Policy-Based Management of the inter-Domain communications Security

El Hamzaoui Mustapha, Abderrahim Sekkaki, and Bahloul Bensassi

Department of Mathematics & Computer science
University Hassan II Aïn-chok, Faculty of sciences
P.O Box 5366, Maarif – Casablanca. Morocco
{m_elhamzaoui, a_sekkaki, bahloul_bensassi}@yahoo.fr

Abstract. Because of the enormous number of enterprises to manage by the inter-domain communication infrastructure manager and the permanent modifications that could occur in this management environment, the security management must be based-policy. In this work we will present a Dynamic Management Environment of the Inter-Domain Communications Security (DMEIDCS) where the Ponder language is used to specify security and management policies. The proposed approach will be characterized by a large opening on the customer by permitting him to interact directly and in real time with the management environment DMEIDCS.

1. Introduction

Because of many reasons like the big extension of the security environments that often makes the management very difficult, the security environment permanent modifications and the customers' unceasing requirements, the management environment of the inter-domain communications security must be dynamic and policy-based.

In general, the inter-domain communications actors don't accept that a third detains a part or the totality of the responsibility of their communications security management. They want to be always an active part in the security management environment and never admit to be marginalized.

In this work we will present a Dynamic Management Environment of the Inter-Domain Communications Security (DMEIDCS) which is based on the Ponder language [3] to specify security and management policies and also is also largely opened on the user to satisfy his requirements.

Ponder is an object-oriented, declarative language for specifying security and management policies for distributed systems. Ponder is used in many research works. Thus, PONDER policies are implemented and validated for Differentiated Services (DiffServ) by using CIM (Common Information Model) as the modelling framework for network resources as this device independent [9]. Ponder language is used also to realize a dynamic adaptation of policies in response to changes could occur within the managed environment [10]. Finally, an integrated toolkit is implemented to specify, deploy and manage policies specified in the PONDER language [4].

2. Presentation of our approach

2.1. Environment of the security dynamic management

The first layer of our DMEIDCS is managed by the manager of the inter-domain communications infrastructure while the second one is own to the DMEIDCS users.

Layer2: Environments of the domains internal security management . The structure of a user environment of the management of the domain internal security is:

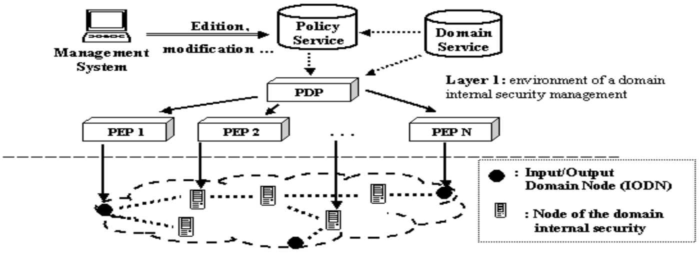
Management System
Edition, modification ...
Policy Service
Domain Service
PDP
Layer 1: environment of a domain internal security management
PEP 1
PEP 2
. . .
PEP N
: Input/Output Domain Node (IODN)
: Node of the domain internal security

Fig.1 Environment of a domain internal security management. A part of the PEPs (IODN) of each user management environment is reserved for the inter-domain communications management.

Layer1: DMEIDCS. The structure of the global DMEIDCS is:

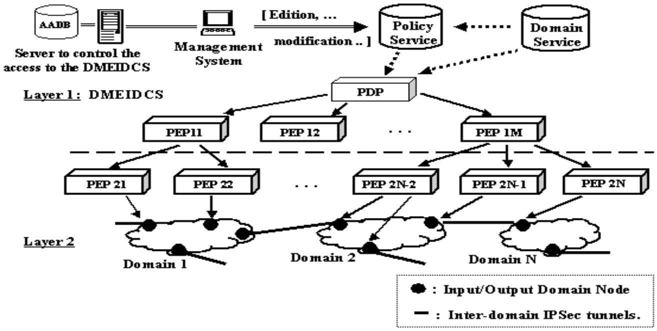

Fig.2 Layer2 PEP2j (j=1..N) that manage the IODNs are the targets of the PEP1i (i=1..M). These latter are the layer1 PEPs that perform the inter-domain communication security policies.

The interactions inter customer-DMEIDCS will occur at the level of layer1 and the DMEIDCS access control is ensured, as it is schematised on the figure2, by a server access control which is provided with an Authentication and Authorisation Database.

2.2. Resources role-based access control

The notion of the role is used in several works on management such as distributed systems management [8], access control management [6] or virtual organization management [7]. The role is strongly related to the concept of position which is primarily a static concept describing a statute in an organization.

