이종의 침입 차단시스템 관리를 위한 중앙 정책 데이터베이스 개발

이 동 영[†] · 정 태 명[††]

요 약

정보통신과 컴퓨터기술의 발전으로 인하여 불법침입으로 인한 정보 파괴, 서비스거부공격 그리고 컴퓨터 바이러스 등에 의한 역기능이 날로 증가하고 있는 추세이다. 또한, 이러한 공격들로부터 네트워크를 보호하기 위해서 침입차단 시스템, 침입탐지 시스템, 접근제어 시스템 등 많은 보안제품들이 개발 및 적용되고 있다. 그러나 이러한 보안 제품들에 대한 관리를 위해서는 많은 작업과 비용이 소요된다. 따라서, 이들 보안제품들에 대한 효율적인 관리와 일관된 보안정책을 적용할 수 있는 정책 기반의 통합보안관리 시스템의 정책모델이 필요하게 되었다. 본 논문에서는 중앙 정책 데이터베이스를 기반으로 대표적인 보안 시스템인 침입차단 시스템을 통합 관리하는 시스템의 구조와 세부 기능들에 대해서 기술하였다. 그리고, 중앙 정책 데이터베이스를 통해 네트워크 상의 각 방화벽 정책을 조정하고 유지하는 네트워크 방화벽 통합 관리 시스템의 핵심 부분인 WISMSF엔진의 구현 기술과 정책 충돌을 정의하고 정책 복구 과정을 제시하였다.

A Development of Central Policy Database for managing Heterogeneous Firewall Systems

Dong-Young Lee[†] · Tai Myoung Chung[††]

ABSTRACT

With a remarkable growth and expansion of Internet, the security issues emerged from intrusions and attacks such as computer viruses, denial of services and hackings to destroy information have been considered as serious threats for Internet and the private networks. To protect networks from intrusions and attacks, many vendors have developed various security systems such as firewalls and intrusion detection systems. However, managing these systems individually demands too much work and high cost. Thus, integrated and autonomous security management for various security products has become more important. In this paper, we present the architecture of the WISMSF (Web-based Integrated Security Management System for Firewalls) and the merits of centralized approach for managing heterogeneous firewalls and implement the prototype of the central policy database that is a component of the WISMSF engine. The WISMSF engine supports an integrated view for policies, the integrity of policies and the easy recovery and addition of policies. And also, we define the policy conflicts of WISMSF and present the policy recovery process to support to the policies consistence.

키워드 : 통합보안관리(Integrated Security Management), 중앙정책 데이터베이스(Central Policy Database), 정책충돌(Policy Conflict)

1. 서 론

정보통신과 컴퓨터 기술의 발전으로 인하여 불법침입으로 인한 정보 파괴와 컴퓨터 바이러스 등에 의한 역기능이 날로 증가하고 인터넷과 같이 범세계적인 네트워크로 연결되어 있는 정보시스템에 대한 위협 역시 급속히 증가하고 있는 추세이다. 컴퓨터 네트워크에 의한 정보화 사회의 역기능으로 악의적인 사용자나 크래커에 의해 각종 주요 정보의 유출 및 파괴, 도용 등 전산 자원과 관련된 범죄사건이 속출하고 있다. 이를 방지하고자 컴퓨터 및 네트워크 보안 기술에 대한 연구가 진행되어 왔으며, 그 결과로 침입탐지 시스템, 네트워크 방화벽 시스템, 접근제어 시스템, 암호화 기술 등이 개발되어 실제 적용되어왔다. 이중, 네트워크 방화벽은 네트워크 단위에 대해 전체적인 정책 관리가 가능하며, 필요한 경우 개개의 호스트 단위에 대해 접근 정책을 설정할 수 있으며, 네트워크의 전단에 위치한 특성으로 부가적인 기능을 수행할 수 있는 장점을 가지고 있기 때문에 네트워크 보안을 위해 각광받고 있다[1-3].

관리 대상 네트워크 규모가 크거나 방화벽이 설치 및 운영되어야 할 네트워크의 경계가 많을 경우, 네트워크 상에 다수의 방화벽이 설치되어야 하며, 다수의 방화벽에 대해 서로 다른 정책을 설정하여 각 네트워크에 대한 보안정책을 달리할

† 정 회 원 : 성균관대학교 정보통신공학부 교수
†† 종신회원 : 성균관대학교 정보통신공학부 교수
논문접수 : 2002년 9월 30일, 심사완료 : 2002년 12월 9일

