２００９年9月２日

SP 構築・運用手順書
(Ver2.0)

(軽井沢セミナー用)

1. 概要

本書は SP の構築手順、および運用方法を説明したものです。

1-1. SP の機能

まず、SP の動作について簡単に説明します。

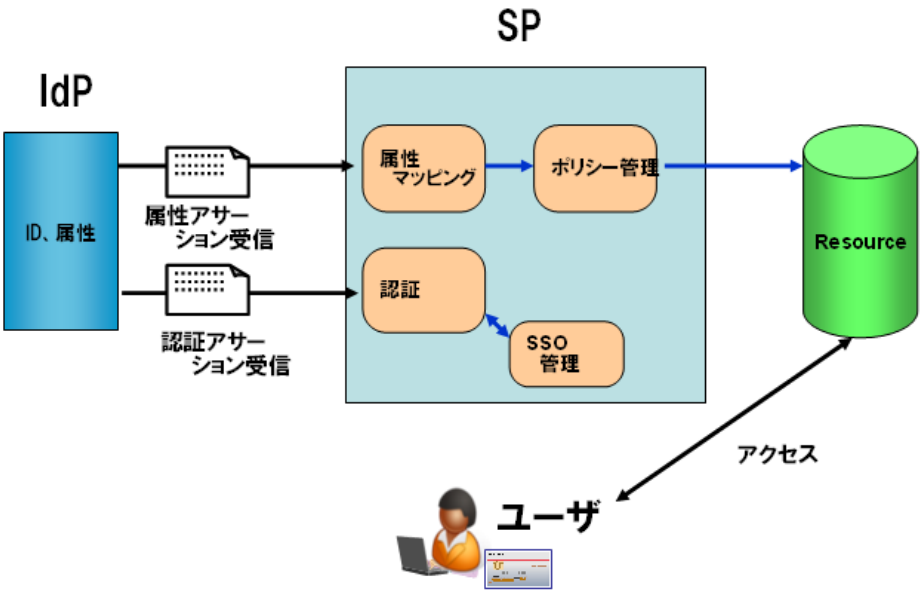

図１　SP の機能ブロック

図１　SP の機能ブロックは、SP の機能を単純化したブロックで示しています。SP は IdP と連携して、以下の２つの動作を行います。

- ユーザの認証を IdP に要求する
- ユーザの属性を安全に IdP から受信して、アプリケーションに渡す

■ 認証要求

ユーザが SP にアクセスすると、SP は IdP にリダイレクトを行い、IdP にユーザの認証を要求します。IdP はこれを受けてユーザの認証を行います。認証方式としては、ID／パスワード認証や、クライアント証明書による認証等の認証方式が設定可能です。

ユーザの認証が行われると、SP は IdP から認証アサーションを受信してユーザを認証したことを確認します。ただし、ここで受信するのはユーザを認証したという事実のみで、そのユーザが誰かという情報は渡されません。

■ 属性の安全な受信

SP は IdP に必要とする属性を要求します。IdP は要求された属性を属性アサーションに入れて SP に送信します。SP はこれを受信して、下記を行います。

- 属性アサーションから属性を取得して、属性の名称を IdP 間で利用する名称から、アプリケーションに渡すための名称に変換する。(図１の属性マッピング機能)
- アプリケーションへのアクセスを許可して良いかどうか、ポリシーを確認します。問題がない場合は、属性をアプリケーションに渡します。(図１のポリシー管理機能)
- アプリケーションでは属性を受け、この属性によりユーザに対する認可判断を行います。

以下の章では、SP の構築手順を示すとともに、上記機能の設定方法、および、これらの機能を用いて SP を運用するための方法について説明します。

1-2. 構築方式について

本書では、貴学にてサーバに OS から shibboleth(SP)までインストール・設定を行い、構築する方式について説明します。

2. インストール

2-1.　貴学にて SP をインストールする場合の構築手順

2-1-1. shibboleth (SP version2.0)の動作要件

- Apache HTTP Server 2.2 以上
- Java 5 以上 (Plone を使用する場合には必要)
  （ただし、CentOS に付属する Gnu Java は利用できません。 Sun の Java を利用してください。）

