Shibboleth IdP 4.1の高度な認証設定

以下でご紹介しているExtendedフローはdeprecatedとなりShibboleth IdPバージョン5で削除される予定です。現行バージョンで当該機能を使っている方はMFAへの移行をご検討ください。

以下で4.1でもまだ有効な簡便な方法を説明していますが、3.3以降であればより汎用的で複雑な挙動が実現できるMFAによる方法もご参照ください。

- 認証フローの階層化
- Password認証フローのExtendedフロー
- 参考

Shibboleth IdP 3以降の高度な認証設定についてのドキュメントです。4.1以降で書式が大幅に変更になりましたのでそれに特化した記述になっております。本ドキュメントはSAML 2.0で認証の切り替えを行うことを目的としており、SAML1は対象外です（LevelXを用いた認証要求はできません）。

SAML1を使うことにより本設定の制約を迂回できることを避けるため、SPにおいてはshibboleth2.xmlにてSAML1の機能を無効化することをお勧めします。

本ドキュメントで使用する認証コンテクストと認証フローの関係を下記に示します。

ここでRemoteUserはパスワード認証とクライアント証明書認証の中間の強度の認証の例として挙げております。実際に運用する場合は別途用意した認証フローで置き換えてください。また、レベルとの対応付けも本ドキュメント独自のもので、例示として使用しています。

認証コンテクスト	略称	認証フロー
urn:mace:gakunin.jp:idprivacy:ac:classes:Level1	Level1	Password
urn:mace:gakunin.jp:idprivacy:ac:classes:Level2	Level2	RemoteUser
urn:mace:gakunin.jp:idprivacy:ac:classes:Level3	Level3	X509

この認証コンテクストとは別に、PasswordやX509は固有の認証コンテクストを持っていますが、ここでは使用しません。

挙動の説明で使用するSPについて下記に示します。

SP	要求する認証
SP_a	なし
SP_b	urn:mace:gakunin.jp:idprivacy:ac:classes:Level1
SP_c	urn:mace:gakunin.jp:idprivacy:ac:classes:Level2
SP_d	urn:mace:gakunin.jp:idprivacy:ac:classes:Level3

認証フローの階層化

設定

1. 既に認証済みの認証フローを優先するために、conf/authn/authn.propertiesのidp.authn.favorSSOをアンコメントしtrueに設定します。

conf/authn/authn.properties

```
 # Whether to prioritize "active" results when an SP requests more than
 # one possible matching login method (V2 behavior was to favor them)
-#idp.authn.favorSSO = false
+idp.authn.favorSSO = true
```

2. 各認証フローのsupportedPrincipalsプロパティに下記を追加します。

認証フロー	supportedPrincipalsプロパティ
Password	継承元のshibboleth.AuthenticationFlowで定義されているsupportedPrincipals, Level1
RemoteUser	Level2, Level1
X509	Level3, Level2, Level1

