Popis systému Single-Sing-On

z pohledu programátorů webových aplikací užívajících SSO
v prostředí Krajského úřadu Plzeňského kraje

Terminologie

V následujícím textu budou jednotlivé strany procesu příhlášení/ověření identity uživatele nazývány takto:

- **SSO**: webová aplikace pracující na serverech Plzeňského kraje, obsahuje webové rozhraní pro komunikaci s *Uživatelem* a SOAP rozhraní pro komunikaci s *Aplikacemi.*
- **Aplikace**: Vaše webová aplikace užívající služby SSO.
- **Uživatel**: Koncový uživatel, tedy člověk sedící u počítače s webovým prohlížečem užívající *Aplikaci*

Základní princip

Cílem SSO je zjednodušit uživatelům přihlašování k aplikacím, tedy umožnit použití jednoho jména a hesla pro více aplikací s možností centrální správy těchto účtů. Uživatel zadává své přihlašovací údaje jen jednou a následně může využívat všechny v SSO zařazené aplikace bez dalších požadavků na identifikaci.

Z pohledu uživatele

Uživatel přistoupí na web aplikace. Pokud doposud není v SSO přihlášen, je přesměrován na webové stránky SSO, kde zadá své přihlašovací údaje. Po úspěšném ověření je uživatel přesměrován zpět do původně požadované aplikace, ta si mezitím z SSO zjistila jeho osobní údaje a uživatel může a aplikaci pracovat.

V případě, že uživatel již v SSO přihlášen je, odpadá zadávání přihlašovacích údajů a celý proces přihlašování je z pohledu uživatele skrytý – aplikace si identitu uživatele zjistí z SSO.

SSO obsahuje některé další funkce, ty Však nejsou pro Vás jako programátory aplikace podstatné. Patří mezi ně registrace nových uživatelů v interní databázi SSO, vyžádání změny hesla či přehlášení na jiného uživatel v případě, že původní uživatel nemá k požadované aplikaci přístup.

Postup použití z pohledu programátora aplikace

Na úvod je třeba sdělit, že za držení session odpovídá Vaše aplikace a očekává se, že komunikace s SSO a ověření uživatele proběhne jen jednou. Následně si Vaše aplikace udržuje informace o přihlášeném uživateli interně ve své session a na SSO se již

neobrací. Podobným způsobem si SSO v rámci své session drží informace o uživateli. Je zřejmé, že uvedený postup neumožňuje žádným způsobem odhlášení uživatele – to proběhne pouze uzavřením prohlížeče a ztrátou session, případně doběhnutím timeoutu dané session na straně serveru.

A teď již k samotnému postupu přihlašování. Uživatel vstoupí do Vaší aplikace a Vy zjistíte, že potřebujete identitu uživatele ověřit. Provedete tedy pomocí http přesměrování či odkazu předání řízení na url

https://sso.plzensky-kraj.cz/saml/TransferService/index.php?TARGET=<url_enkódované_url_vaší_aplikace>&LOGOUT_URL=<url_volané_při_odhlášení_uživatele>

(např. https://sso.plzensky-kraj.cz/saml/TransferService/?TARGET=http%3A%2F%2Fssodemo.muki.cz&LOGOUT_URL=https%3A%2F%2Fssodemo.muki.cz%2Flogout.php)

Parametr target je na straně SSO použit pro zjištění, která aplikace vlastně ověření uživatele požaduje. Je prefixově porovnán s databází aplikací v SSO a pro shodu tedy může obsahovat i delší/přesnější url, než je zapsané v db SSO.

SSO následně vyhodnotí, za se uživatel v rámci jeho session již přihlásil. Pokud ne, provede SSO zobrazení html formuláře s požadavkem na výběr typu ověření uživatele a zadání uživatelského jména a hesla. Pokud SSO vyhodnotní, že přihlašovací údaje byly zadány správně, předá řízení zpět do Vaší aplikace na další url definované v databázi SSO označená jako acs_url, navíc doplní dva parametry: parametr TARGET obsahuje url zaslané parametrem TARGET při počátečním přesměrování do SSO a parametr SAMLart, který slouží jako klíč k převzetí identifikačních údajů z SSO. V případě, že uživatel byl do SSO již dříve přihlášen, žádný formlulář již SSO nezobrazuje a pokračuje stejným přesměrováním do Vaší aplikace, jaké bylo zmíněno výše.

