NEGOTIATION ON A NEW POLICY IN SERVICE

Fereshteh Bayat and Afshin Salajegheh and Yousef Rastegari

M.S. Graduate of Software Engineering,Azad University South Branch,tehran,Iran
Ph.D Assistant Professor of Software Engineering and Computer Science
IAU Tehran South Branch,Tehran,Iran
Ph.D. Candidate of Shahid Beheshti University, Electrical & Computer Engineering
Department, Tehran, Iran

ABSTRACT

During interactions between organizations in the field of service-oriented architecture, some security requirements may change and new security policies addressed. Security requirements and capabilities of Web services are defined as security policies. The purpose of this paper is reconciliation of dynamic security policies and to explore the possibility of requirements of the new defined security policies.

During the process of applying the defined dynamic policy, is checked whether the service provider can accept the new policy or not. Therefore, the compatibility between existing policies and new defined policies are checked, and because the available algorithms for sharing between the two policies, resulted in duplication and contradictory assertion, in this paper for providing a compromise between the provided policy and the new policy, the fuzzy inference method mamdany is used . and by comparing the security level of proposed policy with the specified functionality, the negotiating procedure is done . The difference between the work done in this paper and previous works is in fuzzy calculation and conclusion for negotiations. the advantages of thi work is that policies are defined dynamically and applied to bpel , also can be changed independently of bpel file.

KEYWORDS

Policy,Policy Attachment,Negotiation

1. INTRODUCTION

In general, to determine which web service is appropriate for a specific application, functional capabilities should be adapt able with functional requirements and also non-functional capabilities in Web service should meet non-functional requirements. Consumer and provider of Web services, define their requirements and security policies as XML files named ws-policy. WS-policy provides a basic structure to describe a wide range of requirements and capabilities of Web services. In this paper, the changes are security changes and while applying new policies to processes , check whether the service provider will be accept the new policy or not.

In part 2, the structure of policies is defined. In part 3, the framework will be described and check the ability to dynamically negotiate on new policy. If the negotiation success , the new policy will be dynamically applied. The fuzzy tools of Matlab is used for implementation of proposed method. Section four presents the conclusions and suggestions for future deals.

DOI : 10.5121/ijwsc.2014.5401

2. WS-POLICY STRUCTURE

WS-Policy (Web Service Policy) is used to describe the quality of service. WS- Policy, is a general-purpose model for describing Web service policies, which including blocks to exchange their policies. WS-Policy defines a policy as a set of alternative which each alternative is a set of assertions. indeed assertions describe requirements and functionalities of the Web service. The main structure of a policy in the normal form is as follows:

```
<wsp:Policy … >
  <wsp:ExactlyOne>
    ( <wsp:All> ( <Assertion …> … </Assertion> )* </wsp:All> )*
  </wsp:ExactlyOne>
</wsp:Policy>
```

Listing 1 : normal ws-policy structure

The following example represents the normal form of a policy:

```
<wsp:Policy
    xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy"
        xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy" >
  <wsp:ExactlyOne>
    <wsp:All>
      <sp:Basic256Rsa15 />
    </wsp:All>
    <wsp:All>
      <sp:TripleDesRsa15 />
    </wsp:All>
  </wsp:ExactlyOne>
</wsp:Policy>
```

Listing 2 : ws-policy example

3. PROBLEM PLAN

In order to provide security during data exchange between the services, should service providers and requester agree on their capabilities and requirements. The WS-Policy does not offer a negotiated solution over the web service policies. During interactions between organizations, some Web Service security requirements may be changed and the new security policy is defined. In order to dynamically attach policies to bpel and negotiate on the proposed policy, outlined framework in section 4 is provided.

4 . PROPOSED FRAMEWORK

To attach new policy to BPEL externally and negotiate on policies, outlined framework in Figure 1 is provided. The proposed policy is attached on the two input files and how to attach is reflected. Before the change of policy attachment file, a lock is set on policy file to prevent changes during policy attachment process. Then policies and activities that policies be attached to are identified and a mapping between the scope's activities and the new corresponding policy is created. In order to link a defined external policy with BPEL activities, WS-Policy Attachment structure is used . Attached files are XML files containing “Applies to” element and, “selector”;

the child element. The selector is, an XPATH expression to select an activity within bpel scope. It also contains another element called PolicyReference which includes a reference to a policy. For example, the proposed policy by the name “ATM_new_Policy” . apply to “createTicket” activity as follows:

```
<wsp:PolicyAttachment
xmlns:wsp=”http://schemas.xmlsoap.org/ws/policy/”
xmlns:bpel=”http://schemas.xmlsoap.org/ws/business−process/”>
<wsp:AppliesTo>
<bpat:selector>
//bpel:scope[@name=”TicketCreationUnit”]//bpel:invoke[@operation=”createTicket”]
</bpat:selector>
</wsp:AppliesTo>
<wsp:PolicyReference
URI=” http://schemas.xmlsoap.org/ws/securitypolicy./ATM_new_Policy”/>
</wsp:PolicyAttachment>
```

