SAML a XACML jako nová cesta pro Identity management

SAML and XACML as a New Way of Identity Management

Dagmar BRECHLEROVÁ
Oddělení medicínské informatiky, Ústav informatiky
AVČR, v.v.i.

brechlerova@euromise.cz

INFORUM 2007:13. konference o profesionálních informačních zdrojích
Praha, 22. - 24. 5.2007

ABSTRAKT:
Vzrůstá např. počet webových stránek, které poskytují speciální obsah jednotlivým uživatelům. Jde o to, že konkrétní uživatel dostane pouze takový obsah, který buď vyžaduje, za který zaplatil, je pro něj vhodný nebo má právo jej vidět. Informace jsou tedy poskytovány na základě uživatelova profilu. Aby mohl být poskytnut konkrétní obsah, je nutná identifikace a autentizace uživatele prováděná nejčastěji jménem a heslem nebo jinými bezpečnostními informacemi. Jazyk SAML umožňuje tyto bezpečnostní informace vyměňovat a tak např. realizovat tzv. Single sign on (SSO) mechanismus. Dalším bezpečnostním jazykem je XACML, který umožňuje vytvořit přesné politiky pro přístup ke zdrojům, tj. kdo, kdy, kam a odkud k čemu může přistupovat. Příspěvek se zabývá užitím kombinace těchto dvou jazyků pro přesný přístup uživatele ke zdroji. Zkombinováním těchto 2 technologií je možné dosáhnout toho, že pouze určitý uživatel (či skupina) se dostane v určitou dobu k přesně definovanému zdroji informací za přesně daných podmínek a smí (či nesmí) poté vykonávat přesně dané akce.

KLÍČOVÁ SLOVA:
SAML, XACML, autentizace, autorizace

ABSTRACT

Today, the number of websites that are offering personalised web content (some special information for somebody) are growing rapidly. To provide personalised content,websites require users to identify themselves, which is typically done by giving the user an account with an associated user name and password or other secure information. Security Assertion Markup Language (SAML) is an open standard defined by the OASIS. SAML provides a common language that online entities can use to universally share and exchange security information. Extensible Access Control Markup Language (XACML) is an XML based language for access control. The policy language is used to express acces control policies who can do what. SAML and XACML can be used for very complex identity management, for acces control policy. This paper is about the possibility how to use XACML and SAML for identity management. One model of authetication and authorization by XACML and SAML is presented here, too.

KEY WORDS
SAML, XACML, authentication, authorization

Úvod

XML security je rychle se rozvíjejícím standardem bezpečnostních technologií, které mají zabezpečit některé základní bezpečnostní požadavky (integrita, utajení, nepopíratelnost). Celý systém byl popsán v příspěvku autorky na Inforu 2007 [7]. V tomto příspěvku jsou nejdříve podrobněji popsány technologie SAML a XACML, poté jejich možný způsob propojení a nakonec je použití demonstrováno na příkladu.

