「saml-uid」は全ての言語に対して「ID」と表示します。「saml-mail」は日本語の場合は「メールアドレス」と表示し、それ以外の言語では「Email address」と表示します。なお、送信に必要な「属性マップ」の設定は既に実施されているものとします。

1. OpenAM に管理者ユーザーでログインします。
2. 上部メニューの「連携」タブを開きます。
3.「トラストサークル設定」の「エンティティープロバイダ」から、IdP を選択してクリックします。

4.「表明処理」タブを開きます。
5.「送信する属性の同意」の「同意が必要な属性」に追加し、保存します。
    - saml-uid|ID
    - saml-mail|Email address
    - saml-mail|ja|メールアドレス

以上で完了です。

7.2 動作確認

実際に SP の URL にアクセスして動作を確認します。 SP にアクセスすると OpenAM のログイン画面が表示されます。そして、認証に成功すると次のような同意画面が表示されます。

図 1　同意画面

同意する場合は「次回送信時にもう一度確認する。」、「このサービスに送信する属性が変更された場合、もう一度確認する。」、「連携している全てのサービスに対して全ての属性の

送信を許可し、今後この画面を表示しない。」のいずれかを選択して「同意」ボタンをクリックします。同意後、属性情報を含むアサーションが応答され、SP にアクセスできるようになります。

「拒否」ボタンをクリックした場合は次の拒否画面が表示されます。この場合、SAML のフローは OpenAM で中断されます。SP にはアクセスできません。

図 2　拒否画面

同意の際に「このサービスに送信する属性が変更された場合、もう一度確認する。」を選択した場合、OpenAM の設定変更等で SP に送信する属性が追加されない限りは同一 SP へのアクセスでは同意を求められません。また、「連携している全てのサービスに対して全ての属性の送信を許可し、今後この画面を表示しない。」を選択した場合、他の SP へのアクセスにおいても同意が求められることはありません。もう一度、同意画面を表示したい場合は、ログイン画面の「属性送信の同意画面を表示する」にチェックを入れてログインを実施します。すると、ログイン後に同意画面が再度表示されます。

図 3　ログイン画面

7.3 同意画面のレイアウト

同意画面のレイアウトについて説明します。

7.3.1 表示項目

同意画面のいくつかの表示項目はメタデータの情報を利用します。メタデータを変更することでサービス名の変更やロゴの追加が可能です。

図 4　同意画面の表示要素

No	表示項目	対応するメタデータ
1	サービス名	<mdui:DisplayName>, <AttributeConsumingService> の<ServiceName>, SP のエンティティ ID
2	ロゴ	<mdui:Logo>
3	サービスの詳細	<mdui:Description>, <AttributeConsumingService> の<ServiceDescription>

サービス名等を含むメタデータの例を以下に示します。

```
<EntityDescriptor entityID="https://roles.example.co.jp:443/openam" xmlns="urn:oa
sis:names:tc:SAML:2.0:metadata">
  <SPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protoc
ol">
    <Extensions>
      <mdui:UIInfo xmlns:mdui="urn:oasis:names:tc:SAML:metadata:ui">
        <mdui:DisplayName xml:lang="ja">サービス名</mdui:DisplayName>
        <mdui:DisplayName xml:lang="en">Service Name</mdui:DisplayName>
        <mdui:Description xml:lang="ja">サービスの詳細</mdui:Description>
        <mdui:Description xml:lang="en">Service Description</mdui:Description>
        <mdui:Logo height="40" width="240">https://sp.example.co.jp/logo.png</mdu
i:Logo>
      </mdui:UIInfo>
    </Extensions>
    ... 省略...
   </SPSSODescriptor>
</EntityDescriptor>
```

7.3.2 画面カスタマイズ

同意画面には以下の HTML ファイルが利用されます。このファイルを編集することで画面のカスタマイズが可能です。

```
/opt/osstech/share/tomcat/webapps/openam/XUI/templates/user/SAML2ConsentTemplate.
html
```