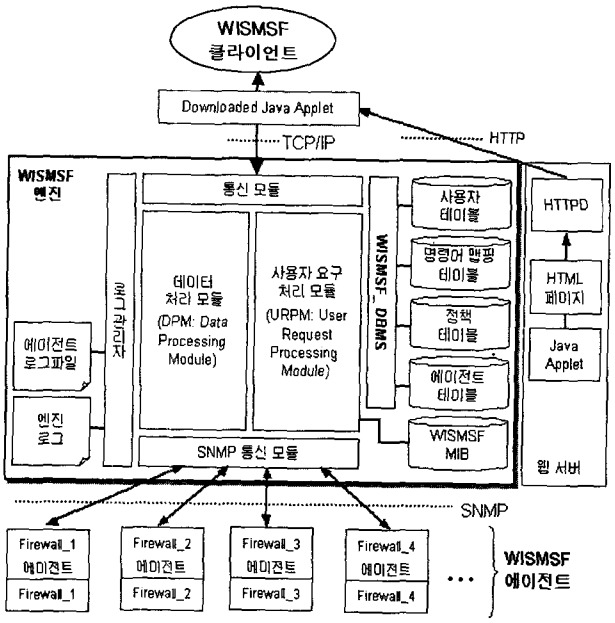
필요가 있다. 그러나, 다수의 방화벽이 존재하는 상태에서 각 방화벽의 정책 일관성을 유지하기란 쉬운 일이 아니며, 정책 일관성이 결여된 경우 네트워크 보안에 오히려 악영향을 초래할 수 있다. 그리고, 보안 관리자가 각 방화벽에 설정되어 있는 다수의 정책들과 서로 영향을 주는 정책들의 관계를 파악하고 관리하기 위해서는 많은 시간과 비용이 든다. 이러한 관리 문제를 해결하기 위해서 근래 상품화된 방화벽은 거의 모두가 원격 관리 인터페이스를 지원하여 원격의 한 지점에서 다수의 방화벽을 관리할 수 있도록 하고 있거나, 같은 방화벽 제품군을 하나의 통합된 관리 인터페이스를 통해 중앙 집중적으로 관리할 수 있는 기능을 포함하고 있다. 그러나, 네트워크 방화벽이 보호해야할 네트워크와 네트워크 경계의 특성에 따라 동작 방식이 다른 방화벽을 설치해야 할 경우도 있으며, 비용 절감과 성능을 위해 특정 플랫폼에 존재하는 패킷 필터링을 이용한 방화벽을 사용할 수도 있다[2, 3]. 이와 같이, 다수의 이질적인 방화벽을 사용하는 경우, 보안 관리자로부터 각각의 방화벽에 설정되어있는 다수의 정책들에 의해서 정책들간의 에러, 누락 또는 관리자들로부터의 요구사항에 대한 겹침과 논리적인 충돌에 의해서 정책 충돌이 발생한다[4-7]. 이는 정책의 주체, 행위 그리고 대상의 관계에서 상호 겹침으로 인하여 발생하며, 이는 통합보안관리 시스템(ISMS : Integrated Security Management System)을 개발하는데 있어서 해결해야 할 중요한 문제이다. 이에 본 논문에서는 이기종의 방화벽 시스템을 통합 관리하는 웹 기반의 방화벽 통합 관리 시스템(WISMSF : Web_based Integrated Security Management System for Firewalls)의 전체 구조와 보안정책의 중앙 관리를 지원하는 보안관리 엔진(security management engine)의 구현 기술과 중앙 정책 데이터베이스에 대해서 언급하고자 한다.

본 논문의 구성을 살펴보면, 2장에서는 웹 기반의 방화벽 통합보안관리 시스템(WISMSF : Web_based Integrated Security Management System for Firewalls)의 구조와 주요 기능에 대해서 살펴보고, 3장에서는 WISMSF의 중앙 정책 데이터베이스의 테이블 구조와 각 테이블 간의 상관 관계 및 정책 처리 과정을 상세히 기술한다. 마지막으로 4장에서는 결론 및 향후계획에 대해서 언급하고자 한다.

2. WISMSF의 개요

2.1 WISMSF의 구조와 동작 메커니즘

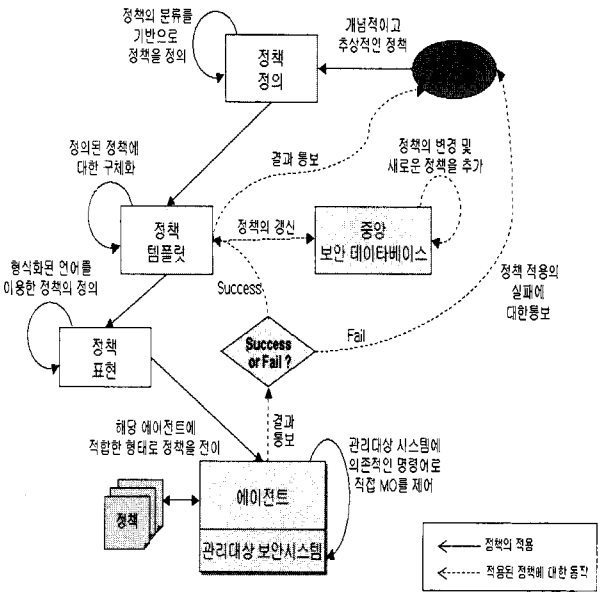
웹 기반의 방화벽 통합 보안관리 시스템(WISMS : Web-based Integrated Security Management System)은 분산 환경에서 이 기종의 보안 시스템들에 대해서 중앙 관리를 목적으로 한다. WISMS의 구성을 살펴보면, 크게 통합 보안관리 시스템에 대한 웹 인터페이스를 제공하는 클라이언트와 클라이언트를 관리하고 보안 관리자에 대한 통신 채널 및 접근 제어 기능을 수행하는 엔진 그리고, 해당 보안 시스템들을 직접 제어하는 보안 에이전트로 구성된다[8-10]. 따라서, 보안정책에 대해서 전문적인 지식이 부족한 사용자의 추상적인 정책 설정 요구에 대해서 보안 관리엔진에서 이를 수행하며, 추가로 보안 시스템에 대해서 통합 관리를 하고자 할 경우에는 기존의 보안 시스템들의 재구현이나 수정이 필요없이 그에 해당하는 에이전트[10]를 추가함으로써 이들을 수용할 수 있는 확장성을 갖고 있다. (그림 1)은 대표적인 보안 시스템인 침입차단 시스템을 통합 관리하는 WISMSF(Web_based Integrated Security Management System for Firewalls)의 구조를 나타낸 것이다.



(그림 1) WISMSF의 구조

WISMSF의 동작을 살펴보면, WISMSF의 보안관리자는 자바 언어를 이용하여 구현된 WISMSF 클라이언트를 통해 개념적이며 추상적인 보안정책 설정을 요구를 WISMSF엔진에게 보낸다. 그리고, WISMSF 엔진은 요구되는 보안정책을 수용할 수 있는 보안 시스템을 제어하는 WISMSF 에이전트를 선정하여 SNMP 메시지를 이용하여 보안 관리자의 정책을 전달한다. 마지막으로, WISMSF 에이전트는 정책 적용 성공 여부를 WISMSF 엔진에게 통보하고 이를 보안 관리자에게 그 결과를 제공한다. WISMSF의 정책 설정 및 적용은 계층적인 정책 모델을 적용하고 있으며, (그림 2)는 계층적 구조를 갖는 WISMSF의 동작 메커니즘을 나타낸 것이다.