conf/authn/authn.properties

```
@@ -56,10 +56,11 @@
 # Unset if using customized Principals per validator
 #idp.authn.Password.addDefaultPrincipals = true
 # The Principal collection below is the typical default if not otherwise noted.
-#idp.authn.Password.supportedPrincipals = ¥
-#    saml2/urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport, ¥
-#    saml2/urn:oasis:names:tc:SAML:2.0:ac:classes:Password, ¥
-#    saml1/urn:oasis:names:tc:SAML:1.0:am:password
+idp.authn.Password.supportedPrincipals = ¥
+    saml2/urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport, ¥
+    saml2/urn:oasis:names:tc:SAML:2.0:ac:classes:Password, ¥
+    saml1/urn:oasis:names:tc:SAML:1.0:am:password, ¥
+    saml2/urn:mace:gakunin.jp:idprivacy:ac:classes:Level1
 # Validators are controlled in password-authn-config.xml

 #### Password Backends ####
@@ -97,7 +98,9 @@
 idp.authn.RemoteUser.supportedPrincipals = ¥
  saml2/urn:oasis:names:tc:SAML:2.0:ac:classes:X509, ¥
  saml2/urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient, ¥
- saml1/urn:ietf:rfc:2246
+ saml1/urn:ietf:rfc:2246, ¥
+ saml2/urn:mace:gakunin.jp:idprivacy:ac:classes:Level2, ¥
+ saml2/urn:mace:gakunin.jp:idprivacy:ac:classes:Level1

 #### RemoteUserInternal ####

@@ -137,7 +140,10 @@
 idp.authn.X509.supportedPrincipals = ¥
     saml2/urn:oasis:names:tc:SAML:2.0:ac:classes:X509, ¥
     saml2/urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient, ¥
-    saml1/urn:ietf:rfc:2246
+    saml1/urn:ietf:rfc:2246, ¥
+    saml2/urn:mace:gakunin.jp:idprivacy:ac:classes:Level3, ¥
+    saml2/urn:mace:gakunin.jp:idprivacy:ac:classes:Level2, ¥
+    saml2/urn:mace:gakunin.jp:idprivacy:ac:classes:Level1

 #### X509Internal ####
```

3. conf/relying-party.xmlのshibboleth.DefaultRelyingParty内のSAML2.SSOにdefaultAuthenticationMethodsプロパティを設定します。

conf/relying-party.xml

```
    <bean id="shibboleth.DefaultRelyingParty" parent="RelyingParty">
        <property name="profileConfigurations">
            <list>
                <bean parent="Shibboleth.SSO" p:postAuthenticationFlows="attribute-release" />
                <ref bean="SAML1.AttributeQuery" />
                <ref bean="SAML1.ArtifactResolution" />
-               <bean parent="SAML2.SSO" p:postAuthenticationFlows="attribute-release" />
+               <bean parent="SAML2.SSO" p:postAuthenticationFlows="attribute-release">
+                   <property name="defaultAuthenticationMethods">
+                       <list>
+                           <bean parent="shibboleth.SAML2AuthnContextClassRef"
+                               c:classRef="urn:mace:gakunin.jp:idprivacy:ac:classes:Level1" />
+                       </list>
+                   </property>
+               </bean>
                <ref bean="SAML2.ECP" />
                <ref bean="SAML2.Logout" />
                <ref bean="SAML2.AttributeQuery" />
                <ref bean="SAML2.ArtifactResolution" />
                <ref bean="Liberty.SSOS" />
            </list>
        </property>
    </bean>
```

必要であればRelyingPartyOverridesのほうのSAML2.SSOにも同様の設定を追加してください。

4. この設定で問題がある場合、つまり許容されているレベルでもより高レベルの認証が求められる場合は、authn.propertiesの idp.authn.*.order を調整してください。より弱い認証のorderを小さい数にすれば、より高い優先度で表示されるようになります。

挙動

未認証の場合

1. SP_aもしくはSP_bからIdPにリダイレクトされると、Level1のPassword認証フローのログインページが表示されます。
2. SP_cからIdPにリダイレクトされると、Level2のRemoteUser認証フローのためのログインページやダイアログが表示されます。
3. SP_dからIdPにリダイレクトされると、Level3のX509認証フローのログインページが表示されます。

Level1が認証済みの場合

1. SP_aもしくはSP_bからIdPにリダイレクトされると、Level1のPassword認証フローが認証済みのためユーザ同意画面もしくは認証後のSPの画面が表示されます。
2. SP_cからIdPにリダイレクトされると、Level2のRemoteUser認証フローのためのログインページやダイアログが表示されます。
3. SP_dからIdPにリダイレクトされると、Level3のX509認証フローのログインページが表示されます。

Level2が認証済みの場合

1. SP_aもしくはSP_bからIdPにリダイレクトされると、Level2のRemoteUser認証フローが認証済みのためユーザ同意画面もしくは認証後のSPの画面が表示されます。
2. SP_cからIdPにリダイレクトされると、Level2のRemoteUser認証フローが認証済みのためユーザ同意画面もしくはSPの画面が表示されます。
3. SP_dからIdPにリダイレクトされると、Level3のX509認証フローのログインページが表示されます。