このファイルは Handlebars テンプレートエンジンを利用しており、以下の変数がメタデータから展開されます。複数の記載があるものは先に記載された要素が優先されます。

変数	対応するメタデータ
spDisplayName	<mdui:DisplayName>, <AttributeConsumingService> の<ServiceName>, SP のエンティティ ID
spLogoURL	<mdui:Logo>
spDescription	<mdui:Description>, <AttributeConsumingService> の<ServiceDescription>
spInformationURL	<mdui:InformationURL>

変数	対応するメタデータ
spPrivacyStatementURL	<mdui:PrivacyStatementURL>
spOrganizationDisplayName	<Organization> の<OrganizationDisplayName>

7.4 監査ログ

送信属性同意機能の監査ログについて説明します。

7.4.1 有効化

送信属性同意機能の監査ログは以下の手順で有効化します。

1. OpenAM に管理者ユーザーでログインします。
2. 上部メニューの「設定」→「グローバルサービス」を開きます。
3.「Audit Logging」をクリックします。
4.「Global CSV Handler」をクリックします。
5.「General Handler Configuration」の「Topics」で「SAML2 Attribute Consent」をチェックします。
6.「Save」ボタンをクリックします。

以上で完了です。

7.4.2 出力内容

監査ログを有効にした場合は、/var/opt/osstech/lib/tomcat/data/openam/openam/log/saml2consent.csv に同意内容が出力されます。

```
"2022-12-12T06:57:51.301Z","AM-SAML2-CONSENT-AGREED",
"a37325a0-3c99-4ea2-bc1d-6f75bb4f661e-1354",
"id=test0001,ou=user,dc=openam,dc=osstech,dc=co,dc=jp",
"/","https://sso.example.co.jp:443/openam",
"https://sp.example.com/sp","[""saml-uid"",""saml-mail""]","ASK_IF_CHANGE"
```

主な項目と格納内容は以下の通りです。

【項目】		【出力例】
timestamp	監査ログ出力日時	2020-06-11T12:58:53.457+09:00

【項目】		【出力例】
eventName[*2]	イベント名	AM-SAML2-CONSENT-AGREED
userId	ユーザーの ID	id=test0001,ou=user,dc=openam,dc=osstech,dc=co,dc=jp
realm	レルム	/
idp	IdP のエンティティ ID	https://sso.example.co.jp:443/openam
sp	SP のエンティティ ID	https://sp.example.com/sp
attributes	同意した属性	[“saml-uid”.”saml-mail”]
consentType[*3]	同意の種類	ASK_AGAIN

7.5 その他

7.5.1 エンドポイント

送信属性同意機能では以下のエンドポイントを利用します。 Apache 等でアクセス制限を行っている場合は、ユーザーが各エンドポイントを利用できるように設定を変更してください。

- /openam/json/saml2/metadata/uiinfo
- /openam/json/saml2/consent/*

[*2] 同意した場合 AM-SAML2-CONSENT-AGREED、拒否した場合 AM-SAML2-CONSENT-REJECTED となる

[*3] 同意の際に「次回送信時にもう一度確認する。」を選択した場合 ASK_AGAIN、「このサービスに送信する属性が変更された場合、もう一度確認する。」は ASK_IF_CHANGE、「連携している全てのサービスに対して全ての属性の送信を許可し、今後この画面を表示しない。」では ASK_NEVER となる。また、拒否した場合は REJECTED となる。

8 その他の SAML 設定

8.1 リモート SP のデフォルト設定を定義する

OpenAM (SAML IdP) に SAML SP を登録する際に設定される初期値を予め定義しておくことが可能です。この機能は osstech-openam14-14.5.0-0 以上で利用可能です。

