IT 시스템의 다중 수준 보안을 위한 관리 환경 연구*

김 점 구**

요 약

복잡한 환경에서 안전한 IT환경은 하나이상으로 분리된 데이터의 그룹으로 나뉘어 같은 시스템에 상주하지 않게 함으로서 그 목적을 얻을 수 있다. 이러한 시스템의 사용자는 특정 데이터에 접근하는 방법과 등급을 다르게 하여야 한다. 정보구조, 물리적 구조, 사용자 권한 및 응용 프로그램 보안 정책을 위한 유지 보수 등은 이러한 환경 관리를 더욱 복잡하게 하고 보안 관리자의 수의 증가를 가져온다.

본 논문은 이러한 환경관리를 위한 CM 툴을 시스템 공학의 CASE 툴을 모델로 하여 제안하고자 한다. 시스템 공학의 모델링 기법은 다중 정보 보안을 처리하는 데 사용할 수 있다. SE의 CASE 툴 모델은 동일 시스템에 대한 논리와 물리적인 관리를 쉽게 할 수 있는 중요한 구성 요소를 가지게 된다. CASE 툴의 확장된 영역은 물리적인 CM 툴을 사용자 친화적이고 안전한 IT 시스템의 관리 환경의 문제점을 해결하는 기본을 제공하게 될 것이다.

Configuration Management for Multi-Level Security Information Technology Systems

Jeom Goo Kim**

ABSTRACT

In a complex, secure IT system environment there will be groups of data that be segregated from one another, yet reside on the same system. Users of the system will have varying degrees of access to specific data. The Configuration Management(CM) of the information architecture, the physical architecture, user privileges and application security policies increases the complexity for operations, maintenance and security staff. This pager describes(current work to merge the capabilities of a network CM toll with those of a Computer Aided System Engineering(CASE) tool. The rigour of Systems Engineering(SE) modelling techniques can be used to deal with the complexities of multi-level information security. The SE logical and physical models of the same system are readily tailorable to document the critical components of both the information architecture and physical architecture that needs to be managed. Linking a user-friendly, physical CM tool with the extended capabilities of a CASE tool provide the basis for improved configuration management of secure IT systems.

Key words : CASE Tools, Security, System Engineering

접수일 : 2010년 10월 10일; 채택일 : 2010년 12월 9일

* 본 논문은 남서울대학교 2007년도 학술연구비 지원에 의해 연구되었음.

** 남서울대학교 컴퓨터학과 교수

1. Introduction

Security configuration management is an information system involves both a logical and a physical component. the logical component deals with the identification of the information to be secured, the users of the information and the security policies governing the access and care of the information. The physical component deals with the identification of the system hardware and software that will allow access to as well as protect the information.

Configuration management is required because changes to the logical and physical elements of an existing system are inevitable. the purpose of configuration management is to ensure that these changes take place in a manner that does not undermine the integrity of the system

Modern information systems will contain information objects (data) with varying degrees of sensitivity which may be available to a community of users with varying access privileges. The system may be physically diverse with security mechanisms distributed amongst a number of hardware and software components. Some information may be accessible from a restricted number workstations while other information may be accessible anywhere on the system, Information with new security requirements may be added to the system. These security interdependency aspects of current and future systems increase significantly the complexity of the configuration management task.

This pager describes work underway to employ Computer-Aided systems Engineering (CASE) tools combined with physical configuration management tools to document the security-related attributes of an information system. CASE tools have been recognized as an essential element in the design and development of complex engineering programs. Many large programs, especially in the aerospace industry require, their use to track system requirements through the analysis, design, build, test and life-cycle support phases of the program. These tools are directly applicable to tracking the logical security architecture requirements of large, complex information systems, By interfacing a CASE tool to a tool designed specifically to document the physical architecture of an IT system, a more comprehensive and cohesive description of security attributes will be achieved, This will result in a more robust mechanism to provide secure configuration management and can be applied form the system concept phase and carried on through the in-service system support phases.

As a basis for discussion, a brief description will be given of a multi-level security concept. This will be followed by a description of how CASE tools can be used to link the security requirements of the logical information architecture model to the physical security mechanisms documented in the CM tool.

2. Configuration Management Principles

Configuration management is a sound engineering practice that is widely used in the development and life-cycle support of complex systems. It provides assurance that the system in operation is in fact the system supposed to be in use. CM consists of four tasks : identification,

control, status accounting and auditing, For every change that is made to a system, the design and requirements of the changed version should be identified. The control task of configuration management is performed by subjecting every change to hardware, software and documentation to review and approval process. Configuration status accounting is responsible for recording and reporting on configuration changes. Configuration audit is the process of verifying that the operation of the system after any changes is consistent with the specification, Organizations that understand information technology security will always maintain an up-to-date description of their information system and log all changes to the system, no mater how small.

