Formal Specification of Common Criteria Based Access Control Policy Model

Manpreet Singh [1] and Manjeet S. Patterh [2]

(Corresponding author: Manpreet Singh)

Department of Computer Engineering, Punjabi University, Patiala, India[1]

Department of Electronics and Communication Engineering, Punjabi University, Patiala, India[2]

(Email: msgujral@yahoo.com)

(Received Sept. 20, 2008; revised and accepted May 12, 2009)

Abstract

One of the major threats that an enterprise Information system networks are facing today is the Insider threat. As part of the Insider Threat study, lack of an effective access control mechanism is identified as one of the major causes that facilitated IT sabotage. In this paper we propose a network access control meta model as per ISO/IEC security evaluation criteria - Common Criteria to provide a framework for implementing an Insider threat protection security solution for network computing environment. We used formal specification notation language Z to specify the proposed model. The paper concludes with a case study along with model verification.

Keywords: Access control, evaluation criteria, formal methods, security policy model

1 Introduction

Insiders, by virtue of legitimate access to their organizations' information, systems, and networks, pose a significant risk to the organization. The Insider Threat Study [1, 6, 15] provided the first comprehensive analysis of the insider threat problem. As part of the Insider Threat study lack of an effective access control mechanism is identified as one of the major causes that facilitated IT sabotage. Ninety three percent of the insiders in the IT sabotage cases exploited insufficient access controls. Other causes of Insider threat include System misconfigurations, disgruntled employees and overloaded system administrators etc. In this paper we focus on access control element of the insider IT sabotage problem.

To protect enterprise information system network from insider threats, we need enterprise level security technologies. In this paper we adopt evaluation criteria based approach to develop a policy oriented access control meta model to protect enterprise networks against Insider threats. In this paper we develop a formal security policy model as per security evaluation criteria Common Criteria (CC) [8, 9] to provide a formal framework for implementing an Internal threat protection security solution against unauthorized access in network computing environment. The present practice followed by information security vendors is to develop their proprietary products and later getting it evaluated against Evaluation Criteria's 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 [26]. The security evaluation criteria Common Criteria (CC) defines the Security Policy as the set of rules that regulate how resources are managed, protected and distributed within a target computing environment, expressed by the security functional requirements. In the CC, the Security Policy Model (SPM) is a structured representation of the security policy to be enforced by the security product. The SPM represents the precise relationship between the security requirements and the function specifications. The SPM shows that the security functions satisfy the security requirements.

In this paper we use SP and SPM to mean Network Security Policy (NSP) and Network Access Control Policy Model (NAC-PM), respectively. In our previous work [25] 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 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.

2 Motivation and Related Work

Internal attacks that continue undetected can cause serious harm to an organization. Perhaps most significant, they can expose the personal information of customers or employees. A breach of this kind - whether it is identity theft, inappropriate use of data or the sale of sensitive information - can leave an organization legally liable for associated damages and subject to regulatory fines. In addition, a company's competitive position could suffer if an insider uses intellectual property or trade secrets for unauthorized purposes. Attacks may also be designed to extort money or damage an organization's reputation. If they lead to IT downtime or damaged systems, they can also disrupt business operations and reduce the value of IT investments. With so much at stake, it is becoming increasingly important to address the threat of insider attacks - *before* they occur.

In this section we give an account of related work. Currently, there exist a rich set of formal security models that can translate enterprise security objectives. This includes Discretionary Access Control (DAC) [18], Mandatory Access Control (MAC) [2, 5], Role-Based Access Control (RBAC) [23, 24, 10], OrBAC [14] etc. Of the numerous recognized access control policies, today's Security products rigidly limit enforcement to a small subset of known policies. Most of the major information technology vendors are offering products that incorporate some form of RBAC. For example, all major DBMS products support RBAC. Microsoft has brought RBAC to the Windows operating systems by introducing Authorization Manager in Windows Server 2003 [21]. But to provide comprehensive Internal threat protection, multiple access control policies are required, and RBAC alone is not sufficient to address the security requirements of an organization. Moreover, in current scenario RBAC based products has failed to meet the organization strong and emerging security objectives. The cause of the failure is the assumption on which all RBAC based products are designed. 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 real world 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 results in ad-hoc Enterprise policies and has been identified as major cause for easier realization of Internal threats. To overcome the above problem, meta model based approach is required that can serve as a unifying framework for specifying and comprehensively enforcing any access control policy.

