Policy-based Management of a Virtual Laboratory Communications Security

Abderrahim Sekkaki, El Hamzaoui Mustapha, and Bahloul Bensassi

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

Abstract. The main objective of Virtual LABoratories (VLABs) is to solve the various time and/or space type problems that prevent the normal working of real laboratories but the VLABs services are permanent target of different dgerous. In this paper, we present a dynamic Public Key Infrastructure (PKI) to secure the inter users and a VLAB communications. Our VLAB is constituted of a set of Manipulations STations (MSTs) and their common security server (SS). Our PKI environment is composed of a PKI Server (PKIServ) to manage the users-VLAB communications security and a Monitoring Service (MS) to automate its functioning. A prototype has been implemented with CORBA environment and same experimental results are presented.

1. Introduction

In spite of the enormous sevices provided by the VLABs such as simulation, remote experimentation and training etc…, they are still targets of various attacks and threats such as spying, piracy and destruction.

In this work, we will propose a solution to secure the communications that could take place between a VLAB and their users. This solution will be based on Domains [11], a PKI environment, and Ponder policy specification language [2] to specify managemeny and securiyu policies.

Our virtual laboratory is constituted of a set of MSTs distributed geographically and of their Security Server (SS) that controls and manages the reources access [4] :

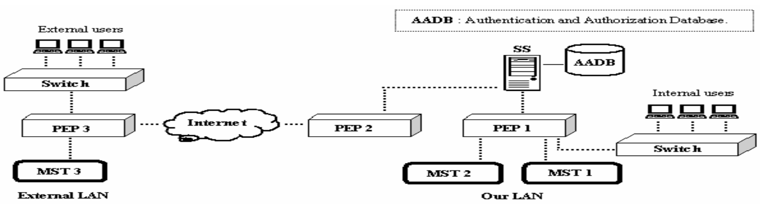


Fig. 1. Seen of the main components of our VLAB

A MST could be composed of a set of databases as taped and simulated manipulation databases, real manipulation databases, and local security databases. A part or the whole of these databases could be distributed applications.

2. Ponder policy specification language

Ponder [2] was developed at Imperial College and it is an object-oriented, declarative language for specifying security and management policies for distributed system [7][8][3].

The basic Ponder specifications concerne access control, obligation policy, constraints, and composite policies. Moreover, for Ponder all managed objects must be organised in domains. The organisation in domain of our VLAB components is:

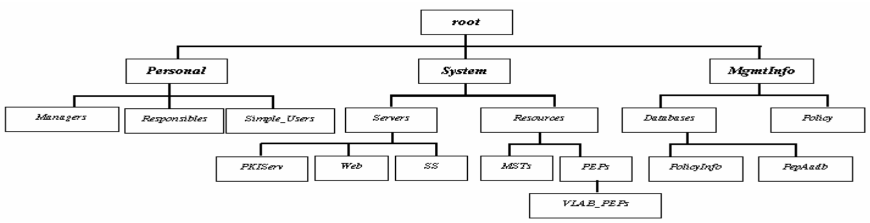


Fig. 2. Organization of our VLAB components

3. Our approach

Our solution is based on a PKI environment (fig.3) that used the RSA algorithm [10], and it discharges users from all security management tasks:

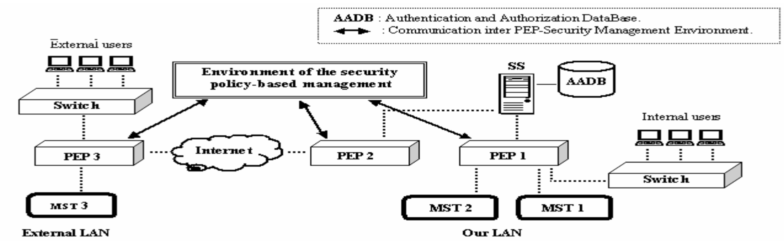


Fig. 3. Our VLAB and its security policy-based management environment

The PKI manage the RSA security policy parameters and when it decides the RSA policy parameters to apply it places them at the disposal of its PEPs (fig.3). Moreover; the users PEPs which perform the security of the inter users-VLAB communications must implement the encrypting/decrypting methods corresponding to the RSA policy managed by our PKI environment.

The communications between the PKI environment components are assured through an ORB Bus. The PKIServ is provided with two databases: a database (PolicyInfo.mdb) to contains the RSA policy information and parameters (Tables RSA_param & PEPs_Needs) and a database (PepAadb.mdb) to contain the necessary information on the users PEPs (Table PEPs_info).