1. OpenAM に管理者ユーザーでログインします。
2. SAMLv2 リモート SP 初期値設定サービスの設定画面を開きます。
   - レルムに関わらず同じ初期値を設定する場合
     - 上部メニューの「設定」タブから「グローバルサービス」を開きます。
     - 「SAMLv2 リモート SP 初期値設定」を開きます。
   - レルムごとに初期値を設定する場合
     - 対象のレルムを開きます。
     - 左のサイドメニューの「サービス」を開きます。
     - 「サービスの追加」を押下します。
     - サービスタイプに「SAMLv2 リモート SP 初期値設定」を選択し、「作成」を押下します。

   ※ SAML SP を作成するレルムに 「SAMLv2 リモート SP 初期値設定サービス」が存在しない場合には、グローバルサービスの 「SAMLv2 リモート SP 初期値設定」 が参照されます。
3. 各項目に初期値を設定します。設定可能な項目は下記の 3 つです。
   - **表明の暗号化**
     - SAML アサーションの暗号化有無を設定します。
     - 対応する SAML SP 設定：「表明コンテンツ > 暗号化> 表明」
   - **属性送信の同意画面を表示する**
     - 属性送信同意画面の表示有無を設定します。
     - 対応する SAML SP 設定：「表明処理 > 送信する属性の同意 > 属性送信の同意画面を表示する」
   - **ポリシーに基づくエンドポイントの保護**
     - OpenAM のポリシー機能による、アクセス制御の実施有無を設定します。
       * 設定にあたり、別途 OpenAM の SAML エンドポイントポリシーの定義が必要です。ポリシー設定方法は「ポリシーの設定」を参照ください。
     - 対応する SAML SP 設定：「高度 > ポリシー> ポリシーに基づくエンドポイ

ントの保護」

4.「保存」または「設定の保存」を押下します。

定義した初期値設定を SAML SP の設定に反映する方法は SP の登録方法によって異なります。

- 共通タスクからの登録 (レルム > SAMLv2 プロバイダを作成> リモートサービスプロバイダを登録)
  - プロバイダ登録画面で「初期値の継承」を有効にして SAML SP を登録します。
- 連携からの登録 (連携タブ > エンティティーのインポート)
  - 拡張メタデータを指定せずに SAML SP を登録します。
- `ssoadm` コマンドからの登録
  - import-entity の `-x(--extended-data-file)` オプションを指定せずに SAML SP を登録します。
- `ssoadm.jsp` からの登録 (import-entity)
  -「Extended entity configuration to be imported:」を空欄にしたまま SAML SP を登録します。

8.2 連携の持続性を無効にする

ユーザーエントリに sun-fm-saml2-nameid-info 及び sun-fm-saml2-nameid-infokey 属性を保存しないようにするには、下記の手順で設定を変更します。

1. OpenAM に管理者ユーザーでログインします。
2. 上部メニューの「連携」タブを開きます。
3.「トラストサークル設定」の「エンティティープロバイダ」から、対象の SP を選択してクリックします。
4.「NameID の書式」の「Disable NameID persistence」にチェックします。
5. 画面右上の「保存」ボタンをクリックします。

以上で完了です。既に属性が保存されている場合は削除する必要があります。

8.3 NameID や属性で連携する値について

SP に送る SAML 認証応答メッセージの NameID や属性情報は、データストア (LDAP サーバー) の属性値の他に OpenAM のセッションプロパティの値やスクリプトで設定した値を送信することができます。セッションプロパティの値を送信する機能は osstech-openam14-14.0.0-54

以降、スクリプトで設定した値を送信する機能は osstech-openam14-14.2.0-16 以降のバージョンで利用可能です。 NameID や属性情報は「NameID フォーマット (SAML 認証応答メッセージ内のパラメーター名)=属性名」で設定しますが、この属性名が「LDAP 属性 (ユーザープロファイル属性)」の名前であれば LDAP の属性値、「セッションプロパティ」のキーであればセッションプロパティの値、「スクリプト」で設定したマッピングのキーであれば対応する値が SP に送信されます。 OpenAM が NameID と属性情報の値を決定する際の優先順位は「LDAP 属性」>「セッションプロパティ」>「スクリプト」で、同じ名前の定義が存在した場合はより優先順位の高いものの値が反映されます。