Traditionally, configuration management was applied only to the system because systems supported just one security policy. With the advent of multiple security policies being supported on distributed systems, the configuration management process must be extended to also include the information.

3. Open Systems with Multi-Level Information Security Requirements

The basis of our work is the assumption that information system will be widely distributed and will have to support multiple security policies, The following four statements capture what we believe will define the requirements of many future systems :

1. Information systems must support information processing under multiple security policies of any complexity or types.
2. Information systems must be sufficiently protected to allow distributed information processing (including distributed information system management) among multiple hosts on multiple networks in accordance with open systems architectures.
3. Information processing among users with different security attributes employing resources with varying degrees of security protection.
4. Information systems must be sufficiently protected to allow connectivity via common carrier communications systems.

A description of how security may be implemented to accommodate th above four points is contained in the Department of Defence Technical Architecture Framework for Information Management(TAFIM)[2]. The basic concept is to deal with the date, or information objects and the physical network system as two separate entities.

Information is segregated into *Information Domains*, Each Information Domain is subject to a security police describing how the information is to be protected and who may access the information. Users must be constrained to processing and distributing the information on physical systems that meet the security policy of the Information Domain. This may be done automatically by computers which will let only authorized users access the information on a network, by physically isolating the information, or through doctrinal mechanisms that prohibit users from distributing the information to those not authorized to have it.

The *Information System, which is made up of*

communications, processing and storage elements, must be implemented in such a way that the assets of each Information Domain are protected. The general approach is to allow information objects onto a system only if the system can meet the security requirements of the domain. The objects in each Information domain must be kept strictly isolated from one another unless a deliberate and authorized transfer is invoked, The Information system must be designed and administered to fulfil this requirement.

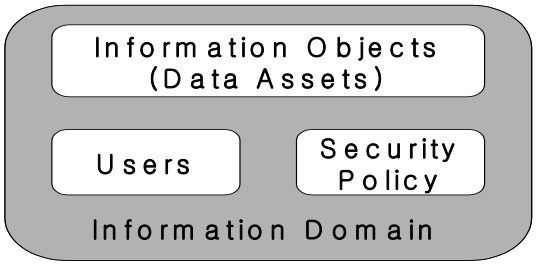
3.1 Information Domains

An information domain is made up of three entities: information objects; users; and a security policy. Users are those who are authorized to access the information objects. The Security Policy of the Information Domain defines: membership requirements; security management requirements; the required security services; and interdomain transfer requirements, (Figure 1) characterizes an information domain.

For each Information Domain, it is necessary to determine the security services required to maintain the confidentiality, integrity and availability of the deposited information. There are seven security services to be applied :

Authentication
confidentiality
Access Control
Audit
Integrity
Availability
Non-Repudiation

The degree, of strength, of each category must be sufficient to meet the needs established for the Information Domain under consideration.



