同じ値が再割り当てされないeduPersonTargetedIDの生成方法

eduPersonTargetedID(ePTID)を生成するときにはLDAP上の属性としてuidを代表とした属性値が利用されますが、これらの属性値では再割り当ての問題があります。

例えばuidとして test001 を使っていた人が異動になり、さらに年月が経って同じuidを使いたいという人が現われた場合にはそのまま割り当ててしまうことはできません。これはSP 側で uid=test001 という属性値を基に生成されたePTIDで個人を識別していた場合に、再割り当て前の人物と、再割り当て後の人物を区別できず同一人物とみなして再割り当て前のアカウントの情報を利用してしまうこと（当人が意図しないなりすまし）が起こるためです。

ePTIDでStoredIDを利用している場合には失効処理を行うことで新しいePTIDを生成できることから、再割り当てされる属性値をソースとした上で再割り当て時に失効することでも対処可能です。今回は別の方法として、LDAP上のuidの付加情報としてLDAPエントリの作成時間(createTimestamp)を加えた値をソースとしてePTIDを生成する方法を紹介します（ComputedIDをベースに設定方法を紹介していますが、StoredIDの場合も同様です）。

例えば <uid>-<createTimestamp> のように2つの属性値をハイフンでつなげて test001-20130314110740Z といった値をソースとしてePTIDを生成すれば、uid再割り当てごとの失効処理が不要となります。ただし、再割り当ての際に必ず「作成時間」が変更されるようにアカウント作成処理をすることが前提となります。（LDAPエントリを再利用するような運用ではcreateTimestampが変更されない可能性があります）

- eduPersonTargetedIDのAttributeDefinitionはデフォルトのままで利用可能です。

/opt/shibboleth-idp/conf/attribute-resolver.xml の設定

```
    <!-- Attribute Definition for eduPersonTargetedID (computedID) -->
    <resolver:AttributeDefinition xsi:type="ad:SAML2NameID" id="eduPersonTargetedID"
                                  nameIdFormat="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent" sourceAttributeID="
computedID">
        <resolver:Dependency ref="computedID" />
        <resolver:AttributeEncoder xsi:type="enc:SAML1XMLObject"
                                   name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10" encodeType="false" />
        <resolver:AttributeEncoder xsi:type="enc:SAML2XMLObject"
                                   name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10" friendlyName="eduPersonTargetedID" encodeType="
false" />
    </resolver:AttributeDefinition>
```

- Template Attribute Definitionで uid-createTimestamp の文字列を返すAttributeDefinitionを定義して、ComputedID用DataConnectorのsourceAttributeID, Dependencyで参照できるようにします。ここでは「templateePTID」という名前を用います。

/opt/shibboleth-idp/conf/attribute-resolver.xml の設定

```
    <!-- Computed targeted ID connector -->
    <resolver:DataConnector id="computedID" xsi:type="dc:ComputedId"
                            generatedAttributeID="computedID"
                            sourceAttributeID="%{idp.persistentId.sourceAttribute}"
                            salt="%{idp.persistentId.salt}">
        <resolver:Dependency ref="%{idp.persistentId.sourceAttribute}" />
    </resolver:DataConnector>
    ↓以下の行を追加
    <resolver:AttributeDefinition id="templateePTID" xsi:type="Template" xmlns="urn:mace:shibboleth:2.0:resolver:ad">
        <resolver:Dependency ref="myLDAP" />

        <Template>
            <![CDATA[
                ${uid}-${createTimestamp}
            ]]>
        </Template>

        <SourceAttribute>uid</SourceAttribute>
        <SourceAttribute>createTimestamp</SourceAttribute>
    </resolver:AttributeDefinition>
```

/opt/shibboleth-idp/conf/saml-nameid.properties の設定

```
# For computed IDs, set a source attribute and a secret salt:
idp.persistentId.sourceAttribute = templateePTID   ← 変更
```

- LDAPから追加でcreateTimestampを取得するためにLDAP DataConnectorにReturnAttributesを定義します。

/opt/shibboleth-idp/conf/attribute-resolver.xml の設定

```
    <resolver:DataConnector id="myLDAP" xsi:type="dc:LDAPDirectory"
        ldapURL="%{idp.attribute.resolver.LDAP.ldapURL}"
        baseDN="%{idp.attribute.resolver.LDAP.baseDN}"
        principal="%{idp.attribute.resolver.LDAP.bindDN}"
        principalCredential="%{idp.attribute.resolver.LDAP.bindDNCredential}"
        useStartTLS="%{idp.attribute.resolver.LDAP.useStartTLS:true}">
        <dc:FilterTemplate>
            <![CDATA[
                %{idp.attribute.resolver.LDAP.searchFilter}
            ]]>
        </dc:FilterTemplate>
        <dc:ReturnAttributes>* createTimestamp</dc:ReturnAttributes>   ← 追加(dc:FilterTemplateの直後である必要があります)
    </resolver:DataConnector>
```