弊社では追加プラグインパッケージにて「ユーザープロファイル属性 (LDAP の属性) + “固定の文字列”」や「ユーザープロファイル属性 (LDAP 属性) の SHA-1 ハッシュ値」をセッションプロパティにセットする機能を用意しています。この機能を利用することで加工した値を SAML の NameID や 属性に含めて SP に送信することが可能です。 プラグインについては別ドキュメント (認証 POST プロセスクラスリファレンスマニュアル) を参照ください。

スクリプトの設定方法については「属性定義用スクリプトを利用する」を参照ください。

8.4 属性定義用スクリプトを利用する

「属性定義用スクリプト」を利用して値の生成を行うことで、任意の値を NameID や連携する属性情報の値として SAML SP に送ることができます。

事前準備としてスクリプトで設定する属性名を決め、NameID 値マップと属性マップに設定しておきます。設定した属性名はこの後の手順「5」でスクリプトを作成する際に使用します。属性名は「NameID や属性で連携する値について」に記載されている優先順位を考慮して決める必要があります。

- NameID 値の生成をスクリプトで行う場合
  - 「NameID として利用する属性を変更する」の手順で「NameID フォーマット=属性名」の属性名に任意の文字列を設定します。
- 属性情報の属性値の生成をスクリプトで行う場合
  - 「SAML 応答メッセージに属性情報を付加する」の手順で「SAML 認証応答メッセージ内のパラメーター名=属性名」の属性名に任意の文字列を設定します。

スクリプトは下記の手順で作成します。

1. OpenAM に管理者ユーザーでログインします。
2. 対象のレルムを選択します。
3. 左のサイドメニューの「スクリプト」を開き、「スクリプト一覧」の画面で「新規ス

クリプト」ボタンを押下します。

4. 「名前」に任意の名前を入力し、「スクリプトタイプ」に「SAML Attribute Resolution」を設定して「作成」を押下します。
5. 「スクリプト」のテキストボックスにスクリプトを入力します。スクリプトの記述方法は「属性定義用スクリプトを作成する」を参照ください。
6. 「設定の保存」を押下します。
7. 上部メニューの「連携」タブを開きます。
8. 「トラストサークル設定」の「エンティティープロバイダ」にある IdP のエンティティ ID をクリックします。
9. 「表明処理」の「属性定義用スクリプト」に「4」で作成したスクリプトを設定します。
10. 「保存」を押下します。

以上で完了です。 スクリプトで LDAP 属性を利用する場合は、別途データストアの「LDAP ユーザーオブジェクトクラス」および「LDAP ユーザー属性」に利用する属性のオブジェクトクラスと属性名を追加する必要があります。

8.4.1 属性定義用スクリプトを作成する

スクリプトでの様々な条件分岐や柔軟な値の生成などを実現するため、属性定義用スクリプトではスクリプト内で利用可能な定数が予め 6 つ定義されています。

定義名	主な用途
logger	`Federation` デバッグログへの出力
httpClient	HTTP Client API
identity	認証済みユーザーの LDAP 属性取得
hostEntityID	IdP のエンティティ ID による条件分岐
remoteEntityID	SP のエンティティ ID による条件分岐
attributes	属性名と対応する値のエントリの追加

以下に NameID と属性情報の値を設定するスクリプトの作成例と設定例を記載します。

ここでは、スクリプトを使って NameID と属性情報の値を設定するものとします。「属性定義用スクリプトを利用する」の事前準備の手順でそれぞれのマップを設定します。

- IdP の NameID 値マップに「urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified=nameid」と設定します。
- IdP の属性マップに「mail=mail」「mobile=script-mobile」「cookie-name=cookie-name」と設定します。