Nyní již tedy Vaše aplikace ví, že proběhlo úspěšné přihlášení do SSO, ale netuší, kdo je vlastně přihlášen.

Zjištění těchto údajů z SSO provede Vaše aplikace pomocí buď pomocí SOAPových volání či přes JSONové URL (preferovaný způsob).

Pro zjištění pomocí JSONu aplikace sestaví URL ve tvaru https://sso.plzensky-kraj.cz/JSONserver/?AssertionArtifact=<SAMLart>. Hodnotu SAMLart obdrží skript během přesměrování z SSO jako parametr. Po zavolání metodou GET je vrácena JSON odpověď.

Ukázka scriptu pro zjistění údajů pomocí JSON (verze .php)

```
<?php

  session_start();
  $art_id = $_REQUEST['SAMLart'];
  $art_json = file_get_contents("https://sso-tst.plzensky-kraj.cz/JSONserver/?
AssertionArtifact=$art_id");
  $art = json_decode($art_json);
  if ($art->Status!="Success") die("Interni chyba - overeni se nezdarilo");

  $_SESSION['user']=$art->User;
```

```
$_SESSION['logged']=1;
$_SESSION['fid']=$art->User->FID;
header("Location: ".$_REQUEST['TARGET']);
```

Identita uživatele je vrácena v $art->User.

Pro zajištění identity uživatele přes SOAP je třeba zavolání dvou SOAP funkcí. Obě jsou popsány WSDL definicí na http://sso.plzensky-kraj.cz/xml/saml.wsdl. První funkce se jmenuje samlAuthenticationQuery(), očekávaným parametrem je struktura s položkami (typy viz zmíněné WSDL)

AssertionArtifact: parametr SAMLart vrácený z SSO (klíč k údajům)
MajorVersion: 1
MinorVersion: 1
RequestID: identifikace požadavku, doporučený tvar je <IP_klienta>.<unix_timestamp>
IssueInstant: cas dle ISO 8601 (2010-07-25T07:59Z);

SSO vrátí opět strukturu, důležitá je položka Status->StatusCode->Value, pokud má hodnotu „Success“, je současně vrácen seznam povolených profilů pokud je SSO pro aplikaci eviduje a NameIdentifier, který budeme potřebovat v následujícím volání funkce samlAttributeQuery – vstupním parametrem je opět struktura

MajorVersion: 1
MinorVersion:1
AttributeQuery->Subject->NameIdentifier->_ : identifikátor vrácený předchozím voláním

Vrácené jsou pak osobní údaje uživatele evidované v SSO, v případně ověření uživatele v systému ePUSA volání navíc vrací i údaje vedené tam.

Příklad SOAP volání

Požadavek pro samlAuthenticationQuery()

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:oasis:names:tc:SAML:1.0:protocol">
<SOAP-ENV:Body><ns1:Request RequestID="109.81.175.23.1374744385" MajorVersion="1"
MinorVersion="1" IssueInstant="2013-07-25T09:26:25Z">
<AssertionArtifact>MTA5LjgxLjE3NS4yMy4xMzc0NzQ0Mzgz</AssertionArtifact>
</ns1:Request>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Odpověď z samlAuthenticationQuery()

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:oasis:names:tc:SAML:1.0:protocol">
<SOAP-ENV:Body><ns1:Response InResponseTo="109.81.175.23.1374744385" MajorVersion="1"
MinorVersion="2" IssueInstant="2013-07-25T09:26:25Z">
<Status>
        <StatusCode Value="Success"/>
</Status>
<Assertion Issuer="sso.plzensky-kraj.cz">
        <Conditions NotBefore="2013-07-25T09:26:25Z" NotOnOrAfter="2013-07-25T10:26:25Z"/>
        <AuthenticationStatement AuthenticationInstant="2013-07-25T09:26:25Z">
                <Subject>
                        <NameIdentifier Format="rn:oasis:names:tc:SAML:1.1:nameid-
format:emailAddress">muki</NameIdentifier>
                <SubjectConfirmation>
                <ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:artifact</ConfirmationMethod>
```

