

Bezbednost Servisno Orjentisane Arhitekture

Aleksandar Nedeljković¹

Matematički fakultet, Studentski Trg 16
11000 Beograd, Srbija
mi09036@alas.matf.bg.ac.rs

Abstract. Servisno Orjentisana Arhitektura (SOA) je jedan od napopularnijih konceptata za implementiranje računarskih sistema. Medjutim suočava se sa velikim bezbednosnim izazovima i mnogim standardima koji pružaju sigurnost. U ovom radu diskutovaću o bezbednosnim rizicima Servisno Orjentisane Arhitekture koji su povezani sa vaznim aspektima sigurnosti kao što su autentifikacija, autorizacija, poverljivost, integritet i neporecivost. Takodje ću predstaviti mehanizme koji se koriste od strane pružaoca usluga za rešavanje sigurnosnih problema.

Keywords: *Sigurnost, Servisno Orjentisana Arhitektura (SOA), Kerberos, Infrastruktura javnog ključca (PKI), WS-Security, XML Digitalni potpis, SAML, XACML*

1. Uvod

Servisno Orjentisana Arhitektura (SOA) je jedan od najpopularnijih IT trendova, koji je definisan na sledeći način: *Servisno Orjentisana Arhitektura (SOA) je dizajn obrazac koji se sastoji od labavo spregnutih, vidljivih, višekratnih interoperabilnih platformskih usluga u kojima je svaka usluga dobro definisan standard. Svaka od ovih usluga može biti vezana ili nepovezana u bilo koje vreme i po potrebi* [5]. Medjutim, kao što je definisano, SOA ima labavo spojenu funkciju, koja čini SOA-u otvorenom na bezbednosne izazove. To znači da SOA mora da ispuni nekoliko zahteva. Glavni zahtevi su pretraživost servisa, autentifikacija servisa, proveru identiteta korisnika, kontrola pristupa, poverljivost, integritet, dostupnost i privatnost. Da bi se obezbedila sigurnost u labavo vezanom SOA okruženju, zajednica otvorenih standarda koja je kreirala veb servise razvila je niz bezbednosnih standarda, što je jedan od najaktivnijih i široko prihvaćenih implementacija Servisno Orjentisane Arhitekture.

Usluge su opisane na jeziku standardne definicije, imaju objavljen interfejs i komuniciraju jedni sa drugima tražeći izvršenje svojih operacija kako bi kolektivno podržali zajednički poslovni zadatak ili proces [4].

SOA je metodologija za postizanje interoperabilnosti aplikacija i ponovne upotrebe IT sredstava koje karakterišu jak arhitektonski fokus na idealnom nivou apstrakcije, postavljanje infrastrukture i biblioteka za višekratnu upotrebu (W3C definicija) [2]. Ona takodje obuhvata podršku za organizovanje i korišćenje resursa koji su pod kontrolom različitih administracija.

Neophodno je vreme da preduzeća brzo reaguju na poslovne promene sa efikasnošću i prednostima postojećih ulaganja u aplikacije i aplikacije infrastrukture za rešavanje novijih poslovnih zahteva. Rešenje koje se aktivno koristi za rešavanje ovih zahteva je Servisno Orjentisana Arhitektura koja omogućava preduzećima da priključe nove usluge ili nadogradnju postojećih usluga u zrnasti način za obradu novih poslovnih zahteva. SOA pruža

moгуćnost da se usluge koriste preko razliĉitih kanala i otkriva postojeća preduzeća i njihove aplikacije kao usluge, što je osnovni gradivni blok Servisno Orjentisane Arhitekture.

2. Sigurnosni problemi

Priroda SOA-e ĉini sisteme podložnijim bezbednosnim pretnjama zbog sledećih razloga.

1. Servisi mogu na primer da daju pristup desktop aplikaciji i neke aplikacije koje normalno ne bi bile dostupne klijentu zbog servisnog interfejsa su javno izložene celom svetu. Vlasnik ima najmanju kontrolu ko može da koristi uslugu.
2. Podaci su izloženi širokom spektru korisnika. Zaštita podataka tokom tranzita i skladištenja je važna zbog obezbeđivanja integriteta podataka i privatnosti.
3. Podaci putuju kroz heterogene sredine, imajući različite politike, tehnologije i mrežne protokole. Stoga je teško integrisanje i udruživanje razliĉitih mera bezbednosti za implementaciju.
4. Povezivanje kod Servisno Orjentisane Arhitekture nije od taĉke do taĉke (eng. *"point-to-point"*)
5. Ovaj sistem je osetljiv na ponovne napade koji jednostavno ponavljaju važeću potpisanu poruku i dobijaju neovlašćeni pristup.