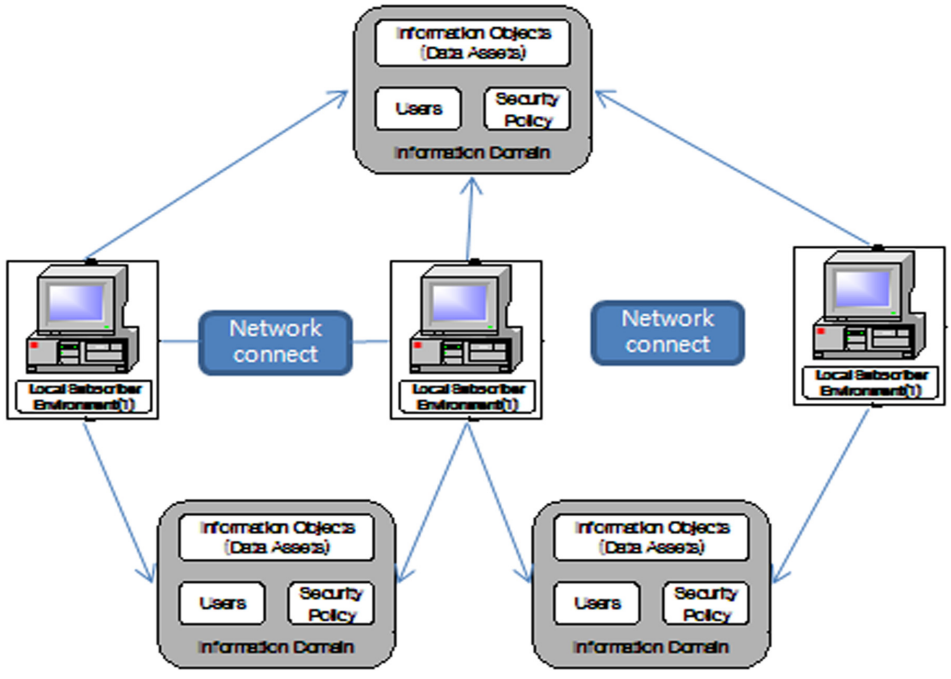
(Figure 1) Information Domain

3.2 Information Systmes

Information Systems include all hardware, software and communications networks. Systems may be confined to one location or distributed over large distances, Most systems today and in the future will use commercial-off-the-shelf hardware and software components. Communications will occur over commercial nets such as the internet or telephone systems.

The Information System can be thought of as having two basic parts : communication networks (CNs); and Local Subscriber Environments (LSEs). The LSEs contain all elements that are under control of the host organization and include workstations, LANs, routers, servers, etc. What constitutes an LSE is discretionary and may be defined, for example, as all assets in a building or perhaps 0ll assets in a department. The CNs connect one LSE to another.

Security services may be embedded in an Information System at different levels. Some of the services may be implemented in software and some in hardware. Physical isolation may also be used. With the trend toward using third party



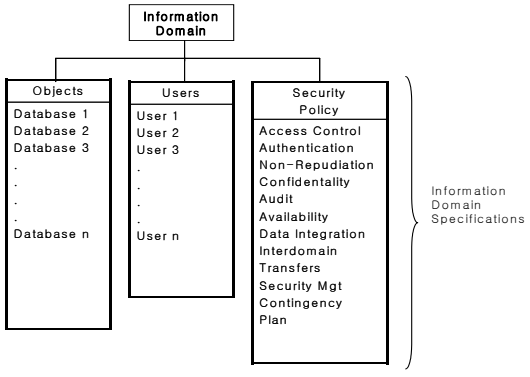
(Figure 2) Relationship of Information Domains and Information Systems

communications networks as an integral part of the system, it is prudent to rely only on the availability of the communications networks. All other security requirements should be taken care of in the Local Subscriber Environments (LSEs) attached to th network.

Relationship of Information Domains to Information Systems Information Domains are distinct from Information systems. Information Domains are no bounded by networks. but rather by the presence of their assigned information assets and specifically authorized users. They may be supported by any information system that can meet the protection and delivery requirements of the Information Domain security policy. The diagram on the next page shows how three Information Domains are supported by three LSEs on single communication network. One Information Domain is accessible by all three LSEs while the other two are each accessible by two LSEs. For example, LSE2 must meet the security requirements of all three Information Domains. LSE1 and LSE3 need only meet the security requirements of the two domains it supports. These concepts are described in much greater detail in refernce 2.

4. Security Configuration Management of Information Architectures

Managing configuration, in the traditional sense, involves identifying and keeping track of the hardware and software configuration items



(Figure 3) The Information Domain as a Configuration Item

(and documentation) that make up the Information System, In the information architecture described above, each Information Domain can be considered to be a configuration item as well. The properties (or specifications) of the Information Domain can be placed under configuration management, as well as the "interfaces" between information domains (interdomain transfers) and LSEs. For information architectures with a large number of configuration

<Table 1> summarizes some of the pertinent aspects of configuration management as they relate to multi-level security information architectures.

4.1 Using a CASE Tool for Security Configuration Management

In many large, complex engineering systems, the functional and physical design is beyond the capability of a single person to comprehend. Systems Engineering methodologies have been developed to decompose the design into manageable components for development and then reassemble them to produce a working system. To keep track of all system requirements that become allocated to components and the interfaces between components, sophisticated CASE tools have been applied. The features and capabilities of these CASE tools are directly applicable to se-

〈Table 1〉 Secure CM Tasks

CM Task	Comments
Configuration Identification	Identify all security Ci's to be controlled, including : • Information Domains • hardware(LSEs) • Software including firmware • Distribution of Security mechanisms in hardware and software • Physical Interfaces(network specs) • Domain Interfaces(what information is allowed on a LSE)
Change Control	Implement procedures to control changes to all controlled CIs. Control how hardware and software are updated, and by whom. for Information Domains, Control membership and information objects.
Status Accounting	Maintain a record of all changes to the information architecture.
Auditing	Audit all actions and maintain logs as specified by the security policies of the Information Domains.