2-1-2. OS をインストールする

① OS での設定

- OS：CentOS 5.3
  インストーラでインストールするもの。
  Web サーバー (HTTP のみ)
  unixODBC
  その他のパッケージがある場合は必要に応じてインストールしてください。
  ただし、Java 開発は後の手順で別にインストールします。

- ネットワーク設定
  環境に合わせ、ホスト名・ネットワーク・セキュリティを設定して下さい。
  SP では shibd サービスが通信を行います。

② DNS へ登録する

新しいホスト名と IP アドレスを DNS に登録してください。

③ 時刻同期を設定する

ntp サービスを用い、貴学環境の ntp サーバと時刻同期をしてください。

※ Shibboleth では、通信するサーバ間の時刻のずれが約５分を越えるとエラーになります。

2-1-3. shibboleth のインストール

① shibboleth-SP 関連のインストールファイルのダウンロード

http://shibboleth.internet2.edu/downloads.html から shibboleth-SP 関連のインストールファイルをダウンロードします。以下は 2009/7/30 時点での最新版です。OS が 64bit 版の場合、64bit 対応のインストールファイルをご使用ください。

【対象ファイル】

log4shib-1.0-2.i386.rpm	xerces-c-3.0.1-1.i386.rpm
log4shib-debuginfo-1.0-2.i386.rpm	xerces-c-debuginfo-3.0.1-1.i386.rpm
log4shib-devel-1.0-2.i386.rpm	xerces-c-devel-3.0.1-1.i386.rpm
log4shib-doc-1.0-2.i386.rpm	xerces-c-doc-3.0.1-1.i386.rpm
opensaml-2.2-1.i386.rpm	xml-security-c-1.5.0-1.i386.rpm
opensaml-debuginfo-2.2-1.i386.rpm	xml-security-c-debuginfo-1.5.0-1.i386.rpm
opensaml-devel-2.2-1.i386.rpm	xml-security-c-devel-1.5.0-1.i386.rpm
opensaml-docs-2.2-1.i386.rpm	xmltooling-1.2-1.i386.rpm
shibboleth-2.2-3.i386.rpm	xmltooling-debuginfo-1.2-1.i386.rpm
shibboleth-debuginfo-2.2-3.i386.rpm	xmltooling-devel-1.2-1.i386.rpm
shibboleth-devel-2.2-3.i386.rpm	xmltooling-docs-1.2-1.i386.rpm
shibboleth-docs-2.2-3.i386.rpm	

② インストール

上記の shibboleth-SP 関連のファイルをインストールします。

```
# rpm -ivh log4shib-1.0.3-1.1.i386.rpm ¥
      xerces-c-3.0.1-1.i386.rpm ¥
      xml-security-c-1.5.1-3.2.i386.rpm ¥
      xmltooling-1.2.2-1.i386.rpm ¥
      opensaml-2.2.1-1.i386.rpm ¥
      shibboleth-2.2.1-2.i386.rpm
```

※ 依存関係上、上記の順番でインストールする必要があります。

＜セミナー注： セミナーでは６４ビット版を利用します。＞

その他のパッケージは必要に応じてインストールしてください。

unixODBC がインストールされていないと依存関係チェック時にエラーが表示されます。

RedHat 系でパッケージをインストールする場合は、以下のコマンドを実行してください。

```
# yum install unixODBC
```

その他の情報は、以下のサイトから入手してください。

http://www.unixodbc.org/

＜セミナー注： セミナーでは実習時に配布します。＞

③ httpd 設定

/etc/httpd/conf.d/ssl.conf にて、 ServerName を設定します。

```
ServerName upkishibSP.nii.ac.jp:443    ← ホスト名を設定
```

④ shibd 自動起動設定

shibd を OS 起動時に自動起動するには、以下のコマンドで設定します。

(オプションは マイナス ‘ー‘ が２つ必要です)

```
# chkconfig --add shibd
# chikconfig --level 345 shibd on
```