But when the policy combination is required, flexibility is hard to support and implement. So better solution is to restructure the security policy model to the support the new emerging security requirements of the network computing environments. In literature we found some work in this direction but they are limited to DBMS level or Operating system level [3, 4, 13, 17]. Ferraiolo et al. [17] discussed the issues related to meta policy models. Given the large diversity and types of access control policies being used in enterprise computing environment, unified framework is the only solution to for specifying and comprehensively enforcing any access control policy.

In this paper we used Common Criteria as a basis to model a unified access control framework for providing protection against access control oriented insider threats. In literature we found some work in support of our proposed approach. The research work based on Common Criteria, primarily focused on requirement engineering which is the first step of software development life cycle and prerequisite for model development. In [20], Mellado et al. proposed a process that integrates Common Criteria into the software life cycle so that it unifies the concepts of requirement engineering and security engineering. Lee et al. [19] in their work developed a Common Criteria based security engineering process to achieve high assurance. In [27] Vetterling et al. proposed secure systems development based on common criteria. Morimoto et al. [22] proposed a security specification verification technique based on the international standard ISO/IEC Common Criteria. Keblawi et al. [16] 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 [7] 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 [25] 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. We used formal specification notation language Z [12] to specify the proposed model.

3 The Formal Model Framework

Our security requirement specification framework for the formal Network Access Control - Policy model consists of the following.

1) Formal Model of Network Security Policy.

2) Formal Specification of Security functional components.

3) Verification of formal model of Network Security policy.

For Internal threat protection in network computing environments, the network security policy model presently focus only on two major families of Security functional policies(SFPs): Access Control SFPs and Information Flow Control SFPs. Access control SFPs base their policy decisions on attributes of the users, resources, subjects, and objects. These attributes are used in the set of rules that govern operations that subjects may perform on objects. Information Flow Control SFPs base their policy decisions on the attributes of the subjects and information within the scope of control and the set of rules that govern the operations by subjects on information. The attributes of the information may be associated with the attributes of the container or may be derived from the data in the container. The attributes stay with the information as it is processed by the TOE Security Functionality.

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 Data Model

4.1.1 Basic Sets

1) Network Subject
   An active entity in the system, which can be a user or an application process operating on behalf of users. The set of all subjects is called *NSUB*.

2) Network Object
   This 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*.

4.1.2 Cartesian Product Type

In the NAC-PM model Cartesian product type is used to specify different authorization capability list. In the following the example of the Cartesian product types used in the model is presented.

1) Connection Authorization List
   The specification of the network connection authorization list as Cartesian product type is as follows.

$$NConnAuthLists : \mathbb{P}(NOBJ \times \mathbb{P}AUTHMODE).$$

2) Access Authorisation List
   The specification of the network access authorization list as Cartesian product type is as follows.

$$NAccLists : \mathbb{P}(IEOBJ \times \mathbb{P}ACCMODE).$$

4.1.3 Relations

In the NAC-PM model relations are used to represent the association between different network entities. In the following the example of the relations used in the model are presented.