The communications MS-PKIServ and PEPs-PKIServ are in the form of remote methods invocation (fig.4):

```
module PkiServ {

// Interface of methods invoked by the Monitoring Service:
      interface PkiServ_MS {
          oneway void changePolicyParam();
          oneway void modifyTables();
      };

// Interface of methods invoked by the PEPs:
     interface PkiServ_PEPs {
           string getSecurityParam(in string pep_id, in string pep_passwd);
     };
};
```

Fig. 4 The idl file (PkiMgmt.idl).

In our implementation, the RSA security policy (pol_RSA) parameters are stored in the table RSA_param.

We estimated a period time (T_{RSA}) for modifying the RSA_param table content and we chose also to change five times the applied security parameters for each T_{RSA}. Thus, after the expiry of these parameters, the MS invokes on the level of PKIServ the method changePolicyParam(). The corresponding Ponder specification is :

```
inst  oblig   obligpol_ChangePolicyParam {
   on           EventChangePolicyParam() ;
   Subject      s = System/Servers/PKIServ ;
   Target       t = MgmtInfo/Databases/PolicyInfo;
   do           policy_param[]= selectParam() -> registry(t.PEPs_Needs,policy_param[]); } ;
```

After the reception of the event EventChangePolicyParam(), the subject PKIServ selects firstly from the Table RSA_param the security parameters and stores them in the variable policy_param[]. Secondly, the subject registries these parameters in the table PEPs_Needs to put them in the disposal of the PEPs. The implementation is :

```
......
public void changePolicyParam(){
           policy_param = selectRSAParam();
}
```

To modify permanently the content of the static table RSA_param, the MS invokes, at each T_{RSA}, the method modifyTable() on the level of PKIServ (fig.4). The corresponding Ponder specification is :

```
inst  oblig   obligpol_tableContent {
   on           EventChangeTables() ;
   Subject      s = System/Servers/PKIServ ;
   Target       t = MgmtInfo/Databases/PoliciesInfo;
   do           supp(t.RSA_param) -> registryParam (t.RSA_param)  ; }
```

After the reception of the event EventChangeTable(), the subject PKIServ suppresses the content of the table RSA_param of the target PolicyInfo and records after in it ten new recordings. The corresponding implementation is :

```
...... //program
public void modifyTable() {
   changeRSAParamTable();        \\ Methode to change the RSA_param Table contents.
 }
... // program.
```

The method changeRSAParamTable() removes the content of the table and RSA_param and records afterwards in it ten new recordings.

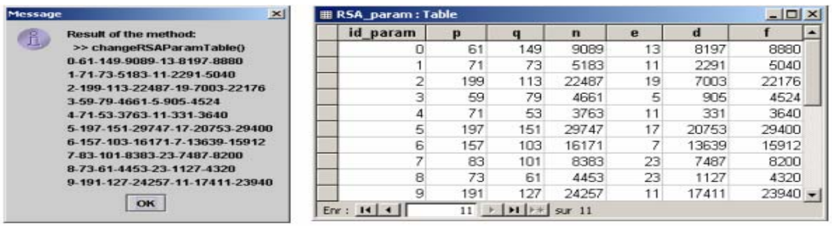
These PEPs call periodically the PKIServ through the method geSecurityParam() (fig.4) to get the pol_RSA parameters to apply. The arguments of these method (fig.4) are the PEP identifier and password. One could also use certificate to reinforce the security on this level [12]. The corresponding implementation is :

```
......... // program
public String getSecurityParam(String peplogin, String peppasswd, String pepsec){

   boolean b1,b2;
   b1 = authentication(peplogin,peppasswd);
   b2 = authorization(peppasswd,pepsec);
   String resp="";

   if( b1==true){
            if (b2==true) {resp=getSecParam();}
            else {resp = "Warning : Unauthorized PEP";}
         }
   else { resp = "Warning : Failed Authentication ";}

   return resp;

 } // End of the method getSecurityParam().
......... // program
```

Fig.5 Implementation of the method authentication()

The PKIServ checks firstly the PEP identity through the method authentication() and checks afterwards the PEP authorization through the method authorization(). The checking is based on the consultation of the table PEPs_info.

