중앙 정책 관리를 통한 방화벽 통합 관리 시스템의 개발

(Implementation of Integrated Firewall Management System by Central Policy Management)

김동수, 정태명
성균관대학교 전기전자 및 컴퓨터공학부
dskim@rtsl.skku.ac.kr, tmchung@ece.skku.ac.kr

요 약

네트워크 상에 다수의 방화벽이 설치되어 있을 경우, 각각의 방화벽에 대한 정책을 설정하고 관리하는 데 있어서 각 방화벽의 정책이 서로 충돌하거나, 하나의 방화벽에 설정된 정책이 다른 방화벽에 영향을 줄 수 있다. 이의 결과로, 네트워크 방화벽의 존재가 무의미해 지거나 네트워크의 정상적인 동작을 방해할 가능성도 존재한다. 따라서, 네트워크 방화벽 정책의 중앙 집중적인 관리가 필요한데, 각 방화벽이 서로 다른 제품일 경우, 각 방화벽의 중앙 집중적인 정책 관리는 더더욱 어려우며, 보안 관리자의 업무를 가중시키고 혼란을 초래하여 문제 발생의 소지를 내포한다. 본 논문에서는 이러한 문제점을 해결하고자 이질적인 방화벽을 중앙 관리하기 위한 시스템의 전체적인 구조를 제시하고 관리자의 요구를 처리하며, 중앙 정책 데이터베이스를 통해 네트워크 상의 각 방화벽 정책을 조정하고 유지하는 네트워크 방화벽 통합 관리 시스템의 핵심 부분인 관리 엔진의 구현 기술에 대해 논한다.

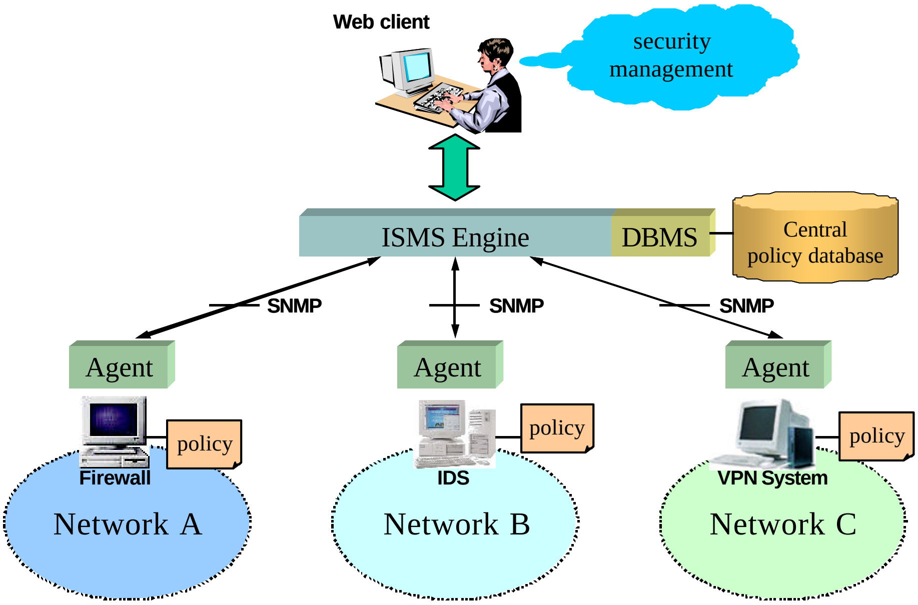
1. 서론

오늘날, 컴퓨터 통신 기술의 발달로 경제, 사회, 정치 분야에서 컴퓨터를 이용한 업무처리 및 정보 관리 등 인간 생활의 많은 부분이 이에 의존하고 있으며, 컴퓨터 통신을 이용한 다양한 정보교류가 사회를 지탱하고 있는 근간이 되고 있다. 그러나, 컴퓨터 네트워크에 의한 정보화 사회의 역기능으로 악의적인 사용자나 크래커에 의해 각종 주요 정보의 유출 및 파괴, 도용 등 전산 자원과 관련된 범죄 사건이 속출하고 있다[7]. 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. 이 중, 네트워크 방화벽(이하 방화벽)은 네트워크 단위에 대해 전체적인 정책 관리가 가능하며, 필요한 경우 개별적인 호스트 단위에 대해 접근 정책을 설정할 수 있으며, 네트워크의 전단에 위치한 특성으로 부가적인 기능을 수행할 수 있는 장점을 가지고 있기 때문에 네트워크 보안을 위해 각광 받고있다[8, 9, 10].