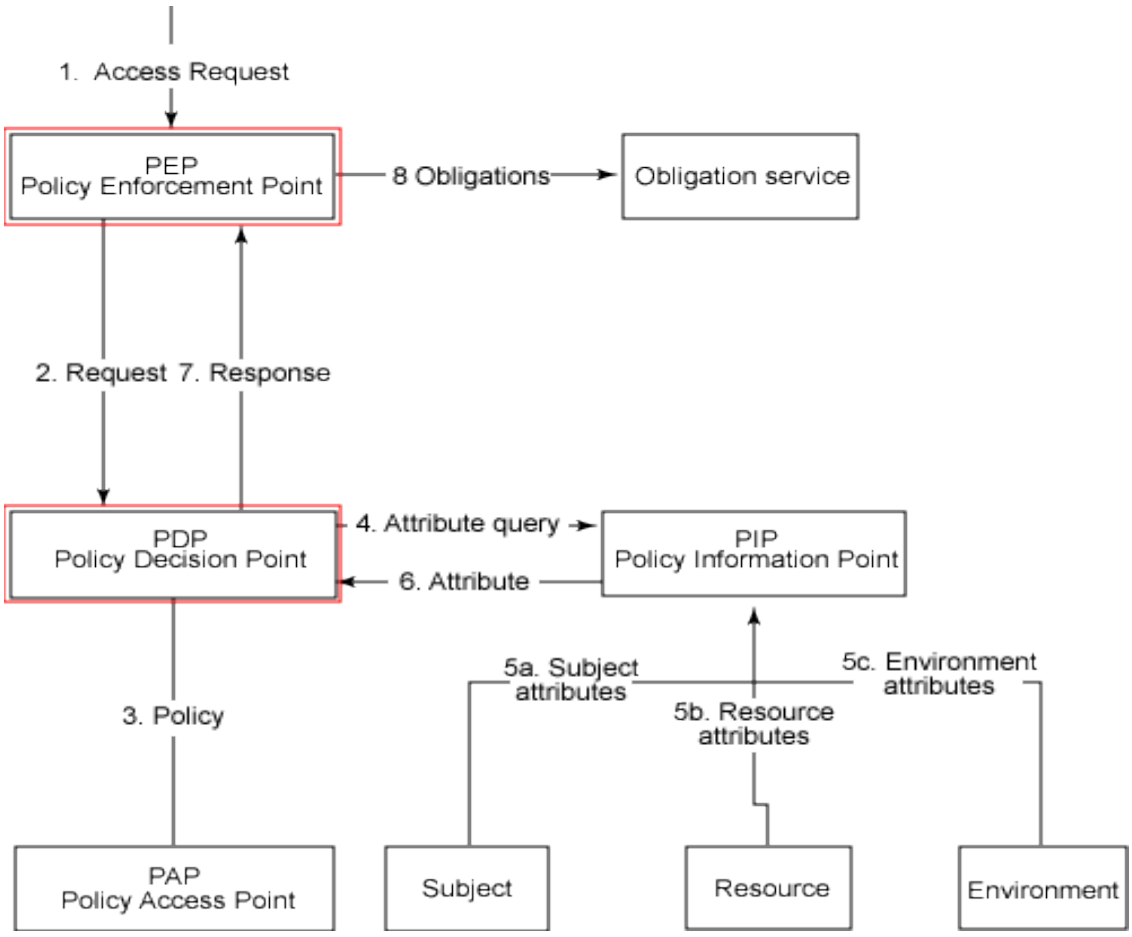
XACML

Jedním z nedůležitějších leč špatně řešitelných požadavků dnešní bezpečnosti je bezpečnost vlastních sítí. Ve většině organizací se jedná o jedno z nejkritičtějších míst bezpečnosti. Důležitou roli zde hraje správa přístupu, což je schopnost přesně definovat, který uživatel má mít možnost přístupu k jakým informacím či datům, kdy, jaká má konkrétní práva atd. Pokud je organizace umístěna v jedné budově, jde obvykle tuto úlohu splnit. Ovšem organizace (škola, nemocnice atd.) má obvykle více budov, tj. více segmentů sítě. Navíc segmenty mohou vznikat i zanikat. V tu chvíli je bezpečnost sítí těžko řešitelná a XACML by měl pomoci tuto situaci řešit. Navíc musíme počítat třeba v případě zdravotnického zařízení se zcela vnějšími uživateli, jako jsou pacienti, lékaři z jiných nemocnic, pracovníci pojišťoven atd, Vedle tohoto motivu pro vývoj XACML je motiv rozšiřování XML jako obecného mechanismu pro výměnu dat.
XACML (Extensible Access Control Markup language) je iniciativa vedená skupinou OASIS [3] určená na vyjádření bezpečnostní politiky pro přístup (autentizaci a autorizaci) k XML dokumentům a datovým zdrojům. Souvisí se SAML (viz dále) a to tak, že SAML poskytuje mechanismus pro šíření autentizačních a autorizačních informací mezi servery a službami, zatímco XACML je autentizační a autorizační informací. Idea XACML je ta, že XML dokument nebo samotný SOAP vzkaz může popisovat politiku přístupu, tj. kdo má mít přístup k čemu atd.
Cílem je standardizovat jazyk pro popsání autentizace a přístupových politik v XML syntaxi. Standardní jazyk pro kontrolu přístupu vede k nízkým nákladům, protože není potřeba vyvíjet jazyk pro určitou aplikaci nebo psát politiky kontroly přístupu ve více jazycích. Pomocí XACML je možné vytvářet politiky kontroly přístupu z těch, které byly vytvořeny jinými stranami. XACML definuje slovník pro specifikaci předmětu, práv subjektu a podmínek. Jeden standardní jazyk pro řízení kontroly přístupu tak může nahradit několik jiných jazyků jednotlivých aplikací. XACML je OASIS standard, který popisuje jednak jazyk pro psaní politik a jednak pro psaní dotaz / odpověď (obojí je napsané v XML). Jedna XACML politika může pokrýt mnoho zdrojů, to zabrání nekonsistentním politikám. XACML dovoluje jedné politice odkazovat na jiné, to je důležité pro velké organizace.