Cobcerning the execution, the invocation of the method modifyTable() that changes the content of the static table RSA_param gave us the following results:



id_param	p	q	n	e	d	f
0	61	149	9089	13	8197	8880
1	71	73	5183	11	2291	5040
2	199	113	22487	19	7003	22176
3	59	79	4661	5	905	4524
4	71	53	3763	11	331	3640
5	197	151	29747	17	20753	29400
6	157	103	16171	7	13639	15912
7	83	101	8383	23	7487	8200
8	73	61	4453	23	1127	4320
9	191	127	24257	11	17411	23940

Fig.6 Modification of the contents of the table RSA_param

All our next examples and executions will be based on the content of this table. In the same way, The invocation of the method changePolicyParam() gave as result :

Fig.7 Result of the invocation of the method changePolicyParam()

The possible replies that could receive the PEPs are :

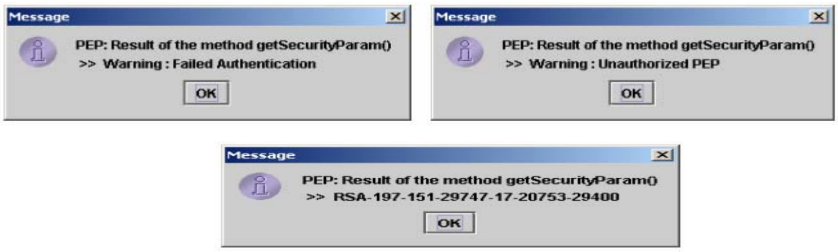


Fig.8 the possible replies of the invocation of the method getPolicyParam()

4. Related works

The majority of works dealt with VLABs presents platforms and approachs in layers where security and management are realized on the level of particular layers. In this context, an important platform of telecommunication in three layers is presented in [9] where management and security tasks are dealt with on the level of the adaptation layers. An other virtual laboratory platform in five layers is given in [1] where management and security are deal with on the level of the second layer (Core Middleware) that contains Globus [security, job management, etc.] and GRACE.

To solve the heterogeneity problem of our VLAB (interconnection platforms), we used CORBA objects to secure the inter users-VLAB communications. Our approach belongs to the cryptography and management works [5] realised inside our group. It could be generalized to secure the inter–domain communications and also more opened on the users [6]. An opened PKI allows to users to select automatically their desired security policy parameters.

5. Conclusion

In this work, we have presented a dynamique and centralised solution to secure a virtual evironment. The proposed approach was composed of a PKI Server to mange entirely the security environment and a monitoring system to automate the functioning. This approach discharges users from all security management tasks. To realise these tasks, we in implemented, on the level of the users' PEPs, the necessary encrypting/decrypting methods corresponding to the environment security policies.

Our perspective will be to extent our solution to be able to support symmetrical and asymmetrical cipher and to be more opened on their users.

6. References

1.Buyya, R., Branson, K., Giddy, J., Abramson, D.: The Virtual Laboratory : a toolset to enable distributed molecular modelling for drug design on the World-Wide Grid, Concurrency and computation : Practice and Experience,15:1-25 (DOI:10.1002/cpe.704), (2003).
2. 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 (2001) 29-31
3. 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) pp.213-218
4. El Hamzaoui, M., Sekkaki, A., Bansassi, B.: The Policy-Based Management of a Virtual Laboratory Security, GRES'2005 - Gestion de REseau et de Service, 6ème Colloque Francophone, Luchon-France, 28 Février-03 Mars (2005).
5. El Hamzaoui, M., Sekkaki, A., Bansassi, B.: Policy-based Resolution of the Diffie-Hellman protocol vulnerability, IEEE/IFIP 4th Latin American Network Operations and Management Symposium (LANOMS), Porto Alegre, Brazil, August (2005) 29-31
6. El Hamzaoui, M., Sekkaki, A., Bansassi, B.: Policy-Based Management of the inter-Domain communications Security, IEEE/IFIP 4th Latin American Network Operations and Management Symposium (LANOMS), Porto Alegre, Brazil, August (2005) 29-31
7. 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)
8. 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)
9. Pierre, S., Kassouf, M. : Towards a Telecommunication Platform for Supporting Distributed Virtual Laboratories, International JI. Of educational telecommunications 7(2),157-194, (2001).
10. Rivest, R., Shamir, A., Adleman, L.:A Method for Obtaining Digital Signatures and Public-Key Cryptosystems Communications of the ACM, 21 (2), February (1978) pp. 120-126,
11. Sloman, M.: Policy Driven Management for Distributed Systems. Journal of Network and Systems Management, Plenum Press (1994) 2(4):333-360
12. Westphall, C.B., Sekkaki, A., Alvarez, L.M., and W.T. Watanabe. Extending TINA Secure On-Line Accounting Services, Journal of Network and Systems Management (JNSM), Vol.11, No.4, December (2003).