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 보안 정책을 달리할 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 설정되어있는 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 보호 하여야 할 네트워크와 네트워크 경계의 특성에 따라 동작 방식이 다른 방화벽을 설치해야 할 경우도

있으며, 비용 절감과 성능을 위해 특정 플랫폼에 존재하는 패킷 필터링(packet filtering) 기능을 이용한 방화벽을 사용할 수도 있다[9, 10]. 이와 같이, 다수의 이질적인 방화벽을 사용하는 경우 현재로써는 각 방화벽을 따로 관리할 수밖에 없으며, 이로 인해 앞서 언급한 정책 일관성 문제 등의 관리상의 어려움이 그대로 존재한다.

본 논문에서는 이러한 이질적인 방화벽과 보안 제품들을 중앙 집중적으로 관리하기 위해 설계되고 구현된 통합 보안 관리 시스템(integrated security management system, ISMS)의 전체적인 개요를 설명하고 정책 중앙관리를 지원하기 위한 엔진 부분의 구현 기술에 대해 언급한다. 2장에서는 방화벽 정책 중앙 관리 시스템의 전체 개요 및 관리 엔진의 구조와 방화벽 정책을 중앙 집중적으로 관리하여 얻어지는 장점에 대해서 설명하며, 3 장에서는 방화벽 정책 중앙관리 엔진이 사용하는 데이터베이스 내 테이블의 구조와 각 테이블의 상관관계를 기술하고, 4 장에서는 실제 관리자가 요구한 정책에 따라 방화벽 정책 중앙관리 엔진이 수행하는 세부 동작과 유사시 혹은, 각 방화벽의 재운영시에 정책이 복구되는 과정을 설명하며 마지막으로, 5 장에서는 결론 및 향후 계획에 대해서 논한다.



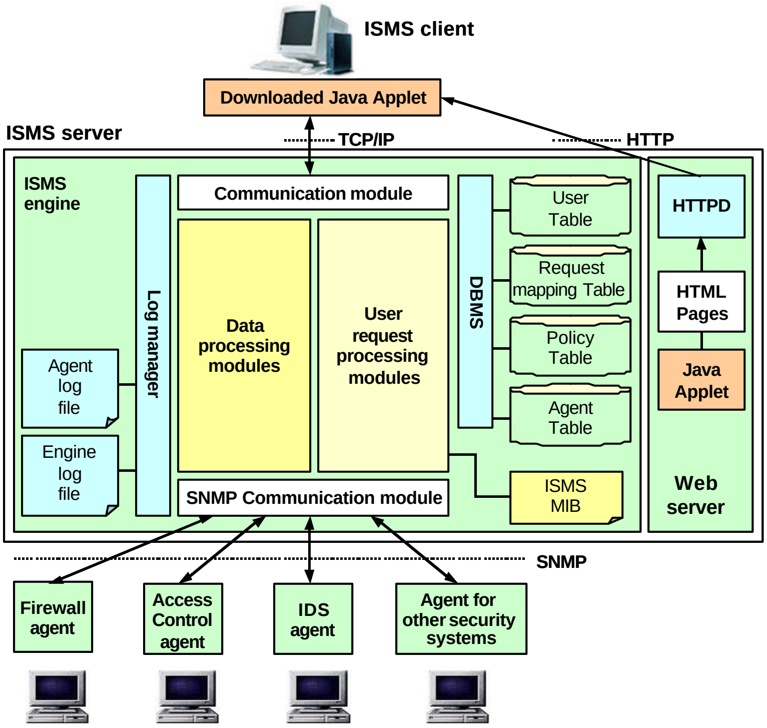
(그림 1) 통합 보안 관리 시스템의 개념

2. 방화벽 정책 중앙관리시스템의 개요

본 논문에서 설명하는 방화벽 정책 중앙관리 엔진은 현재 구현이 진행중인 통합 보안관리 시스템(ISMS)의 일부이며 일차적 구현 결과이다[14, 18]. (그림 1)은 이러한 통합 보안 관리 시스템의 전체적인 개념을 나타낸 것이다.

통합 보안관리 시스템은 네트워크에 산재되어 설치되어 있는 각 보안 제품들을 보안 제품 관리 전담 에이전트, 전체적인 관리 동작을 수행하는 하나의 통합 관리 엔진 그리고, 웹 인터페이스를 제공하는 클라이언트의 세 구성 요소를 통해 관리자가 다수의 보안 제품에 대한 관리 동작을 손쉽게 수행하고 각 제품들이 지원하는 보안 정책 및 접근 제어 정책 사이의 무결성을 보장할 수 있는 시스템이다[14, 15, 16, 17, 18].