2.1. Sigurnosna pitanja

Osnovna pitanja bilo koje bezbednosne infrastrukture su autentifikacija, autorizacija i poverljivost. Fokus se proširuje tamo gde su infrastrukture izložene kao u sluĉaju SOA-e. U ovom sluĉaju to su problemi neporecivosti i integriteta poruke.

1. **Autentifikacija** – Predstavlja proveru validnosti identiteta subjekta. Subjekt može biti korisnik, veb servis, računar ili aplikacija. Autentifikacija je prvi korak u kontroli prava pristupa. Da bi se omogućila kontrola prava pristupa neophodno je da sistem identifikuje subjekat i da poseduje izvestan nivo uverenja u autentiĉnost identiteta subjekta. Medjusobna autentifikacija predstavlja dvosmernu autentifikaciju i omogućava dokazivanje identiteta obe strane uključene u komunikaciju.
2. **Autorizacija** – Predstavlja odredjivanje prava i dozvola koje korisnik poseduje. Nakon autentifikacije, provere identiteta, sistem mora odrediti prava koja korisnik poseduje i koje su njegove mogućnosti korišćenja sistemskih resursa.
3. **Poverljivost** – Pošto se podaci dele preko razliĉitih usluga i preko razliĉitih domena postoje velike šanse da su podaci izloženi neželjenim primaocima uprkos strogim merama zaštite podatka.

4. **Integritet** — Validnost integriteta podataka predstavlja tehnologiju dokazivanja da poruka nije menjana od strane neautorizovanog subjekta. Zbog mogućnosti zloupotreba poruka preko TCP/IP mreža neophodno je koristiti digitalne potpise, autentifikacione kodove ili hash algoritme da bi se potvrdila validnost integriteta poruke.
5. **Nemogućnost poricanja** — Posledica digitalnog potpisivanja poruke i predstavlja legalni dokaz da je subjekat potpisao poruku. Digitalni potpis kriptografski vezuje identitet potpisivača sa sadržajem potpisanih podataka. Korišćenje kriptografije javnim ključem omogućava neoporecivost pošiljaoca da je on potpisao poruku.

2.2. Tehnologija koja narušava SOA bezbednost

SOA je pod uticajem primene ranjivosti Veb aplikacija koje su obično izgrađene na vrhu veb protokola. Klasične bezbednosne ranjivosti su one koje se mogu iskoristiti bez upotrebe novijih veb tehnologija. Predstaviću neke od uobičajenih veb bezbednosnih propusta koji utiču na SOA aplikacije.

1. **SQL injekcije** — omogućavaju napadaču da izvršava neautorizirane SQL upite iskorištavajući neproveravanje ulaza u veb aplikacijama koje grade dinamičke SQL upite. Ranjivost je prisutna kod korisničkog unosa koji se nekorektno proverava.
2. **XML Spoljašnji Unos** — DTD funkcionalnost koja je dostupna u XML-u se koristi za definisanje sintakse elemenata dokumenata. On takodje omogućava spoljašnjim podacima da budu ugrađeni u XML dokument. Navodjenjem lokalnog fajla, neke XML funkcije mogu biti napravljene za omogućavanje neovlašćenog pristupa informacijama iz lokalnog fajl sistema.
3. **XML napad uskraćivanjem usluga**(eng. *Denial of Service*) — Ovaj napad koristi prednost povlačenja u entitete koji su definisani u DTD-u. Povlačenje entiteta rekurzivno izaziva crpljenje memorije i na taj način onemogućava servis da obradjuje dalje zahteve.
4. **Ponovni napadi** (eng. *Replay attacks*) — SOA je zaštićena politikama koje obezbeđuju digitalno potpisane zahteve servisa. Ovaj sistem je ranjiv na ponovne napade koji jednostavno ponavljaju validno potpisanu poruku čime se dobija neovlašćeni pristup.