2-1-4. サービス起動・停止方法

- httpd の起動方法

  service httpd start

- shibd の起動方法

  service shibd stop

- httpd の停止方法

  service httpd stop

- shibd の停止方法

  service shibd stop

3. 運用・設定・カスタマイズ

3-1. 接続までに必要なセッティング

3-1-1. shibboleth の設定

デフォルトでは shibboleth は /etc/shibboleth ディレクトリにインストール
されます。変更する各設定ファイルも 同ディレクトリ配下にあります。
また、ログファイルは /var/log/shibboleth ディレクトリに出力されます。

・ shibboleth2.xml ファイル

/etc/shibboleth/shibboleth2.xml ファイルを以下の様に変更します。

※「<Host name="sp.example.org">」を検索し、場所を特定してください。(行番号は参考です)

```
 62            <Host name="upkishibSP.nii.ac.jp">     ← ホスト名変更
 63                <Path name="secure" authType="shibboleth" requireSession="true"/>
 64            </Host>
(省略)
 77      <ApplicationDefaults id="default" policyId="default"
 78          entityID=https://upkishibSP.nii.ac.jp/shibboleth-sp  ← ホスト名変更
 79          REMOTE_USER="eppn persistent-id targeted-id"
 80          signing="false" encryption="false"
 81          >
```

※「Default example directs」を検索し、場所を特定してください。(行番号は参考です)

```
104            <!-- Default example directs to a specific IdP's SSO service
                                              (favoring SAML 2 over Shib 1). -->
105            <SessionInitiator type="Chaining" Location="/Login"    isDefault="true"
                      id="Intranet"  relayState="cookie"
106                   entityID="https:// upkishibIdP.nii.ac.jp /idp/shibboleth">
                                  ↑
                       ※metadataに設定されているIdPのentityIDの内容を設定してください。

107                <SessionInitiator type="SAML2" defaultACSIndex="1"
                                                template="bindingTemplate.html"/>
108                <SessionInitiator type="Shib1" defaultACSIndex="5"/>
109            </SessionInitiator>
(省略)                                                          ↓追加
129            <SessionInitiator type="Chaining" Location="/DS" isDefault="false" id="DS"
                                                                  relayState="cookie">
130               <SessionInitiator type="SAML2" defaultACSIndex="1" template=
                    "bindingTemplate.html" acsByIndex="false"/>
131                <SessionInitiator type="Shib1" defaultACSIndex="5" acsByIndex="false"/>
132              <SessionInitiator type="SAMLDS" URL="https://upkishibDS.nii.ac.jp/ds/WAYF"/>
133            </SessionInitiator>                              ↑DS 名
```

※「Example of locally maintained metadata」を検索し、場所を特定してください。(行番号は参考です)

```
217             <!-- Example of locally maintained metadata. -->
218             <!-- -->
219             <MetadataProvider type="XML" file=" /etc/shibboleth /partner-metadata.xml"/>
220             <!-- -->                          ↑コメントを外し、メタデータのパスを書きます
221         </MetadataProvider>
```

3-1-2. Apache の設定とメタデータの作成

① サーバ証明書申請

「UPKI オープンドメイン証明書自動発行検証プロジェクト」の利用の手引きにおける「加入者編」をご覧いただき、サーバ証明書を申請します。

下記のサイトをご参照ください。

https://upki-portal.nii.ac.jp/docs/odcert/howto/ee

証明書の交付までには数日を要するので、お早めに申請してください。

② 入手したサーバ証明書を元に、以下のファイルに設定してください。

■ /etc/httpd/conf.d/ssl.conf

```
(省略)
SSLCertificateFile /etc/shibboleth/cert/server.crt ←サーバ証明書の格納先
SSLCertificateKeyFile  /etc/shibboleth/cert/server.key ←サーバ秘密鍵の格納先
#SSLCACertificateFile /etc/pki/tls/certs/ca-bundle.crt ←コメントアウト
SSLCACertificatePath /etc/shibboleth/cert/CA     ←CA 証明書の格納先
(省略)
```

/etc/shibboleth/cert/CA ディレクトリが無い場合は作成してください。このディレクトリには、ファイル名をハッシュ値とした中間 CA 証明書を配置します。

詳しくは、「サーバ証明書インストールマニュアル」を参照してください。

https://upki-portal.nii.ac.jp/docs/odcert/document/install

■ /etc/shibboleth/shibboleth2.xml

```
(省略)
<!-- Simple file-based resolver for using a single keypair. -->
    <CredentialResolver type="File"
        key="cert/server.key" certificate="cert/server.crt"/>
                                          ←サーバ証明書, 秘密鍵の格納先
```