Politika a množina politik

Kořenem XACML politik je Politika (Policy) nebo Množina politik (PolicySet). Množina politik je kontejner, který může obsahovat další politiky (Policy) nebo množinu politik (PolicySet) a také odkazy na politiky v okolí (vzdálené politiky). Politika reprezentuje jednotlivou politiku řízení přístupu, vyjádřenou pomocí množinu pravidel.(set of Rules) Každý XACML dokument s politikou obsahuje právě jednu politiku nebo PolicySet jako „kořenový XML tag“ tj. kořen daného XML.
V typickém XACML scénáři subjekt (tj. uživatel nebo pracovní stanice) chce udělat

nějakou akci na konkrétním zdroji. Subjekt podá žádost entitě, která chrání zdroj (např. web server, souborový systém atd.). Tato entita se nazývá PEP (Policy Enforcement Point). PEP zformuluje dotaz (za použití XACML jazyka) založený na atributech subjektu, akci, požadovaném zdroji a dalších informacích náležejících dotazu. PEP potom pošle tento dotaz PDP (Policy Decision Point), který se podívá na dotaz a nějakou politiku, kterou aplikuje na dotaz a přijde s odpovědí, zda může být přístup povolen. Odpověď vyjádřená v XACML jazyku je vrácena PEP, který pak povolí nebo zakáže přístup. PEP i PDP mohou být buď v jednotlivé aplikaci nebo mohou být distribuovány na několika serverech. Navíc k poskytnutí dotaz / odpověď a jazyku politik, XACML také poskytuje další prostředky, speciálně nalezení politiky, která se aplikuje pro daný dotaz a ověření, zda porovnáním žádosti a dané politiky je přístup povolen nebo ne.
PDP a PEP jsou tedy 2 kořenové konceptuální prvky XACML modelu. PDP je zpracovávací „stroj“, který chápe, jak vyhodnotit politiku založenou na dotazu. PEP je (typicky) element specifický pro aplikaci, který si fyzicky vynucuje přístup ke zdroji a generuje dotaz (žádost) na PDP.
Jak PDP a PEP komunikují? Záleží na modelu. V některých systémech jsou PDP a PEP umístěny ve stejné aplikaci. V jiných jsou separovány, ale stále ještě na stejném stroji, jinde mohou být rozmístěny v síti. V každém z těchto případů můžeme najít užití standardních formátů dotazu a odpovědi nebo nějaké uživatelské prezentace.



Obr. 1- hlavní komponenty XACML

SAML

SAML (Security Assertion Markup Language of Structured Information) tento jazyk je vyvíjen OASIS. [1] Cílem příslušné skupiny OASIS je vyvinout standard pro výměnu autentizačních a autorizačních informací. SAML umožňuje přechod autentizačních a autorizačních informací mezi zúčastněnými stranami a poskytuje tzv. "prosazení " důvěry. Tato aplikace může „prosadit", že jde o určitého uživatele a ten má navíc určitá privilegia. SAML dokument může být digitálně podepsán pomocí XML signature nebo /a zašifrován pomocí XML šifrování. SAML poskytuje distribuci informace mezi určitou platformou a organizací a je proto jedno, kolik bodů prochází. Jedná se tedy např. o systém jednoho přihlášení. Např. nějaký portál autentizuje Alici a ví, že Alice má určitou roli. Portálová aplikace toto připojí např. do tvrzení v SOAP zprávě s dotazem na další webovou službu. Další webová služba se podívá na portálovou identitu, ověří digitální podpis portálu a povolí nebo zakáže přístup uživatele vzhledem k jeho roli.