또한, 침입차단 시스템들을 통합 관리하는 WISMSF 엔진은 보안 관리자 보낸 요구 메시지를 어떤 WISMSF 에이전트가 처리할 수 있는가를 결정하여 해당 WISMSF 에이전트에게 사용자 요구 메시지를 SNMP 메시지로 변환하여 보내고, WISMSF 에이전트로 부터 그 결과를 받아 사용자에게 결과를 되돌려 주는 동시에 WISMSF 엔진이 관리하는 정보의 갱신을 수행한다.



(그림 2) WISMSF의 동작 메커니즘

WISMSF 에이전트와 WISMSF 엔진의 통신은 SNMP를 이용하며, WISMSF 에이전트는 방화벽 제품 정보, 방화벽의 정책, 방화벽의 정책 적용 대상이 되는 네트워크에 대한 정보 등을 SNMP MIB[11-13]의 형태로 관리하고 있다. 따라서, WISMSF 엔진은 정책 설정 요구에 대해서는 SNMP Set-Request 메시지를 사용하여 정책과 관련된 MIB의 값을 설정하며, 정보 요구에 대해서는 SNMP GetRequest 메시지를 WISMSF 에이전트에게 전송하여 정보를 가져온다. 그리고, WISMSF 에이전트는 새로운 방화벽 시스템을 위해서나 이질적인 다수의 방화벽을 위해 각각 새로이 구현될 필요 없이 방화벽과 직접적인 제어 메시지를 주고받는 부분의 갱신을 통해 쉽게 재사용 될 수 있도록 구현되어 확장성, 이식성이 뛰어나다는 장점을 갖는다.

본 논문에서 제시하는 WISMSF의 엔진은 웹 기반의 통합 보안관리 시스템(WISMS)의 일부이며 이는 프로토타입으로 구현한 결과이며, 침입차단 시스템방 이외의 타 보안 시스템의 정책 설정 지원을 위해 엔진의 기능 확장이 진행되고 있다.

2.2 WISMSF의 중앙정책 데이터베이스를 이용한 정책관리의 특징

WISMSF의 중앙 정책 데이터베이스를 이용하여 이기종의 방화벽 정책에 대한 중앙 집중적 통합관리의 특징을 살펴보면 다음과 같다.

2.2.1 정책에 대한 통합적인 뷰를 제공

WISMSF의 보안관리자는 통합된 인터페이스를 통해 제공되는 네트워크 전반에 걸친 접근 정책을 통합적으로 파악할 수 있다. 네트워크 상에서 방화벽 정책에 따른 문제점이 발생하였을 경우나, 사용자 요구로 정책의 수정이 필요한 경우에 관리자는 방화벽의 정책을 전체적으로 점검하여 문제가 발생할 수 있는 정책의 유무와 정책 수정에 따른 결과를 예측할 수 있다.

2.2.2 정책에 대한 무결성 보장

전반적인 정책 검사에 의해 일차적으로 관리자의 판단으로 정책 무결성에 대한 검사를 할 수 있으며, WISMSF 엔진에 의해 정책 무결성 검사를 통해 다수의 방화벽에 흩어진 정책에 대한 무결성을 보장할 수 있다.

2.2.3 정책에 대한 복구가 용이

방화벽에 문제가 발생하였을 경우나, 여타 다른 이유로 방화벽의 정책이 손상되었을 경우, 중앙에서 관리되는 정책을 이용하여 특정 방화벽의 정책을 이전의 상태로 복구할 수 있다.

2.2.4 부가적인 정책 제어 기능의 제공

외부적인 요소가 방화벽 정책을 제어함으로써 이 외부 요소가 방화벽 정책에 다른 항목을 추가하여 이 항목에 따라 정책을 제어할 수 있다. 예를 들어, 단순 패킷 필터링 방화벽이 시간대별로 정책 적용하거나 해제하는 기능이 없는 경우, 중앙의 정책 관리자가 설정된 시간에 정책의 적용 및 해제 메시지를 보내어 정책을 시간대별로 적용할 수 있는 기능이 추가된다.

3. WISMSF의 중앙 정책 데이터베이스

3.1 테이블 구조

WISMSF엔진의 DBMS으로는 MySQL[14, 15]을 사용하고, 사용자 정보, 정책 정보, 각 방화벽 관리를 담당하는 에이전트 정보 등을 관리하기 위해 각 정보들을 테이블 형태로 구성하여 데이터베이스에 저장하며, 정책 설정과 직접 관련된 테이블들은 다음과 같다.

3.1.1 사용자 테이블(User Table)

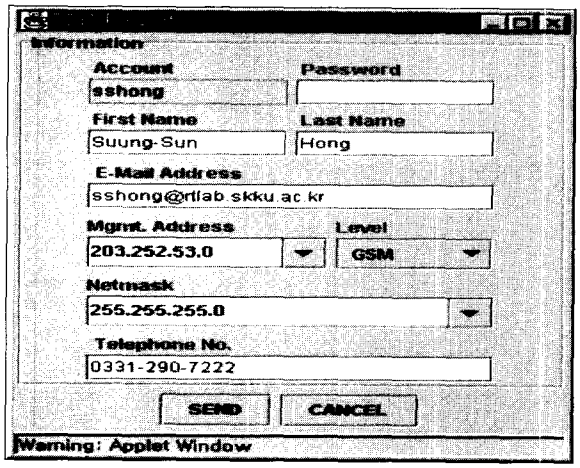
사용자 테이블은 방화벽 정책 중앙관리 시스템의 사용이 허가된 사용자들의 정보를 관리하며, 테이블의 속성에서 'level' 필드는 각 사용자의 관리 등급을 나타낸다. 이 관리 등급에 따라 사용자는 관리 영역이 차별화 된다. 사용자 테이블의 내용은 <표 1>과 같다.

