International Journal of Security and Its Applications
Vol. 3, No. 2, April, 2009

A Formal Policy Oriented Access Control Model for Secure Enterprise Network Environment

Manpreet Singh,
Punjabi University Patiala
msgujral@yahoo.com

Manjeet Singh Patterh
Punjabi University Patiala
mspattar@yahoo.com

Tai-Hoon Kim
Hannam University, Daejeon, 306-819, Korea
taihoonn@hnu.kr

Abstract

In this paper we use Security Evaluation Criteria as basis to develop the Network access control model for enterprise wide network computing environment. The Network access control model addresses both the access control and information flow control requirements of the enterprise network system. The security architecture of the model attempts to ensure authorized access to network resources and secure flow of information between network entities. The underlying concept of the Network access control model relies on the separation of the access control mechanism from the access control policy. This enables support for multiple access control policies within a single model specification. A further advantage of Network Access control model is that it is highly extensible, since it can be augmented with any new policy that a specific application or a user may require. The precision property is satisfied as network access control model is written in a formal mathematical notation. The property of simplicity is satisfied as only the security properties related to network computing system are modeled.

Keywords: *Access Control, Security Policy, Formal Methods, Evaluation Criteria.*

1. Introduction

We need enterprise level security technologies to protect enterprise information system network from internal threats. In this paper we adopt evaluation criteria based approach to develop a security policy model to protect enterprise information networks against internal threats. The advantage of using evaluation criteria based approach is that it provides requisite level of assurance through standard guidelines and processes.

As per security evaluation criteria approach high level of assurance requires formal methods and security policy models. In this paper we develop a formal security policy model as per security evaluation criteria Common Criteria (CC) [5-6] to provide a formal framework for implementing an Internal threat protection security solution in network computing environment.

The present practice followed by information security vendors is to develop their proprietary products and later getting it evaluated for getting security certification. But in this

paper we propose a reverse engineering approach in which we make an evaluation criterion as a basis for formulation of security requirement specification and then using these specifications for developing a security model. It would be more beneficial if ISO/IEC Common Criteria for Security Evaluation is used in the development phase rather than being used only for evaluation [1].

In our previous work [4] we used evaluation criteria Common Criteria as a basis to identify the security requirements to provide internal threat protection in network computing environment and corresponding security functional components to satisfy these requirements. The security functional components mean security functions which enforce security. In this paper we develop Network Access Control Policy Model (NAC-PM) to formally specify these security functional components and shows that security functional components satisfies the security requirements.

The paper begins by surveying the related work in the literature. The Section 3 describes the structural components of formal specification framework. The development process is described in section 4. Model verification is presented in section 5. In section 6, the concluding remarks with future scope of work are presented.

1.1. Motivation and Related Work

In this section we give an account of related work in the area of security policy models. The precise and explicit specification of security policies is important in order to achieve the organizational security objectives. We found major focus on RBAC model [2-3]. Most major information technology vendors are offering products that incorporate some form of RBAC.

RBAC assumes that all permission needed to perform a job function can be neatly encapsulated. In fact, role engineering has turned out to be a major obstacle for achieving a strong security in network computing environment. The challenge of RBAC is the contention between strong security and easier administration. Most of the RBAC products claim for providing an easier administration. Easier administration means fewer roles to manage with users operating with multiple roles. Assigning multiple roles to the user has been identified as major cause for easier realization of internal threats.

Another limitation of RBAC is that it does not leave access control to the discretion of the users and roles.Therfore, RBAC is difficult to use for supporting Discretionary Access Control policy which is one of the major security functional requirement to achieve internal threat protection.

To overcome the limitation of the RBAC, RBAC products need to be combined with rule based and other more tested access control methods to achieve the most practical solution. But when the policy combination is required, flexibility is hard to support and implement. So better solution is to restructure the security model to the support the new emerging security requirements of the network computing environments.