3-1-3. メタデータの更新

以下の UPKI イニシアティブのサイトのリポジトリから SP 用メタデータテンプレートをダウンロードし、必要な項目を変更します。

https://upki-repo.nii.ac.jp/Template

ダウンロードしたメタデータテンプレートを下記のように変更してください。entityID などの URL に関する変更は、本ガイドに沿ってインストールしている場合は、ホスト名の部分だけの変更で十分です。

```
(中略)
<EntityDescriptor entityID="https://YourSPSite.ac.jp/shibboleth-sp"> ←ホスト名
 <!-- An SP supporting SAML 1 and 2 contains this element with protocol support as shown. -->
 <SPSSODescriptor
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol urn:oasis:names:tc:SAML:1.1:protocol">
   <Extensions>
     <!-- Extension to permit the SP to receive IdP discovery responses. -->
     <idpdisc:DiscoveryResponse
xmlns:idpdisc="urn:oasis:names:tc:SAML:profiles:SSO:idp-discovery-protocol"
       index="1" Binding="urn:oasis:names:tc:SAML:profiles:SSO:idp-discovery-protocol"
       Location="https://YourSPSite.ac.jp/Shibboleth.sso/DS"/>
   </Extensions>
   <KeyDescriptor>
     <ds:KeyInfo>
     <ds:X509Data>
     <ds:X509Certificate>
MIIDOzCCAiOgAwIBAgIUdTJ6oiEccCjrtDyDaeBXT1RpfPcwDQYJKoZIhvcNAQEF
BQAwHzEdMBsGA1UEAxMUdXBraXNoaWIxMS5uaWkuYWMuanAwHhcNMDgwNzA4MTAz
Iyv7HVc9KtwkVAooF/LMPP9ofkZeuqzpc8T6Wg+zaUUsIKhEDhon4Zb/rt9tS3vB
                    ****************
                    ****************
zKbGNi+5a05w/wLkxpEaP8NPTHkbN3E+EXQDik7QQOqGJO+JEUYLAPO6HTGGCs5i
YU+cTQ5QSgjfsSwcZQt6ljQUzlyhKOAWnazbrRGVfCV1woY10hkpmGMSb4Jjxo6E
61psWSAHlehx6L2F9Eat
         </ds:X509Certificate>
       </ds:X509Data>
     </ds:KeyInfo>
   </KeyDescriptor>
   <!--
   This tells IdPs where and how to send authentication assertions. Mostly
   the SP will tell the IdP what location to use in its request, but this
   is how the IdP validates the location and also figures out which
   SAML version/binding to use.
   -->
   <AssertionConsumerService index="1" isDefault="true"
     Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
     Location="https://YourSPSite.ac.jp/Shibboleth.sso/SAML2/POST"/>  ←ホスト名
   <AssertionConsumerService index="2"
     Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-SimpleSign"
     Location="https://YourSPSite.ac.jp/Shibboleth.sso/SAML2/POST-SimpleSign"/>  ←ホスト名
   <AssertionConsumerService index="3"
     Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"
     Location="https://YourSPSite.ac.jp/Shibboleth.sso/SAML2/Artifact"/>  ←ホスト名
   <AssertionConsumerService index="4"
     Binding="urn:oasis:names:tc:SAML:1.0:profiles:browser-post"
     Location="https://YourSPSite.ac.jp/Shibboleth.sso/SAML/POST"/>  ←ホスト名
   <AssertionConsumerService index="5"
     Binding="urn:oasis:names:tc:SAML:1.0:profiles:artifact-01"
     Location="https://YourSPSite.ac.jp/Shibboleth.sso/SAML/Artifact"/>  ←ホスト名
 </SPSSODescriptor>
```