1) Network Subject Reference
   A set of network objects is associated to a network subject through the relation $NSubRef$: $NSUB \longrightarrow \mathbb{P}NOBJ$. The relation NSubRef can be specified as follows:

[**NSubRef**]
$dom : NSubRef \rightarrow \mathbb{P}Nsub$
$ran : NSubRef \rightarrow \mathbb{P}Nobj$

$\forall NSubRef : NSUB \rightarrow \mathbb{P}NOBJ \bullet$
$domR = \{nsub : Nsub; nobj : Nobj | nsub\underline{R}$
$nobj \,\mathrm{dim}\, \bullet nsub\} \wedge$
$ranR = \{nsub : Nsub; nobj : Nobj | nsub\underline{R}$
$nobj \bullet nobj\}$

2) Network Subject Security
   A network subject is associated to a set of access class through the relation $NSub \longrightarrow \mathbb{P}ACLS$

   The relation NSubSec can be specified as follows:

[**NSubSec**]
$dom : NSubSec \rightarrow \mathbb{P}Nsub$
$ran : NSubRef \rightarrow \mathbb{P}Acls$

$\forall NSubSec : NSUB \rightarrow \mathbb{P}ACLS \bullet$
$domR = \{nsub : Nsub; ac : Acls | nsub\underline{R}ac \bullet$
$nobj \dim \bullet nsub\} \wedge$
$ranR = \{nsub : Nsub; ac : Acls | nsub\underline{R}ac \bullet$
$ac\}$

3) Authorized Role
A User is associated to a set of roles through the relation *AuthRole*: $USER \longrightarrow \mathbb{P}RSET$.

The relation AuthRole can be specified as follows.

[**AuthRole**]
$dom : AuthRole \rightarrow \mathbb{P}Nsub$
$ran : AuthRole \rightarrow \mathbb{P}Rset$

$\forall AuthRole : NSUB \rightarrow \mathbb{P}Rset \bullet$
$domR = \{nsub : Nsub; role : Rest | nsub\underline{R}role$
$nobj \dim \bullet nsub\} \wedge$
$ranR = \{nsub : Nsub; role : Rest | nsub\underline{R}role$
$\bullet role\}$

4) Network Connection Authorization
A Network Subject on behalf of Authorized User is associated to a list of authorized network entities through the function *NConnAuth*: $NSUB \longrightarrow NCONNAUTHLIST$. The relation NConnAuth can be specified as follows:

[**NConnAuth**]
$dom : NConnAuth \rightarrow \mathbb{P}Nsub$
$ran : NConnAuth \rightarrow \mathbb{P}NConnAuthList$

$\forall NConnAuth : NSUB \rightarrow \mathbb{P}NConnAuthList \bullet$
$domR = \{nsub : Nsub; authlist :$
$NConnAuthList | nsub\underline{R}authlist \bullet nsub\} \wedge$
$ranR = \{nsub : Nsub; authlist :$
$NConnAuthList | nsub\underline{R}authlist \bullet authlist\}$

5) Network Information Access
A Network Subject on behalf of Authorized User is associated to a list of authorized information entities through the function $InfoAcc$: $NSUB \longrightarrow NACCLIST$. The relation NInfoAcc can be specified as follows:

[**NInfoAcc**]
$dom : NInfoAcc \rightarrow \mathbb{P}Nsub$
$ran : NInfoAcc \rightarrow \mathbb{P}Nacclist$

$\forall NInfoAcc : NSUB \rightarrow \mathbb{P}NAccList \bullet$
$domR = \{nsub : Nsub; acclist : NAccList | nsub$
$\underline{R}NAccList \bullet nsub\} \wedge$
$ranR = \{nsub : Nsub; acclist : NaccList | nsub$
$\underline{R}acclist \bullet authlist\}$

4.2 Formal State Model

4.2.1 Abstract State

The formal model we describe here is state machine based model .We shall refer to this model as a 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 NACPM is specified in the following schema.

S: $\mathbb{P}STATE$ $\quad Nop$: $\mathbb{P}NOP$
$Systran$: $\mathbb{P}NOP \times S \twoheadrightarrow S$

$domSystran \subseteq \mathbb{P}NOP \times S$
$ranSystran \subseteq S$

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*. The following schema captures the abstract state of the NACPM model.