In literature we found some work based on Common Criteria primarily focusing on requirement engineering which is the first step of software development life cycle and prerequisite for model development. In [7], Mellado et al proposed a process that integrates Common Criteria into the software lifecycle so that it unifies the concepts of requirement

engineering and security engineering. Lee et al [8] in their work developed a common Criteria based security engineering process to achieve high assurance. In [9] Vetterling et al. proposed secure systems development based on common criteria. Morimoto et al. [10] proposed a security specification verification technique based on the international standard ISO/IEC Common Criteria. Keblawi et al. [11] in their work explained with case study how Common Criteria can be applied to specify security requirements in large systems. Our major source of inspiration behind our proposed approach is the recent work[12] in which Cheng et al. emphasized on the need for a systematic security engineering environment to provide designers, developers, users, and maintainers with standard, formal, and consistent supports for design, development, operation, and maintenance of information systems with high security requirements.

In our research work our target system of evaluation is a network computing environment. In our previous work [4] we derived network interpretation of security functional components defined as a part of standard security evaluation criteria. The derived network interpretation components are used in this paper as basis to develop an access control framework for secure network computing environment.

2. Network System Security Issues and Challenges

To many people including some expert's, network security is encryption and the sole purpose of network security is to prevent wiretapping. Through this paper it would become clear that the total network security problem is far more than the wiretapping threat, and encryption is just part of the solution. Some of the important issues and challenges related to network security are as follows.

***Research Focus*:** The major issue is the focus of research community. Secure networks have been studied much less than secure computer systems, and few practical examples of them exist. It is easy to find pieces of network security solutions particularly concentrating on the use of cryptography. Finding an example of an integrated secure enterprise network system is much harder. The problem is not in the technology but in the lack of an accepted security policy model for network system. For a single computer system it is easy to describe a generally accepted security model as being composed of well defined system subjects, objects and applications. But with the network system things are much more complex.

***Network Subjects and Objects*:** Network system is constructed as a hierarchy of layers, each of which implements a specific type of network service. Defining network subjects and objects relevant to the network system security requirements becomes a complex task as different layers in the architecture support different network subject, object and concepts. For example, if we consider ISO layer 5, 6 and 7, network users may be treated as network subject and files, mail etc can be considered as network object. In case of ISO layer 4, processes and application may be treated as network subject and network connection may be treated as network object. In case of layer 3, network hosts, terminal can be treated as network subject and packets may be treated as network objects. As there are several options for the interpretation of network subjects and objects, standard definitions are required for the identification of subjects and objects in network environment.

***Network System Environment*:** In order to describe additional challenges involved in designing secure enterprise network environment, let us consider a typical network

environment. Network system is a collection of network components that include hardware, firmware and software necessary to provide a desired functionality. Network hardware includes network hosts, terminals, network communication devices and network printers etc. A network component provides all or a portion of the total functionality required of a network system. This view of network system give arises to additional security requirements such as communication security, denial of service and transmission security. These requirements normally do not arise in the case of standalone systems. Making provisions for these additional security requirements further complicate the formulation of the security model.

Network System View: In the context of network system, the selection of an appropriate abstraction of network system for policy enforcement requires the understanding of operational and technical characteristics of the environment in which a network exists. If we consider a single trusted system view of network system, then the overall network security policy need to be decomposed into policy elements and these policy elements further needs to be allocated to the appropriate component. Reliable enforcement of overall network policy rests on how these policy elements are distributed and allocated to various components. If we consider a multiple system view of the network system, then the components of the network system need independent management and accreditation. In this case reliable enforcement of network policy rests on the interconnection rules between independently accredited network components and their accreditation ranges.

Network Connection Policy: In the case of networking environment, the overall network security requirements include controlling the establishment of authorized connections across the network. The connection control security requirements are specified through connection control policy. From an overall network perspective connection control policy may be is in accordance with connection control security requirements, but specifying such a policy in terms of component level abstraction is a challenging task.

***Network Information Flow Policy*:** In the case networking environment, the overall network security requirements include controlling the flow of information between network entities. The information flow control requirements are specified through information flow control policy. The role of the network as a whole in controlling information flow may be more easily understood, but extending information flow control policy to the reference monitor requirements of individual components in the network is not a straightforward task.

Network System Architecture: The network system architecture must demonstrate the linkage between the network system view at abstract level and its realization in the individual components of the network. Designing such architecture is a challenging task as it requires defining security functionality at the component level.