〈표 1〉 사용자 테이블

Field	Data Type	Description
user_id	INT	• 사용자 식별을 위한 식별값
name	VARCHAR(20)	• 사용자의 관리 시스템 Login ID
passwd	CHAR(10)	• 관리 시스템 Login 패스워드
level	ENUM	• 사용자의 관리 등급(NM, GSM, TSM)
network	CHAR(15)	• 사용자가 관리하는 네트워크 주소
description	TINYTEXT	• 사용자에 대한 설명

방화벽 정책 중앙관리 시스템에서 시스템 사용자는 관리 권한에 따라 크게 네트워크 관리자(Network Manager, NM),

일반 보안 관리자(General Security Manager, GSM), 최상층 보안 관리자(Top-level Security Manager, TSM)로 그 등급이 나뉘어 지며, 각각의 관리 등급에 따라 중앙관리 시스템에게 요구할 수 있는 내용들이 달라진다. 각 사용자들이 할 수 있는 내용을 간략히 살펴보면, 네트워크 관리자는 자신이 관리하는 네트워크 내의 통신 소통에 및 서비스 제공에 문제가 있을 경우, 네트워크에 대한 정책이 어떻게 설정되어있는지를 검사하기 위해 정책 열람과 로그, 통계 정보의 열람 권한만이 있으며, 정책 설정 권한은 없다. 일반 보안 관리자는 자신이 담당하고 있는 네트워크에 대한 정책 설정의 책임이 있으며 정책 설정 영역이 자신의 담당 네트워크로 국한된다. 최상층 보안 관리자는 자신이 속한 단체가 사용하고 있는 네트워크 전반에 대한 정책 수립에 대한 책임이 있으며 전체 네트워크 영역에 대한 보안정책을 설정할 수 있다. (그림 3)은 WISMSF의 사용자 관리 화면을 나타낸 것이다.



(그림 3) WISMSF의 사용자 관리 화면

3.1.2 에이전트 테이블(Agent Table)

에이전트 테이블은 각 방화벽의 직접적인 제어를 담당하고 있는 관리 에이전트들에 대한 정보를 관리하기 위한 테이블이며, 그 내용은 <표 2>와 같다.

<표 2> 에이전트 테이블의 내용

Field	Data Type	Description
agent_id	INT	• 에이전트 식별을 위한 식별값
name	VARCHAR(20)	• 에이전트 이름
type	ENUM	• 에이전트가 관리하는 방화벽의 형태 (예, pkt_filter, app_gw, circuit_gw, stateful_inspection)
ext_addr	CHAR(15)	• 에이전트 관리 경계의 외부 네트워크 주소
int_addr	CHAR(15)	• 에이전트 관리 경계의 내부 네트워크 주소
community	VARCHAR(20)	• 에이전트의 SNMP community

이 테이블은 주로 각 관리 에이전트가 담당하는 방화벽의 관리 범위에 있는 네트워크를 식별하기 위해서 사용된다. 즉, 사용자의 정책 관리 요구에 대해 해당 정책을 처리할 수 있는 방화벽을 담당하는 에이전트를 찾기 위한 테이블로 사용된다. 또한, 에이전트 테이블은 각 에이전트와 SNMP를 이용한 통신을 하기 위해 사용되는 SNMP 커뮤니티(community) 정보를 포함한다.

3.1.3 정책 테이블(Policy table)

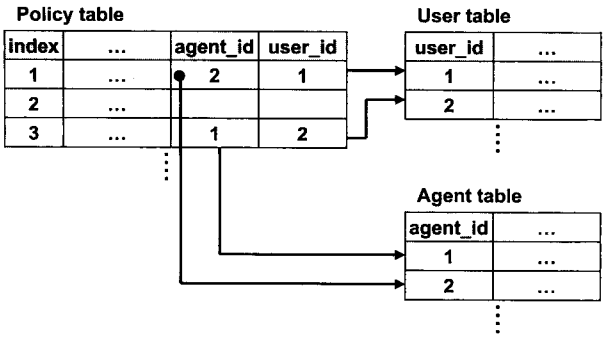
정책 테이블은 각 방화벽을 위한 정책 목록을 관리하기 위한 테이블이며 직접 방화벽에서 정책 설정과 관련된 내용 즉, 정책의 종류, 근원지 주소, 근원지 포트 번호, 목적지 주소, 목적지 포트 번호, 프로토콜 등이 있다. 그리고, 해당 정책을 관리하는 에이전트가 명시되어 해당 정책이 설정되어 있는 방화벽을 관리하는 에이전트 정보를 얻어낼 수 있도록 되어 있으며, 정책을 설정한 사용자의 ID가 명시되어 정책을 설정한 사용자의 정보를 얻을 수 있도록 되어 있어서 사용자 등급에 따라 정책 수정 및 재설정 등의 요구를 처리하는데 필요한 권한 검사를 위한 정보를 제공한다. 정책 테이블의 내용은 <표 3>과 같다.

<표 3> 정책 테이블의 내용

Field	Data Type	Description
index	INT	• 정책 식별을 위한 식별값
policy	ENUM	• 정책(permit, deny)
state	ENUM	• 정책 적용 상태(enable, disable)
src_addr	CHAR(15)	• 근원지 주소
src_port	CHAR(5)	• 근원지 포트 번호(0~65535)
dst_addr	CHAR(15)	• 목적지 주소
dst_port	CHAR(5)	• 목적지 포트 번호(0~65535)
protocol	ENUM	• 프로코콜(IP, TCP, UDP, ICMP)
service	CHAR(20)	• 서비스 명(Telnet, WWW, FTP..)
s_time	DATETIME	• 정책 적용 시작시간
e_time	DATETIME	• 정책 적용 종료시간
day	ENUM	• 정책 적용 요일(mon, tue, wed, thu, fri, sat, sun)
notice	ENUM	• 정책 위반 발생시 처리방법(log, alarm, log_alarm)
c_time	DATETIME	• 정책 생성시간
m_time	DATETIME	• 정책 변경시간
agent_id	INT	• 정책과 관련된 에이전트의 ID
user_id	INT	• 정책을 설정한 사용자 ID
comment	TINYTEXT	• 정책에 관한 설명