Level3が認証済みの場合

1. SP_aもしくはSP_bからIdPにリダイレクトされると、Level3のX509認証フローが認証済みのためユーザ同意画面もしくは認証後のSPの画面が表示されます。
2. SP_cからIdPにリダイレクトされると、Level3のX509認証フローが認証済みのためユーザ同意画面もしくは認証後のSPの画面が表示されます。
3. SP_dからIdPにリダイレクトされると、Level3のX509認証フローが認証済みのためユーザ同意画面もしくは認証後のSPの画面が表示されます。

Password認証フローのExtendedフロー

設定

1. 認証フローの階層化の設定を行ってください。
2. conf/authn/authn.properties のauthn/PasswordにExtendedフローで利用するLevel2, Level3を追加します。

conf/authn/authn.properties

```
     saml2/urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport, ¥
     saml2/urn:oasis:names:tc:SAML:2.0:ac:classes:Password, ¥
     saml1/urn:oasis:names:tc:SAML:1.0:am:password, ¥
-    saml2/urn:mace:gakunin.jp:idprivacy:ac:classes:Level1
+    saml2/urn:mace:gakunin.jp:idprivacy:ac:classes:Level1, ¥
+    saml2/urn:mace:gakunin.jp:idprivacy:ac:classes:Level2, ¥
+    saml2/urn:mace:gakunin.jp:idprivacy:ac:classes:Level3
 # Validators are controlled in password-authn-config.xml
```

3. conf/authn/password-authn-config.xmlでExtendedフローのbeanをアンコメントし、下記の設定を行います。もし当該ファイルにこの部分が存在しなければ、最終行の1つ上にこの部分を挿入してください。
    - shibboleth.authn.Password.ExtendedFlowsのc:_0に、ExtendedフローとするRemoteUserとX509を設定します。
    - shibboleth.authn.Password.PrincipalOverrideに、Password認証フローで認証するLevel1を追加します。Level2やLevel3を除いているところがポイントです。

conf/authn/password-authn-config.xml

```
     <!--
     Configuration of "extended" login methods to offer in the password login form.
     The String bean is a regular expression identifying the flows to offer. These flows
     must also be enabled at the "top" level to be available for use.
     The ExtendedFlowParameters bean can be used to transfer custom parameters from the
     login form into the context tree for use later by other flows.
     The last bean provides the set of custom Principals to use for results produced by the
     Password flow itself. You would use this if you need the Password flow to run as a shell
     to run the "extended" login methods, but want to limit its own results more narrowly.
     -->
-    <!--
-    <bean id="shibboleth.authn.Password.ExtendedFlows" class="java.lang.String" c:_0="" />
+    <bean id="shibboleth.authn.Password.ExtendedFlows" class="java.lang.String" c:_0="RemoteUser|X509" />
     <util:list id="shibboleth.authn.Password.ExtendedFlowParameters">
     </util:list>
     <util:list id="shibboleth.authn.Password.PrincipalOverride">
         <bean parent="shibboleth.SAML2AuthnContextClassRef"
             c:classRef="urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport" />
         <bean parent="shibboleth.SAML2AuthnContextClassRef"
             c:classRef="urn:oasis:names:tc:SAML:2.0:ac:classes:Password" />
         <bean parent="shibboleth.SAML1AuthenticationMethod"
             c:method="urn:oasis:names:tc:SAML:1.0:am:password" />
+        <!-- -->
+        <bean parent="shibboleth.SAML2AuthnContextClassRef"
+            c:classRef="urn:mace:gakunin.jp:idprivacy:ac:classes:Level1" />
     </util:list>
-    -->
```