2.1 Network System Evaluation Criteria

This section is a concise introduction to the work related to formulation of the network evaluation criteria for secure network system. The evaluation criteria provide a basis for specifying specific security requirements and formulation of formal security models. The Department of Defense (DoD) published Trusted Computer System Evaluation Criteria (TCSEC) to provide a means of evaluating specific security features and requirements in standalone computer systems. The TCSEC was the first widely used formal evaluation methodology, and subsequent methodologies built and improved on it over time. The basic

philosophy of the protection described in the TCSEC requires that the access of subjects to objects be mediated in accordance with an explicit and well defined security policy. After the publication of TCSEC, efforts started in the direction of extending the TCSEC concepts to network systems. The National Computer Security Centre, Trusted Network Interpretation (TNI) of TCSEC is considered as significant effort in the development of network system evaluation criteria.

The TNI is the interpretation of the TCSEC for network system i.e. it extends the guidance provided for standalone computer system to network system. The TNI provided a standard to manufacturers as to what security features to build into their new and planned network products. The TNI offered two approaches: evaluation of the networks and evaluation of network components. The network approach addressed centralized networks with a single accreditation authority, policy and Network trusted computing base. In the first part of the TNI, the TCSEC criteria was interpreted for networks, and one could evaluate a network at the same levels offered by the TCSEC. The second part of the TNI offered evaluation of network components based on the specific functionality that the component offered. A network component may be designed to provide a subset of the security functions of the network as whole.

The TCSEC and its interpretations provided a process for security evaluation of commercial products and heightened the awareness of the commercial sector to the needs of security in computing environment. After being widely used for nearly two decades, the inadequacies and criticisms stimulated a wave of new approaches that addressed many areas of concern, including limitation of scope, binding of assurance and functionality and inflexibility in selection of requirements, to name the most significant ones. New methodologies were developed to address these issues. Most notable of these were the Information Technology Security Evaluation Criteria (ITSEC) in Europe, the Canadian Trusted Computer Product Evaluation Criteria (CTCPEC), and the Federal Criteria (FC) in the United States. These foundational methodologies have culminated in the Common Criteria (CC)[5-6], which today has world-wide support.

In the CC, the TCSEC concept of a trusted computing base is generalized and replaced with the term TSF (TOE Security Functionality).The TSF is a set consisting of all hardware, software, and firmware of the TOE(product or system) that must be relied on for the correct enforcement of the security policy. In CC there are 11 classes of security functional requirements, each having one or more families. For Network System Evaluation Criteria, we consider the class User Data Protection (Class FDP) as an important class as it includes the two important security functional requirements for network computing environment. The Class FDP includes two different types of security policies. These are access control and information flow control polices. The difference between these two types of policies is essentially that an access control policy makes decisions based on discrete sets of information, such as access control lists or access permissions, whereas an information flow control policy addresses the flow of information from one repository to another. A discretionary access control policy is an access control policy and mandatory access control policy is an information flow control policy. These families are also represented in other methodologies, but they are generalized in the CC, for flexibility.

In this paper we will be addressing only these two security functional requirements families in our security policy model for network systems [Singh and Patterh,2007].

Informally, in the context of network systems, the above two security functional requirements families can be described as follows:

Network Discretionary Access Control: The network discretionary access control policy is based on the identity of the network components, implemented in the form of an authorization list .The authorization list determines whether a connection is allowed to be established between network entities. The individual components may in addition impose their own controls over their users. This is in conformance with the Common Criteria security functional requirements.

Network Information Flow Control: The network information flow control policy requires appropriate labeling mechanism to be present for both network components and network information units. A access policy based on the labels of network components is used to control the establishment of network connection between two entities. A policy based on the labels of network information units is used to control the flow of information between different network subjects and objects, when performing operations involving information transfer over the network. This is in conformance with the Common Criteria security functional requirements.

In the next section we begin with the design of network security policy model in conformance with the basic principles described in the Trusted Network Interpretation and Common Criteria.

3. The Formal Model Framework

Our security requirement specification framework for the formal Network Security Policy model consists of the Formal Model of Network Security Policy, Formal Specification of Security functional components and Verification of formal model of Network Security policy. The formal model of network security policy is divided into three models for structural representation. These are data model, state machine model, and policy model. Formal specification of security functional components is provided for identifying consistency between network security policy model and security function specifications. Verification of formal model is for ensuring consistency and completeness of the network security policy model.