スクリプトで「nameid」「mail」「script-mobile」「cookie-name」を下記の表に従って設定します。「nameid」「script-mobile」「cookie-name」と同一の名前を持つユーザープロファイル属性、「nameid」「mail」「script-mobile」「cookie-name」と同一の名前を持つセッションプロパティは存在しないものとします。

設定名	値
nameid	SP のエンティティ ID が `https://sp.example.co.jp/emp` の場合は LDAP 属性の `employeeNumber`、その他の SP の場合は `uid` の値
mail	LDAP 属性の `mail` が存在する場合は `mail` の値、存在しない場合は `employeeNumber` の値と`@example.co.jp` を結合した値
script-mobile	ハイフンありまたはなしの形式で格納された携帯電話番号をハイフンありの形式に統一した値
cookie-name	OpenAM の API を利用して取得した OpenAM のセッション Cookie 名

上記の表の定義に従って作成したスクリプトは下記のようになります。(作成例は JavaScript で記述しています。)

```
// nameid
setNameID();
// mail
setMail();
// script-mobile
setMobilePhoneNumber();
// cookie-name
setCookieName();

function setNameID() {
    var attributeName;
    if (remoteEntityID === "https://sp.example.co.jp/emp") {
        attributeName = "employeeNumber";
    } else {
        attributeName = "uid";
    }
    // ユーザープロファイル属性の uid または employeeNumber から属性値を取得
    var nameIDValue = identity.getAttribute(attributeName);
    // nameid として格納
    putEntry("nameid", nameIDValue);
}

function setMail() {
```

```
    /*
    ユーザープロファイル属性の mail が存在する場合、
    スクリプトで何らかの値を mail に設定しても
    ユーザープロファイル属性の mail の値が利用される

    ユーザープロファイル属性の mail が存在しない場合に
    マッピングする値を下記スクリプトで生成・設定する
    */

    // ユーザープロファイル属性の employeeNumber から属性値を取得
    var empNum = identity.getAttribute("employeeNumber");
    var mail = empNum.isEmpty() ?
            null : empNum.iterator().next() + "@example.co.jp";
    // mail として格納
    putEntry("mail", mail);
}

function setMobilePhoneNumber() {
    // ユーザープロファイル属性の mobile から属性値を取得
    var mobile = identity.getAttribute("mobile");
    if (!mobile.isEmpty()) {
        mobile = mobile.iterator().next();
        if (/^0[789]0\d{4}\d{4}$/.test(mobile)) {
            var m = [];
            m.push(mobile.substring(0,3));
            m.push(mobile.substring(3,7));
            m.push(mobile.substring(7));
            mobile = m.join("-");
        }
        mobile = /^0[789]0-\d{4}-\d{4}$/.test(mobile) ? mobile : null;
    }
    // script-mobile として格納
    putEntry("script-mobile", mobile);
}

function setCookieName() {
    // OpenAM のサーバー情報を API で取得
    var response = httpClient.get(
            "https://sso.example.co.jp/openam/json/serverinfo/*", {
        cookies: [],
        headers: []
    });
    // レスポンスからエンティティを取得
    var serverInfo = response.getEntity();
```

```
    var json = JSON.parse(serverInfo);
    // セッションクッキー名を取得
    var cookieName = json ? json.cookieName : null;
    // cookie-name として格納
    putEntry("cookie-name", cookieName);
}

function putEntry(name, values) {
    if (!name || !values
            || values instanceof java.util.HashSet && values.isEmpty()
            || values instanceof Array && !values.length) {
        // 属性値が空の場合は attributes に格納しない
        // 追加しないエントリのキーをデバッグログに出力（エラー）
        logger.error("Key name: " + name
                + " could not be added to the result map.");
        return;
    }

    if (typeof values === "string" || values instanceof String) {
        // 文字列型の属性値 -> Set に変換
        var set = new java.util.HashSet();
        set.add(values);
        values = set;
    } else if (values instanceof Array) {
        // Array オブジェクトの属性値 -> Set に変換
        values = new java.util.HashSet(values);
    }
    if (logger.messageEnabled()) {
        // 追加するエントリのキーと値をデバッグログに出力（メッセージ）
        logger.message("Key name: " + name + ", values: " + values.toString()
                + " has been added to the result map.");
    }

    // attributes にエントリを追加
    // put メソッドの第一引数には文字列型の属性名、第二引数には Set 型の属性値を渡す
    attributes.put(name, values);
}
```