SAML je systém založený na XML pro výměnu autentizačních a autorizačních informací. SAML dovoluje entitám udělat tvrzení týkající se identity, atributů a práv subjektů (obvykle se jedná o lidského uživatele) k druhým entitám, jako je partnerská organizace nebo nějaká aplikace.

Výhody SAMLu

Je to neutralita vzhledem k platformě. Bezpečnost je více nezávislá na aplikační logice. SAML nevyžaduje, aby informace o uživatelích byly udržovány a synchronizovány mezi adresáři. SSO uživatelé se mohou identifikovat u Identity providera (IP) bez následné autentizace u Service providera (SP). Identity federation se Samlem dovoluje bez porušení soukromí využít služby uživatelům přesně na míru. SAML dovoluje redukovat náklady ohledně násobné autentizace, protože toto břemeno je přesunuto na IP. Riziko je přesunuto na IP, který dělá management identity. To je lépe , než aby to dělal SP.

Technologie SAMLa její části – tvrzení, protokoly, spojení, profily [8]

1. definuje syntaxi a sémantiku XML zpráv obsahujících **tvrzení** (assertion) ve formě XML, je to tedy jazyk na vytváření tvrzení o identitách. Jde o autentizační a autorizační tvrzení.
2. definuje **protokoly** žádostí a odpovědí mezi žádající a vydávající stranou pro výměnu bezpečnostních informací. Tedy protokol, kterým je možno takové tvrzení ověřit u důvěryhodné autority. Autorizační požadavek: má určitá identita právo přistupovat k tomuto zdroji? Odpověď: tato identita má právo číst zdroj. Je zde řada možností, např. zda jméno identity je zaregistrované, zda není povolení k její činnosti již ukončené, a mnoho dalších.
3. definuje pravidla pro užití tvrzení se standardy pro **transport**, např. definuje, jak SAML tvrzení můžeme transportovat ve zprávě SOAP přes HTTP. Jde tedy o napojení na přenosový protokol. Může jít o HTTP, SMTP, JMS nebo SOAP.
4. **Profil** definuje omezení či rozšíření použití SAMLU pro nějakou aplikaci. Např. Web Browser SSO profil specifikuje, jak SAML autentizační tvrzení je použito při komunikaci mezi IP a SP. Web SSO profile definuje, jak SAML

dotaz/ odpověď protokol se použije v kombinaci s http Post, apod.

Tvrzení SAMLu neprovádějí autentizaci, SAML ani nedefinuje nějaký nový způsob autentizace či autorizace, ale SAML slouží k obalení, zapouzdření tohoto procesu autentizace a jeho přenosu. Cílem SAMLU je tedy poměrně široké řešení bezpečnosti, které jde využít v rozsáhlém IS, kde jde o to zajistit autentizaci a autorizaci identit v tomto IS, samozřejmě musí zde existovat nějaký mechanismus na ověření tvrzení jazyka SAML. Je tak možno vytvořit ověřenou digitální identitu, která jde použít v jiném systému. Je ovšem nutné, aby si oba systémy důvěřovaly navzájem nebo aby zde existovala autorita, které důvěřují oba systémy.

.

SAML tvrzení- assertion

Je to XML dokument, který obsahuje bezpečnostní informace. Existují celkem 3 možnosti výroků o subjektu, což může být osoba nebo program, tj. zde se užívá subjekt ve smyslu bezpečnostních modelů. Assertion může obsahovat všechny tyto výroky najednou nebo nemusí. SAML definuje celkem 3 typy tvrzení.
Jsou to autentizace, atribut a autorizace.

1.Autentizace - definuje , jakým způsobem byla identita autentizována, např. heslem a jménem, certifikátem X.509 aj. Toto tvrzení typicky vydává SAML autorita zvaná identity provider.
2.Tvrzení o vlastnostech- atribut. To je další dodatečná informace o identitě - např. ročník studenta apod. Informace o lékaři, z jakého je oddělení atd..
3.Rozhodnutí o autorizaci (anglicky Authorization Decision)
Tvrzení o tom, ke kterým zdrojům údajů a služeb má identita právo přistupovat a jakým způsobem - číst, psát, zapisovat atd.

