MOŽNOSTI ŘEŠENÍ BEZPEČNÉHO PŘÍSTUPU POMOCÍ XML SECURITY

Dagmar Brechlerová
PEF KIT ČZU, Praha 6, Suchdol, brechlerova@pef.czu.cz

ABSTRAKT:
Otázky identifikace uživatele, autentizace a autorizace jsou v rozsáhlých systémech velmi problematické. Příspěvek se zabývá řešením tohoto problému pomocí XML security, kde existují části SAML a XACML, které tomuto účelu slouží. Příspěvek popisuje tyto nové technologie, jejich využití a propojení.

KLÍČOVÁ SLOVA:
SAML, XACML, autentizace, autorizace

ABSTRACT
Problems of identification, authentication and authorization are very complex in large systems. This paper deals with solution of this problem by XML security. Special languages (parts of XML security) exist here, they are SAML and XACML, and they are intended for this reason. These new technologies are presented here, usage of SAML and XACML, communication SAML and XACML. One model of authetication and authorization by XACML and SAML is presented here, too.

KEY WORDS
SAML, XACML, authentication, authorization

ÚVOD

V rozsáhlém informačním systému, do kterého mají přístup jak vnitřní uživatelé (zaměstnanci, studenti..), tak případně vnější uživatelé (zákazníci, studenti jiných škol..) je jedním ze zásadních problémů řízení přístupu ke zdrojům. Jde o to, povolit každému přístup jen k tomu zdroji, kam je to záhodno. Pokud se jedná navíc třeba o osobní data, mohou mít přístup pouze lidé, k tomu pověření, aby se organizace nedostala do rozporu se zákonem, Nastavit tato práva, tj. kdo, kdy, odkud, kam může přistupovat a jaké tam navíc může provozovat akce, je velmi složité. V XML security [7] jsou k tomu určeny 2 technologie a to jazyk XACML a jazyk SAML. Jejich kombinací lze dosáhnout velmi podrobné autentizace a autorizace. Dále jsou popsány tyto dvě technologie a jejich propojení.

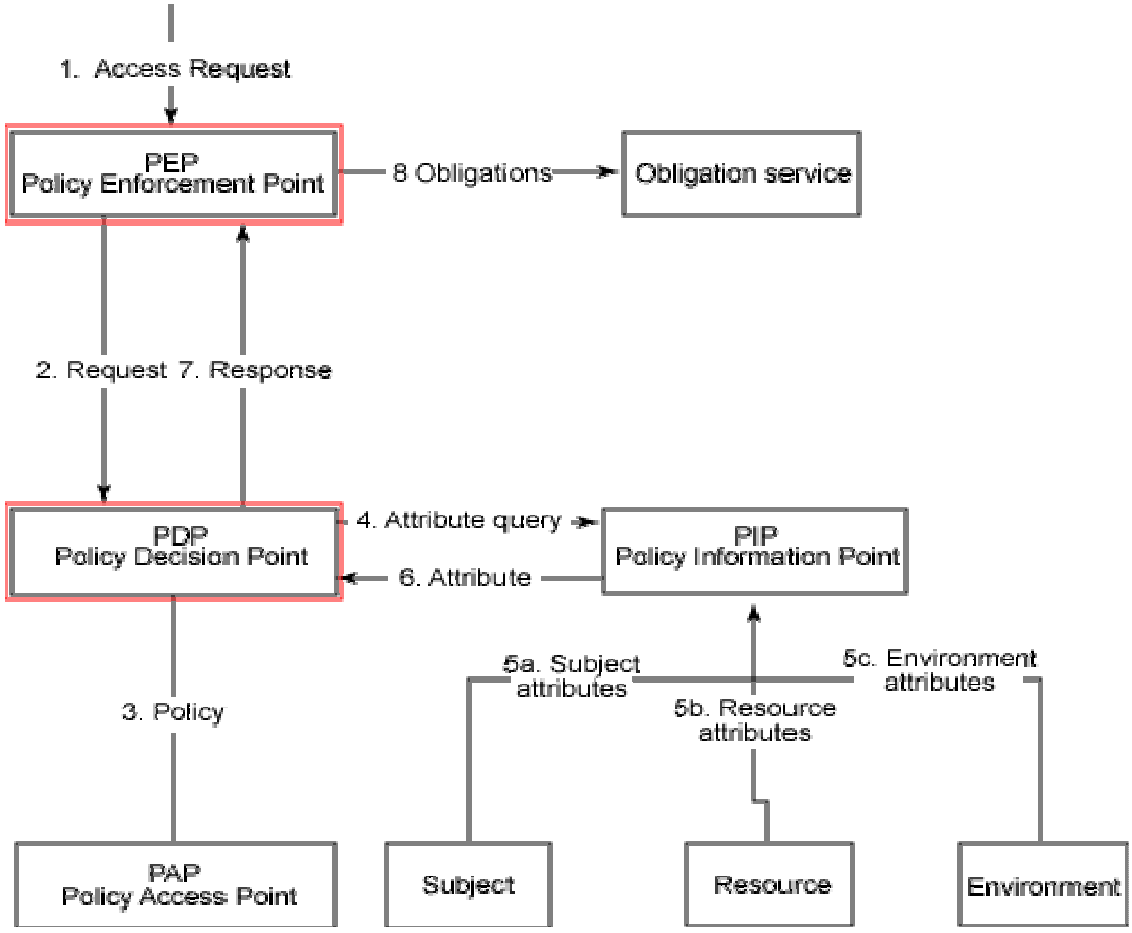
XACML posledni verze 2.0 [2]