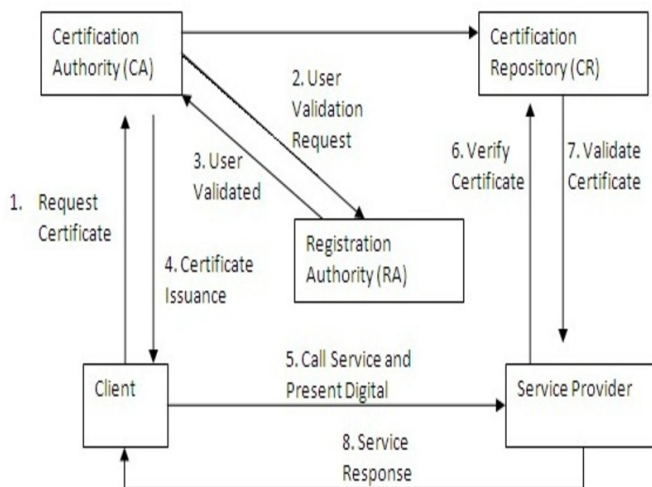
3. Postojeća rešenja SOA bezbednosti i njihove kritike

Mere bezbednosti su dizajnirane i primenjuju se za različite aspekte bezbednosti (kao što je pomenuto u odeljku 2). Najkorišćeniji i najsveobuhvatniji mehanizmi koji se bave SOA bezbednošću su *Kerberos* i *Infrastruktura javnog ključa (PKI)*.

3.1. Infrastruktura javnog ključa

Infrastruktura javnog ključa (PKI) je skup hardvera, softvera, ljudi, polisa i procedura neophodnih za kreiranje, upravljanje, distribuiranje, korišćenje, skladišćenje i pozivanje

digitalnih sertifikata. Bazira se na kriptografiji javnog ključa za enkripciju. PKI kreira digitalne sertifikate koji mapiraju javne ključeve u entitete, sigurno skladišteći te sertifikate u centralni repozitorijum i poziva ih po potrebi.



Slika 1. Tok podataka infrastrukture javnog ključa (PKI)

PKI tok podataka [3] [1] [2]

Korak 1 – Korisnik omogućava akreditivne Certification Autoriti-ju (CA) i zahteve za sertifikat

Korak 2 – Certification Autoriti (CA) kontaktira Registration Authority (RA) da bi proverio korisničke akreditivne. Ukoliko je korisnik autentifikovan (RA) ga prosledjuje (CA) za izdavanje sertifikata.

Korak 3 – Certification Autoriti (CA) obuhvata probleme korisnikovog javnog ključa i rok upotrebe sertifikata.

Korak 4 – Korisnik predaje sertifikat pružaocu usluge dok zahteva uslugu

Korak 5 – Pružaoac usluge verifikuje sertifikat i ako je sertifikat validan započinje se komunikacija

Infrastruktura javnog ključa (PKI) se bavi većinom bezbednosnih izazova koje nameće SOA kao što su provera identiteta, poverljivost, integritet i neporecivost.

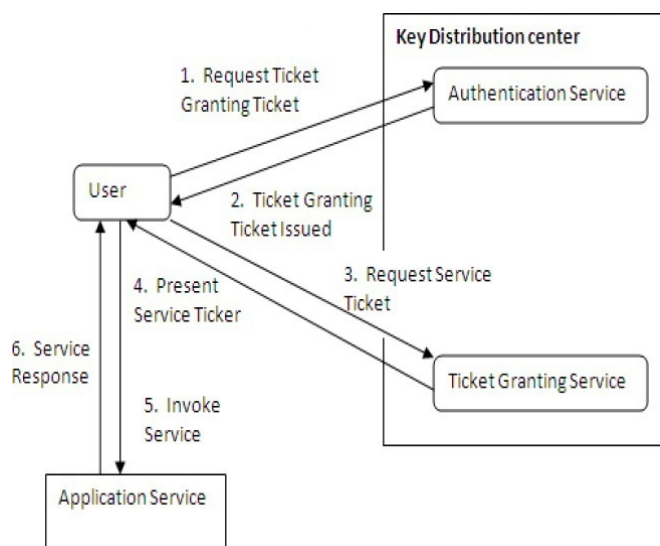
Slabosti infrastrukture javnog ključa (PKI)

1. PKI je tehnika koja zahteva mnogo resursa u smislu procesorskog vremena i memorije pa se ne može lako integrisati na uređajima slabe snage koji imaju mogućnost povezivanja na internet kao što su mobilni telefoni
2. PKI se ne bavi ovlašćenjem korisnika za izvršavanjem određene radnje ili pozivanje servisa

3. Svaki sajt uključuje poverenje u svoje korisnike i u sertifikate za ovlašćenja. Ukoliko poverenje između bilo koga od njih ne postoji tada uticaj toga može biti potencijalno opasan.