入手した証明書に変更

```
  <!-- This is just information about the entity in human terms. -->
  <Organization>
    <OrganizationName xml:lang="en">Your SP</OrganizationName>  ←組織名
    <OrganizationDisplayName xml:lang="en">Your SP</OrganizationDisplayName>  ←組織表示名
    <OrganizationURL xml:lang="en">http://Your HomePage/</OrganizationURL>  ←組織 URL
  </Organization>
  <ContactPerson contactType="Your ContactType"> ←管理者ポジションを[technical, support, administrative,
billing, other]から選択
    <GivenName>Your GivenName</GivenName>  ←管理者名
    <SurName>Your SurName</SurName>  ←管理者名
    <EmailAddress>Your Email Address</EmailAddress>  ←管理者の e-mail アドレス(メタデータは公開されるので alias
名などを推奨：システム運用基準 4.2 項参照)
  </ContactPerson>

</EntityDescriptor>
 (中略)
```

完成した新しい SP 用のメタデータを、ヘルプデスク（upki-sso-help@nii.ac.jp）へ送付してください。

ヘルプデスクでは、送付していただいたファイルをもとに、DS に登録するとともに共用メタデータを更新します。

3-1-4. SP への接続確認

① httpd サービスと、shibd サービスを再起動します。

```
# service httpd restart
# service shibd restart
```

② SP にアクセスします。

サーバ上のブラウザで、設定した SP にアクセスします。

https://localhost/Shibboleth.sso/Status

（サーバ名は必ず localhost として下さい）

画面上に ok が表示されれば SP に接続が確認出来ました。

＜セミナー注： セミナーの環境ではブラウザをインストールしていないので、
この確認は実行できません。＞

3-1-5. IdP との SP 接続確認

接続する IdP の設定変更も必要となります。設定変更は IdP の管理者に依頼して下さい。

① SP にテスト用のファイルを用意します。ファイルの内容は以下の１行です。

/var/www/html/secure/phpinfo.php

```
<?php phpinfo(); ?>
```

② SP のメタデータに IdP への接続設定を追加します。

直接リダイレクトする IdP のメタデータ

/opt/shibboleth-idp/metadata/idp-metadata.xml

に記述された、その IdP の<EntityDescriptor> ～ </EntityDescriptor>部分と同じ内容を全て、SP の /etc/shibboleth/partner-metadata.xml に追加します。

```
<EntitiesDescriptor Name="urn:mace:shibboleth:testshib:two"
(省略)
     <EntityDescriptor entityID="https:// upkishibIdP.nii.ac.jp/idp/shibboleth">
(省略)                                          ↑ 直接リダイレクトする IdP
     </EntityDescriptor>
(省略)
</EntitiesDescriptor>
```

③ 追加した SP の設定を IdP に追加します。

SP の/etc/shibboleth/partner-metadata.xml に記載された この SP の

<EntityDescriptor> ～ </EntityDescriptor>部分と同じ内容を全て、IdP の

/opt/shibboleth-idp /metadata/idp-metadata.xml に追加します。

```
<EntitiesDescriptor Name="urn:mace:shibboleth:testshib:two"
(省略)
     <EntityDescriptor entityID="https:// upkishibSP.nii.ac.jp/idp/shibboleth">
(省略)                                          ↑ この SP のホスト名
</EntityDescriptor>
(省略)
</EntitiesDescriptor>
```

④ ブラウザから SP の①で用意したファイルへアクセスします。

https://upkishibSP/secure/phpinfo.php

⑤ IdP にログイン後、表示内容を確認します、

IdP のログイン画面が表示され、ID, パスワードを入力してログインした後、表示される環境変数に、IdP で公開する設定とした値(LDAP に保存されている eduPersonPrincipalName など)が含まれていることを確認します。

これが、SSO により IdP から渡されたユーザの属性情報となります。

表示例)

PHP Variables

variable	value
_SERVER[“unscoped-affiliation”]	faculty

⑥ /etc/shibboleth/shibboleth2.xml ファイルに、接続する DS を設定します。
IdP へ直接リダイレクトせず、DS を用いる設定を行います。