上記の設定例のとおりに設定した場合の SAML レスポンス (一部省略) は下記のようになります。

- SP のエンティティ ID が https://sp.example.co.jp/emp、認証したユーザーの LDAP 属性が「uid=user0001」「employeeNumber=osstech0001」「mail=0001@example.co.jp」「mobile=08012345678」のとき

```
<samlp:Response ...>
  ...
  <saml:Assertion ...>
    ...
    <saml:Subject>
      <saml:NameID
          Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified"
          NameQualifier="https://sso.example.co.jp/openam"
          SPNameQualifier="https://sp.example.co.jp/emp"
          >osstech0001</saml:NameID>
    ...
    </saml:Subject>
    ...
    <saml:AttributeStatement>
      <saml:Attribute Name="mobile">
        <saml:AttributeValue ...>080-1234-5678</saml:AttributeValue>
      </saml:Attribute>
      <saml:Attribute Name="cookie-name">
        <saml:AttributeValue ...>iPlanetDirectoryPro</saml:AttributeValue>
      </saml:Attribute>
      <saml:Attribute Name="mail">
        <saml:AttributeValue ...>0001@example.co.jp</saml:AttributeValue>
      </saml:Attribute>
    </saml:AttributeStatement>
  </saml:Assertion>
</samlp:Response>
```

- SP のエンティティ ID が https://sp.example.co.jp/sp、認証したユーザーの LDAP 属性が「uid=user0001」「employeeNumber=osstech0001」「mobile=080-1234-5678」のとき

```
<samlp:Response ...>
  ...
  <saml:Assertion ...>
    ...
    <saml:Subject>
      <saml:NameID
          Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified"
          NameQualifier="https://sso.example.co.jp/openam"
          SPNameQualifier="https://sp.example.co.jp/sp"
          >user0001</saml:NameID>
      ...
```

```
    </saml:Subject>
    ...
    <saml:AttributeStatement>
      <saml:Attribute Name="mobile">
        <saml:AttributeValue ...>080-1234-5678</saml:AttributeValue>
      </saml:Attribute>
      <saml:Attribute Name="cookie-name">
        <saml:AttributeValue ...>iPlanetDirectoryPro</saml:AttributeValue>
      </saml:Attribute>
      <saml:Attribute Name="mail">
        <saml:AttributeValue ...>osstech0001@example.co.jp
        </saml:AttributeValue>
      </saml:Attribute>
    </saml:AttributeStatement>
  </saml:Assertion>
</samlp:Response>
```

スクリプト実行時、作成したスクリプトに問題がある場合は Federation デバッグログに下記のようなエラーが出力されます。エラーが出力された場合は再度スクリプト編集画面を開き、修正します。

```
ERROR: IDPSSOUtil.getAttributesMapFromScript: Failed to execute the script.
```

8.5 SAML2 メタデータの自動更新を設定する

日次でメタデータの URL を参照してローカルディレクトリ内に保存し、その記載内容に沿うように SAML エンティティを自動更新できる機能があります。自動更新の内容としては、メタデータに記載されているエンティティの新規追加又は設定変更と、メタデータから記載が抜けたエンティティの削除を行います。
この機能は osstech-openam14-14.5.0-0 以上で利用可能です。

本機能を利用するための設定方法を以下に記します。

1. OpenAM に管理者ユーザーでログインします。
2. SAML2 メタデータの自動更新サービスの設定画面を開きます。
   - レルムに関わらず同じ初期値を設定する場合
     - 上部メニューの「設定」タブから「グローバルサービス」を開きます。
     -「SAML2 メタデータの自動更新」を開きます。
   - レルムごとに初期値を設定する場合