Domain 'IPsec_user'. The customers' domain Personal/User/IPsec_User contains important information on the customers. Then, the access to this domain must be role-based controlled:

```
type role User_Manager(set tgt) {

  type auth+ DomAccessCtrl(target tgt1){
     action add(),remove(), enable(),disable();}
  inst auth+ Dom_User_CtrlAcces = DomAccessCtrl(tgt);
  type oblig User_Supervisor (target tgt2) {
     on EvetAdd(); //Event to add a new customer\user
     do Add_User(tgt2);}
  inst oblig Add_User = User_Supervisor(tgt);
  ... //Other policies attributed to the role
 }// End of the role declaration-Domains specification:
Domain admu = Personal/Adm/IPsec_Adm/User_Adm;
Domain users = Personal/User/IPsec_User;
// role instantiation :
inst role r_Users_Mger= User_Manager(users) @admu;
```

Other roles. Because of the domain System/Soft/Agt/IPsec_Agt contains all information on the DMEIDCS IPsec management agents, the access to this domain must be role-based controlled. In the same way, the access to the domain MgmtInfo/Policy/IPsec_Policy which contains significant information on the management and IPsec security policies, must be role-based controlled.

2.3. Scenarios of the possible inter user- DMEIDCS interactions

A DMEIDCS customer can choose and apply IPsec security policies and perform directly some operations on the domain service (PEP addition/suppression and IPsec tunnel establishment/disestablishment).

Moreover, all these tasks must be preceded by a request and the traditional authentication and authorization operations on the level of the access control server (fig.2). The request is very important because it allows sending the necessary customers' parameters to the DMEIDCS in order to perform the requested operation.

Choice and application of policies. To choose an IPsec policy and to apply it on a user PEP (IODN) the specified obligation policy is:

```
inst oblig Policy_Choice{//PMA:Policy Management Agent

  on      EvtPolicySelect(Pol_ref,PEP_ref);
  Subject root/syst/Soft/Agt/IPsec_Agt/PMA;
  do evaluate(Pol_ref,PEP_ref)
     -> IPsecPolicy=selectPolicy(Pol_ref)
     -> IPsecPolicy.enable() -> par[]=calcPar(Pol_ref)
     -> EvtService.GeneratEvt(IPsecPolObligEvt,par[]);}
```

Firstly, the method evaluate() checks the customer parameters and the PEP availability. Secondly, the method selectPolicy() selects the policy referenced by Pol_ref. Then, the policy is activated and its corresponding parameters are also calculated through the method calcPar(). Finally, the obligation event IPsecPolObligEvt is generated to trigger the policy.

Choice and perform of operations. After the successful customer authentication and authorisation an event will be activated automatically, in the monitoring service, to trigger, by transmitting necessary parameters, the obligation policy corresponding to the customer request.

To add a PEP, the obligation policy performing this operation is:

```
inst oblig PEP_ADD{

  on      EvtPEPADD(user_ref,parPEP[]);
  Subject s=root/syst/soft/Agt/IPsec_Agt/PEPAdd_Agt;
  target  t=root/syst/Device/PEP/IPsec_PEP;
   do     checkparam(user_ref,parPEP[]) ->
          ref= attrPEPref(parPEP[])->
          t.ADDPEP(ref);}
```

Firstly, the method checkparam() checks the transmitted parameters. Then, a reference (ref) is attributed to the PEP through the method attrPEPref(). Finally, the PEP will be recorded in the target domain.

To establish an IPSec tunnel the adequate obligation policy is:

```
inst oblig Tunnel_ADD {

 on      EvtTunlADD(parPEP1[],parPEP2[],pol_ref);
 subject s=root/syst/soft/Agt/IPsec_Agt/TunlADD_Agt;
 do   checkPEP(parPEP1[]) -> checkPEP(parPEP2[])
     -> ref_user2 = refUser(parPEP2[])
     -> agr_user(ref_user2,parPEP2[],parPEP2[],pol_ref)
     -> p=selectPolicy(pol_ref) -> p.enable()
     -> pol_par[] = calcPolpar(parPEP1[],parPEP2[])
     -> ServiceEvt.GenerateEvt(obligPolEvt,pol_par[]);}
```

The parameters of each PEP are checked by the method checkPEP(). The method refUser() extracts the second connected party reference. The second connected party agreement and the availability of his PEP are negotiated through the method agr_user(). Afterwards, the selection and the activation of the IPsec tunnel configuration policy are performed respectively through the methods selectPolicy() and enable(). The parameters corresponding to this policy are calculated through the method calcPolpar() and finally, the policy itself is triggered by the obligation event obligPolEvt().