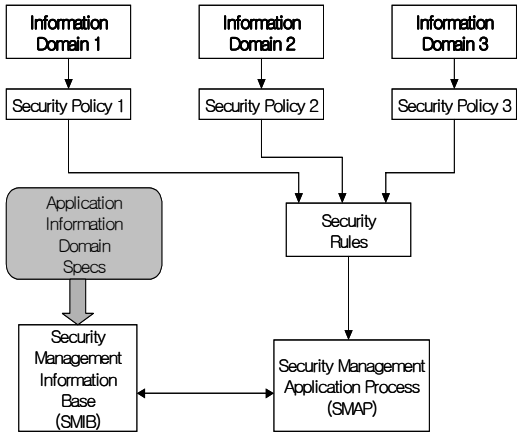
curity configuration management.

(Figure 3) shows the components of an Information Domain. The listings of information objects, users and the security policy form the "specification" of the domain. The specifications can be documented and retained in a CASE tool in the same manner as a hardware or software configuration item. The CASE tool provides a convenient mechanism to administer change control, a key requirement of configuration management.

Once the specifications for all Information Domains are entered into the CASE tool, a comprehensive set of security rules can be derived. The CASE tool will automatically check for consistency and maintain a link between each security rule and security policy that requires it. A set of security mechanisms and the strength of the mechanisms can be derived from the security rules and documented.

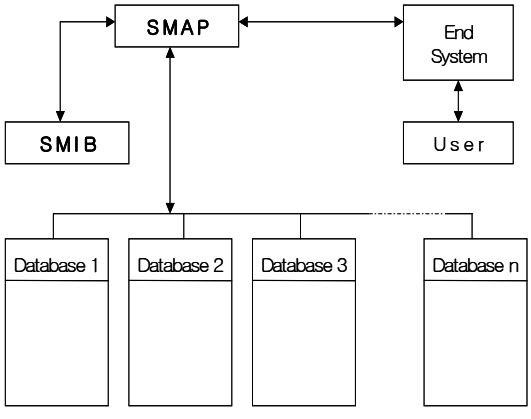
The Security Management Application Process (SMAP) and the Security Management Information Base (SMIB) that will be used in the information architecture can also be specified in the CASE tool. The SMIB will contain items such as user rights and privileges, authentication criteria and security label information to be attached to displayed or printed information, etc. Again the CASE tool will check for consistency and completeness. (Figure 4) shows this concept.

The CASE tool can also describe the process of the information architecture. (Figure 5) show a simple, logical process where the SMAP, based on entries in the SMIB determines whether a particular user, can access information in one or more databases, form a specific end system (workstation). In practice, the functionality of the SMAP will most likely be distributed in various



(Figure 4) Relationship of the SMAP and SMIB to the Information Domains

parts of the system. For example, the process if access control and authentication may be assigned to the end station as well as to one or more servers. The CASE tool can store all of the attributes of the various functions as well as the properties of the links.



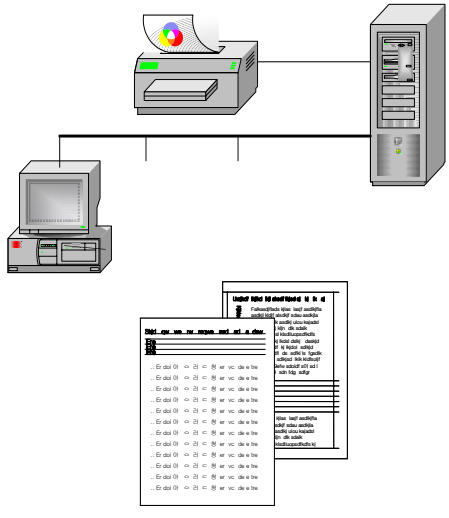
(Figure 5) Logical System Representation Stored in a CASE Tool

4.2 Network Diagramming Tools for Configuration Management

An accurate record of the configuration of the hardware and software in a network is essential to maintain the integrity of the network. Network diagramming tools are available that allow an accurate graphical representation of network components. Some have the added ability to overlay layers containing features such as floor plans or geographical maps to identify the location of each configuration item. This type of graphical interface is very useful for administrators to quickly recognize the physical assets in the system. The configuration management information is maintained in a database structure within the tool. Many tools have the capability to search for specific fields and to manipulate and display data in different formats. Import and export of data is also possible.