[**NACPM**]
$Nsubs : \mathbb{P}NSUB$
$Nobjs : \mathbb{P}NOBJ$
$Authmode : \mathbb{P}AUTHMODE$
$Accmode : \mathbb{P}ACCMODE$
$NConnAuthLists : \mathbb{P}(NOBJ \times \mathbb{P}AUTHMODE)$
$NConnAuth : NSUB \rightarrow NCONNAUTHLIST$
$NAccLists : \mathbb{P}(IEOBJ \times \mathbb{P}ACCMODE)$
$NAcc : NSUB \rightarrow NACCLIST$

$\cup NConnAuthList \subseteq Nobjs \times \mathbb{P}Authmodes$
$domNConnAuth = Nsubs$
$ranNConnAuth \subseteq NConnAuthlists$
$\cup NAccLists \subseteq Ieobjs \times \mathbb{P}Accmodes$
$domNAcc = Nsubs$
$ranNAcc \subseteq NAcclists$

4.2.2 Initial State

In this step, the initialization of the NACPM model is illustrated. Initial State is defined in terms of the abstract state and some extra predicates defining the initial conditions of the system. For more realistic initial state where $Nsub_0 \neq \phi$, $Nobj_0 \neq \phi$, we assume that the initial system state s_0 is defined in such a way that it satisfies all the conditions of the secure state.

4.3 Policy Model

4.3.1 Network Operations-State Based View

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.

[**LoginProp**]
$SysState\_NACPM$
$x : User$

$\forall x) : User | (CurRole(x)) \in (AuthRole(x)) \bullet$
$(NSubSecFunc(x)) dominates$
$(NObjSecFunc(Userlogin(x))) \wedge$
$(NSubSecFunc(x)) dominates (CurrSecFunc(x))\}$

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

[**ConnProp**]
$SysState\_NACPM$
$nsub : Nsub$
$neobj : Nobj \; a : Accmode$

$\forall (nsub) : Nsub | nsub \in domNCurConn \bullet$
$(neobj, a) \in NConnAuth(nsub) \wedge$
$\forall (neobj) \notin NOD, (NSubSecFunc(nsub))$
$dominates$
$(NObjSecFunc(neobj)) \wedge \}$
$\forall (neobj) \notin NOD, (NObjSecFunc(neobj))$
$dominates(NSubSecFunc(nsub))$

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

[**InfoAccProp**]
$SysState\_NACPM$
$neobj : Nobj \; ieobj : IEobj$

$\forall (iebj) : IEobj; (neobj) : Nobj | ieobj \in$
$Nexplore(neobj) \bullet$
$(NObjSecFunc(neobj) dominates$
$(NObjSecFunc(ieobj))$

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

[**UserRoleProp**]
$SysState\_NACPM$
$u : User$

$\forall u \in User \bullet (CurRole(u) \in AuthRole(u))$

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

$$SecState_N ACPM \overset{=}{\wedge} LoginProp \wedge ConnProp$$
$$\wedge InfoAccProp \wedge UsrRoleProp.$$

4.3.2 Secure State

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.*

4.3.3 Network Operations - Operation Based View

At the outermost level of the specification, the system is considered to be modelled 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. 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.

We here give example of with some fundamental operations related to network objects like addition, deletion and manipulation of security attributes.

1) Addition of Network Object
The addition operation with security constraints is illustrated in *Add_nobj schema*.

[**Add_nobj**]
$\triangle SecState\_NACPM$
$nsub\mathbb{P} : Nsub$
$neobj\mathbb{P} : Nobj$
$a\mathbb{P} : Authmode$

$neobj\mathbb{P} \notin Nobj \Rightarrow NObj‘ = Nobj \cup \{neobj\mathbb{P}\}$
$NSub‘ = Nsub$
$\forall nsub\mathbb{P} \in NSub‘ \bullet \langle neobj\mathbb{P}, a\mathbb{P}\rangle \notin$
$NConnAuth(nsub)‘?$
$\langle neobj\mathbb{P}, a\mathbb{P}\rangle \notin NCurConn(nsub)‘$
$\forall nsub\mathbb{P} \in NSub‘; \forall obj\mathbb{P} \in Nobj \bullet$
$NConnAuth(nsub)‘ = NConnAuth(nsub)$