- 対象のレルムを開きます。
- 左のサイドメニューの「サービス」を開きます。
- 「サービスの追加」を押下します。
- サービスタイプに「SAML2 メタデータの自動更新」を選択し、「作成」を押下します。

※ 該当のレルムに SAML2 メタデータの自動更新サービスが存在しない場合にはグローバルサービスの設定が参照されます。

3. 各項目に初期値を設定します。設定可能な項目は下記になります。

項目名	設定値
メタデータの URL とバックアップ先のマッピング	[キー] メタデータを取得する URL [値] 対応するバックアップファイルの保存先 ※ add ボタンで追加します。
実行時刻	1 日 1 回更新処理をおこなう時刻を hh:mm:ss の形式で指定します。
実行するサーバーの URL	更新処理をおこなうサーバーです。冗長構成の場合に指定します。
対象とするロール	更新対象とするエンティティのロールです。(すべて,IdP のみ,SP のみ, なし)
対象とするエンティティ	対象とするエンティティ ID を複数指定可能です。「対象とするロール」に含まれない場合も更新対象となります。
除外するエンティティ	除外とするエンティティ ID を複数指定可能です。「対象とするロール」に含まれている場合も除外対象となります。
エンティティの新規登録を許可する	メタデータ内の対象エンティティを連携先に自動で追加します。
有効期限をチェックする	メタデータの有効期限をチェックし、期限を過ぎていた場合は更新しません。
署名を検証する	メタデータの署名を検証し、検証出来なかった場合は更新しません。 ※署名用の鍵を OpenAM のキーストアにインポートしておく必要があります。

4. 「保存」または「変更の保存」を押下します。

設定は上記で完了です。更新処理は実行時刻に開始します。実行時刻を待たずに即座に開

始したい場合は下記の URL にアクセスすることで可能です。

- 最上位レルムの場合：

```
https://openam01.example.co.jp/openam/saml2/jsp/reload.jsp
```

- usr レルムの場合：

```
https://openam01.example.co.jp/openam/saml2/jsp/reload.jsp?realm=/usr
```

9 改版履歴

- 2019 年 12 月 11 日 リビジョン 1.0
  - 初版作成
- 2020 年 10 月 22 日 リビジョン 1.1
  - 初期設定ガイドで構築した場合の URL に変更
  - 「エンドポイント URL の調整」を追記
  - 「注意 2」に修正されたバージョンがあることを追記
  - G Suite の表記を新しい Google Workspace に変更
- 2021 年 03 月 10 日 リビジョン 1.2
  - 「NameID や属性で連携する値について」を追記
- 2022 年 07 月 14 日 リビジョン 1.3
  - 表紙の社名を OSSTech 株式会社に変更
- 2022 年 10 月 28 日 リビジョン 1.4
  - 「ベース URL」について追記
  - 「SAML SP 毎に異なる NameID を設定する」を追記
  - 「属性定義用スクリプトを利用する」を追記
- 2022 年 12 月 5 日 リビジョン 1.5
  - 「リモート SP のデフォルト設定を定義する」を追記
  - 「SAML 用 OpenAM ポリシー作成手順」を追記
- 2022 年 12 月 16 日 リビジョン 1.6
  - 「NameID の変更」及び「属性情報の連携」を「その他の SAML 設定」から移動して新たな章に変更
  - 「送信属性同意機能の設定」を追記
- 2022 年 12 月 20 日 リビジョン 1.7
  - 「SAML2 メタデータの自動更新を設定する」を追記
- 2022 年 12 月 22 日 リビジョン 1.8
  - 「ポリシーベースアクセス制御の設定」を追記
- 2023 年 3 月 8 日 リビジョン 1.9
  - 「SAML2 メタデータの自動更新を設定する」に補足追加
- 2023 年 4 月 21 日 リビジョン 2.0
  - 誤字の修正