3.2. Kerberos

Osnovnu verziju Kerberosa je formirao MIT (*Massachusetts Institute of Technology*). Do danas se pojavilo više verzija ovog protokola, a najpopularnije su MIT verzija 4 i verzija 5. MIT je projektovao Kerberos da zaštiti mrežne servise omogućene od strane projekta Atina. Zasnovan je na ranijem *Needham-Schroeder* protokolu simetričnih ključeva. Protokol je nazvan po biću Kerberos iz grčke mitologije koji je monstruzni troglavi čuvar Hada. Kerberos koristi kriptografiju simetričnih ključeva. Unakrsna autentifikacija je korisna komponenta Kerberos-a čiji je cilj da omogući bezbedan pristup usluga preko organizacionih ciljeva.



Slika 2. Kerberos tok podataka

1. Korisnik šalje akreditive Authentication Service-u i zahteva *Ticket Granting Ticket*
2. AS verifikuje korisnika sa podacima. Nakon verifikacije generiše se ključ za šifrovanje i vreme isteka sesije koje najčešće iznosi 8 sati. AS šalje ključ sesije koji će se koristiti između korisnika i TGS-a. Ključ sesije se šifrjuje korisnikovim tajnim ključem i TGS tajnim ljučem pa se zatim šalje korisniku.
3. Korisnik šalje TGT i ID servisa ka Ticket Granting servisu
4. TGS dekriptuje TGT i oporavlja sesije ključa koji će se koristiti sa klijentom u budućoj komunikaciji

5. TGS generiše ključ sesije (*Client-Service Key*) koji će se koristiti od strane klijenta za dalju komunikaciju sa servisom. TGS šifruje to sa servisnim tajnim ključem i šalje nazad klijentu
6. Klijent šalje šifrovani *Service ticket* (dobijen od TGS-a) ka servisu i zahteva operaciju
7. Servis dešifruje potvrdu i dobija ključ sesije. Taj ključ se u budućnosti koristi za komunikaciju sa klijentom sve do isteka sesije

Sigurnosna pitanja kojima se bavi Kerberos su autentifikacija, poverljivost i integritet.

Mane Kerberos protokola

1. Kerberos ne može da se koristi u interakciji sa ne Kerberos sistemima
2. Postojeći servisi zahtevaju modifikaciju da bi rukovali Kerberos protokolom
3. Centar za distribuciju ključeva može biti jedna od tačaka neuspeha. Ukoliko je ugrožen ceo Kerberos sistem je pod rizikom

4. Sigurnosni standardi veb servisa

Postoji veliki broj standarda veb servisa, a ja sam se fokusirao na glavnim standardima za koje smatram da su najviše važni da bi se razumela SOA arhitektura. Osnovni bezbednosni ciljevi fokusiraju se na bezbednost poruke kao poverljivosti, integriteta i autentičnosti podataka koji su pokriveni od strane tri osnovna bezbednosna standarda: *WS-Security*, *XML Digitalnih potpisa* i *XML Enkripcije*.

4.1. WS-Security

WS-Security je predložen kao standard od strane Microsoft-a i IBM-a 2002. godine. Uspostavljen je kao OASIS standard 2004. Ovaj standard definiše poboljšanja SOAP poruka u cilju obezbeđivanja sigurnosti u porukama koje se prenose između korisnika i servisa. Zbog toga su dodatne informacije uključene u zaglavlju SOAP poruka na osnovu daljih specifikacija kao što su XML Enkripcija za poverljivost poruke, XML Digitalni potpis za integritet poruke i još mnogo toga. WS-Security je korišćen za primenu širokog spektra različitih bezbednosnih tehnologija i modela kao što su *X.509* sertifikat i *Kerberos*.

Nedostaci WS-Security standarda

1. WS-Security koristi XML Digitalni potpis za potpisivanje ne susednih delova SOAP poruka. Stoga su sva ograničenja XML Digitalnih potpisa takodje primenljiva i na WS-Security.
2. WS-Security omogućava više bezbednosnih zaglavlja sa istim imenom u okviru iste SOAP poruke. Ovo stvara veliki propust koji može biti iskorišćen od strane napadača.
3. WS-Security ne predlaže nikakvu novu tehnologiju bezbednosti. Medjutim on navodi kako postojeće bezbednosne tehnologije mogu da se iskoriste za sigurnu razmenu SOAP poruka.