2) Access Class Assignment
The access class assignment operation with security constraints is illustrated in *Set_Acls_nobj schema*

[**Set_Acls_nobj**]
$\triangle SecState\_NACPM$
$nsub\mathbb{P} : Nsub$
$neobj\mathbb{P} : Nobj$
$a\mathbb{P} : Authmode$
$su\mathbb{P} : User$
$ac\mathbb{P} : Acls$

$(CurRole(su)) \in (AuthRole(su))$
$\forall nsub\mathbb{P} \in Nsub; neobj\mathbb{P} \in Nobj \bullet \langle neobj\mathbb{P}, a\mathbb{P}\rangle$
$\notin NCurConn(nsub) \wedge$
$\{ac\} \notin (NObjSecFunc(neobj)$
$\Rightarrow (NObjSecFunc(neobj)‘ =$
$(NObjSecFunc(neobj) \cup \{ac\}$

3) Deletion of Network Object
The deletion operation with security constraints is illustrated in Delete_nobj schema. This is an operation, when invoked results in the removal of *neobj*.

We now consider operations related to network subjects authorization like addition of new authorization.

[**Delete_nobj**]
$\triangle SecState\_NACPM$
$nsub\mathbb{P} : Nsub$
$a\mathbb{P} : Authmode$
$au\mathbb{P} : User$

$(CurRole(su)) \in (AuthRole(su))$
$neobj\mathbb{P} \in Nobj$
$\forall nsub\mathbb{P} \in Nsub; \langle neobj\mathbb{P}, a\mathbb{P}\rangle \notin$
$NCurConn(nsub) \Rightarrow Nobj‘ = Nobj\ \{neobj\mathbb{P}\}$
$NSub‘ = Nsub$
$\forall nsub\mathbb{P} \in NSub‘; \forall obj\mathbb{P} \in Nobj‘ \bullet$
$NConnAuth(nsub)‘ = NConnAuth(nsub)$

4) Addition of New Authorization
The operation for adding new authorization with security constraints is illustrated in Set_Auth schema. This is an operation, when invoked results in the creation of new capability for network subject *nsub*.

[**Set_Auth**]
$\triangle SecState\_NACPM$
$nsub\mathbb{P} : Nsub$
$neobj\mathbb{P} : Nobj$
$a\mathbb{P} : Authmode$
$su\mathbb{P} : User$

$(CurRole(su)) \in (AuthRole(su))$
$nsub\mathbb{P} \in Nsub; neobj\mathbb{P} \in Nobj$
$a \subseteq AuthMode \Rightarrow$
$NConnAuth(nsub)‘ = NConnAuth(nsub) \cup$
$\langle neobj\mathbb{P}, a\mathbb{P} \rangle \wedge$
$Nsub‘ = Nsub \wedge$
$Nobj‘ = Nobj$
$\forall x\mathbb{P} \in Nsub‘; \forall y\mathbb{P} \in Nobj‘; (x, y) \neq (nsub, neobj)$
$\Rightarrow NConnAuth(x)‘ = NConnAuth(x)$

5) Deletion of Authorization
The operation for deleting new authorization with security constraints is illustrated in *Delete_Auth* schema.