The PEP suppression depends on its state. Thus, if the PEP is not integrated in an IPsec tunnel the suppression will be direct:

```
inst oblig Free_PEP_SUPP {
 on       EvtFreePEPSUPP (parPEP[]);
 subject  root/syst/soft/Agt/IPsec_Agt/TunlSupp_Agt;
 target   t= root/syst/Device/PEP/IPsec_PEP;
  do      checkPEP(parPEP[])->
          ref=refPEP(parPEP[]) -> t.SuppPEP(ref);}
```

Firstly, the PEP parameters are checked y the method checkParPEP(). Then, the PEP reference is extracted through the method refPEP(). Finally, the PEP will be suppressed by removing its reference from the target domain.

If the PEP is integrated in an IPsec tunnel the agreement of the second connected party will be necessary. Finally, to disestablish an IPsec tunnel the mechanisms require only the disablement of the applied policy instead of the PEP suppression.

3. Evaluation of our work

Many works are investigated to the IPsec protocol. Thus, Barrère et al. have presented an approach [2] which belongs to the Architectures of management of the multi plate-forms dynamic VPNs and also relates to the problems of the IPsec security policy distribution. Al-Chaal has presented an approach [1] which is a centralized solution, where everything, including VPN creation, deployment and membership management, is under the control of a single Management Operation Point (MO). Finally, the mechanisms and the main uses of the IPSec protocol are displayed in [11][12].

Concerning our work, the particularity of the proposed approach is the use of the IPsec protocol, as tool, to develop an opening environment of management of the inter-domain communications.

4. Conclusion

We have presented through this work a Dynamic Management Environment of the Inter-Domain Communications Security (DMEIDCS) which is based on two layers. The use of layers is significant to facilitate management and to specify the functions and responsibilities according to each level. Our solution is characterized by its large opening on the customers that is considered as an active entity in the DMEIDCS.

5. References

1. Al-Chaal, L.: Dynamic and Easily Manageable Approach for Secure IP VPN Environments. Ph.D.Dissertation, Institut National Polytechnique de Grenoble, France, February (2005)
2. Barrère, F., Benzekri, A., Grasset, F., Laborde, R., Raynaud,Y.: Distribution de politiques de sécurité IPsec, GRES'01-Gestion de Réseau et de Service, 4ème Colloque Francophone, Marrakech-Morocco, December (2001)

3. Damianou, N., Dulay, N., Lupu, E., Sloman, M.: The Ponder Policy Specification language. Proc. Policy 2001, International Workshop on Policies for Distributed Systems and Networks, Bristol, United Kingdom, January 29-31 (2001)
4. Damianou, N., Dulay, N., Lupu, E., Sloman, M.: Tools for Domain-Based Management of Distributed Systems. IEEE/IFIP Network operations and management symposium (NOMS2002), Florence,Italy,15-19 April,(2002) 213-218
5. S. Kent, and R. Atkinson. "Security Architecture for the Internet Protocol", RFC 2401, November (1998)
6. Lupu, E.,Marriott, D., Sloman, M.,Yialelis, N.: A Policy Based Role Framework For Access Control. 1st ACM/NIST Role Based Access Control workshop, Gaithersburg, USA, December (1995)
7. Lupu,E., Milosevic, Z., Sloman, M.: Use of Roles and Policies for Specifying, and Managing a Virtual Enterprise. Proceedings of the 9th IEEE International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises (RIDE-VE'99), Sydney, Australia, March 23-24, (1999)
8. Lupu, E., Sloman, M.: Towards A Role Based Framework For Distributed Systems Management. Journal of Network and Systems Management, vol.5, no.1, Plenum Pess, Systems Management, Plenum Press, (1997) 2(4):333-360
9. Lymberopoulos, L., Lupu, E., Sloman, M.: PONDER Policy Implementation and Validation in a CIM and Differentiated Services Framework. 9th IEEE/IFIP Network Operations and Management Symposium (NOMS 2004), Seoul, Korea, May (2004)
10. Lymberopoulos, L., Lupu, E., Sloman, M.: An Adaptive Policy-Based Framework for Network Services Management. Journal of Network and Systems Management, Vol.11, No.3, September (2003)
11. hsc: http://www.hsc.fr/ressources/articles/ipsec-tech/, last update : 23 October 2002.
12. TCP/IP Guide: http://www.tcpipguide.com/free/t_IPSecModesTransportandTunnel.htm.