통합 보안관리 시스템의 관리 동작을 간단히 설명하면 우선, 관리자는 자바 언어를 이용하여 구현된 웹 인터페이스를 통해 개념적이며 추상적인 보안 정책 설정을 요구를 통합 보안관리 엔진에게 보낸다. 그리고, 통합 보안관리 엔진은 요구되는 보안 정책을 수용할 수 있는 보안제품 관리 에이전트를 선정하여 SNMP 메시지를 이용해 보안제품 관리 에이전트에게 정책을 전달한다. 마지막으로, 에이전트는 정책 적용 성공 여부를 통합 보안관리 엔진에게 보내어 관리자가 그 결과를 볼 수 있도록 한다[14, 18].



(그림 2) ISMS 의 전체 연결과 엔진의 기능 모듈

2.1. 통합 보안관리 엔진

통합 보안관리 엔진은 사용자의 정책 설정 요구에 대해 적절한 보안제품을 선정하고 내부적으로 관리자가 요구하는 정책이 기존의 보안 정책과 충돌하지는 않는지, 다른 보안 정책에 영향을 주지는 않는지 등의 보안 정책 무결성 보장을 위한 동작을 수행하며, 정책 설정 요구를 해당 보안제품을 담당하는 에이전트에 전달하는 가장 핵심적인 역할을 하는 구성 요소이다.

본 논문에서 설명하는 방화벽 정책 통합 관리 엔진은 이러한 통합 보안관리 엔진의 시험적인 구현 결과이며 현재, 방화벽 이외의 타 보안 시스템의 정책 설정 지원을 위해 엔진의 기능 확장이 진행되고 있다. 따라서, 방화벽 정책 통합관리 엔진은 통합 보안관리 엔진으로 발전할 것이다. (그림 2)은 통합 보안관리 시스템의 전체적인 관계와 엔진의 기능을 중심으로 하는 모듈 구성을 간략하게 나타내고 있다.

방화벽 정책 중앙관리 시스템을 이용하여 방화벽 정책을 통합하여 중앙 집중적으로 관리하는 장점으로는 다음과 같은 것이 있다.

■ 정책의 전체적인 파악

관리자는 통합된 인터페이스를 통해 제공되는 네트워크 전반에 걸친 접근 정책을 한 눈에 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.

■ 정책의 무결성 보장

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

■ 정책 복구 용이

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

■ 부가적인 정책 제어 기능 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.

방화벽 정책 통합 관리 엔진(이하 관리 엔진)의 역할은 사용자가 보낸 요구 메시지를 어떤 에이전트가 처리할 수 있는가를 결정하여 해당 에이전트에게 사용자 요구 메시지를 SNMP 메시지로 변환하여 보내고, 에이전트로부터 그 결과를 받아 사용자에게 결과를 되돌려 주는 동시에 관리 엔진이 관리하는 정보의 갱신을 수행한다.

에이전트와 관리 엔진의 통신은 SNMP 를 이용하며, 에이전트는 방화벽 제품 정보, 방화벽의 정책, 방화벽의 정책 적용 대상이 되는 네트워크에 대한 정보 등을 SNMP MIB 의 형태로 관리하고 있다. 따라서, 관리 엔진은 정책 설정 요구에 대해서는 SNMP SetRequest 메시지를 사용하여 정책과 관련된 MIB 의 값을 설정하며, 정보 요구에 대해서는 SNMP GetRequest 메시지를 에이전트에게 전송하여 정보를 가져온다[11, 12, 19, 20, 21].