[**Delete_Auth**]
$\triangle SecState\_NACPM$
$nsub\mathbb{P} : Nsub$
$neobj\mathbb{P} : Nobj$
$a\mathbb{P} : Authmode$
$su\mathbb{P} : User$

$(CurRole(su)) \in (AuthRole(su))$
$nsub\mathbb{P} \in Nsub; neobj\mathbb{P} \in Nobj$
$a \subseteq AuthMode \Rightarrow$
$NCurConn(nsub)' \Rightarrow$
$NConnAuth(nsub)\ \langle neobj\mathbb{P}, a \rangle$
$\wedge NSub‘ = Nsub$
$\wedge Nob‘ = Nobj$
$\forall x \in Nsub‘; \forall y\mathbb{P} \in NSub‘; (x, y) \neq (nsub, neobj)$
$\Rightarrow NConnAuth(x)‘ = NConnAuth(x)$

6) Network Connection Request
The operation for establishing connection with network object with security constraints is illustrated in *Connect_nobj_req* schema

[**Connect_nobj_req**]
$\Xi SecState\_NACPM$
$nsub\mathbb{P} : Nsub$
$neobj\mathbb{P} : Nobj$
$a\mathbb{P} : Authmode$
$su\mathbb{P} : User$
$rep! : Report$

$(CurRole(su)) \in (AuthRole(su)) \wedge$
$nsub\mathbb{P} \in Nsub \wedge \langle neobj\mathbb{P}, \mathbb{P} \rangle$
$\in NConnAuth(nsub)$
$\wedge (CurSecFunc(nsub)) dominates$
$NCurConn(nsub)'$
$\Rightarrow NConnAuth(nsub)\ \langle neobj\mathbb{P}, a \rangle$
$(NObjSecFunc(neobj))$
$\Rightarrow rep! = Nop\_Allowed$
$(CurRole(x)) \notin (AuthRole(x)) \vee$
$nsub\mathbb{P} \notin Nsub \vee \langle neobj\mathbb{P}, a\mathbb{P} \rangle$
$\notin NConnAuth(nsub)$
$\vee (CurSecFunc(nsub)) notdominates$
$(NObjSecFunc(neobj))$
$\Rightarrow rep! = Nop\_Denied$

In the next section we consider the verification of the propose 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. Before we begin with phase level verification let us formally specify the change of system state. The change of network system state can be specified as follows.

$$\triangle SecState_N ACPM \mathrel{\overline{\overline{\wedge}}} SecStat_N ACPM \\ \wedge SecStat_N ACPM‘.$$

Sometimes the state of the system is left unaffected by an operation, particularly if an error is detected or it is a status operation:

$$\Xi SecState_N ACPM \mathrel{\overline{\overline{\wedge}}} [\triangle SecState_N ACPM \\ | \theta SecState\_NACPM‘ = \theta SecState_N ACPM].$$

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 a insecure state. We can now formally state this as follows:

Constraint 1. The system described by network access control model NACPM satisfies the security conditions of User Login Phase if initial state s_0 satisfies the User Login Constraint.

The network operation performed during this phase can be specified as $NOP \overset{=}{\wedge}$ Login. The change of state on the execution of the network operation can be specified as

$$\triangle\ SecState\_NACPM \overset{=}{\wedge} NOP|`SecState\_NACPM \\ \wedge SecState\_NACPM`.$$

Initially there are no logged in users in the network system.

$$InitState\_NACPM \overset{=}{\wedge} [SysState\_NACPM|Users = \phi].$$

We assume here the initial state to be secure state. When Users login into network system with Login Property conditions satisfied, the state of the system will remain in secure state.

$$SecState\_NACPM \overset{=}{\wedge} IniState\_NACPM \wedge LoginProp.$$

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 NACPM and starting at initial state s_0, a system is said to be secure if the initial state s_0 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_j need to satisfy the security condition of the Connection Property. We can now formally state the model constraint during the connection phase as follows.

The network operation performed during this phase can be specified as follows

$$NOP \overset{=}{\wedge} Login \vee Connect/_nobj/_req \vee Connect/_nod/_req.$$

The change of state on the execution of the network operation can be specified as follows

$$\Delta SecState\_NACPM \overset{=}{\wedge} NOP|`SecState\_NACPM \\ \wedge SecState/_NACPM`.$$

Initially there are no connection requests for network resources in the network system. This can be specified as follows.

$$InitState\_NACPM \overset{=}{\wedge} [SysState\_NACPM \\ |NCurrConn = \oslash].$$

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. This can be specified as follows

$$SecState\_NACPM \overset{=}{\wedge} IniState\_NACPM \\ \wedge LoginProp \wedge ConProp.$$

Constraint 2. The system described by network access control policy model NACPM satisfy the security conditions of Network connection Phase if the initial state s_0 satisfies the Connection Establishment Constraint.