4.2. XML Digitalni potpis

XML potpis je deo W3C standarda koji obezbeđuje integritet poruke i ne odbacivanje XML dokumenata i XML elemenata. XML potpis se ponekad naziva još XML-SIG ili XML-DSIG. Oslanja se na tehnologiju javnih ključeva u kojoj se heš elementi ili skup XML elemenata potpisuju privatnim ključem potpisnika i stoga mogu biti proverene od strane trećih lica obavljanjem kriptografskih operacija na heš podatke sa javnim ključem pošiljaoca. NDA ugovori (eng. *Non-disclosure agreement*) su relativno jednostavan način čuvanja poverljivosti informacija, ali potrebno je naglasiti da oni ne sprječavaju krađu informacije već vlasniku informacije daju pravni oslonac za tužbu u slučaju zloupotrebe. Poruka može imati elemente koji su digitalno potpisani od strane više različitih strana, i potpis može kriptografski spojiti elemente. Moguće je spojiti identitet, autorizaciju i vreme iz *WS-security* naslova i SOAP zahtev, ovim sprečavamo napade ponovnim korišćenjem istih poruka.

```

1  <Signature>
2    <SignedInfo>
3      <CanonicalizationMethod />
4      <SignatureMethod />
5      <Reference>
6        <Transforms>
7        <DigestMethod>
8        <DigestValue>
9      </Reference>
10     <Reference /> etc.
11   </SignedInfo>
12   <SignatureValue />
13   <KeyInfo />
14   <Object />
15 </Signature>

```

Osnovna struktura XML Digitalnih potpisa

Nedostaci XML Digitalnih potpisa

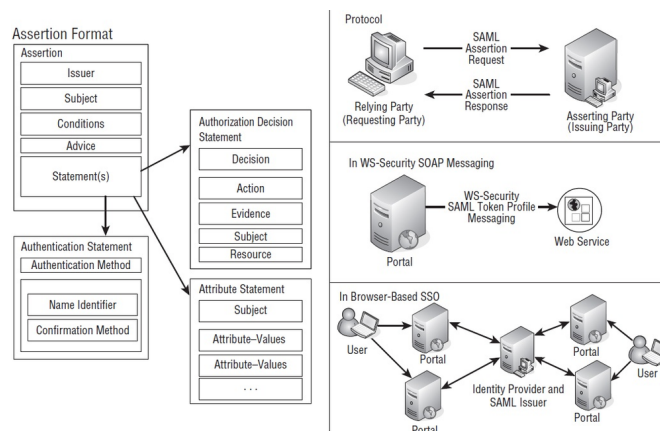
XML reprezentuje podatke korišćenjem drvolike strukture. XML Digitalni potpisi omogućavaju nepovezanim objektima XML skupa podataka odvojeno potpisivanje. Potpisan objekat može biti označen korišćenjem indirekcije (URI-ja) kao reference potpisanom elementu. Ovo indirektno referenciranje ne daje nikakve informacije o stvarnoj lokaciji potpisanog objekta. Dakle, potpisan objekat može biti lako premešten a da *Signature value* i dalje ostaje na snazi. U slučajevima kada je lokacija objekta podataka važna u tumačenju semantike podataka ovo može da bude iskorišćeno od strane napadača da stekne neovlašćen pristup nad neovlašćenim resursima. Ovo je glavno ograničenje XML Digitalnih potpisa.

4.3. SAML

SAML (*The Security Assertion Markup Language*) je okruženje bazirano na XML-u i koristi se za uspostavljanje autentifikacije i naziva subjekta kao i atributa informacije.

Koristi se u razne svrhe i obično u kombinaciji sa drugim standardima uključujući WS-Security SOAP Messaging, WS-Trust, WS-Federation, XACML i ID-WSF (*Liberty Identity Web Services Framework*). Ovaj standard se razvija oko koncepta tvrdnje (*assertion*), koji predstavlja izjavu o subjektu. Može biti tvrdnja o autentifikaciji subjekta (identitet subjekta), spisku autorizacijskih prava subjekta ili izraz autorizacijske odluke koja daje pristup nekom resursu. Poverenje u SAML tvrdnju je bazirano na poverenju u entitet koji zahteva tu istu tvrdnju. Od usvajanja 2002. godine ovaj standard je značajno uznapredovao i sa verzijom 2.0 pruža podršku za federativni identitet.