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 čemu, kdy, jaká konkrétní práva atd. Pokud jsou naše zařízení 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. 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 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 speciálním 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 části tohoto vztahu, 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

Příklad politiky

Dále je příklad jednoduché politiky (Policy), která používá rysy diskutované výše.
Její Target říká, že tato Policy je aplikována pouze na žádosti pro server „Sample Server“. Policy má pravidlo (Rule) s Targetem, který vyžaduje akci „Login“ a Condition, který je aplikován pouze, jestliže Subject se pokouší logovat mezi 9 ráno a 5 odpoledne. Tento příklad může být rozšířen tak, aby zahrnul další pravidla pro jiné akce. Pokud není první pravidlo aplikováno, pak je použito default Rule, které vždy vrací Deny (Rules jsou vyhodnoceny po řadě.)

```
<Policy PolicyId="SamplePolicy"
        RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">

   <!—Tato  Policy  se aplikuje pouze na zadost pro the SampleServer -->
   <Target>
    <Subjects>
      <AnySubject/>
    </Subjects>
    <Resources>
      <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
        <AttributeValue
```

```
DataType="http://www.w3.org/2001/XMLSchema#string">SampleServer</AttributeValue>
        <ResourceAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"
                                AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"/>
      </ResourceMatch>
    </Resources>
    <Actions>
      <AnyAction/>
    </Actions>
  </Target>
 <!—Toto Rule se podívá, zda se smí Subject prilogovat -->
  <Rule RuleId="LoginRule" Effect="Permit">
    <!—Pouiziva se toto  Rule, pokud je akce  login -->
    <Target>
      <Subjects>
        <AnySubject/>
      </Subjects>
      <Resources>
        <AnyResource/>
      </Resources>
      <Actions>
        <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">login</AttributeValue>
          <ActionAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"
                                AttributeId="ServerAction"/>
        </ActionMatch>
      </Actions>
    </Target>
    <!—Dovoluje se logovani pouze mezi 9am a 5pm -->
    <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-greater-than-or-equal"
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-one-and-only">
          <EnvironmentAttributeSelector
DataType="http://www.w3.org/2001/XMLSchema#time"
                                  AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-
time"/>

        </Apply>
        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#time">09:00:00</AttributeValue>
      </Apply>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-less-than-or-equal"
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-one-and-only">
          <EnvironmentAttributeSelector
DataType="http://www.w3.org/2001/XMLSchema#time"
                                  AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-
time"/>
        </Apply>
```

```
        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#time">17:00:00</AttributeValue>
      </Apply>
    </Condition>

  </Rule>
  <!—Zde se daji vlozit jina Rules pro jine akce -->

  <!-- A final, "fall-through" Rule that always Denies -->
  <Rule RuleId="FinalRule" Effect="Deny"/>

 </Policy>
```

SAML poslední verze 2.0 2005 [1]