Všechna tvrzení mohou obsahovat informace o verzi tvrzení, definici identity, popis vydávající autority, XML podpis tvrzení atd.

Použití SAMLu

Protože SAML tvoří základ pro komunikační bezpečnost a informace o identitě, je jeho použití možné v řadě různých způsobů. Dále jsou některé nejvýznamnější.

Jediné přihlášení SSO – uživatel se přihlásí na tom.com a je autentizován. Později se chce přihlásit na joe.com. Bez užití SSO by musel své údaje zadávat znovu. Pokud je užit SAML, pak joe.com pošle požadavek na tom.com s dotazem, zda se již uživatel na tom.com autentizoval. tom.com odpoví prohlášením, že ano, uživatel je autentizován. Poté joe.com zpřístupňuje své zdroje, aniž vyžaduje znovu přihlašovací informace. SAML dovoluje web SSO skrytě komunikaci pomocí autentizačních tvrzení, které se posílají z 1. web site na další.

Autorizace založená na atributech

Podobně jako SSO informace o subjektu se posílají od jednoho web situ k druhému. Zde ale jde o informace o atributech přihlášeného spíše než o konkrétní identitu. Nebo je přímo žádoucí, aby konkrétní identita byla skryta, ale aby bylo jasno, že jde o studenta té a té university apod., lékaře té a té kliniky.

Distribuovaná transakce
Uživatel použije přihlášení k nějaké službě na www.a1.cz a poté chce jinou službu od www.b1.cz. Uživatel poté může předat informace o svém profilu na www.a1.cz serveru www.b1.cz. Ten pošle tzv. SAML tvrzení serveru www.a1.cz, ve kterém bude chtít veškeré informace, které o uživatelovi má. www.a1.cz tyto informace pošle ve formě tzv. tvrzení.

Autorizační služba
Pomocí této služby je možno zasílat tvrzení, zda je někdo k něčemu autorizován, např. k platbě, objednání atd. Tedy někdo chce provést za univerzitu nějakou akci. Poskytovatel služby požádá danou universitu o potvrzení, zda daný uživatel smí akci za univerzitu provést.

SAML je velmi flexibilní a je určen k použití dalším standardům. Jedná se např. o Liberty Aliance, Shibboleth projekt a WS security. Tyto aplikace použily právě SAML

Příklad na spolupráci XACML a SAML u Access control problému

Je mnoho způsobů možné spolupráce SAML a XACML. Záleží na tom, kde jsou např. umístěny jednotlivé části XACML, jak daná aplikace vypadá a na mnoha dalších rysech. Následující hypotetický příklad ukazuje situaci, kdy se používá SAML i XACML, předpokládejme, že jde o přístup k nějaké aplikaci. Používá se proxy a dále v aplikaci existuje tzv. autentizační a autorizační server (AA). Úlohou proxy je poskytnout jeden vstupní bod do systému. Dále je proxy použit jako autentizační bod. Po zadání jména a hesla proxy pustí uživatel k té aplikaci, kam má právo. Kam je tzv. autorizován. Atributy uživatele a připojené bezpečnostní politiky jsou na AA.

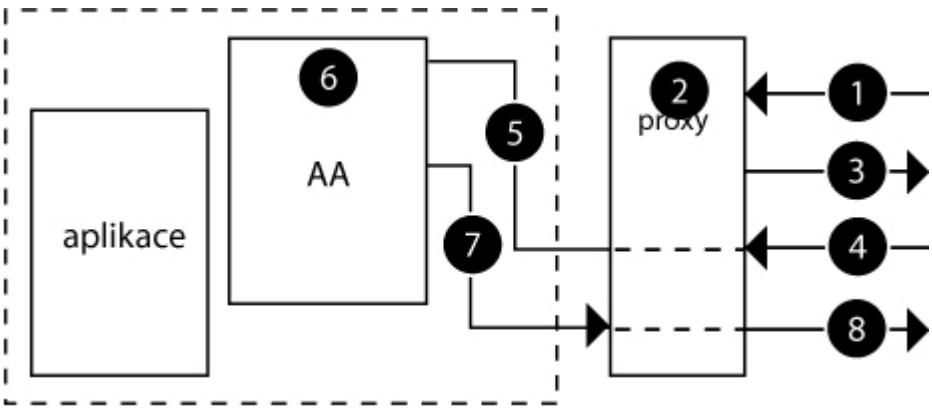
V této souvislosti je nutno řešit následující otázky:

1. Kdo rozhoduje, zda uživatel má mít přístup ke nějakému zdroji? Je to proxy, AA nebo zdroj sám?
2. Jak se rozhodnutí udělá?
3. Jak je definována přístupová politika? V jaké podobě? Kde se uchovává?
4. Jak je politika vynucena? Kdo ji vynucuje – proxy, AA nebo zdroj?
5. Jak jsou posílány bezpečnostní informace? Jak vypadá bezpečnostní vzkaz? Jakou má podobu? Kdo ho posílá a komu?
6. Jak jsou tyto bezpečnostní informace samy zabezpečeny?

Konkrétní uspořádání záleží na tom, zda je použit SAML a XACML nebo pouze SAML. Zda jsou použity nějaké přídavné části. Dále na tom, jak jsou rozmístěny jednotlivé části XACML (PEP atd...). Dále záleží na tom, které profily ze SAMLU jsou použity.

Model má 2 části a to autentizační a a autorizační. Každá část je řešena zvlášť. Autentizace probíhá od okamžiku prvního kontaktu uživatele se systémem do výběru aplikace, s kterou chce pracovat. Autorizační část řeší akce, jak a zda je uživateli povolen či odepřen přístup ke zdrojům. V tomto modelu AA je považován za PDP

(viz dříve). Proxy je považován za PEP. Aplikace sama není SAML ani XACML. K AA je připojena PAP, která v sobě obsahuje základní XACML politiky.[6]



Obr.2 Autentizace

Politika je napsána v XACML formátu a jsou v ni vyjádřeny kdo smí dělat v aplikaci jakou operaci.
Dále jsou krátce popsány body autentizace.

1. Uživatel přistupuje na portál, přes který se chce dostat do aplikace. Stránka portálu (případně webová aplikace) je umístěna na proxy. Tj. uživatel napíše např. http://www.nejakaorganizace.cz/aplikace
2. Proxy dostává http dotaz se žádostí o hlavní stránku portálu. Za předpokladu, že se jedná o 1. návštěvu, tak nejsou ještě uloženy coocies.
3. Proxy žádá uživatele o autentizace. Jméno a heslo nebo smart karta nebo jiný způsob autentizace. Proxy posílá uživatelovi stránku s formulářem pro vyplnění jména a hesla.
4. Uživatel posílá zpět heslo, jméno apod. Uživatel vyplní formulář a pošle ho.
5. Proxy dostane tato data ve formě HTML formuláře, dostane z něj heslo a jméno a pošle je AA. Jak probíhá přesně transport, zde neřeším.
6. AA dostane heslo a jméno, ověří korektnost. Metodu opěr nerozebírám. AA vygeneruje tvrzení o autentizaci v jazyce SAML. Toto tvrzení je podobné dříve uvedenému.
7. Výsledek autentizace je poslán nazpět do proxy. Dále je poslán seznam zdrojů a uživatel si z nich vybere.
8. Na základě minulých akcí, proxy formuluje odpovídající HTML obsah a pošle ho zpět prohlížeči uživatele.