Slika 3. je primer korišćenja SAML-a kao protokola za zahtev/odgovor (*request/response*) u potražnji tvrdnji za odluke o pravima pristupa. Pokazuje korišćenje SAML novčića u WS-Security i konceptualnu upotrebu SAML tvrdnji u browser-based SSO okruženju [6].



Slika 3. Korisnici SAML-a u SOA preduzećima

Vidimo tri tipa izjave u tvrdnji:

1. *Izjava autentifikacije* – Pruža informacije o tome kako, kada i od strane koga je izvršena autentifikacija subjekta
2. *Izjava atributa* – Prikazuje spisak atributa koji se odnose na subjekat
3. *Izjava autorizacijske odluke* – Koristi se za prikaz prava subjekta da pristupi određenom resursu

SAML takodje definiše zahtev/odgovor (*request/response*) protokol za tražnju tvrdnji, obično se koristi za autorizaciju.

SAML novčići se mogu koristiti za slanje poruka u SOA okruženjima, obično u saradnji sa WS-Security ili drugim protokolima.

SAML 2.0 pruža podršku za SSO. Pojedini tipovi SSO-a podržani od strane SAML-a su:

1. Identitetni SSO - Identitet korisnika je registrovan u oba domena
2. Atributni SSO - Kontrolu prava pristupa resursima izvršavaju autorizacijski atributi

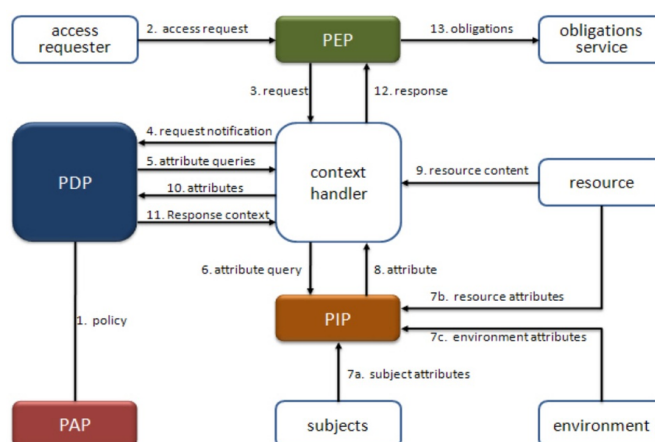
Anonimni SSO se postiže prosledjivanjem SAML izjava o atributima, takodje postoji mogućnost korišćenja pseudonima radi veće privatnosti. SAML Web browser SSO profil određuje kako tvrdnje o autentifikaciji pregovaraju izmedju pružaoca identiteta i pružaoca usluga [6].

4.4. XACML

XACML(*eXtensible Access Control Markup Language*) je ekspresivni i fleksibilni jezik zasnovan na XML-u. Koristi se za dostavljanje polisa o kontroli prava pristupa resursima i uključuje zahtev/odgovor (request/response) protokol za slanje upita o pravima pristupa. On opisuje zahteve kontrole prava pristupa i može se proširivati definisanjem novih funkcija, tipova podataka i logikom.

XACML profil specifikuje pet glavnih grešaka za obradu odluke o pristupu:

1. *Policy Administration Point* **PAP** je repozitorijum politika i obezbedjuje politike za PDP
2. *Policy Enforcement Point* **PEP** je interfejs celog okruženja za spoljašnji svet. Prima zahteve pristupa i ocenjuje ih uz pomoć drugih učesnika i dozvoljava ili odobrava pristup reursu
3. *Policy Decision Point* **PDP** je glavna tačka za donošenje odluka o pravima pristupa. Prikuplja sve neophodne informacije od drugih učesnika i donosi odluku
4. *Policy Information Point* **PIP** je tačka gde se potrebni atributi za procenu polise dobijaju od nekoliko eksternih ili internih učesnika
5. Upravljač konteksta (*context handler*)



Slika 4. XACML učesnici i tok podataka

Slika 4. ilustruje učesnike i tok podataka. PAP kreira XACML polise za resurse i čini ih dostupnim PDP-u. Kada subjekat zatraži pristup resursu PEP šalje zahtev o pristupu upravljaču konteksta(context handler), koji prenosi zahtev PDP-u. PDP traži pristup atributima subjekta, resursa i okruženja da bi mogao da donese odluku o pravima pristupa. Zahtev za atributima se šalje PIP-u i on ih vraća. Kada PDP dobije informaciju vraća autorizacijsku odluku, kroz upravljač konteksta, PEP-u. PEP donosi odluke na osnovu obaveza koje se dobijaju od strane servisa obaveza(*obligation service*).