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 insecure state during this phase if the execution of network operations is allowed by NACPM 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 insecure state from secure state.

We can now formally state the model constraint during the network operation phase as follows. The network operation performed during this phase can be specified as follows

$$NOP \overset{=}{\wedge} Login \vee Connect/_nobj/_req \vee \\ Connect/_nod/_req \vee Read/_ieobj/_req \vee \\ Append/_ieobj/_req \vee Add/_nobj \vee Set/_Acls/_nobj \vee \\ Add/_nsub \vee Set/_Acls/_nsub \vee Set/_Auth \vee \\ Delete/_Auth \vee Delete/_nsub \vee Delete/_nobj \vee \\ Set/_nusr/_role \vee Delete/_nusr/_role.$$

The change of state on the execution of the network operation can be specified as follows.

$$\triangle SecState\_NACPM \overset{=}{\wedge} NOP|`SecState\_NACPM \\ \wedge SecState\_NACPM`.$$

The state of the system will remain in secure state if following is satisfied. When there is a request for user level operation, user must have an authorised connection and when there is a request for administrative operation, user must have administrative privilege to perform it. This can be formally specified as follows.

For User Level Operation

$$SecState_N ACPM \overset{=}{\wedge} SecState_N ACPM \wedge LoginProp$$
$$\wedge ConnProp \wedge InfoAccprop.$$

For Administrative Operation

$$SecState\_NACPM \overset{=}{\wedge} SecState\_NACPM$$
$$\wedge UserRoleProp.$$

Constraint 3. The system described by network access control model NACPM satisfy the security conditions of Network operation Phase if Connection Establishment Constraint is satisfied before executing network operations and secondly the network operation security conditions are satisfied before their execution. After defining the constraints for three phases, we are now in a position to state the security theorem to show that a system described by NACPM is secure. The security conditions are as follows.

- The initial state is secure and
- Every request for network resource access, information transfer and network administrative task satisfies the constraints stated in the phases discussed above.

6 Conclusion

In this paper, the key components of Network Access Control Policy Model are formalized in order to be sharp, precise and prevent their multiple interpretations. 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 symbolic computational environment to produce an animation of the formal specification to further refine the framework.

References

[1] S. R. Band, D. M. Cappelli, L.F. Fischer, A. P. Moore, E.D. Shaw, and R.F. Trzeciak, *Comparing Insider IT Sabotage and Espionage: A Model-Based Analysis*, Software Engineering Institute Technical Report CMU/SEI-2006-TR-026, Carnegie Mellon University, Dec. 2006.

[2] D. E. Bell, and L. J. LaPadula, *Secure Computer Systems: Unified Exposition and Multics Interpretation*, Technical Report ESD-TR-73-306, The MITRE Corporation, 1976.

[3] D. Bell, "Modeling the multipolicy machine," *Proceedings NewSecurity Paradigms Workshop*, pp. 2-9, 1994.

[4] E. Bertino et al., "Supporting multiple access control policies in database systems," *Proceedings IEEE Symposium on Research in Security and Privacy*, pp. 94-109, 1996.

[5] K. J. Biba, *Integrity Consideration for Secure Computer Systems*, Technical Report MTR-3153, The MITRE Corporation, 1975.

[6] D. M. Cappelli, A. G. Desai, A. P. Moore, T. J. Shimeall, E. A. Weaver, B. J. and Willke, "Management and education of the risk of insider threat (MERIT): Mitigating the risk of sabotage to employers' information, systems, or networks," *Proceedings of the 24th International System Dynamics Conference*, Nijmegen, Netherlands, July 2006.

[7] J. Cheng, G. Yuichi, S. Morimoto, H. A. Daisuke, "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.

