Journal of ELECTRICAL ENGINEERING, VOL. 61, NO. 1, 2010, 20–28

A FRAMEWORK FOR TRANSLATING A HIGH LEVEL SECURITY POLICY INTO LOW LEVEL SECURITY MECHANISMS

Ahmed A. Hassan * — **Waleed M. Bahgat** **

Security policies have different components; firewall, active directory, and IDS are some examples of these components. Enforcement of network security policies to low level security mechanisms faces some essential difficulties. Consistency, verification, and maintenance are the major ones of these difficulties. One approach to overcome these difficulties is to automate the process of translation of high level security policy into low level security mechanisms. This paper introduces a framework of an automation process that translates a high level security policy into low level security mechanisms. The framework is described in terms of three phases; in the first phase all network assets are categorized according to their roles in the network security and relations between them are identified to constitute the network security model. This proposed model is based on organization based access control (OrBAC). However, the proposed model extend the OrBAC model to include not only access control policy but also some other administrative security policies like auditing policy. Besides, the proposed model enables matching of each rule of the high level security policy with the corresponding ones of the low level security policy. Through the second phase of the proposed framework, the high level security policy is mapped into the network security model. The second phase could be considered as a translation of the high level security policy into an intermediate model level. Finally, the intermediate model level is translated automatically into low level security mechanism. The paper illustrates the applicability of proposed approach through an application example.

K e y w o r d s: network security, security modeling, security policy, security mangement, OrBAC model

1 INTRODUCTION

Today networks are complex connections of resources that are often difficult to be managed effectively. That leads to the corporate network consists of many individual network security components such as firewall, active directory, Intrusion detection system *etc.* Each security component has its own policy and enforcement mechanisms. Also, each security policy has many types like access control, and auditing. Enforcement of each network security component policy to low level security mechanisms is a very difficult task for some reasons. First, security policy may be ambiguous that leads to the incorrectness of its translation into low level mechanisms. Second, the security policies may conflict with each other. Third, policies are quickly become obsolete, thus maintaining security policies is never-ending and time consuming. Finally, there is no way to assess the implemented security mechanisms with respect to the proposed security policy. To overcome these difficulties, this paper introduces a framework of an automation process that translates a high level security policy into low level security mechanisms. The proposed framework is described in terms of three phases; in the first phase, all network assets are categorized according to their roles in the network security and relations between them are identified to constitute the network security model. Through the second phase, the high level security policy is mapped into the network security model identified in the first phase. The second phase could be considered as a translation of the high level security policy into an intermediate model level. Finally, the intermediate model level is translated automatically into a vendor-specific security mechanism. The proposed model is based on organization based access control (OrBAC). However, the proposed model extend the OrBAC model to include not only access control policy but also some other administrative security policies like auditing policy. Also the proposed model introduces modularity of the high level security policies to the corresponding unique low level policy.

The remainder of the paper is structured as follows: In Section 2 we discuss related work. In Section 3 we introduce our proposed framework for the automation of security policy. In Section 4 we illustrate the applicability of our framework through an application example. The paper is ended with a conclusion and future work.

2 RELATED WORK

There are two approaches to automate the implementation of security mechanisms; the requirement engineering approach and the policy based approach. In all work of the requirement engineering approach, researchers try to add security features to existing models like UML, TROPOS, *etc.* These models are not native security models,

* Department of Computer Science and Informatics, Taibah University Al Medina Al Munawara, Saudi Arabia, Ahmadtdm@gmail.com
** Department of Communications, Misr Engineering and Technology Institute, Al Mansoura, Egypt, Wabahgat@yahoo.com

ISSN 1335-3632

so they model the security policy locally not from global view of thinking. Besides, their work did not support automating the translation of high level security policy into low level mechanisms. For these reasons and space limitation we will focus in presenting the related work in policy based approach. The policy based researchers work can be classified into model driven policy based approach and policy specification languages. In the next subsection we will focus on the researchers work in both approaches.

2.1 Model Driven Policy Based Approach

One of the most comprehensive treatments of security policy in networks with many firewalls and distinct security policies for sub-networks is the Firmato [1–2]. Firmato is a firewall management toolkit with a model definition language, a model compiler, translating global knowledge of the model into firewall-specific configuration files, and a graphical firewall rules illustrator. In Firmato, the connectivity results of a change have to be computed off-line and the engine has to be re-run on the changed input. An entity-relationship model is used to specify both the access security policy and the network topology and makes use of the concept of roles to define network capabilities. In this approach there is some mixing between the net topology and the access security policy to be enforced so that the role concept becomes ambiguous. Indeed, the authors are bounded to introduce the "group" concept with an unclear semantics; sometimes group is used to design a set of hosts and sometimes it stands for a role. This can lead to some difficulties to assign network entities to the model entities. They introduce notions of "open group" to authorize inheritance of permissions and "closed group" to prohibit it. The reason is the fact that concept of group is not well defined.