(Figure 6) shows a simple diagram which demonstrates some of the features of a tool named netViz.

In this particular tool, a diagram consists of nodes and links. Things such as workstations, servers, printers, microwave relay stations etc. are nodes. The lines that interconnect the nodes are links. The configurable database associated with it for information storage. For complex systems, any node can be expanded into a subsidiary diagram showing more details. For a system spread across the country, the top level nodes might be cities on the network, displayed on a map overlay. Each city node would contain a subsidiary diagram with buildings or LSEs. Each LSE would contain a subsidiary diagram of the network inside, including end systems, routers, servers, LANs and so forth.



(Figure 6) Network Diagram Showing Underlying Database Capability

The underlying database can be configured to suit the user. This capability is very useful for keeping track of security mechanisms 조ich may be distributed throughout the network. The security mechanisms performed by specific nodes in the system can be documented in the database associated with each node along with pertinent hardware and software information. The database attached to a node or link moves with it's parent if the parent is moved.

The capability to search for and group fields provides the ability to easily identify where specific security mechanisms reside in a network. This type of capability can be useful in both managing and analysing the security status of a system.

〈Table 2〉 Specifying the Presence of Information Domains on LSEs

Information Domain	Domain Name	LSE-a	LSE-b	LSE-c	LSE-d
1	Procedures	x		x	x
2	Finance	x			
3	Payroll	x			
4	Project A		x	x	
5	Project B			x	
6	Project C				x
7	Security Mgt	x	x	x	x

5. Combining a CASE Tool with a Network Diagramming Tool

In a multi-level security information architecture, it is necessary to ensure that the information Domain can be protected in accordance with its security policy by the hardware and software in the system, <Table 2> illustrates an organization where the information in a number of discrete domains must be accessible from (and only from)selected LSEs in the organization.

Over time, the configurations of the LSEs will change as new hardware and software is changed or updated. New Information Domains may be created as well. The process of verifying that the security mechanisms in each LSE are adequate and remain adequate can be greatly simplified if the capabilities of a CASE Tool can be integrated with those of a network diagramming tool.

As previously mentioned, the CASE tool can contain all the security requirements of the Information Domains and can relate these to security rules and mechanisms in the information architecture. Since both the CASE tool and the Network diagramming tool use databases to store information, it is possible to import or export information between the two tools if the database structures are kept compatible. This allows the potential of automatically verifying that an LSE has the required security mechanisms in place to satisfy the security policy of an Information Domain, even when the security mechanisms are distributed in various hardware and software CI's of the LSE.

Another function that shows great promise is a "auto-discovery." Currently, some network diagramming tools will search a network and report the existence and location of hardware and software assets. There is the potential to extend this type of function to search for security requirements in the logical architecture contained in a CASE tool and verify that they are compatible with the mechanisms built into the physical architecture.

6. Conclusions

Configuration management for multi-level secure information systems involves identification, control, status accounting and auditing for not only the hardware and software configuration items but also for the Information Domains supported by the system. Assurance that these functions adequately cover all required aspects in complex systems is greatly facilitated by using automated tools.

Our work has shown that it is possible to integrate a CASE tool and a network diagramming tool. The CASE tool is useful for capturing the

logical system and information architecture. The network diagramming tool is useful for capturing and viewing the physical architecture. Integration of the two will improve the reliability of security configuration management. Presently, this work is not sufficiently advanced to demonstrate that the concept will provide rigorous security assurance, however it dose show the utility of the approach to greatly reduce the possibility of error or omission.

References

[1] A Guide to Understanding Configuration in Trusted Systems, National Computer Security Center, NCSC-TG-006 VERSION-1, Vol. 28, March, 1988.

[2] Technical Architecture Framework for Information Management, Volume 6 : DoD Goal Security Architecture, Version 2.0, 30, June, 1994.

[3] The CASE tool used in this work was: System Architect, Popkin Software and Systems, Inc. 11, Park Place, New York, NY 10007-2801.

[4] netViz, Quyen Systems, nc. 1300 Piccard Drive, Suite 108, Rockville, MD 20850.

김 점 구

광운대학교 전자계산학과 (이학사)

광운대학교 전자계산학과 (이학석사)

한남대학교 컴퓨터공학과 (공학박사)

(주) 제성프로젝트 연구원

(주) 시사컴퓨토피아 인터넷사업본부장

현재 남서울대학교 컴퓨터학과 교수