Listing 3 : policy attachment

And content of the new policy of ATM_new_Policy are:

```
<wsp:Policy
Xmlns:wsu = “http://schemas.xmlsoap.org/ws/securitypolicy”
Wsu:Id=”ATM_new_Policy”>
<wsp:ExactlyOne>
<wsp:All>
   <sp:AlgorithmSuite>
       <sp:Basic256/>
     </sp:AlgorithmSuite>
     <sp: AuthenticationToken>
        <sp:UsernameToken/>
      </sp: AuthenticationToken >
  </wsp:All>
</wsp:ExactlyOne>
</wsp:Policy>
```

Listing 4 ATM_new_Policy

Figure 1 : proposed framework for new policy attachment and negotiation

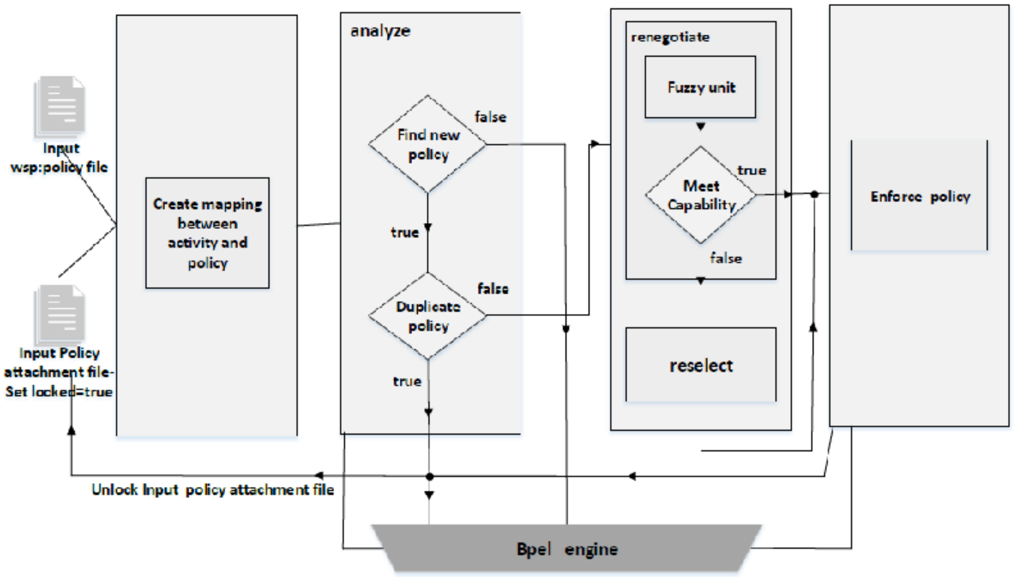

Figure 1 : proposed framework for new policy attachment and negotiation

We will describe Figure 1 at below:

Analyze: when it turns to perform an bpel activity within the particular bpel scope, searching begins in the mapping file,to determine whether new policy for proposed activitiy is defined or not. If the policy is not defined, bpel engine is notified to continue its work. If the new policy is defined, then it is surveyed that the proposed policy is new to attach to the activity or is already applied.

Renegotiate: In this part, fuzzy calculations are done for all alternatives in the proposed policy. If the security level for at least one alternative is supported by provider, negotiation will be done, if not ,another supplier is reselect.

Fuzzy Unit: in order to negotiate for accepting the new policy, the degree of provided security by the new policy is calculated according to the fuzzy calculations and then compared with provider's capabilities.

Table 1 : Algorithm_Suite

Table 1 : Algorithm_Suite

AlgorithmSuite	Assigned_number
Basic256	**16**
Basic192	**15**
Basic128	**14**
TipleDes	**13**
Basic256Rsa15	**12**
Basic192Rsa15	**11**
Basic128Rsa15	**10**
TripleDesRsa15	**9**
Basic256Sha256	**8**

Basic192Sha256	**7**
Basic128Sha256	**6**
TripleDesSha256	**5**
Basic256Sha256Rsa15	**4**
Basic192Sha256Rsa15	**3**
Basic128Sha256Rsa15	**2**
TripleDesSha256Rsa15	**1**
No-algorithm	**0**

Table 2 : AuthenticationToken

AthenticationToken	Assigned_number
X509Token	**9**
KerberosToken	**8**
SamlToken	**7**
RelToken	**6**
SecureConversationToken	**5**
SecurityContextToken	**4**
SpnegoContextToken	**3**
IssuedToken	**2**
UsernameToken	**1**
No-algorithm	**0**

Table 1 represents a sequence of algorithms and the sequence of tokens are described in Table 2. Algorithm Suite and Authentication Token are assertion types of policy. The left column of table 1 is from the strongest to the weakest algorithm and the left column of table2 is from the strongest to the weakest authentication token . For example TripleDesSha256Rsa15 and Username Token are the weakest .[4] For each input variable Algorithm Suite, Authentication Token and security output variable, fuzzy sets are defined in accordance with membership functions in Figures 2, 3 and 4.