Hassan and Hudec have introduced Role base Network Security (RBNS) model [3] that can be used as an intermediary level between high-level policy form and low-level firewall rule-base. The main concept of RBNS model is that network services are assigned to roles and hosts are made members of appropriate roles thereby acquiring the roles' network services. The authors keep from the RBAC model only the concept of *role*. Indeed, the specification of network entities and role and permission assignments are not rigorous and does not fit reality. In particular, (1) all RBNS relations are binary even though an access control security goal and its equivalent filtering rule are always a triple (source, service, target). This leads to a loss of information: *permissions* are missing in RBNS model although authors consider the assignment of a service to an IP address as a permission which is semantically weak. (2) The model is dedicated to the firewall security component only and does not have the flexibility to support other network security components. In addition, the model is dedicated only to access control policy.

The work done by F. Cuppens et al [4–7] is very impressive. They introduce the main features of the Organization-Based Access Control model. The concept of organization is brought in as the central component of their model. In this manner, the policy specification is completely parameterized by the organization so that it is possible to handle simultaneously several security policies associated with different organizations. They define role, activity, view and organization hierarchies and analyze inheritance of both permissions and prohibitions through these hierarchies. Also they try to provide a clear semantics link between an abstract access control model, and its implementation into specific security components. Unfortunately they only apply the model to the firewall security component. However, they do not mention how the model could be applied to other network security components like active directory, intrusion detection system *etc.* In addition the OrBAC is concerned only with access control policy. Also, there is no concrete link between the security policy item and its implementation, thus the implemented security mechanisms cannot be reviewed or assessed.

2.2 Policy Specification Based Approach

Policy specification languages are an attempt to formalize the intent of the owner into a form that can be read and interpreted by machines [8]. For the study of current policy specification languages, we have considered the following languages:

KAoS [9] is a collection of services and tools that allow for the specification, management, conflict resolution, and enforcement of policies. KAoS uses ontology concepts encoded in OWL [10] to build policies. The KAoS Policy Service distinguishes between authorization policies and obligation policies. The applicability of the policy is defined by a set of conditions or situations whose definition can contain components specifying required history, state and currently undertaken action. In the case of the obligation policy the obligated action can be annotated with different constraints restricting possibilities of its fulfillment.

LaSCO [11] attempts to express constraints on objects. LaSCO policies are specified as logical expressions and as directed graphs. The auditing operations and control aggregation problems are indirectly expressed by LaSCO. However, there is no direct way of specifying confidentiality and integrity in LaSCO. From the grid's point of view, the delegation is not supported by the LaSCO syntaxes.

In [12–13], two trust management applications are presented: the PolicyMaker and its successor KeyNote. Both of these applications are used to answer signed queries of the form "does a set of requested actions r, supported by credential set C, comply with policy P?", where the credentials can be public key certificates with anonymous identity. Both policies and credentials are predicates specified as simple C-like and regular expressions.

Some proposals express access control policies as XML documents as exemplified by XACML [14]. XACML is an XML specification for expressing policies for information

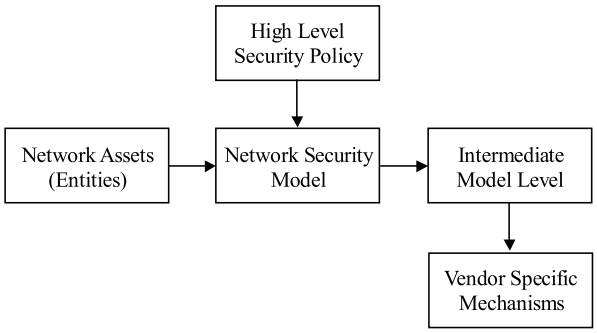

Fig. 1. Proposed Framework for Automating Security Policy

access over the Internet and is being defined by the organization for the Advancement of Structured Information Standards (OASIS) technical committee. The language provides XML with a sophisticated access control mechanism that enables the initiator not only to securely browse XML documents but also to securely update each document element.

Ponder [15] is a declarative, object-oriented language developed for specifying management and security policies. Ponder permits to express authorizations, obligations, information filtering, refrain policies, and delegation policies. Ponder can describe any rule to constrain the behavior of components, in a simple and declarative way.

An extension to ponder is Kava [16], which is a metaobject protocol that has been developed to allow flexible enforcement of security policies upon compiled code. The existing Kava implementation expresses security policies through a combination of Kava's binding specification and the policy representation used by the security model being enforced by Kava. This approach requires every method or field that is the subject of a policy to be individually listed therefore it is not an ideal approach to reusing policies. The aim of this work is to explore using a policy language that would make it easier to specify and reuse security policies.

However, Formal approaches suffer from being non intuitive and do not easily map to implementation mechanisms. They assume a strong mathematical background that makes of them difficult to use and understand. They also do not specify the policy by facts which lead to major difficulties in the administration.

3 PROPOSED FRAMEWORK

The proposed framework is described in terms of three phases; in the first phase all network assets are identified to constitute the network security model, which is based on OrBAC model. Through the second phase the high level security policy is mapped into the network security model. By the end of the second phase we get less abstracted, more detailed, representation of the high level security policy which is called intermediate level security policy. In the third phase the intermediate level of the security policy can be automatically translated into low level security mechanism.