※「Default example directs」を検索し、場所を特定してください。(行番号は参考です)

```
    104             <!-- Default example directs to a specific IdP's SSO service
                                                (favoring SAML 2 over Shib 1). -->
    105             <SessionInitiator type="Chaining" Location="/Login" isDefault="false"←false とする
                          id="Intranet"  relayState="cookie"
    106                   entityID="https:// upkishibIdP.nii.ac.jp /shibboleth">
    107                 <SessionInitiator type="SAML2" defaultACSIndex="1"
                                                 template="bindingTemplate.html"/>
    108                 <SessionInitiator type="Shib1" defaultACSIndex="5"/>
    109             </SessionInitiator>
   (省略)                                                          ↓true とする
    129             <SessionInitiator type="Chaining" Location="/DS" isDefault="true" id="DS"
                                                                  relayState="cookie">
    130               <SessionInitiator type="SAML2" defaultACSIndex="1" template=
                        "bindingTemplate.html" acsByIndex="false"/>
    131                 <SessionInitiator type="Shib1" defaultACSIndex="5" acsByIndex="false"/>
    132               <SessionInitiator type="SAMLDS" URL="https://upki-test-ds.nii.ac.jp/ds/WAYF"/>
    133             </SessionInitiator>
```

＜セミナー注： DS の URL は、https://seminar-ds.nii.ac.jp/ds/WAYF として下さい。
＞

＜セミナー注２： 下記を設定して下さい。＞

attribute-map.xml に下記を追記。

＝＝＝＝＝＝＝＝＝＝＝＝

```
<Attribute name="urn:oid:2.16.840.1.113730.3.1.241" id="displayName"/>
<Attribute name="urn:oid:1.3.6.1.4.1.32264.1.1.1" id="jasn"/>
<Attribute name="urn:oid:1.3.6.1.4.1.32264.1.1.2" id="jaGivenName"/>
<Attribute name="urn:oid:1.3.6.1.4.1.32264.1.1.3" id="jaDisplayName"/>
<Attribute name="urn:oid:1.3.6.1.4.1.32264.1.1.4" id="jao"/>
<Attribute name="urn:oid:1.3.6.1.4.1.32264.1.1.5" id="jaou"/>
```

＝＝＝＝＝＝＝＝＝＝＝＝

attribute-policy.xml では、

<Filter out undefined affiliations ... の ScopingRules 等をコメントアウト

3-2.　構築後のカスタマイズ

3-2-1. IdP-アプリケーション間で受け渡す属性の追加方法

/etc/shibboleth/attribute-map.xml 内に、該当する属性があるか確認してください。

※ ほとんどの属性が attribute-map.xml にて定義されています。

attribute-map.xml で定義されている属性は、IdP がリリースすると、無変換でアプリケーションに送られます。

attribute-map.xml で定義されていない場合については、以下に「displayName」属性をマッピングする例で示します。

① スキーマの確認

- LDAP サーバ上の/etc/openldap/schema 配下にスキーマファイルがあります。
- 「displayName」属性は、/etc/openldap/schema/inetorgperson.schema にて以下のように定義されています。

```
（中略）
# displayName
# When displaying an entry, especially within a one-line summary list, it # is useful to be able
to identify a name to be used.  Since other attri- # bute types such as 'cn' are multivalued,
an additional attribute type is # needed.  Display name is defined for this purpose.
attributetype ( 2.16.840.1.113730.3.1.241
        NAME 'displayName'
        DESC 'RFC2798: preferred name to be used when displaying entries'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )
（中略）
```

② /etc/shibboleth/attribute-map.xml への登録

```
<Attributes xmlns="urn:mace:shibboleth:2.0:attribute-map"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
（中略）
    <Attribute name="urn:mace:dir:attribute-def:displayName" id="displayName"/>
   <Attribute name="urn:oid:2.16.840.1.113730.3.1.241" id="displayName"/> ←①のoid
（中略）
</Attributes>
```

3-2-2. メタデータの自動更新設定方法

設定ファイルを変更し、shibd を再起動することでメタデータの自動更新設定に変更します。

① メタデータ格納用ディレクトリの作成

メタデータ格納用のディレクトリを作成します。

```
#mkdir /etc/shibboleth/metadata
```

※作業は root 権限で実行する必要があります。

② shibboleth2.xml ファイルの変更

従来の設定では、/etc/shibboleth/shibboleth2.xml ファイルに、メタデータをファイルから読込む設定を記述していました。

```
<MetadataProvider type="XML" file="/etc/shibboleth/partner-metadata.xml"/>
```