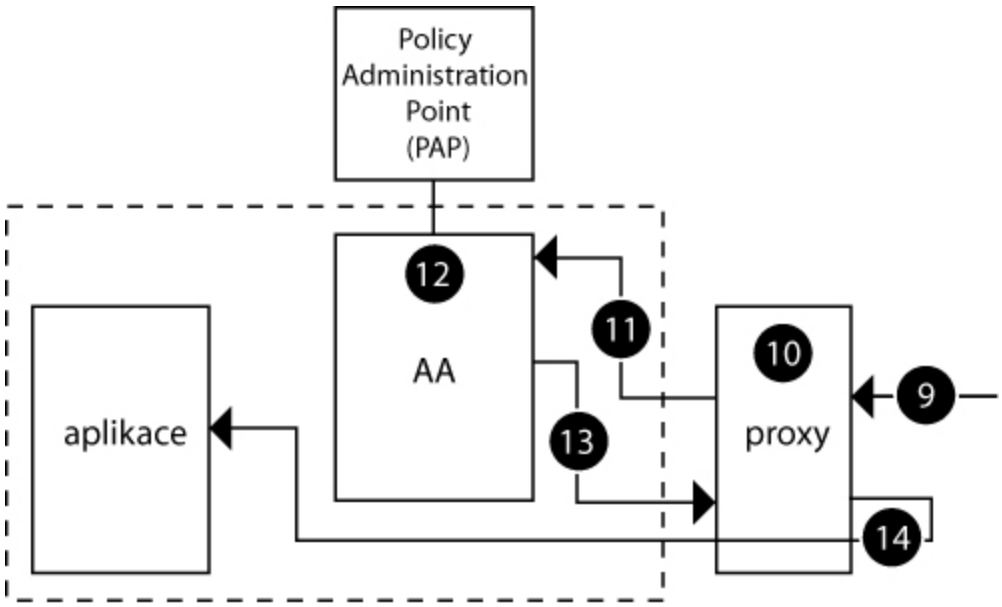
Autentizační tvrzení, které vygeneruje AA

<saml:Assertion …
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion“
MajorVersion="1" MinorVersion="1"
AssertionID="....."
Issuer=" https://www.nejakaorganizace.cz/AA/."
IssueInstant="....">
<saml: Conditions

```
        Not Before="....."
        NotOnOrAfter="....."/>
<saml:AuthenticationStatement
AuthenticationMethod="password"       (pomocí autentizace M- zde heslem)
AuthenticationInstant="2006-12-03T10:02:00Z">     (Čas T)
                <saml:Subject>        (Subjekt   Dagmar)
                        <saml: NameIdentifier
                        Format="X509SubjectName">
                        Dagmar
                        </saml: NameIdentifier>
                </saml:Subject>
        </saml:AuthenticationStatement>
</saml:Assertion>
```

Autorizace

9. Uživatel vybere aplikaci, ke které chce přistupovat. Žádost jde přímo na proxy.
10. Proxy obdrží žádost (HTTP GET), poté zkonstruuje XACML žádost. Pro nedostatek prostoru zde XACML žádost neuvádíme, jedná se opět o konstrukci v XACML jazyku, kde je popsáno kdo chce přistupovat k jakému zdroji a jakou dělat akci. Tedy proxy je to, co komunikuje s uživatelem.
11. Proxy posílá XACML žádost do AA
12. AA obdrží XACML žádost, konzultuje PAP, který je připojený k AA a hledá se politika, relevantní k příchozí žádosti. Na základě nalezené politiky je určena odpověď.
13. Odpověď je odeslána zpět do proxy.
14. Na základě odpovědi proxy buď dovolí nebo zakáže přístup k požadovanému zdroji.



obr.3 Autorizace

Proxy zde tedy působí jako PEP. SAML procesy jsou zcela izolovány od aplikace. Politiky jsou popsány v XACML:Na druhou stranu, pokud by proxy z nějakého důvodu zhavaroval, tak žadatel má přímý přístup k aplikaci.

Příklad politiky

Dále je příklad jednoduché politiky (Policy), která používá rysy diskutované výše. Takto zhruba by mohla vypadat politika pro přístup ke zdroji. Její Target říká, že tato Policy je aplikována pouze na žádosti pro https://www.nejakaorganizace.cz/apl. Do aplikace na dané adrese nesmějí zapisovat subjekty z kategorie data administrators. Čili zdroj (resource) nesmí být zpřístupňován k zápisu (action) danému subjektu (data administrátoři)

```
<Policy PolicyId="SamplePolicy"
        RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
algorithm:deny-overrides">
   <Target/>
    <Rule RuleId =" ….. " Effect =" Deny">
   <Target>
 <Subjects>
   <Subject>
     <SubjectMatch
         MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
       <AttributeValue
       DataType="http://www.w3.org/2001/XMLSchema#string">
                          Data Administrators
        </AttributeValue>
       <SubjectAttributeDesignator
        DataType="http://www.w3.org/2001/XMLSchema#string"
                              AttributeId="urn:oasis:names:tc:xacml:1.0:subject-category"/>
       <SubjectAttributeDesignator
     </SubjectMatch>
   </Subject>
</Subjects>

    <Resources>
       <Resource>

     <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
       <AttributeValue
       DataType="http://www.w3.org/2001/XMLSchema#stringAny URI">
                          https://www.nejakaorganizace.cz/apl
       </AttributeValue>
       <ResourceAttributeDesignator
        DataType="http://www.w3.org/2001/XMLSchema#stringAny URI"
         AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"/>
       </ResourceAttributeDesignator>
     </ResourceMatch>
       </Resource>
   </Resources>
```

```
<Actions>
   <Action>
     <ActionMatch
      MatchId ="urn:oasis:names:tc:xacml:1.0:function:string-equal">
       <AttributeValue
          DataType="http://www.w3.org/2001/XMLSchema#string">
          write
         </AttributeValue>
       <ActionAttributeDesignator
          AttributeId =" urn:oasis:names:tc:xacml:1.0:action : action-id">
         DataType="http://www.w3.org/2001/XMLSchema#string""/>
      </ActionMatch>
     </Action>
    </Actions>
  </Target>

 </Rule>
  </Policy>
```