3.1 Components of Formal Network Security Policy Model

Data Model: The data model introduces the basic sets such as network subjects and objects that represent entities of the network security policy.

Formal State Machine Model: The state machine model specifies the secure state of the underlying formal model of security. It comprises of the model entities defined in the data model and the invariant relationships between these entities.

Policy model: The policy model specifies, through definition of network operation, how operations on secure state are constrained in order to satisfy the network security policy.

4. Formal Security Policy Model – A Case Study

4.1. Formal State Model - Abstract State

The formal model we describe here is state machine based model. We shall refer to this model as Network Access Control Policy Model (NAC-PM) .We consider network system as a collection of entities and values. The set of relationship at any time between entities and values constitutes the state of the system. The state of the system changes whenever any of these relationship changes. Let us denote the set of possible states of the system with S. Some subset of S consists of exactly those states in which the system is authorized to reside. So whenever the system state is in authorized state, the system is secure. In addition, we also need to ensure that the system state is always an element of authorized state. Formally Network Access Control Policy Model (NAC-PM) is specified in the following schema.
We define network policy model, NAC-PM as follows:

NAC-PM = $\langle S, NOp, ST, s_0 \rangle$ where

S : is a set of states
NOP : is a set of operation
ST : is a system transition function
s_0 : is an initial state.

The set *NOP* describes the network operations related to connection control, information manipulation and flow control. The transformation function *Systran()* describes the transition from one state to another state by applying one or a sequence of operations from the set *NOP*

4.2. Model State Variables

In Step 1 we need to define security relevant state variables. Each state $s \in S$ of the system at any one time is expressed as an element of

S = {NSub, NObj, AM, SF, SR, RA, UL, CD}

Now let us briefly describe some of the terms involved in the state definition. An active entity in the system, which can be a user or an application process operating on behalf of users, is define as Network Subject. The set of all subjects is called *NSUB*. The network object set includes the set of all entities designated as object in enterprise network system. We consider an object to be any resource in the system that can be assigned access rights. The set of all objects is called *NOBJ*. Formally; a set of objects is associated to a subject through the function *NsubRef*.

The set AM of access control matrices indicates which subject has the access right to which network object. The set comprises of 3-tuples (Auth (Nsub, Nobj, a), Conn (Nsub, Nobj, a) and Acc (Nsub, Nobj, a)). The null value $\{\phi\}$ for *'a'* designates empty set, i.e. no access right entry exists for (*nsub, nobj*).The authorization access triple Auth (Nsub, Nobj, a) is a set of accesses that indicates the subject nsub $\in$ Nsub has an access right to connect to the network object neobj $\in$ Nobj. The connection access triple Conn (Nsub, Nobj, a) gives the current set of authorized connections at given state. The Access triple Acc (Nsub, Nobj, a) is a set of accesses that indicate the nsub $\in$ Nsub has an access right to access Information entity say ieobj $\in$ Nobj.The entries in the above access matrices is based on the discretionary policy associated with each network device.

The security association function SF is the set of 3-tuples (sf_s, sf_o, sf_c) used to bind each entity to an access class ac $\in$ Acls. The sf_s : Nsub $\rightarrow$ Acls is a clearance function used to bind each subject (user and process) to access class. Thus ac = sf_s (nsub) represents the clearance of nsub $\in$ Nsub. The sf_c : Nsub $\rightarrow$ Acls is a current access class function used to bind each subject to a access class representing the current access class of nsub such that sf_c (nsub) $\in$ Acls and $sf_c(nsub) \leq sf_s(nsub)$. The sf_o : Nobj $\rightarrow$ PS (Acls) is a classification function used to bind each object to one or more access class where PS denotes the power set of Acls. This classification of nobj $\in$ Nobj can be represented as sf_o (nobj) = ac.

The subject referencing function SR: Nsub $\rightarrow$ PS (Nobj) is a mapping which indicates the set of objects referenced by a subject where PS denotes the power set of Nobj.