3.1 Proposed Framework Model

On the heart of the proposed framework is the network security model. The proposed framework model for modeling the security policy is based on the OrBAC model. The main idea behind this model is to consider the corporate network as organizations. Each organization has its own abstract levels (roles, activities and views) and concrete levels (subjects, actions and objects), the hierarchy for the organizations and their abstract levels is an important step for the model. Some extensions to the OrBAC model are made to satisfy the following objectives:

- Encapsulating all network security components of the corporate network.
- Modularity of the concrete policy.
- Extending the OrBAC model to support more policy types other than access control policy.

To achieve the objective of encapsulation of all the corporate network security components, we introduce new links between the network security component and its concrete level. The purpose of that links is to distinguish between the concrete levels of each network security component. To realize these links, the following new definitions are introduced.

DEFINITION 1. Relation Relevant_subject. Relevant_subject is a relation over domains $Org \times S$. If Org is an ancestor organization and S is a subject, then Relevant_subject(Org, S) means that S is a relevant subject in organization Org.

DEFINITION 2. Relation Relevant_action. Relevant_action is a relation over domains $Org \times A$. If Org is an ancestor organization and A is an action, then Relevant_action (Org, A) means that A is a relevant action in organization Org.

DEFINITION 3. Relation Relevant_object. Relevant_object is a relation over domains $Org \times O$. If Org is an ancestor organization and O is a subject, then Relevant_object(Org, O) means that O is a relevant object in organization Org.

Modularity means that, the low level mechanisms translation is in the level of unique concrete policy item. The modularity leads to easy way to review the implemented low level mechanisms with respect to the high level concrete policy. Further, it will facilitate the maintenance process. To achieve this modularity of the policy, a new set representing the concrete policy identifier I is added to the OrBAC model. In addition we redefine the permission and prohibition relations to include the new set.

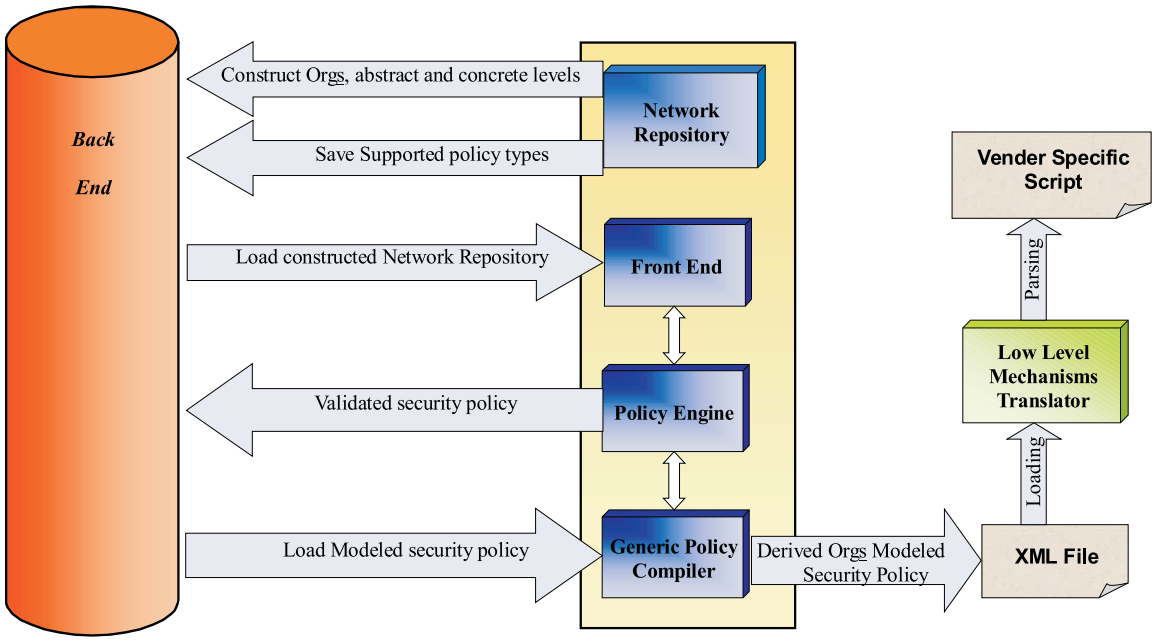

Fig. 2. Proposed Framework Hierarchical Design

DEFINITION 4. Relation Permission. Permission is a relation over domains $(I \times Org \times R \times A \times V \times C$. More precisely, if I is the concrete policy identifier, Org is an Organization, R is a role, V is a view, A is an activity, and C is a context then Permission $(I \times Org \times R \times A \times V \times C)$ means that For the concrete policy identifier I organization Org grants role R the positive authorization to perform activity A on view V in context C.

In the same spirit, the prohibition relation can be redefined.