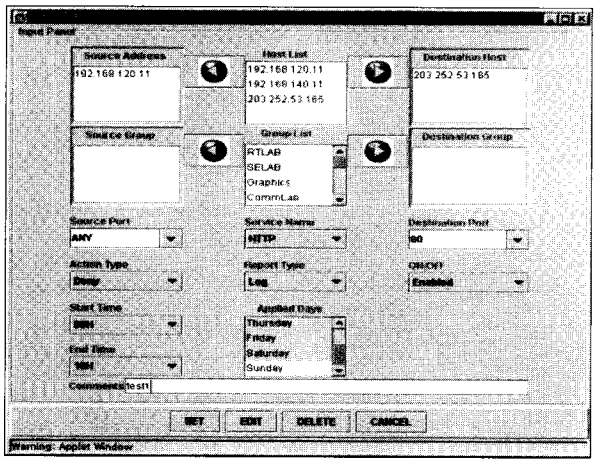
따라서, 정책 관리에 사용되는 이 세 가지 테이블은 상호 연관관계를 갖게되며, (그림 3)은 사용자 테이블, 에이전트 테이블 그리고 정책 테이블 간의 상관관계를 나타낸 것이다.

이와 같이 테이블간의 관계를 이용하여 네트워크에 흩어져 있는 각각의 방화벽에 설정되어 있는 정책을 집중관리 할 수



(그림 4) 각 테이블 간의 관계

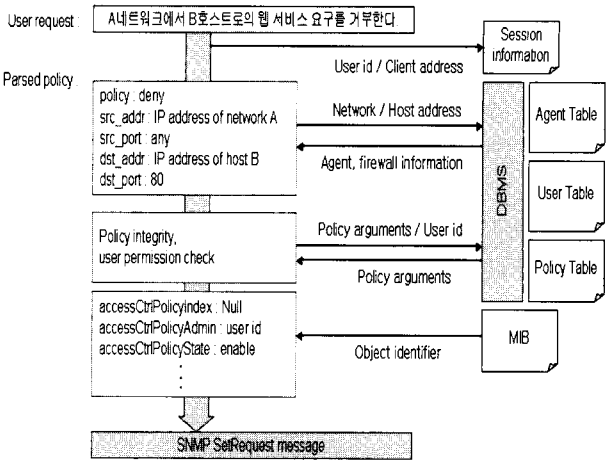
있다. DBMS가 지원하는 각 테이블에 대한 동작들을 이용하여 특정 방화벽에 설정된 모든 정책을 검색할 수 있으며, 특정 사용자가 설정한 모든 정책내용을 볼 수도 있다. 정책의 중앙 관리를 위한 DBMS로 MySQL을 사용한 이유로 MySQL은 한 테이블 당 10000개 이하의 tuple에 대한 동작 수행에 가장 좋은 성능을 보이며 자유롭게 이용 가능한 공개 DBMS 라는 장점 때문이다. 그리고, 관리 엔진을 위한 자료들을 SQL을 지원하는 DBMS를 이용하여 관리하는 이유는 차후 네트워크가 증설되고 그에 따라 관리 대상 방화벽의 수가 증가하여 데이터베이스의 규모 확대가 필요하거나 성능 개선의 목적으로 다른 DBMS로의 교체를 유연하게 수용할 수 있게 하기 위해서이다. (그림 5)는 WISMSF의 정책 설정 화면을 나타낸 것이다.



(그림 5) WISMSF의 정책 설정 화면

3.2 요구된 정책의 처리과정

사용자의 정책 설정 요구를 받은 WISMSF 엔진과 중앙 정책 데이터베이스는 정해진 순서에 따라 정책 설정 요구를 처리하며, (그림 6)은 요구된 정책 "A네트워크에서 B호스트로의 웹 서비스 요구를 거부한다."에 대한 처리과정을 나타낸 것이다. 이때 사용자가 관리 시스템에 로그인하면 관리 클라이언트를 종료할 때까지 엔진은 해당 클라이언트와의 세션 정보를 관리한다.



(그림 6) 요구된 정책의 처리과정

① 사용자의 정책 설정 요구 메시지에서 호스트 혹은 네트워크 주소 추출한다.
② 현재 요구를 보낸 사용자가 해당 호스트나 네트워크에 대한 정책 관리 권한이 있는지를 검사한다. 수신한 요구 메시지에 대한 사용자 정보는 접속한 사용자들에 대한 세션 정보를 관리를 통해서 알아낸다.
② 에이전트 테이블에서 해당 호스트/네트워크 접근 정책에 대한 책임이 있는 에이전트를 검색하여 찾아낸다.
③ 정책 설정의 대상이 되는 방화벽에 동일한 정책이 존재하는지를 검사한다.
④ 사용자의 정책 설정 요구 메시지 내의 정책 설정에 필요한 인자들을 추출한다.
⑤ 추출된 각 인자들에 해당하는 OID와 인자들을 커플링하여 SNMP SetRequest 메시지를 구성한다.
⑥ 에이전트에게 SNMP 메시지를 전송하여 성공적으로 응답을 수신한 경우 해당 정책을 데이터베이스에 저장한다.

3.3 WISMSF의 정책 충돌의 분류 및 정책 복구

WISMSF 엔진은 사용자의 정책 설정 요구에 대해 적절한 보안제품을 선정하고 내부적으로 관리자가 요구하는 정책이 기존의 보안정책과 충돌하지는 않는지, 다른 보안정책에 영향을 주지는 않는지 등의 보안정책 무결성 보장을 위한 동작을 수행한다. 분산 환경에서 통합 보안관리 시스템 개발에 있어서 정책을 정의하고 이에 대한 충돌 문제는 매우 중요한 문제이다[16-18]. 정책 충돌은 정책의 누락, 에러 또는 관리자들로부터의 요구사항에 대한 겹침과 논리적인 충돌에 의해서 발생하며, 본 논문에서는 WISMSF의 정책 충돌을 동일한 정책에 의한 충돌, 상반된 정책의 충돌 그리고, 포함 관계 정책 충돌로 정의하였다.