The role assignment function RA is the set of 2-tuples (ra_a, ra_c) used to identify the roles of the user. The ra_a : Users $\rightarrow$ PS (Rset) is a role assignment function that gives the authorized set of roles for the user where PS denotes the power set of Rset. The ra_c : Users $\rightarrow$ Rset is a current role function that gives the current role of a user.

The user login function UL: Users $\rightarrow$ T is a function which gives the terminal in which a user is logged on. The content mapping function CD: IE $\rightarrow$ {DATA} is a function which maps the set of information entities into the set of Data strings. It gives the contents of information entities.

4.3. Secure State

In this step, list of legal network operation are defined. The fundamental approach used here is to capture the security constraints of the system and express them from two different points of view: The state based and Operation based. Describing two overlapping perspectives means that a certain amount of duplication can arise, but this also gives two natural approaches to validation. With two level of constraint specification, it is easier to be able to cross-check two such views than to work with a single complex view. In this section we focus on the state based view followed by operation based view in next section.

Our primary goal of presenting the state based view is to define the secure state for the enterprise network system. For this purpose, firstly we need to identify all the properties of the secure network state. In order to identify these security properties we need to consider the security condition during the different phases of User interaction with enterprise network system. After going through different phases of Network system operations, the security properties of the secure state may be summarized as follows.

- Login Property
- Connection Property
- Information Access Property
- Authorized User Role Property

These different security properties must hold in any secure state for all the network entities. We begin with the login property.

1. Login Property: The Login property with security constraints is statically represented in *LoginProp schema*. *A state satisfies user login constraint if* $\forall x \in Users$

1. $sf_s(x) \geq sf_o(UL(x))$
2. $sf_s(x) \geq sf_c(x)$
3. $ra_c(x) \in ra_a(x)$.

2. Connection Property: The Connection property with security constraints is statically represented in *ConnProp* schema.

A state satisfies connection establishment constraint if $\forall$ *(nsub, neobj)* $\in$ *Conn (Nsub, Nobj, a) such that* $a \neq \phi$,

1. *(nsub,neobj, a)* $\in$ *Auth (Nsub, Nobj, a) such that* $a \neq \phi$,
2. $neobj \notin NOD \Rightarrow sf_c(nsub) \geq glb(ac_1, ac_2, ..., ac_n)$ *where* $sf_o(neobj) = \{ ac_1, ac_2, ... ac_n \}$.
3. $neobj \in NOD \Rightarrow sf_o(neobj) \geq sf_s(nsub)$.

where $glb(ac_1, ac_2, ..., ac_n)$ *represents greatest lower bound.*

3. Information Access Property: The Information Access property with security constraints is statically represented in *InfoAccProp* schema.

A state satisfies information control constraint if $\forall f \in IE, \forall z \in NE$

1. $(f, CD(f)) \in explore(z)$, *where explore(z) is a set of ordered pair* $\{(f_1, v_1), (f_2, v_2), ... (f_n, v_n)\}$, *and* v_i *is displayed on the network entity object z which can be host, terminal or a network output device. Each* f_i *is an information entity object and* v_i *is the result of applying the content mapping function to* f_i .
2. $sf_o(z) \geq sf_o(f)$.

4. Authorized User Role Property: The Authorized User Role property with security constraints is statically represented in *UserRoleProp* schema.

A state satisfies user role constraint if $\forall u \in Users$

1. $ra_c(u) \in ra_a(u)$.

After defining the different security properties, we are now in position to define the secure state of the system.

A state s is Secure if

1. *s satisfies the User Login Constraint.*
2. *s satisfies the Connection Establishment Constraint.*

3. *s satisfies the Information Control Constraint.*
4. *s satisfies the User Role Constraint.*

4.4. Network Operation Constraints

At the outermost level of the specification, the system is considered to be modeled by the initial state followed by an arbitrary sequence of legal operations. Operations on the system will cause a change of state. There are invariants which relate the before and after states for all operations on the system. Here we use the convention of placing the prime symbol ` in front of a state variable to refer to the new state. Unprimed variables refer to the value in the old state. We begin with description of administrative level operation followed by user level operation. Network administrative operation are used to manipulate security attributes of the subjects and objects, addition and deletion of subjects and objects and all other administrative tasks to ensure secure state of network computing environment.