DEFINITION 5. Relation Prohibition. Prohibition is a relation over domains $(I \times Org \times R \times A \times V \times C)$. More precisely, if I is the concrete policy identifier, Org is an Organization, R is a role, V is a view , A is an activity and C is a context, then Permission $(I \times Org \times R \times A \times V \times C)$ means that For the concrete policy identifier I organization Org grants role R the negative authorization to perform activity A on view V in context C.

Generalizing the OrBAC model to support more policy types other than access control policy is achieved by introducing some new relations. For example to support the audit policy, we will introduce a definition for the audit relation as the following:

DEFINITION 6. Relation Audit . Audit is a relation over domains $(I \times Org \times R \times A \times V \times C)$. More precisely, if I is the concrete policy identifier, Org is an Organization, R is a role, V is a view, A is an activity, and C is a context then Audit $(I \times Org \times R \times A \times V \times C)$ means that for the concrete policy identifier I organization Org will audit all events resulted when the role R do activity A on view V in context C.

3.2 Proposed Framework Hieratical Design

Figure 2, describes a detailed hierarchical design for the proposed framework. Our proposed framework consists of five main stages:

3.2.1 Network Repository

Network repository is responsible for storing the representations of the various network elements in the network being administered. The network repository is responsible for the following:

- Defining the organizations and their hierarchy of the corporate network.
- Defining concrete level (subjects, actions and objects) for each predefined organization.
- Defining abstract level (roles, activities and views) associated with each organization and their hierarchy.
- Assigning of the roles, activities and views to predefined subjects, actions and object.
- In addition the network repository also holds the predefined policy types (access control- audit *etc*) associated with each organization.

The whole network repository Information is stored in the back end of the proposed framework which most commonly is a database.

3.2.2 Front End

The front end is considered the management console. We can say the front end is responsible for the following:

- Defining the network security policies associated to each organization and policy type.

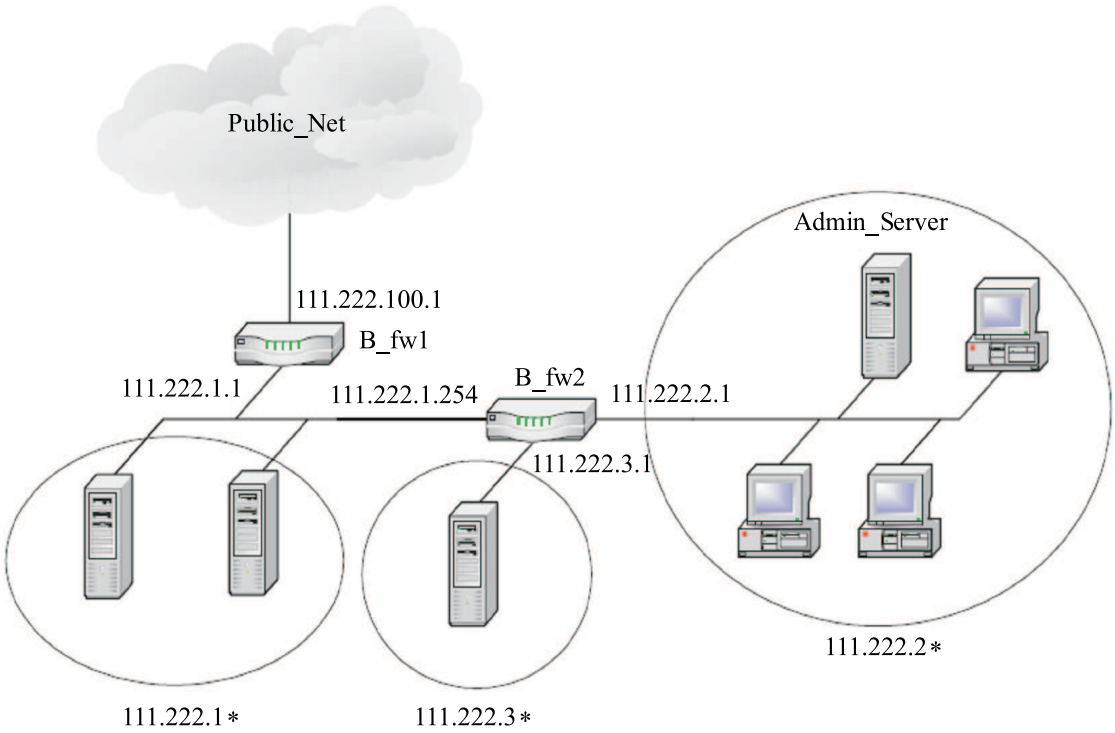

Fig. 3. Application Network Example

Table 1. Network example listed subjects

subject name	IP and mask	Port
S-Admin_Server	111.222.2.2/32	*
S-Admin_Gtw	111.222.3.2/32	*
S-Web_Server	111.222.1.2/32	80
S-FTP Server	111.222.1.2/32	21
S-Mail server	111.222.1.2/32	25
S-DNS server	111.222.1.3/32	53
S-Interface1	111.222.100.1/32	*
S-Interface2	111.222.1.1/32	*
S-Interface3	111.222.1.254/32	*
S-Interface4	111.222.3.1/32	*
S-Interface5	111.222.2.1/32	*
S-Private Hosts	111.222.2.*/24	*
S-Public hosts	*.*.*.*/0	*

- Defining the modeled security policy.
- Linking the concrete security policy with the modeled security policy.