3.3.1 WISMSF의 정책 구성 인자

WISMSF의 정책 P(x)은 정책을 설정하는 관리자 Mag_Obj(x), 관리자의 보안정책을 직접 실행하는 목표 시스템 Tar_Obj(x), 그리고 정책 실행 대상이 수행하는 정책의 행위 Action(x)로 구성된다. 즉, P(x) = {Mag_Obj(x) × Tar_Obj(x) × Action(x)}라고 정의할 수 있으며, 이를 간략하게 P(x) = {M(x), T(x), A(x)}라고 표현한다. 그리고 기존에 적용된 정책을 P(old)라 하고 새로운 정책은 P(new)라고 한다. P(old) = {Mag_Obj(old) × Tar_Obj(old) × Action(old)}, P(new) = {Mag_Obj(new) × Tar_Obj(old) × Action(new)}로 표현된다. 그리고, 각 각의 정책 P(x)는 발생 시점에 해당하는 시간을 측정한 정책 시간 Policy_Time(x)인 PT(x)를 갖게되며, 이를 통해서 정책의 동시성 문제를 해결하는데 적용한다.

3.3.2 동일한 정책에 의한 충돌

보안관리자가 요구한 정책과 동일한 정책이 이미 존재하고 있을 경우 정책의 충돌이 발생하며, 이를 동일한 정책에 의한 충돌이라고 한다. 그리고, 동일한 정책 대상 T(old) = T(new)에 동일한 정책의 행위 A(old) = A(new) 경우를 말하며, 이는 M(new) = M(old)인 경우와 M(new) ≠ M(old)로 구분 할 수 있다. <표 4>은 동일한 정책 충돌의 분류를 나타낸 것이다.

〈표 4〉 동일한 정책 충돌의 분류

M(x)	T(x)	A(x)
M(new) = M(new)	T(new) = T(old)	A(new) = A(old)
M(new) ≠ M(old)		

3.3.3 상반된 정책에 의한 충돌

기존의 정책과 새로이 추가 요구되는 정책이 서로 상반되는 경우에 발생하는 정책 충돌을 말한다. 즉, (그림 7)과 같이 동일한 관리자 M1 또는 서로 다른 관리자 M1, M2에 의해서 동일한 정책 적용 대상 T1에게 서로 상반되는 정책의 행위 Negative_Action(x)과 Positive_Action(x)을 적용하는 경우를 말한다. <표 5>와 같이 상반된 정책 충돌은 M(new) = M(old)와 M(new) ≠ M(old)인 경우로 분류할 수 있다.

〈표 5〉 상반된 정책 충돌의 분류

M(x)	T(x)	A(x)
M(new) = M(old)	T(new) = T(old)	A(new) = Positive_Action A(old) = Negative_Action
		A(new) = Negative_Action A(old) = Positive_Action
M(new) ≠ M(old)		A(new) = Positive_Action A(old) = Negative_Action
		A(new) = Negative_Action A(old) = Positive_Action

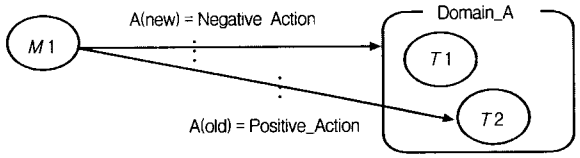
3.3.4 포함관계에 의한 상반된 정책 충돌

기존의 정책에 관여된 객체와 새로이 추가 요구된 정책에 관여된 객체가 서로 포함관계를 갖는 경우, 이를 포함 관계의 정책에 의한 정책 충돌이라고 정의한다. 이 경우 한 정책의 영향 범위가 다른 정책을 포함하여 포함되는 정책의 효력을 상실하게 되므로 포함되는 쪽의 정책은 그 존재 의미가 없어지므로 불필요한 정책으로 남게 된다. 특히, P(old)와 P(new)가 서로 상반된 정책 행위를 가지면서 상호 포함관계를 갖는 정책 충돌은 WISMSF에서는 빈번하게 발생하고 있다. <표 6>은 서로 상반된 행위를 갖는 포함관계 정책 충돌의 분류를 나타낸 것이다.

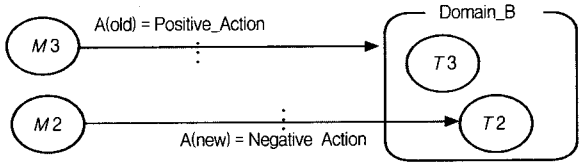
〈표 6〉 포함관계 정책 충돌의 분류

M(x)	T(x)	A(x)
M(new) = M(old)	T(new) ⊂ T(old)	A(new) = Positive_Action A(old) = Negative_Action
		A(new) = Negative_Action A(old) = Positive_Action
	T(new) ⊃ T(old)	A(new) = Positive_Action A(old) = Negative_Action
		A(new) = Negative_Action A(old) = Positive_Action
M(new) ≠ M(old)	T(new) ⊂ T(old)	A(new) = Positive_Action A(old) = Negative_Action
		A(new) = Negative_Action A(old) = Positive_Action
	T(new) ⊃ T(old)	A(new) = Positive_Action A(old) = Negative_Action
		A(new) = Negative_Action A(old) = Positive_Action

<표 6>에 정의된 포함관계 정책의 경우 정책의 발생시간 즉, PT(x)에 따라서, 정책 충돌의 검출 및 해결 과정이 상이하다. (그림 7)은 정책의 발생 시점 PT(x)가 이 서로 상이하고, 포함관계에 의한 상반된 정책 충돌의 예를 나타낸 것이다.



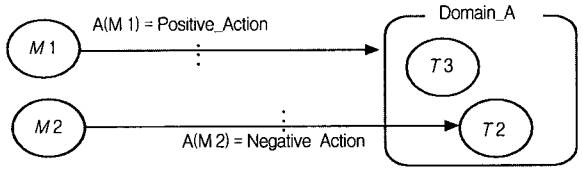
(a) 동등한 관리자에 의한 포함관계 정책 충돌



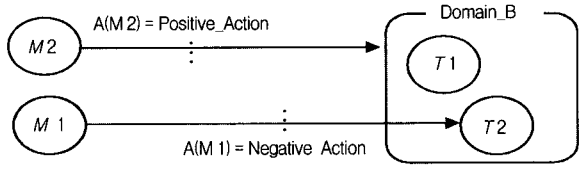
(b) 서로 다른 관리자에 의한 포함관계 정책 충돌

(그림 7) 포함관계 정책 충돌의 예

그리고, (그림 11)과 같이 서로 포함 관계에 있는 정책 적용 목표들에 서로 다른 보안관리자 M1, M2로부터 동시에 정책을 설정하는 즉, PT(M1) = PT(M2)이고, 서로 상반된 정책의 설정을 요구할 경우에도 포함관계의 정책 충돌이 발생한다.