We here give example of some fundamental operations related to network objects like addition, deletion and manipulation of security attributes. The addition operation, when executed by a $nsub \in Nsub$ with a classification *ac* results in the creation of *neobj* such that $sf_o(nobj) = sf_c(nsub) = ac$ where $ac \in Acls$. We can put the security requirements of this operation as follows.

Add_nobj (neobj, ac)
{
If $neobj \notin Nobj$
Then $`Nobj = Nobj \cup \{neobj\}$
Set sf_o (neobj) = sf_c (nsub) = ac and,
$\forall nsub \in Nsub,$ (nsub, neobj, a) $\in$ `Auth (nsub, neobj, a) such that a $= \phi$,
$\forall nsub \in Nsub,$ (nsub, neobj, a) $\in$ `Conn (nsub, neobj, a) such that a $= \phi$.
}

The operation for configuring authorization mode with security constraints is illustrated in Set_auth_mode schema.

Set_auth_mode *(nsub, neobj, a)*
{
$\forall nsub \in Nsub,$ (nsub, neobj, a) $\in$ Auth (nsub, neobj, a)
such that a $= \phi$, where $neobj \in Nobj$.
$\forall nsub \in Nsub,$ Set (nsub, neobj, a) $\in$ `Auth (nsub,
neobj, a) such that a $\neq \phi$,
$R_{NA} \in ra_a(sub)$ and $ra_c(sub) = R_{NA}$.
}

where R_{NA} represents network administrator role. Similarly the schemas for other network operation can be defined.

User level operations are used by network authorized users for information access and manipulation. The purpose of these user level network operations is to constrain the types of changes that the system user may make. An example user level operation is as follows. The connect request operation for network object, when executed by a $nsub \in Nsub$ with a classification *ac* allows a network subject to connect to a remote network entity $neobj \in Nobj$. The network subject set include the set of Users (U) and all application processes (AP) executing on behalf of the users i.e. Nsub = U $\cup$ AP where $nusr \in Nsub$. The security requirements for this operation can be defined as follows.

Connect_nobj_req *(nsub, neobj)*
{
$nsub \in Nsub$, (nsub, neobj, a) $\in$ Auth (nsub, neobj, a) such that a $\neq \phi$
If neobj $\notin$ NOD *then* sf_c (nsub) $\geq$ glb $(ac_1, ac_2, ..., ac_n)$
where sf_o (neobj) = { $ac_1, ac_2, ...ac_n$ }.
Set (nsub, neobj, a) $\in$ `Conn (nsub, neobj, a) such that a $\neq \phi$ and $neobj \in `SR(nsub)$
Else if neobj $\in$ NOD *then* sf_o (neobj) $\geq$ sf_s (nsub).
Set (nsub, neobj, a) $\in$ `Conn (nsub, neobj, a) such that a $\neq \phi$ and $neobj \in `SR(nsub)$
}

The connection request operation for information entity object, when executed by a $nsub \in Nsub$ with a classification *ac* allows a network subject to connect to a remote information entity object $ieobj \in IE$ .Before the *nsub* links itself with a *ieobj* in a network entity *neobj*, a network connection should exist between *nsub* and *neobj* i.e. connect constraint must be satisfied for (nsub, neobj) connection. The security requirements for this operation can be defined as follows.

Connect_iobj_req (*nsub, ieobj@neobj*)
{
$nsub \in Nsub$, (nsub, ieobj, a) $\in$ Acc (nsub, ieobj, a)
such that a $\neq \phi$, $sf_c(nsub) \geq sf_o(ieobj)$
$ieobj \notin SR(sub)$ where $sub \in Nsub$
Set $ieobj \in `SR(nsub)$ where $nsub \in Nsub$
}

In the next section we consider the verification of the proposed model.

5. Model Verification

The model verification consisted of two parts: the definition of an initial state, and an informal argument that each state transition function could produce a valid, secure final state when applied to a valid, secure start state. The second part of model verification requires critical examination of all those phases of system functionality during which system may

undergo a state transition. The three major phases identified for model verification are Login Phase, Connection Phase and Network Operation Phase. Our aim here is to examine the security properties of the network system as it undergoes state transition and verify that the network system satisfies all the required security properties.