3.2.3 Policy Engine

The policy engine is considered the "brain" of the proposed framework and is responsible for:

- All necessary validation to the security policy and modeled security policy to avoid the inconsistence and incorrectness of the security policy.
- Implementing powerful strategy to manage the conflict between permission policies and prohibition policies.

3.2.4 Generic Policy Compiler

The generic policy compiler is responsible for:

- Generating the modeled security policy for all derived organizations. The modeled security policy belongs to the
  i. Both role and view belongs to the derived organization.
  ii. Both role and view does not belong to any other derived organization. That means that it should be included in all derived organizations.
- Compiling the modeled security policy into generic policy form for the derived organizations.
- Translating the generic policy form into intermediate language like XML script.

3.2.4 Low Level Mechanisms Translator

The low level mechanisms translator is the final stage of our proposed mechanism translator is responsible for:

- Importing the intermediate generic policy generated in the previous stage.
- Parsing the generic policy through a vendor specific compiler.
- Generating vendor specific configuration script to be mapped to network security component.

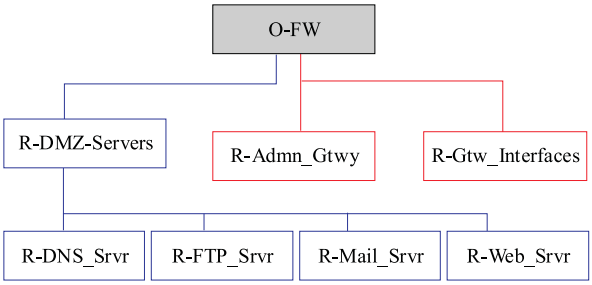

Fig. 4. Roles Hierarchy for O-FW Organization

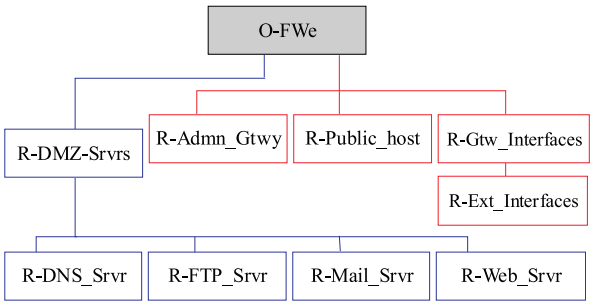

Fig. 5. Role Hierarchy for O-FWe Organization

4 APPLICATION EXAMPLE

4.1 Environment

To illustrate our approach, we reuse the example used in Firmato [18]. The corporate network is shown in Fig. 3. There is an external firewall, which guards the corporation's Internet connection Firewall. Behind it is the DMZ, which contains the corporation's externally visible servers. In our case these servers provide http/https (web), FTP, SMTP (e-mail), and DNS services. The corporation actually only uses two hosts to provide these services, one for DNS, and the other for all the other services. Behind the DMZ is the internal firewall which guards the corporation's intranet. This firewall actually has three interfaces: one for the DMZ, one for the corporate network zone, and a separate interface connecting to the firewall administration host. Within the corporate network zone, there is one distinguished host, Admin_Server, which provides the administration for the servers in the DMZ.

The policy has the following goals:

1. Internal corporate hosts can access all the resources on the Internet.
2. External hosts can only access the servers in the DMZ.
3. The DMZ servers can be updated only by the web administrator host admin server. Other corporate hosts have the same privileges as Internet hosts with respect to the DMZ servers.
4. The firewall gateway interfaces are only accessible from the fw admin host and are otherwise inaccessible to any host.

4.2 Constructing Network Repository

To construct network repository we define the organizations and their hierarchy, abstract levels and their hierarchy in addition to defining the concrete levels and associating concrete level to abstract level.

4.2.1 Organization

Our approach considers the corporate network consists of many organizations. The firewall (O-FW) is a subsequent organization from the corporate network (O-CN). Since the network security policy is actually managed by two firewalls, we shall consider that O-FW has two sub-organizations denoted O-FWi and O_FWe that respectively corresponds to the internal and external firewalls. This organization structure is modeled as:

Sub_Organization (O-FW, O-CN).
Sub_Organization (O-FWi, O-FW).
Sub_Organization (O-FWe, O-FW).