SAML umožňuje přechod autentizačních a autorizačních informací mezi zúčastněnými stranami. SAML 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. SAML poskytuje distribuci informace mezi určitou platformou a organizací a je proto jedno, kolik bodů prochází. Jedná se tedy o systém jednoho přihlášení. Např. nějaký portál autentizuje Alici a ví, že Alice má určitou roli. Portálová aplikace toto připojí do tvrzení v SOAP zprávy 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 Security Assertion Markup Language of Structured Information: tento jazyk je vyvíjen OASIS.[3]. Cílem příslušné skupiny OASIS je vyvinout standard pro výměnu autentizačních a autorizačních informací. Jedním slovem SAML je konstrukce, systém založený na XML pro výměnu autentizačních a autorizačních informací.

Má 3 části:
1. definuje syntaxi a sémantiku XML zpráv obsahujících tvrzení (assertion) ve formě XML.
2. definuje protokoly žádostí a odpovědí mezi žádající a vydávající stranou pro výměnu bezpečnostních informací
3. definuje pravidla pro užití tvrzení se standardy pro transport, např. definuje, jak SAML tvrzení můžeme transportovat ve zprávě SOAP přes http.

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.

SAML tvrzení- assertion – [5]
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. Jsou to autentizace, attribut a autorizace.

Dále je ukázána autentizace. Autentizace (anglicky authentication) — autentizační výrok

říká, že uvedený subjekt S byl autentizován konkrétními prostředky M v konktrétním čase T. Toto tvrzení je reprezentováno elementem <AuthenticationStatement>. Toto se používá pro umožnění SSO.

```
<saml:Assertion ...>
<saml:AuthenticationStatement
AuthenticationMethod="password"       (pomocí autentizace M- zde heslem)
AuthenticationInstant="2006-12-03T10:02:00Z">     (Čas T)
<saml:Subject>        (Subjekt   S)
<saml:NameIdentifier
SecurityDomain="sun.com"
Name="Sang" />
<saml:ConfirmationMethod>
http://...core-25/sender-vouches
</saml:ConfirmationMethod>
</saml:Subject>
</saml:AuthenticationStatement>
</saml:Assertion>
```

Autentizační výrok říká, že subjekt Sang v bezpečnostní doméně sun.com byl autentizován v čase T 3. prosince 2006. Jako prostředek autentizace bylo užito jméno a heslo.

XACML a SAML: Jak jsou si podobné a jak jsou odlišné? [4]

XACML architektura a SAML architektura spolu pevně souvisejí, dotýkají se jedna druhé. Obě spolu sdílejí množství stejných koncepcí a sfér působnosti - autentizace, autorizace, řízení přístupu. Avšak problémy, které obě architektury řeší, jsou odlišné. SAML řeší autentizaci a poskytuje mechanismus pro přechod autentizačních a autorizačních rozhodnutí mezi spolupracujícími entitami, XACML se soustřeďuje na mechanismus návratu autorizačních rozhodnutí. Zatímco SAML poskytuje mechanismus pro vytvoření autentizačních a autorizačních tvrzení (výroků) a mechanismus pro jejich přesun, XACML poskytuje jazyk, který definuje pravidla nutná pro vytvoření autorizačního rozhodnutí.

SAML standard poskytuje rozhraní, které dovoluje třetí straně poslat žádost o autentizaci nebo autorizaci. Ale jak se s touto žádostí interně zachází, to je věcí XACML. XACML ale neřeší jenom způsob zacházení s autorizační žádostí, ale definuje mechanismus pro vytvoření kompletní infrastruktury pro pravidla, politiky a množiny politik pro autorizační rozhodnutí.