1. User Login Phase: The security conditions that need to be satisfied during this phase are rightly specified by User Login Property. As no other operation is executed during this phase, therefore system starting with initial state satisfying security conditions of User Login Property will never go to an insecure state. We can now formally state this as follows:
Model_Constraint 1: The system described by network security policy model NAC-PM = $\langle S, NOp, ST, s_0 \rangle$ satisfy the security conditions of User Login Phase if initial state s_0 satisfies the User Login Constraint.

2. Network Connection Phase: During this phase, network user tries to establish a connection with network resources available at remote network entity after successfully logging onto network system. Before the request for network connection is granted, the mandatory connection conditions and discretionary connection condition must be satisfied. These conditions are rightly specified as a part of Connection Property.

For a system described by NAC-PM and starting at initial state, a system is said to be secure if the initial state satisfies the security condition of Connection Property. On application of sequence of system transition functions, system will undergo transition resulting in a sequence of states $\{s_0, s_1, s_2, ...\}$. To maintain the secure state of the system, every state in a sequence $\{s_0, s_1, s_2, ...\}$ starting from previous secure state s_i need to satisfy the security condition of the Connection Property. We can now formally state the model constraint during the connection phase as follow.

We assume here the initial state to be secure state. When Users requests connection to network resources of network system with Login Property and Connection Property conditions satisfied, the state of the system will remain in secure state.
Model_Constraint 2: The system described by network security policy model NAC-PM = $\langle S, NOp, ST, s_0 \rangle$ satisfy the security conditions of Network connection Phase if a) the initial state s_0 satisfies the Connection Establishment Constraint and b) the system transformation functions $ST : Nsub \times NOp \times S \rightarrow `S$ satisfies the security conditions of network operation NOP.

3. Network Operation Phase: During this phase, the user tries to perform a sequence of network operations involving information transfer from one network entity to another. The first important security condition that is required before executing any network operation is to obtain an authorized network connection. The system may move to an insecure state during this phase if the execution of network operations is allowed by NAC-PM without having an authorized network connection. The second important concern during this phase is the sequence in which network operation are performed. The sequence of network operation may also cause the system to move to an insecure state from secure state. We can now formally state the model constraint during the network operation phase as follow.
Model_Constraint 3: The system described by network security policy model NAC-PM = $\langle S, NOp, ST, s_0 \rangle$ satisfy the security conditions of Network operation Phase if a) Connection Establishment Constraint is satisfied before executing network operations. b) The network operation security conditions are satisfied before their execution. c) The security transition function ST satisfies the security condition required to maintain the secure state.

After defining the model constraints, we are now in a position to state the security theorem to show that a system described by NAC-PM is secure if its initial state is secure and every request for network resource access and information transfer satisfies the conditions stated in model constraint 1, 2 and 3.

We can now formally state the security theorem as follow.

Security Theorem: A system described by the model NAC-PM = $\langle S, NOp, ST, s_0 \rangle$ is secure, if

- The initial state s_0 is a secure state.
- Any system transition ST defined by
  $ST(nsub, ad\_op \mid ul\_op, s) = `s$ satisfies
  1. *Model_Constraint 1*
  2. *Model_Constraint 2*
  3. *Model_Constraint 3*
- $\forall `f \in IE_{`s}, \forall f \in IE_s, \forall `z \in NE_{`s}, \forall z \in NE_s$ $if\,(f, CD(f)) \notin \exp lore(z)$ & $if\,(`f, CD(`f)) \in \exp lore(`z)$
  then $sf_o(`z) \geq sf_o(`f)$.
  where the states s and s` are given as follows:
  s = $\{Nsub_s, Nobj_s, AM, SF, SR, RA, UL, CD\}$
  `s= $\{Nsub_{`s}, Nobj_{`s}, `AM, `SF, `SR, `RA, `UL, `CD\}$
- For all users $\forall u \in Users$, $`ra_c(u) \in `ra_a(u)$.

The security theorem presented above provides the formal statement of network security requirements for any security mechanism to be implemented for secure enterprise network environment.