4.2.2 Concrete level

a) Subjects

In firewall organization, the subjects correspond to host machines. All hosts listed in Fig. 3 are defined and classified in Tab. 1

The previous listed subjects are modeled as the following:

Relevant _subject (O-FW, S-Admin_Server)
Relevant _subject (O-FW, S-Admin_Gtw)
Relevant _subject (O-FW, S-Web_Server)
Relevant _subject (O-FW, S-FTP_Server)
Relevant _subject (O-FW, S-Mail_server)
Relevant _subject (O-FW, S-DNS_server)
Relevant _subject (O-FW, S-Interface[1..5])
Relevant _subject (O-FW, S-Private_Hosts)
Relevant _subject (O-FW, S-Public_hosts)

b) Actions

An action is any implementation of a network service such as http, SNMP or ping. In our model, a service has three elements: a protocol, a source port and a destination port. According to the policy described above the services expressed as action listed in Tab. 2 are needed.

The previous listed subjects are modeled as the following:

Relevant _action (O-FW, A-HTTP)
Relevant _action (O- FW, A-HTTPS)
Relevant _action (O- FW, A-SMTP)
Relevant _action (O- FW, A-SSh)
Relevant _action (O- FW, A-DNS)
Relevant _action (O- FW, A-FTP)
Relevant _action (O- FW, A-Ping)
Relevant _action (O- FW, A-ALL-TCP)

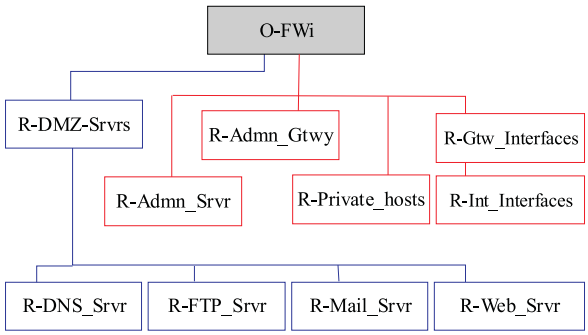

Fig. 6. Role Hierarchy for O-FWe Organization

Table 2. Network example listed actions

Action	Protocol	Port
A-HTTP	TCP	80
A-HTTPS	TCP	443
A-SMTP	TCP	25
A-SSh	TCP	22
A-DNS	TCP	53
A-FTP	TCP	21
A-Ping	TCP	8
A-ALL-TCP	TCP	*

c) Objects

In our example we are modeling the firewall network security component. In the firewall we can consider the subjects are accessing entities, while objects are accessed entities.

4.2.3 Abstract level

a) Roles

The role hierarchy of O-FW organization is shown in Fig. 4. Actually the role hierarchy of the O-FW organization contain the common roles exist in both O-FWi and O-FWe. Each derived organization has its own roles in addition to the common roles in O-FW. Figures 5, 6 show the role organization of both O-FWi and O-FWe.

The previous roles are modeled as the following:

Relevant _Role (O-FW, R-DNS_Srvr)
Relevant _Role (O-FW, R-FTP_Srvr)
Relevant _Role (O-FW, R-Mail_Srvr)
Relevant _Role (O-FW, R-Web_Srvr)
Relevant _Role (O-FWe, R-Public_hosts)
Relevant _Role (O-FWe, R-Ext_Interfaces)
Relevant _Role (O-FWi, R-Private_hosts)
Relevant _Role (O-FWi, R-Int_Interfaces)
Sub_role (O-FW, R-DNS_Srvr, R-DMZ-Srvrs)
Sub_role (O-FW, R-FTP_Srvr, R-DMZ-Srvrs)
Sub_role (O-FW, R-Mail_Srvr, R-DMZ-Srvrs)
Sub_role (O-FWe, R-Ext_Interfaces, R-Gtw_Interfaces)
Sub_role (O-FWi, R-Int_Interfaces, R-Gtw_Interfaces)
Relevant_Role (O-FW, R-DMZ-Srvrs)
Relevant_Role (O-FW, R-Admn_Gtwy)
Relevant_Role (O-FW, R-Gtw_Interfaces)

The predefined subjects are assigned to the roles through the following relations:

Empower (O-FW, S-DNS_Server, R-DNS_Srvr)
Empower (O-FW, S-FTP_Server, R-FTP_Srvr)
Empower (O-FW, S-Mail_Server, R-Mail_Srvr)
Empower (O-FW, S-Web_Server, R-Web_Srvr)
Empower (O-FW, S-Admin_Gtw, R-admn_Gtwy)
Empower (O-FWe, S-Inteface1, R-Ext_Interfaces)
Empower (O-FWe, S-Inteface2, R-Int_Interfaces)
Empower (O-FWe, S-Public_hosts, R-Public_host)
Empower (O-FWi, S-Inteface3, R-Int_Interfaces)
Empower (O-FWi, S-Inteface4, R-Int_Interfaces)
Empower (O-FWi, S-Inteface5, R-Int_Interfaces)
Empower (O-FWi, S-Private_hosts, R-Private_hosts)
Empower (O-FWi, S-Admin_Server, R-Admn_Srvr)

b) Activities

Activities correspond to various services available in the corporate network O-FW. Activities enable us to join together services to which are applied some common authorizations. We define a first activity T-All_tcp that will be associated with all defined the TCP actions. We define also T-Mail activity that will be associated with A-SMTP action, T-Web activity that will be associated with both A-http and A-https actions, T-FTP that will be associated with A-FTP action and T-DNS that will be associated with A-DNS action. In addition we define two other activities, T_admin_to_Gtwy that will be associated with both A-SSH and A-Ping actions and T-gtwy_to_admin that will be associated with A-SSH, A-Ping, A-FTP and A-SMTP. All these activities are relevant in organizations O-FW, O-FWe and O-FWi. The previous activities are modeled as the following:

Relevant _Activity (O-FW, T-All_TCP)
Relevant _Activity (O-FW, T-Mail)
Relevant _Activity (O-FW, T-Web)
Relevant _Activity (O-FW, T-FTP)
Relevant _Activity (O-FW, T-DNS)
Relevant _Activity (O-FW, T_admin_to_Gtwy)
Relevant _Activity (O-FW, T-Gtwy_to_admin)

The predefined actions are assigned to the activities through the following relations:

Consider (O-FW, A-All_TCP, T-All_TCP)
Consider (O-FW, A-SMTP, T-Mail)
Consider (O-FW, A-FTP, T-FTP)
Consider (O-FW, A-HTTP, T-Web)
Consider (O-FW, A-HTTPS, T-Web)
Consider (O-FW, A-SSH, T_admin_to_Gtwy)
Consider (O-FW, A-Ping, T_admin_to_Gtwy)
Consider (O-FW, A-SMTP, T-Gtwy_to_admin)
Consider (O-FW, A-FTP, T-Gtwy_to_admin)
Consider (O-FW, A-SSH, T-Gtwy_to_admin)
Consider (O-FW, A-Ping, T-Gtwy_to_admin)

c) Views

We suggest defining this kind of views as compound atoms having the form to target (r) [37]. We consider that

Table 3. Identified policy goals

Id	Policy goal
1	Internal corporate hosts can access all the resources on the Internet
2	External hosts can only access the servers in the DMZ
3	The DMZ servers can be updated only by the web administrator host admin server. Other corporate hosts have the same privileges as Internet hosts with respect to the DMZ servers
4	The firewall gateway interfaces are only accessible from the fw admin host and are otherwise inaccessible to any

Table 4. Generic policy of policy 1 for organization O-FWI

Source IP	Dest. IP	Source Port	Dest Port	protocol
111.222.2.*/24	*.*.*.*/0	> 1023	*	TCP
111.222.1.3/32	111.222.2.*/24	53	> 1023	TCP
111.222.1.2/32	111.222.2.*/24	25	> 1023	TCP
111.222.2.*/24	111.222.1.3/32	> 1023	53	TCP
111.222.2.*/24	111.222.1.2/32	> 1023	25	TCP
111.222.2.*/24	111.222.1.2/32	> 1023	80	TCP
111.222.2.*/24	111.222.1.2/32	> 1023	443	TCP
111.222.2.*/24	111.222.1.2/32	> 1023	21	TCP

every view defined as to target(r) is relevant in one of the organization of our example if r is a role relevant in this organization. We also consider that if role r_1 is a sub-role of role r_2, then the view to_target (r_1) is a sub-view of view to target (r_2).

4.3 Modeling security policy

The security policy is entered to the front end of our proposed framework. In our example we have four policy goals to be achieved. We assign a unique identifier for each policy goal. This unique identifier is considered the concrete link between the security policy and the modeled one. The unique identifiers for each policy goal are listed in Tab. 3.

Because of the space limitations we will present the modeled security policy for the concrete policy1 (policy with Id equals 1). The other concrete policy is modeled in the same way.

Permission (1, O-FW, R-Private_hosts, ALL_TCP, to_ target (public host)).
Permission (1, O-FW, R-DNS Srvr, DNS, to_target (private hosts)).
Permission (1, O-FW, R-Mail Srvr, mail, to_target (private hosts)).
Permission (1, O-FW, R-Private_hosts, mail, to_target (Mail_Srvr)).
Permission (1, O-FW, R-Private_hosts, T-web, to_target (R-Web_Srvr)).
Permission (1, O-FW, R-Private_hosts, T-FTP, to_target (R-FTP_Srvr)).
Permission (1, O-FW, R-Private_hosts, T-DNS, to_target (R-DNS_Srvr)).

4.4 Generec Policy Translation

To translate the modeled security policy into generic form, we need first to generate the security policy for each derived organization then translating it into generic form. Finally we will translate the generic form into intermediate level. In our example we have two derived organizations (O-FWi, O-FWe) so for space limitation we will generate the modeled security of policy1 for both derived organization as the following:

a) Modeled security Policy for O-FWi Organization

Permission (1, O-FWi, R-Private hosts, T-ALL_TCP, to_target (public host)).
Permission (1, O-FWi, R-DNS_Srvr, T-DNS, to_target (R-Private hosts)).
Permission (1, O-FWi, R-Mail_Srvr, T-Mail, to_target (R-Private hosts)).
Permission (1, O-FWi, R-Private_hosts, T-Mail, to_target (R-Mail_Srvr)).
Permission (1, O-FWi, R-Private_hosts, T-Web, to_target (R-Web_Srvr)).
Permission (1, O-FWi, R-Private_hosts, T-FTP, to_target (FTP_Srvr)).

b) Modeled security Policy for O-FWe Organization

Permission (1, O-FWe, R-Private hosts, T-ALL_TCP, to_target (R-Public host)).