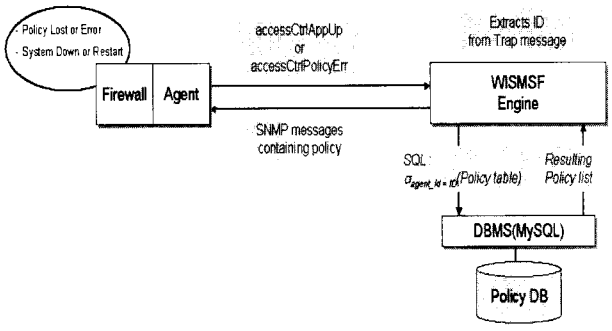
(a) PT(M1) = PT(M2), A(M1) ⇔ A(M2), T(M1)⊃ T(M2)인 경우



(b) PT(M1) = PT(M2), A(M1) ⇔ A(M2), T(M1)⊂ T(M2)인 경우

(그림 8) PT(M 1) = PT(M 2)이고 포함관계에 의한 상반된 정책 충돌

또한, 실제적인 침입차단 시스템 운영 환경에서 갑작스런 정전, 과도한 네트워크 부하, 침입차단 시스템의 자체적인 장애 등으로 인해 침입차단 시스템이 제 기능을 수행하지 못해 재시작 되거나 관리상의 목적으로 침입차단 시스템을 재시작 해야될 필요가 있다. 이 경우, 침입차단 시스템이 동작을 정지한 동안 관리자의 조작에 의해 중앙 관리 시스템의 정책이 변경된 경우 등의 사건에 의해 을 대비하여 중앙 관리 데이터베이스상의 정책과 일관성을 유지해야만 한다. 이를 위해서 각 침입차단 시스템의 정책 관리를 담당하는 에이전트는 침입차단 시스템의 이상 동작 및 시스템 재기동시 이를 엔진에게 SNMP Trap 메시지를 이용하여 알리고 침입차단 시스템의 정책을 재설정한다. 이 절차를 정리하면 다음과 같다. 단, 에이전트는 엔진으로부터 정책을 새로이 받을 때까지 일단 침입차단 시스템의 정책을 기본 정책-모든 서비스 연결 요구 불허-으로 설정한다. (그림 9)는 WISMSF의 정책 복구 과정을 나타낸 것이다.



(그림 9) WISMSF의 정책 복구 과정

① 에이전트가 시스템 재기동이나 침입차단 시스템의 이상을 감지한다.

② 에이전트는 침입차단 시스템의 정책을 모두 삭제하고 기본 정책(모든 연결 불허)으로 설정한다.

③ 에이전트는 SNMP Trap 메시지를 이용하여 관리 엔진에게 사건을 알려 정책 복구를 요구한다.

④ 관리 엔진은 해당 침입차단 시스템에 대해 DB에서 최근에 설정된 정책목록을 추출하고 이를 SNMP Set-Request 메시지를 이용하여 에이전트에게 전송한다.

⑤ 에이전트는 관리 엔진으로부터 전달된 정책을 침입차단 시스템에 설정한다.

⑥ 에이전트는 성공적으로 침입차단 시스템의 정책이 복구된 경우 SNMP GetResponse 메시지를 이용하여 정책 설정이 종료되었음을 통지한다.

⑦ 만약, 복구가 실패한 경우 이를 SNMP trap 메시지를 이용해 엔진에게 알려 정책 재전송을 요구한다.

⑧ 정책 복구가 성공적으로 종료될 때까지 ⑦의 과정을 반복한다.

위와 같은 동작을 거쳐 통합 보안관리 시스템은 문제 발생시에 대비하여 각 침입차단 시스템의 정책을 쉽게 복구할 수 있으며 정책 일관성을 유지할 수 있다. 본 논문에서 프로토타입으로 구현한 WISMSF의 경우 정책 충돌의 해결 방안으로 보안 관리자의 등급과 정책의 우선 순위(PoP : Priority_of_Policy) 그리고 사용자의 역할(RoM : Role_of_Manager) 등과 같은 간단한 방법을 적용하였으며, WISMSF의 정책 충돌 해결 방안에 대한 심도있는 연구는 현재 진행 중이다.

4. 결론 및 향후계획

최근 이종의 분산환경에서는 복잡하고 다양한 방식의 보안관리 및 통신망 관리체계의 집중화, 자동화된 관리체계로의 전환, 그리고 이종간의 보안 시스템들에 대한 효율적이고 통합적인 관리를 위한 정책 관리가 요구되고 있다.

이에 본 논문에서는 네트워크 상에 설치된 침입차단 시스템들에 대한 통합 관리의 장점에 대해 기술하고 실제로 다수의 이질적인 침입차단 시스템의 정책을 중앙에서 관리하기 위해 웹 기반의 클라이언트-엔진-에이전트의 세 요소로 구현된 통합 보안관리 시스템인 WISMSF 기능과 구조에 대해서 설명하였으며, 이 세 요소 중 엔진이 갖는 기능과 동작에 대해 설명하였다.