6. Conclusion

We used network interpretation of the security functional components of Common Criteria to model the access control framework. We used set theoretic notation to model the key components of Network Security Policy Model. The schema describing the basic system elements was large due to multiple security constraints of network computing environment. In our future work our focus is to use logical formalism to produce an animation of the formal specification to further refine the framework.

References

[1] Stoneburner, G. 2005. Developer-Focused Assurance Requirements. Computer 38(7), 91-93,2005.
[2] Sandhu, R. S., Coyne, E., Feinstein, H. and Youman, C. (1996). Role-Based Access Control Models. IEEE Computer.29(2):38-47.
[3] Sandhu, R., Ferraiolo, D. and Kuhn, R. (2000). The NIST Model for Role-Based Access Control: Towards A Unified Standard. In Proceedings of the 5th ACM Workshop on Role-Based Access Control, Berlin, Germany, pp. 47-61, 26-28.
[4] Singh, M. and Patterh, M.Security Functional Components for Building a Secure Network Computing Environment. International Journal of Information Systems Security, 16(6), pp. 332-343, Nov 2007.
[5] Common Criteria for Information Technology Security Evaluation (CC) (2005). version 2.3, ISO/IEC 15408:2005, August.
[6] Common Criteria for Information Technology Security Evaluation (CC). (2006). version 3.1 Revision 1, September.
[7] Mellado, D., Fernández-Medina, E., and Piattini, M. 2007. A common criteria based security requirements engineering process for the development of secure information systems. Computer. Stand. Interfaces 29(2), Feb 2007.

[8] Lee, J., Lee, S., and Choi, B. 2003. A CC-based Security Engineering Process Evaluation Model. In Proceedings of the 27th Annual international Conference on Computer Software and Applications COMPSAC. IEEE Computer Society, Nov 2003.
[9] Vetterling, M., Wimmel, G., and Wisspeintner, A. 2002. Secure systems development based on the common criteria: the PalME project. SIGSOFT Softw. Eng. Notes 27(6),Nov. 2002.
[10] Morimoto, S., Shigematsu, S., Goto, Y., and Cheng, J. 2006. A security specification verification technique based on the international standard ISO/IEC 15408. In Proceedings of the 2006 ACM Symposium on Applied Computing, France, April 2006.
[11] Keblawi, F. and Sullivan, D. 2006. Applying the Common Criteria in Systems Engineering. IEEE Security and Privacy 4, 2 (Mar. 2006), 50-55.
[12] Cheng J. ,Yuichi, G., Morimoto, S. ,Daisuke, .H. A Security Engineering Environment Based on ISO/IEC Standards: Providing Standard, Formal, and Consistent Supports for Design, Development, Operation, and Maintenance of Secure Information Systems. International Conference on Information Security and Assurance pp. 350-354,2008

Authors

Manpreet Singh received the B.E and M.E degree in engineering from Punjab Technical University, Jalandhar, India and Thapar University, Patiala, India respectively. He did his PhD from Punjabi University, Patiala, India. His current interests are computer networks, access control and information security. He has been in teaching and research for the last 10 years. He has published over 18 papers at national and international levels. Presently he is with Faculty of engineering, Punjabi University, Patiala. India. He is life member of ISTE.

Prof. Manjeet S.Patterh, did his Bachelor's degree from Madhav Institute of Technology and Science (MITS), Gwalior (MP) and Master's degree from Birla Institute of Technology and Science (BITS), Pilani, both in Electronics Engineering. He did his PhD from Punjab Technical University Jalandhar. He has published 16 papers in international and national refereed journals and 30 papers in international and national conferences. He is having over 17 years of teaching experience. He is presently working as Professor in department of electronics and communication engineering at Punjabi University Patiala. His current interests are Digital Signal Processing, Wireless Communication Systems and Networking. He is member of IEEE and life member of ISTE, IE (I) and IETE

Prof. Tai-hoon Kim, M.S., Ph.D (Electricity, Electronics and Computer Engineering), currently, Professor of Hannam University, Korea. His research interests include Multimedia security, security for IT Products, systems, development processes, operational environments, etc. He has 15 Years of experience in Teaching & Research. He has already got distinctive Academic Records in international levels. He has published more than 150 Research papers in International & National Journals and Conferences.