Further we will translate the derived organization policy of policy1 into generic form. Table4 shows the translation of policy1 in O-FWi organization into generic form.

The final stage is to translate the generic form shown in the above table into intermediate language like XML script. Afterwards, this XML script is parsed through vendor specific compiler and translated to vendor specific configuration script.

5 CONCLUSION AND FUTURE WORK

This paper introduces a framework for automating the process of translation of high level security policy into low level security mechanisms. A key contribution of our proposed approach is its ability to encapsulate various network security components in a coherent model which is based on the OrBAC model. The proposed model extends OrBAC model to support other types of policies other than access control policy like auditing policy. Modularity of the network security policy to the level of unique policy item is also considered as an important contribution of the proposed model to facilitate managing and maintenance of the policies. In addition, we illustrate an application example to prove the ability of the proposed approach to cross the gap between high level security policy and low level security mechanisms.

Currently, a toolkit is under construction to implement our proposed framework hierarchical design. In addition a verification algorithm is to be developed to prove the

equivalence of the high level security policy and the corresponding low level one.

References

[1] BARTAL, Y.—MAYER, A.—NISSIM, K.—WOOL, A.: Firmato: a Novel Firewall Management Toolkit, IEEE Symposium on Security and Privacy, 1999, pp. 17–31.

[2] MAYER, A.—WOOL, A.—ZISKIND, E. Fang: a Firewall Analysis Engine: In Proc. of the IEEE Symposium on Security and Privacy, 2000.

[3] HASSAN, A.—HUDEC, L.: Role Based Network Security Model: A Forward Step Towards Firewall Management, In Workshop on Security of Information Technologies, Algiers, 2003.

[4] El KALAM, A. A.—El BAIDA, R.—BALBIANI, P.—BENFERHAT, S.—CUPPENS, F.—DESWARTE, Y.—MIEGE, A.—SAUREL, C.—TROUESSIN, G.: Organization Based Access Control, In Proc. of 4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2003), Lake Come, Italy, June 2003, pp. 120-134.

[5] GHORBEL-TALBI, M.—CUPPENS, F.—CUPPENS, N.—BOULAHIA, A.: Managing Delegation in Access Control Models, 15th International Conference on Advanced Computing and Communication (ADCOM'07), Guwahati, Inde, December 2007.

[6] MIEGE, A.: Definition of a Formal Framework for Specifying Security Policies. The Or-BAC Model and Extensions, PhD thesis, Ecole Nationale Superieure des Telecommunications, 2005.

[7] CUPPENS, F.—CUPPENS, N. SANS, T.—MIEGE, A.: A Formal Approach to Specify and Deploy a Network Security Policy, In Proc. of the 2nd Workshop on Formal Aspects in Security and Trust (FAST), Toulouse, France, August 2004.

[8] KANGASLUOMA, M.: Policy Specification Languages, Report of Helsinki University of Technology, Helsinki, Finland, November 1999.

[9] JOHNSON, M.—CHANG, P.—JEFFERS, R.—BRADSHAW, J. *et al*: KAoS Semantic Policy and Domain Services: an Application of DAML to Web Services-Based Grid Architectures, In Proc. of the AAMAS 03 Workshop on Web Services and Agent-Based Engineering, Melbourne, Australia, July 2003.

[10] The Web Ontology Language (OWL), http://www.w3.org/TR/owl-ref/.

[11] HOAGLAND, J. *et al*: Security Policy Specification Using a Graphical Approach, Technical Report CSE-98-3, University of California, Davis Department of Computer Science, 1998.

[12] BLAZE, M.—FEIGENBAUM, J.—IOANNIDIS, J.—KEROMYTIS, A. D.: In Secure Internet Programming: Security Issues for Mobile and Distributed Objects, Springer-Verlag, New York, NY, USA, 1999, pp. 185-210.

[13] BLAZE, M.—FEIGENBAUM, J.—KEROMYTIS, A.: Keynote: Trust Management for Publc-Key Infrastructures, In Proc. of the Security Protocls International Workshop, Cambridge, England, Springer-Verlag LNCS, April 1998, pp. 59–63.

[14] HINE, J.—YAO, W.—BACON, J.—MOODY, K.: An Architecture for Distributed OASIS Services, In Proc. of the Middleware 2000, New York, USA, Lecture Notes in Computer Science,, Springer-Verlag, April 2000, pp. 107–123.

[15] DAMIANOU, N.: A Policy Framework for Management of Distributed Systems, PhD thesis, University of London, 2002.

[16] LU, F.: Enforcing Ponder Policies using Kava, Master thesis, Victoria University of Wellington, 2004.

Received 29 November 2008

Ahmed A. Hassan and **Waleed M. Bahgat** biographies not supplied.