[8] *Common Criteria for Information Technology Security Evaluation (CC)*, version 2.3, ISO/IEC 15408, Aug. 2005.

[9] *Common Criteria for Information Technology Security Evaluation (CC)*, version 3.1 Revision 1, Sep. 2006.

[10] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST standard for role-based access control," *ACM Transactions on Information and System Security*, vol. 4, no. 3, pp. 224-274, 2001.

[11] D. Ferraiolo, and V. Atluri, "A meta model for access control: Why is it needed and is it even possible to achieve?," *Proceedings of the 13th ACM Symposium on Access Control Models and Technologies*, pp. 153-154, New York, NY, 2008.

[12] *Information Technology - Z Formal Specification Notation - Syntax, Type System and Semantics*, ISO/IEC 13568:2002(E), International Standard.

[13] S. Jajodia et al., "A unified framework for enforcing multiple access control policies," *Proceedings ACM SIGMOD Conference*, pp. 474-485, 1997.

[14] A. A. E. Kalam, et al., "Organization based access control," *IEEE 4th International Workshop on Policies for Distributed Systems and Networks*, pp. 120-134, 2003.

[15] M. M. Keeney, E.F. Kowalski, D.M. Cappelli, A.P. Moore, T.J. Shimeall, and S.N. Rogers, *Insider Threat Study: Computer System Sabotage in Critical Infrastructure Sectors*, Joint SEI and U.S. Secret Service Report, May 2005.

[16] F. Keblawi, and D. Sullivan, "Applying the Common Criteria in Systems Engineering," *IEEE Security and Privacy*, vol. 4, no. 2, pp. 50-55, Mar. 2006.

[17] W. Kuhnhauser, and M. Von Kopp Ostrowski, "A framework to support multiple security policies," *Proceedings 7th Annual Canadian Computer Security Symposium*, pp. 1-19, 1995.

[18] B. Lampson, "Protection," *5th Princeton Symposium on Information Sciences and Systems*, pp. 417-429, 1971.

[19] J. Lee, S. Lee, and B. Choi, "A CC-based security engineering process evaluation model," *Proceedings of the 27th Annual international Conference on Computer Software and Applications COMPSAC*, pp. 130-137, IEEE Computer Society, Nov. 2003.

[20] D. Mellado, E. Fernåndez-Medina, and M. Piattini, "A common criteria based security requirements engineering process for the development of secure information systems," *Computer Standard Interfaces*, vol. 29, no. 2, Feb. 2007.

[21] D. McPherson, *Role-Based Access Control for Multitier Applications Using Authorization Manager*, technical report.

[22] S. Morimoto, S. Shigematsu, Y. Goto, and J. Cheng, "A security specification verification technique based on the international standard ISO/IEC 15408," *Proceedings of the 2006 ACM Symposium on Applied Computing*, pp. 1802-1803, France, Apr. 2006.

[23] R. S. Sandhu, E. Coyne, H. Feinstein, and C. Youman, "Role-based access control models," *IEEE Computer*, vol. 29, no. 2, pp. 38-47, 1996.

[24] R. Sandhu, D. Ferraiolo, and R. Kuhn, "The NIST model for role-based access control: Towards a unified standard," *Proceedings of the 5th ACM Workshop on Role-Based Access Control*, pp. 47-61, 26-28, Berlin, Germany, 2000.

[25] M. Singh, and M. Patterh, "Security functional components for building a secure network computing environment," *International Journal of Information Systems Security*, vol. 16, no. 6, pp. 332-343, Nov. 2007.

[26] G. Stoneburner, "Developer-focused assurance requirements," *Computer*, vol. 38, no. 7, pp. 91-93, 2005.

[27] M. Vetterling, G. Wimmel, and A. Wisspeintner, "Secure systems development based on the common criteria: The PalME project," *SIGSOFT Software Engineering Notes*, vol. 27, no. 6, Nov. 2002.

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.

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