Příklad na spolupráci XACML a SAML u Access control problému

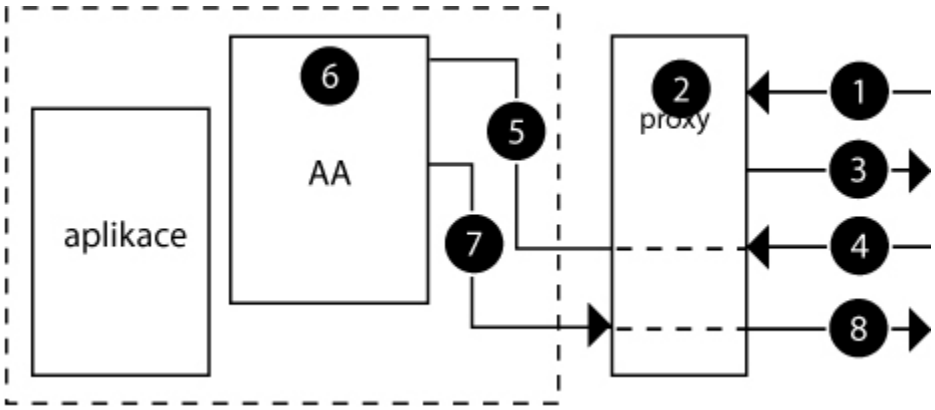
Je mnoho způsobů možné spolupráce SAML a XACML. Záleží na tom, kde jsou např. umístěny jednotlivé části XACML, jak daná aplikace vypadá a na mnoha dalších rysech. Následující hypotetický příklad ukazuje situaci, kdy se používá SAML i XACML, předpokládejme, že jde o přístup k nějaké aplikaci. Používá se proxy a dále v aplikaci existuje tzv. autentizační a autorizační server (AA). Úlohou proxy je poskytnout jeden vstupní bod. 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 zdroji- proxy, AAA nebo zdroj?
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? 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...).

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

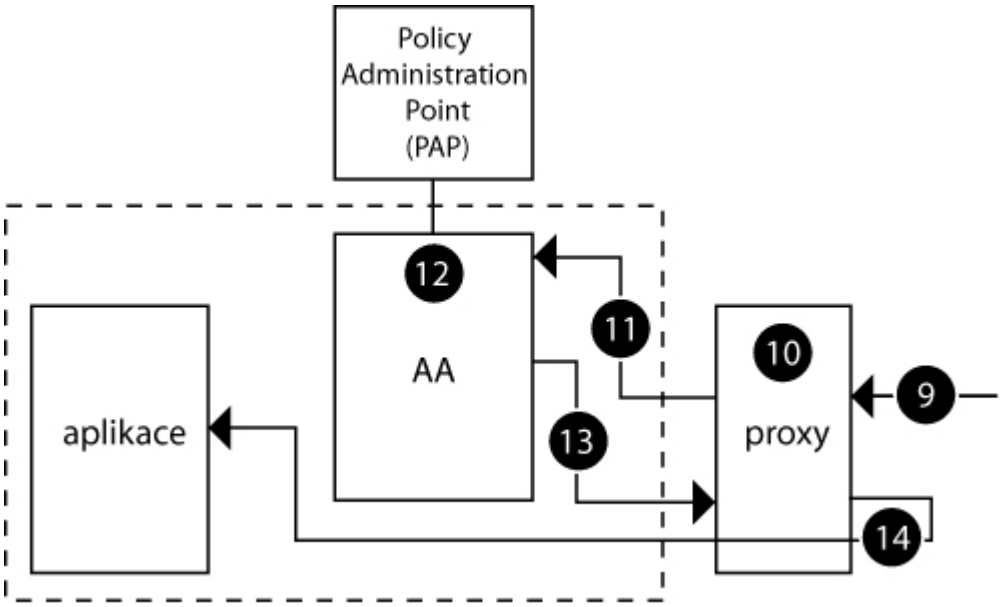
Politika je napsána v XACML formátu a jsou v ni vyjádřeny kdo smí dělat v aplikaci jakou operaci.
Dále jsou krátce popsány body autentizace.

1. Uživatel přistupuje na portál, přes který se chce dostat do aplikace. Stránka portálu je umístěna na proxy. Tj. Uživatel napíše např. http://www.mojefirma.com/apl
2. Proxy dostává http dotaz se žádostí o hlavní stránku portálu.
3. Proxy žádá uživatele o autentizace. Jméno a heslo nebo karta nebo jiný způsob autentizace.
4. Uživatel posílá zpět heslo, jméno apod.
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.

Autorizace

9. Uživatel vybere aplikaci, ke které chce přistupovat
10. Proxy obdrží žádost (HTTP GET), poté zkonstruuje XACML žádost.
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.

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.



obr.3 Autorizace

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. Obě technologie jsou ve velmi rychlém vývoji, ale již nyní se ukazují jako velmi slibné pro tuto oblast.

[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] Blake Dournaee, XML Security, McGraw-Hill Osborne Media, 978-0072193992, 2002