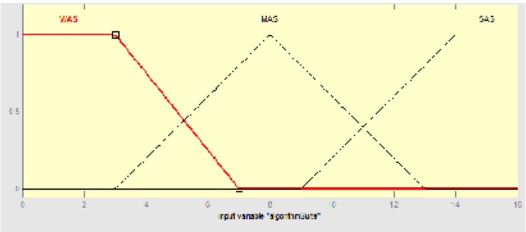

Figure 2 : algorithmSuite membership function

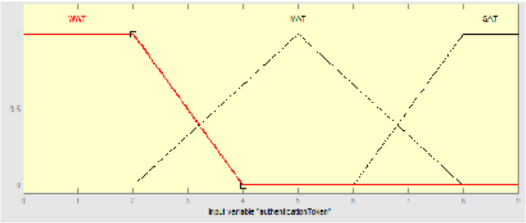

Figure 3 : authenticationToken membership function

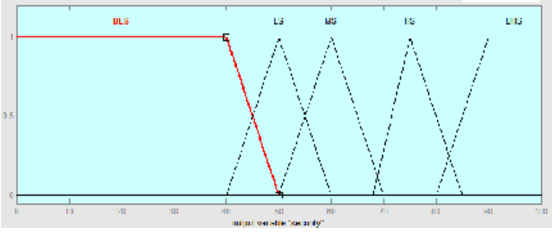

Figure 4 :security membership function

Based on fuzzy calculation steps and defined fuzzy rules, output fuzzy calculation, for the proposed ATM_new_Policy would be accordance with Figure 5.

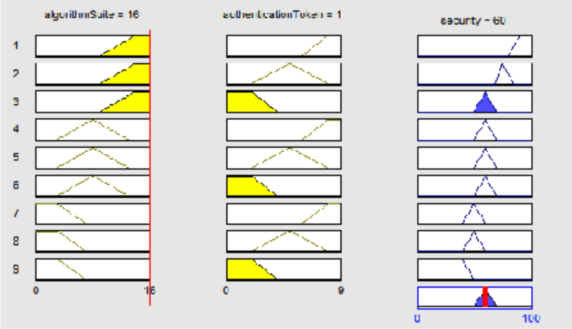

Figure 5 : Fuzzy Mamdani method output in Matlab

The final step is to calculate a value corresponding to the security level which is the center of gravity for the aggregsted area in figure 5. In the example above, as can be seen in Figure 2 Calculated security level is equal to 60. In accordance to obtained number = 60, and the provider capability for the security policy is defined between 60 and 70, then negotiation will be performed. In fact, the fuzzy calculations for all new policy alternatives is done and if at least the security level of one alternative is supported by provider, the negotiation will be done but if the calculated security level of none of the alternatives is not included in the capability range of provider, then another provider will be selected.

Enforce policy: After doing the above steps, bpel engine attachment file corresponding to input attachment file will be modified and during the execution of the corresponding activity in bpel, the proposed policy will apply. Input attachment will be unlocked to be accessible for future changes.

5. CONCLUSIONS

WS-policy is used to specify the security features of web services .

In this paper a framework is proposed to attach a new policy to bpel activitiy dynamically and negotiate between requester and provider . Among the advantages that can point for the proposed framework , is that external attachment of policies to bpel distinct the business process logic from describtion of quality of service . The policies and BPEL files can be changed independently of each other. In addition, the policies can be changed at runtime. It also reduces the complexity of BPEL processes,increase maintainability and changability of bpel processes.

REFERENCES

[1] S.Bajaj,D.Box,F.Chappell “Web Service Policy 1.2 - Framework (WS-Policy) , W3C Member Submission 25 April 2006

[2] G.Della-Libera,M.Gudgin “Web Services Security Policy Language (WS-SecurityPolicy)” ,IBM,Microsoft,RSA,Verisign, July 2005

[3] A.Charfi,R.Khalaf,N.Mukhi “QOS-aware web service composition using non-intrusive policy attachment to bpel”,Springer ,pp.582-593 , 2007

[4] T.Lavarack,M.Coetzee “Considering web services security policy compatibility” , IEEE Information Security for South Africa (ISSA) , august 2010

[5] A.Strunk,S.Reichert,A.Schill “An Infrastructure for supporting Rebinding in BPEL Processes”,IEEE Enterprise Distributed Object Computing Conference Workshops , pp.230-237,Sept.2009

[6] M.Negnevitsky “Artificial Intelligence: A Guide to Intelligent Systems” , Pearson Education , 2009