Iako XACML pruža zahtev/odgovor (request/response) protokol za mnoge od interakcija na slici 4., velika količina informacija može biti prosledjena i SAML novčićima i mnoge razmene mogu koristiti SAML protokol. XACML se prevashodno fokusira na mehanizme koji omogućavaju pristizanje autorizacijskih odluka.

XACML može produžiti autentifikaciju mehanizmon dozvola u portalu veb usluga i koristi prvenstveno njegovo proširenje jezika. Politika servera koristi XACML za izražavanje i skladišćenje politika u predužecu. Budućnost uspeha XACML protoka ostaje da se vidi.

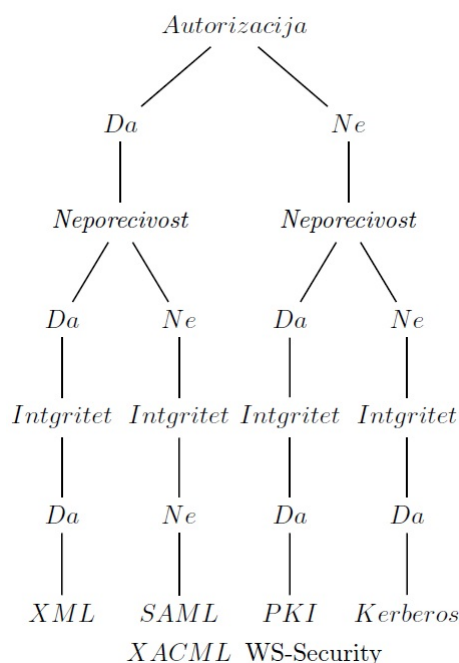
5. Klasifikacioni kriterijum i klasifikaciono drvo

Za klasifikacione kriterijume kod ovog istraživanja su izabrani najčešći sigurnosni izazovi Servisno Orjentisane Arhitekture kao što su autentifikacija, autorizacija, poverljivost, integritet i nemogućnost poricanja. Tabela niže sumira kriterijume koji su korišćeni u procesu odlučivanja najbezbednijeg sigurnosnog standarda.

	Autentifikacija	Autorizacija	Poverljivost	Integritet	Neporecivost
PKI	+		+	+	+
Kerberos	+		+	+	
WS-Security	+		+	+	+
XML Digitalni potpis		+		+	+
SAML	+	+			
XACML	+	+			

Tabela 1: Klasifikacioni kriterijumi

Klasifikaciono drvo rezimira osnovne kriterijume koji se koriste u procesu donošenja odluka za izbor odgovarajućeg bezbednosnog standarda. Za kontrolu pristupa u Servisno Orjentisanoj Arhitekturi brinu se SAML i XACML. Autorizacija je obično pod kontrolom pružaoca usluga. Kerberos sprovodi ključeve sesije koji su bazirani na klijent - server komunikaciji koja je vremenski ograničena i na taj način obezbeđuje integritet. PKI pored integriteta obezbeđuje i neporecivost jer se potpisivanjem privatnih entiteta podatak može koristiti kao neporeciv dokaz da podaci potiču od datog lica.



Slika 5. Klasifikaciono drvo

6. Trendovi i optimalna rešenja za budućnost

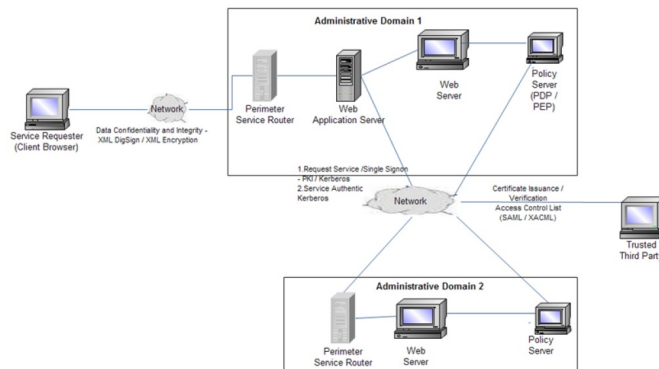
Trendovi se mogu opisati kao veoma specifični delovi tehnologije ili kao prolazne ideje to jest tačke gledišta. Oba pogleda mogu biti veoma moćna pri oblikovanju budućnosti SOA-e. Najveća koris od SOA-e je eksibilnost. SOA omogućava brzu promenu infrastrukture preduzeća kao odgovor na promene u poslovnom okruženju jer je većina poslovnih procesa već dostupna u obliku veb servisa.