WISMSF 엔진은 중앙 정책 데이터베이스를 이용하여 다수의 이질적인 침입차단 시스템에 설정된 정책을 관리하며, 이 데이터베이스 내의 정책 테이블에 대하여 관리자의 요구에 따라 정책을 추가, 삭제, 갱신 등의 동작과 함께 수정된 정책 내용을 적절한 에이전트에게 전달하고 에이전트는 수신한 정책을 침입차단 시스템에 반영한다. 이러한 관리 구조를 통하여 다수의 침입차단 시스템을 위한 정책을 중앙 집중적으로 관리하게 된다. 그리고, WISMSF 엔진이 사용하는

데이터베이스는 정책 테이블뿐만 아니라, 관리자의 요구에 따른 정책을 적절한 엔진에게 전달하기 위해서 에이전트 테이블을 가지며, 관리자가 요구한 정책 내용을 실제 정책 설정 인자로 변환하기 위해 정책 인자 변환 테이블을 가진다. 그리고, 침입차단 시스템에 대한 정보, 침입차단 시스템의 정책 및 침입차단 시스템의 관리 범위내에 있는 네트워크 객체들을 관리하기 위해 정의된 SNMP MIB 값을 관리하기 위한 테이블을 갖고 있다. 각 테이블들은 그 성격에 따라 서로 관련을 가지며 관계형 데이터베이스를 구성한다. 이를 통하여 WISMSF는 다수의 침입차단 시스템 정책 관리의 편의를 제공한다. 또한, WISMSF의 정책 충돌을 정의하고 WISMSF의 관리 대상 침입차단 시스템의 에러 발생시 이에 대한 정책의 복구 과정에 대해서 언급하였다.

향후 계획으로는 침입차단 시스템 정책 중앙 관리 시스템의 각 구성 요소간 안전한 메시지 전송을 보장하고 보안 관리자별 접근 통제 기능을 제공하는 SNMP v3의 적용과 타 보안 시스템으로의 확장 및 구현 그리고, 정책 충돌의 해결 방안에 대한 심도 있는 연구와 구현된 시스템에 대해서 실제 환경에서 성능 테스트를 통한 보완 작업이 필요하다.

참 고 문 헌

[1] William R. Cheswick, Steven M. Bellovin, "Firewalls and Internet Security : repelling the willy hacker," Addison Wesley, 1994.

[2] D. Brent Chapman, Elizabeth D. Zwicky, "Building Internet Firewalls," O Reilly & Associations, Inc., January, 1996.

[3] Chris Hare, Karanjit Siyan, "Internet Firewalls and Network Security," - 2nd ed., New Readers, 1996.

[4] J. Moffett, Morris S. Sloman, "Policy Conflict Analysis in Distributed System Management," Journal of Organizational Computing, Vol.4, No.1, pp.1-22, 1994.

[5] Emil C. Lupu, Morris Sloman, "Conflicts in Policy-Based Distributed Systems Management," Journal of IEEE Transaction on Software Engineering, Vol.25. No.6, pp.852-869, 1999.

[6] Cuppens. F, Cholvy. L, Saurel. C, Carrere. J, "Merging security policies : analysis of a practical example," Computer Security Foundations Workshop, Proceedings. 11th IEEE, pp.123-136, 1998.

[7] Cholvy. L, Cuppens. F, "Analyzing consistency of security policies," Security and Privacy, Proceedings, IEEE Symposium on, pp.103-112, 1997.

[8] 이동영, 김동수, 방기홍, 김홍선, 정태명, "SNMP를 이용한 웹 기반의 통합 보안관리 시스템", KNOM(Korea Network and Operations Management) Review 논문지, Vol.2. pp.1167-1171, 1999.

[9] 이동영, 방기홍, 홍승선, 김동수, "이종의 침입차단 시스템 관리를 위한 웹기반의 통합보안관리시스템 개발", 한국정보보호센타 정보보호 우수논문지 공모전, 응용기술 분야 장려, 1999.

[10] D. Y. Lee, D. S. Kim, K. H. Pang, H. S. Kim, T. M. Chung, "A Design of Scalable SNMP Agent for Managing Heterogeneous Security Systems," NOMS(Network Operations and Management Symposium) 2000, pp.293-294, April, 2000.

[11] William Stallings, SNMP, SNMP v2, SNMP v3, and RMON 1 and 2~3rd ed., Addison Wesley, 1999.

[12] David Perkins, Even McGinnis, Understanding SNMP MIBs, Prentice Hall PTR, 1997

[13] Douglas Hyde, "Web-based Management", 3Com Corp., Technical report, 1997.

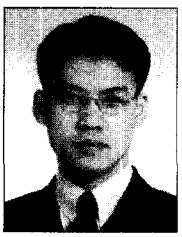
[14] Randy Jay Yarger, George Reese, Tim King, "MySQL and mSQL," O Reilly & Associations, Inc., Janyary, 1999.

[15] http://www.mysql.com/.

[16] Rene Wies, "Using a Classification of Management Policies for Policy Specification and Policy Transformation," Integrated Network Management IV, pp.44-56, 1995.

[17] Rene Wies, "Policy Definition and Classification : Aspects, Criteria, and Examples, Proceeding of IFIP/IEEE International Workshop on Distributed Systems : Operations & Management, Toulouse, France, Oct., 1994.

[18] Miriam J. Maullo and Seraphin B. Calo, "Policy Management : An Architecture and Approach Systems Management," Proceedings of the IEEE First International Workshop on, pp.13-26, 1993.



이 동 영

e-mail : dylee@rtlab.skku.ac.kr
1983년 동아대학교 전자공학(학사)
1998년 성균관대학교 정보공학(석사)
2002년 성균관대학교 컴퓨터공학(박사)
1993년~1997년 기아자동차 중앙기술연구소 연구원
현재 성균관대학교 정보통신공학부 강사
관심분야 : 네트워크 보안, 시스템보안, 네트워크 관리

정 태 명

e-mali : tmchung@ece.skku.ac.kr
1981년 연세대학교 전기공학(학사)
1984년 University of Illinois Chicago, 전자계산학과 학사
1987년 University of Illinois Chicago, 컴퓨터공학과 석사
1995년 Purdue University, 컴퓨터공학 박사
1985년~1987년 Waldner and Co., System Engineer
1987년~1990년 Bolt Bernek and Newman Labs., Staff Scientist
현재 성균관대학교 전기 전자 및 컴퓨터공학부 부교수
관심분야 : 실시간시스템, 네트워크관리, 네트워크 보안, 시스템 보안, 전자상거래