V politice by dále třeba mohly být různé podmínky, kdy je daná akce povolena či zakázána. Mohou zde existovat různá nastavení pro různé skupiny subjektů, odkud je daná akce povolena, např. pouze při logování z určitého zdroje apod. XACML je jazyk velmi rozmanitý, který se dále vyvíjí. Samozřejmě celá politika ale musí být nastavena až po podrobném rozbory situace, kdo kdy odkud v jakou dobu má mít jaké akce povoleny či zakázány.
Pro použití ve zdravotnictví by tak mohly být povolené akce pro lékaře z různých oddělení, pro pacienty a to jednotlivě či různé skupiny. Akce by mohly být povoleny či zakázány v danou denní dobu, v dané datum, v časovém rozmezí apod. Díky možné přesné specifikaci zdroje by tak mohla být napsána velmi podrobná politika resp. politiky pro přesně nastavený přístup.

V jakém stavu je vývoj těchto jazyků?

SAML

V1.0 OASIS standard - listopad 2002. V1.1 září 2003. SAML měl velký úspěch ve vysokém školství, zdravotnictví státní správě, a dalších průmyslových segmentech. Byl široce implementován všemi poskytovateli webových aplikací. V2.0 draft 12. duben 2005 [1], tato verze, má celou řadu nových rysů týkající se např. možnosti šifrovat části tvrzení, řadu nových věcí týkající se ochrany soukromí, apod.
Implementace technologie SAML
Open Saml je to open source implementace jazyka SAML. Vyvíjí ji konsorcium Internet2. Open SAML má jednak verzi v C ++ a jednak v Jave. [9]

XACML v. 2

Posledni verze 2.0 [2], OASIS Standard 2005, dnes práce na V.3. Také existuje open source implementace XACML (Sun).[10]

Závěr

SAML a XACML jsou poměrně nové a stále se vyvíjející jazyky založené na XML. Jejich použitím a propojením vzniká možnost vyřešit složité otázky přístupu ke zdrojům. Některé části již byly úspěšně použity např. v projektu Shibboleth. Jejich nasazení by mohlo pomoci řešit zásadní problémy digitální identity.

[1] http://www.oasis-open.org/specs/index.php#samlv2.0 15.3.2007
[2]http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml 15.3.2007
[3] http://www.oasis-open.org/specs/index.php#xacmlv2.0 15.3.2007
[4] http://www-128.ibm.com/developerworks/xml/library/x-xacml/ 15.3.2007
[5] http://linux456.vsb.cz/~las034/wss/wssecurity.pdf 15.3.2007
[6] Asem Hassan, Conceptual Design of Identity Management in a profile/based access control, Hamburg University of Technology, 2006
[7] http://www.inforum.cz/inforum2006/pdf/Brechlerova_Dagmar.pdf
[8]http://www.oasis-open.org/committees/download.php/13525/sstc-saml-exec-overview-2.0-cd-01-2col.pdf 20.4.2007
[9] http://www.opensaml.org/ 20.4.2007
[10] http://sunxacml.sourceforge.net/ 20.4.2007

Práce je podporována projektem AVČR 1ET2003004 - Informační technologie pro rozvoj kontinuální sdílené péče o zdraví