Najelegantnija i najsigurnija rešenja su obično i najjednostavnija. Što je veća kompleksnost rešenja veća je mogućnost uvođenja grešaka i slabosti. Da bi omogućili kompletnu sigurnost sistema neophodno je da primenimo sve elemente bezbednosti (autentifikaciju, autorizaciju, poverljivost, integritet i nemogućnost poricanja).

Sistem koji je predložen na slici 6. predstavlja sveobuhvatan pristup potreban za osiguranu komunikaciju između servisa i korisnika. Neophodno je da se obezbedi besprekoran razgovor između različitih administrativnih domena.

Glavne komponente predloženog sistema su prikazane ispod:

1. *Service Requester* (najčešće veb pretraživač) – Klijent koji želi da koristi servis
2. *Perimeter Service Router* – prihvata poruke iz spoljnih aplikacija i rutira ih do odgovarajućih veb servisa na privatnoj mreži
3. *Web server* – održava jednostavne usluge u okviru granica organizacije



Slika 6. Predloženi sistem

4. *Web Application Server* – brine o interakcijama različitih veb servisa
5. *Policy Server* – sastoji se od *Policy Enforcement Point* i *Policy Decision Point* politika i interaguje sa trećom stranom za verifikaciju i validaciju sertifikata
6. *Trusted Third Party* – verifikuje, potvrđuje usluge i izdaje sertifikate

7. Zaključak

U ovom istraživačkom radu zaključio sam da ni jedan bezbednosni standard sam po sebi nije dovoljan. Sigurnost standarda i tehnike Servisno Orjentisane Arhitekture dolaze zajedno sa nekim dodatnim temama:

1. SOA forsira visoku interoperabilnost koja smanjuje sigurnost
2. SOA mora da se izbori sa heterogenim sigurnosnim konceptima postojećeg sistema
3. Distributivni proces se prenosi preko mnogo servisa tako da "point-to-point" rešenja nisu dobra za "end-to-end" sigurnost
4. Multiklijent sposobnosti zahtevaju vremensku promenu sigurnosti
5. Sigurnost utiče na performanse servisa

Sljedeće tačke treba uzeti u obzir pri dizajniranju bezbednosti za usluge orjentisane arhitekture:

1. Održavanje sigurnosti od početka
2. Pripremanje, praćenje i primenjivanje sigurnosnih polisa
3. Kreiranje svesti bezbednosti među svim akterima na svim nivoima

Bezbednost se može poboljšati preuzimanjem odgovarajućih mera na nivou operativnog sistema, nivou mreže, aplikacije i nivou baze podataka. Uz proširenje SOA-e ka računarstvu u oblaku, sistemi postaju podložniji bezbednosnim pretnjama. Noviji poslovni modeli koji nastaju zahtevaju eksponencijalno unapredjenje tehnologije. Pored bezbednosnih aspekata koji se razmatraju u ovom radu izazovi vezani za integraciju različitih politika moraju pažljivo da se rešavaju.

References

1. Basic components of a public key infrastructure <http://technet.microsoft.com/en-us/library/cc962020.aspx>
2. Introduction to digital certificates <http://www.verisign.com.au/repository/tutorial/digital/introl.shtml#step1>
3. Jie, W., Arshad, J., Sinnott, R., Townend, P., Lei, Z.: A review of grid authentication and authorization technologies and support for federated access control. *ACM Computing Surveys (CSUR)* 43(2), 12 (2011)
4. Papazoglou, M.P., Van Den Heuvel, W.J.: Service oriented architectures: approaches, technologies and research issues. *The VLDB journal* 16(3), 389–415 (2007)
5. Poonkavithai Kalamegam, P.K., Zayaraz Godandapani, Z.G.: A survey on testing soa built using web services. *International Journal of Software Engineering and Its Applications* 6(4), 91–104 (2012)
6. Rosen, M., Lublinsky, B., Smith, K.T., Balcer, M.J.: *Applied soa: Service-oriented architecture and design strategies* (1 2010), <http://amazon.com/o/ASIN/B00805K7S8/>