この設定をコメントアウトします。

```
<!-- comment out
<MetadataProvider type="XML" file="/etc/shibboleth/partner-metadata.xml"/>
-->
```

新たに、メタデータを自動的にダウンロードする設定を追加します。

```
<MetadataProvider type="XML"
    uri="http://upki-repo.nii.ac.jp/Metadata/upki-fed-metadata.xml"
    backingFilePath="/etc/shibboleth/metadata/backingMetadata.xml"
    reloadInterval="86400" />
```

※必要に応じて 24 時間毎に自動ダウンロードする設定となっています。

③ shibd の再起動

shibboleth2.xml ファイルを変更した後で、shibd を再起動します。

```
#service shibd restart
```

④ 動作確認方法

1．設定変更を行った SP で、ログインできることを確認

通常の手順で、設定変更を行った SP にログインできることを確認します。

2．自動ダウンロードした UPKI-Fed メタデータを確認

ファイルが存在し、タイムスタンプがログイン時の日時に更新されていることを確認します。

```
#ls -l /etc/shibboleth/metadata/backingMetadata.xml
```

3．自動更新の確認

上記確認後、24 時間以上経過したあとで再度ログインし、メタデータファイルのタイムスタンプが更新されていることを確認します。

3-2-3. メタデータ署名の検証設定方法

① 設定方法

メタデータ格納用のディレクトリを作成します。

$SP_HOME/shibboleth2.xml に、設定を追加する必要があります。

なお、shibboleth2.xml を変更後、設定を有効にするには shibd サービスを再起動する必要があります。

検証用証明書をダウンロードして、$SP_HOME/cert/に置きます。

```
<SPConfig>
   <ApplicationDefaults>
     <MetadataProvider type="Chaining">
        <MetadataProvider
          詳細省略
        >   ← "/>"から変更する
        <SignatureMetadataFilter ← 署名検証を行うための設定を追加する
              certificate="/etc/shibboleth/cert/ upki-fed-signer-ca.cer "/> : 検証に用いる証明書
     </MetadataProvider> ← 追加する
   </ApplicationDefaults>
</SPConfig>
```

② 記述例

メタデータを自動ダウンロードし、メタデータの署名検証を行う場合の記述例です。

・ 修正前

```
 <MetadataProvider type="Chaining">

    <MetadataProvider type="XML"
        uri="https://157.1.72.5/UPKIFed/Repository/upki-fed-metadata-signed.xml"
                          backingFilePath="/etc/shibboleth/metadata/backingMetadata.xml"
        reloadInterval="7200"
        />

</MetadataProvider>
```

・ 修正後

```
<MetadataProvider type="Chaining">

    <MetadataProvider type="XML"
        uri="https://157.1.72.5/UPKIFed/Repository/upki-fed-metadata-signed.xml"
        backingFilePath="/etc/shibboleth/metadata/backingMetadata.xml"
        reloadInterval="7200"
        >

        <SignatureMetadataFilter certificate="/etc/shibboleth/cert/upki-fed-signer-ca.cer " />

    </MetadataProvider>

</MetadataProvider>
```

③ 参考資料

参考資料 Shibboleth2.0(SP)の設定ドキュメント

https://spaces.internet2.edu/display/SHIB2/NativeSPReloadableXMLFile

https://spaces.internet2.edu/display/SHIB2/NativeSPMetadataProvider

4. 関連 URL

■ UPKI プロジェクト（UPKI イニシアティブ）
https://upki-portal.nii.ac.jp/

■ 学術認証フェデレーション
https://upki-portal.nii.ac.jp/docs/fed/

■ UPKI オープンドメイン証明書自動発行検証プロジェクト
https://upki-portal.nii.ac.jp/docs/odcert

■ Shibboleth プロジェクト
http://shibboleth.internet2.edu/

■ Shibboleth2.0 Wiki（Shibboleth2.0 の構築、設定に関する公式サイト）
https://spaces.internet2.edu/display/SHIB2/Home

■ Switch.aai （スイスのフェデレーション）
http://www.switch.ch/aai/

■ InCommon（米国のフェデレーション）
http://www.incommonfederation.org/