```
                </SubjectConfirmation>
                </Subject>
        </AuthenticationStatement>
</Assertion></ns1:Response></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

Požadavek pro samlAttributeQuery()

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:oasis:names:tc:SAML:1.0:protocol">
<SOAP-ENV:Body>
<ns1:Request MajorVersion="1" MinorVersion="1">
        <AttributeQuery>
                <Subject>
                        <NameIdentifier>muki</NameIdentifier>
                </Subject>
        </AttributeQuery>
</ns1:Request>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Odpověď z samlAttributeQuery()

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:oasis:names:tc:SAML:1.0:protocol">
<SOAP-ENV:Body><ns1:Response MajorVersion="1" MinorVersion="1" IssueInstant="2013-07-
25T09:26:25Z">
<Status>
<StatusCode Value="Success"/>
</Status><Assertion>
<AttributeStatement>
<Subject><NameIdentifier Format="rn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">muki</
NameIdentifier></Subject>
<Attribute AttributeName="FID"><AttributeValue>muki</AttributeValue></Attribute>
<Attribute AttributeName="FirstName"><AttributeValue>Pavel</AttributeValue></Attribute>
<Attribute AttributeName="MiddleName"><AttributeValue> </AttributeValue></Attribute>
<Attribute AttributeName="LastName"><AttributeValue>Muknšnábl</AttributeValue></Attribute>
<Attribute AttributeName="LastName2"><AttributeValue> </AttributeValue></Attribute>
<Attribute AttributeName="TitleBefore"><AttributeValue> </AttributeValue></Attribute>
<Attribute AttributeName="TitleAfter"><AttributeValue> </AttributeValue></Attribute>
<Attribute AttributeName="Name"><AttributeValue>Muknšnábl Pavel</AttributeValue></Attribute>
<Attribute AttributeName="Email"><AttributeValue>muki@muki.cz</AttributeValue></Attribute>
<Attribute AttributeName="UserID"><AttributeValue>12766</AttributeValue></Attribute>
<Attribute AttributeName="DomainID"><AttributeValue>1</AttributeValue></Attribute>
</AttributeStatement>
</Assertion>
</ns1:Response></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

Odhlašování

Když aplikace předává řízení do SSO může vedle povinného parametru TARGET uvést nepovinně i parametr LOGOUT_URL. Pokud je LOGOUT_URL uvedeno, zařídí SSO během procesu odhlašování přístup na uvedené URL (Je použito pro stažení neviditelného obrázku na odhlašovací stránce – tedy přístup bude proveden z IP uživatele, ale pravděpodobně [z důvodu bezpečnosti nastavené v prohlížeči] nikoli v rámci session uživatele. Tedy je třeba zařídit, aby url obsahovalo nějakou identifikaci session, která má být během odhlašování zrušena. Nedoporučuje použít přímo id session ze session cookie [vhodné použít šifrování či session identifikovat jiným způsobem]).

Odhlášení provede uživatel kliknutím na odkaz https://sso.plzensky-kraj.cz/logout/ - je vhodné, aby byl v každé aplikaci uveden.

Vzorová aplikace

Na vyžádání je k dispozici vzorová php aplikace, demonstrující užití systému SSO. Obsahuje dva soubory – index.php představuje samotnou logiku aplikace, jeho výstupem je zobrazení dat, která od SSO obdrží. Artifact.php je bodem aplikace pro návrat ze sso (jako návratové url je v databázi sso uvedeno http://ssodemo.muki.cz/artifact.php), obsahuje soap komunikaci pro zjištění informací o přihlášeném uživateli.

Verze z 10.6.2019

Pavel Muknšnábl
muki@muki.cz