4. Shibboleth IdP 4.1および4.2にはバグがありますので手動でaddDefaultPrincipalsをfalseにしてください。

conf/authn/authn.properties

```
 # Unset if using customized Principals per validator
-#idp.authn.Password.addDefaultPrincipals = true
+idp.authn.Password.addDefaultPrincipals = false
 # The Principal collection below is the typical default if not otherwise noted.
```

5. Shibboleth IdP 4.0.0および4.0.1をお使いの場合および以降のバージョンでも以前のバージョンからアップデートしている場合は、login.vmにバグがあり追加のボタンが表示されませんので、以下の修正を行ってください。

views/login.vm

```
            #end

            #foreach ($extFlow in $extendedAuthenticationFlows)
-             #if ($authenticationContext.isAcceptable($extFlow) and $extFlow.apply(profileRequestContext))
+             #if ($authenticationContext.isAcceptable($extFlow) and $extFlow.test(profileRequestContext))
                <div class="form-element-wrapper">
                  <button class="form-element form-button" type="submit" name="_eventId_$extFlow.getId()">
                    #springMessageText("idp.login.$extFlow.getId().replace('authn/','')", $extFlow.getId().replace('authn
/',''))
```

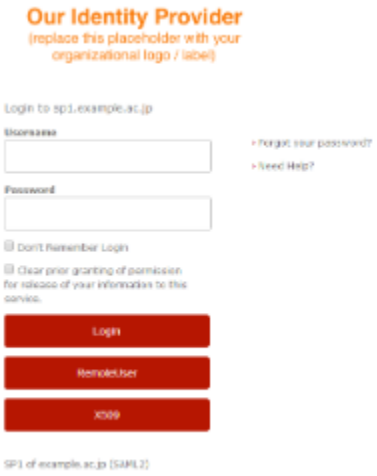
挙動

各SPからIdPにリダイレクトされた時に表示されるPassword認証フローのログインページを下記に示します。

1. SP_aおよびSP_bからの場合
   Password認証フローのための「Username」と「Passowrd」のフォームと「Login」ボタン、およびExtendedフローであるRemoteUser認証フローとX509認証フローのためのボタン「RemoteUser」と「X509」が表示されます。

   「Login」とボタンの形状が同じで紛らわしいですが、Extendedフローを利用する場合は上部のフォーム（「Username」と「Passowrd」）の入力は不要です。

図1. SPaおよびSPbからの場合

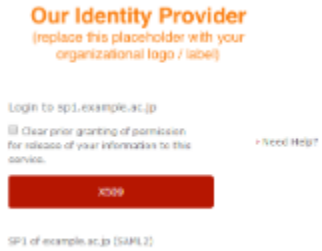


2. SP_cからの場合
   ExtendedフローであるRemoteUser認証フローとX509認証フローのためのボタン「RemoteUser」と「X509」が表示されます。Level1のPassword認証フローは表示されません。

図2. SPcからの場合



3. SP_dからの場合
   Extendedフローのうち`Level3`以上であるX509認証フローのためのボタン「X509」が表示されます。`Level1`のPassword認証フロー、および`Level2`のX509認証フローは表示されません。
   図3. SPdからの場合



参考

高度な認証フローを設定する上で、参考になるドキュメントを下記に示します。

- [Shibboleth wiki] AuthenticationConfiguration
- [Shibboleth wiki] AuthenticationFlowSelection
- [Shibboleth wiki] PasswordAuthnConfiguration|ExtendedFlows
  の"Extended Flows"
- [Shibboleth wiki] Orchestrating Multiple Authentication Methods and Contexts - The Multi-Context Broker (MCB)
- [Shibboleth wiki] Configuring the IdP for the Multi-Context Broker Model
- [Shibboleth wiki] Replicating Multi-Context Broker Functionality (Duo + Username/Password with user-opt-in forcing Duo)
- [Shibboleth wiki] SP-driven Duo opt-in
  （リンク先にsystem/以下のファイルを編集している箇所がありますが推奨されていません）
- 3.3向け
  [Shibboleth wiki] MultiFactorAuthnConfiguration