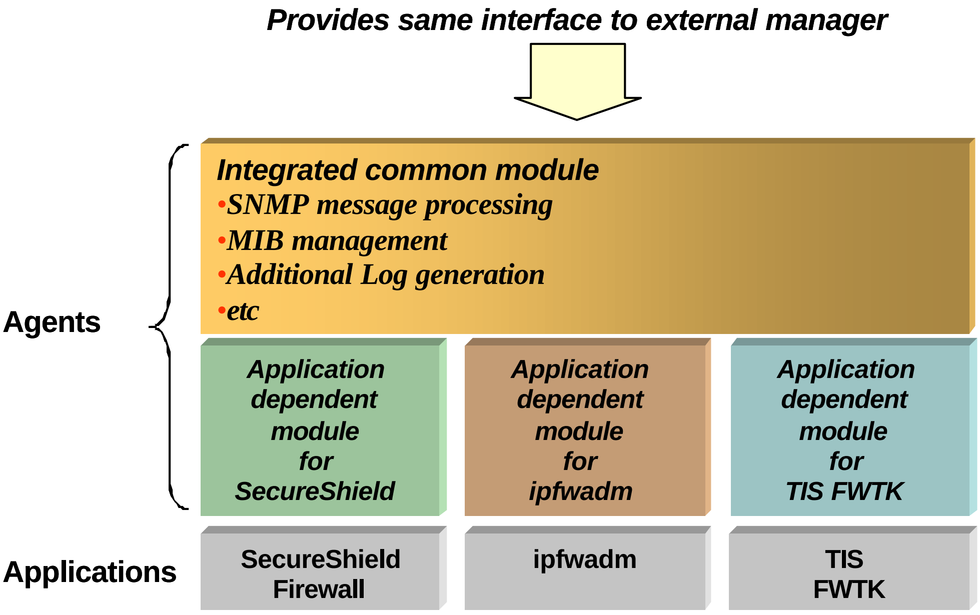
2.2. ISMS 에이전트

에이전트는 분리적인 구조를 통해 새로운 방화벽 시스템을 위해서나 이질적인 다수의 방화벽을 위해 각각 새로이 구현될 필요 없이 방화벽과 직접적인 제어 메시지를 주고받는 부분의 갱신을 통해 쉽게 재 사용될 수 있도록 구현되어 확장성, 이식성이 뛰어나다는 장점을 갖는다[15, 16]. (그림 3)는 이렇게 구현된 에이전트의 개념적인 모습을 나타내고 있다.

현재 방화벽 정책 통합관리 엔진이 관리 대상으로 정하여 시험적으로 관리되고 있는 방화벽은 SecureSoft 사의 SecureShield-수호신 v2.0[24], LINUX kernel packet filter(ipfwadm)[25], TIS Firewall Toolkit[23]의 세 가지가 있다. 이 세 가지 방화벽의 제어를 담당하고 있는 각 에이전트에 대해 동일한 기능을 수행하는 부분이 동일하게 구현되어 있고, 직접 각 방화벽을 제어하기 위한 부분만이 서로 다르게 구현되어 있다. 이러한 구조를 통해서 방화벽을 직접 제어하기 위한 부분만 새로이 구성되면 에이전트를 다른 방화벽을 위해 쉽게 재사용할 수 있다.

3. 관리 엔진 데이터베이스

관리 엔진은 DBMS 로 MySQL 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 테이블로는 다음과 같은 테이블이 존재한다.


Provides same interface to external manager
Integrated common module
•SNMP message processing
•MIB management
•Additional Log generation
•etc
Agents
Application dependent module for SecureShield
Application dependent module for ipfwadm
Application dependent module for TIS FWTK
Applications
SecureShield Firewall
ipfwadm
TIS FWTK

(그림 3) ISMS 에이전트의 개념적인 구성

3.1. 사용자 테이블(User Table)

사용자 테이블은 방화벽 정책 중앙관리 시스템의 사용이 허가된 사용자들의 정보를 관리하며, 테이블의 속성에서 'level'이 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, SM, 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 중앙관리시스템에게 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 보안 정책을 설정할 수 있다.

3.2. 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.

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 SNMP 를 이용한 통신을 하기 위해 사용되는 SNMP community 정보[11, 12, 19]를 포함한다.

<표 2> 에이전트 테이블

Field	Data Type	Description
agent_id	INT	에이전트 식별을 위한 식별값
name	VARCHAR(20)	에이전트 이름
type	ENUM	에이전트가 관리하는 방화벽의 형태 (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

3.3. 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 테이블

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	프로토콜 (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, all)
notice	ENUM	정책 위반 발생 시 처리 방법(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

3.4. 테이블간의 관계와 DBMS

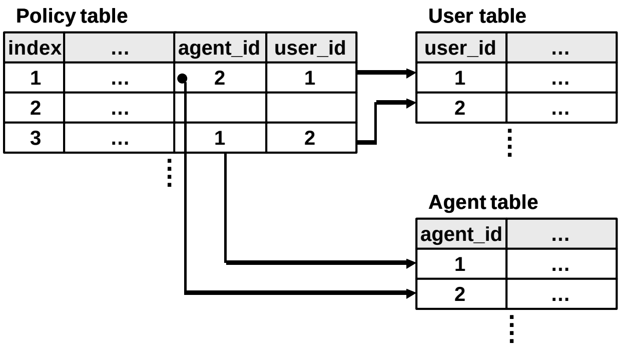
정책 관리에 사용되는 세 가지 테이블의 관계는 (그림 4)과 같이 나타낼 수 있다. 이와 같은 테이블간의 관계를 이용하여 네트워크에 흩어져 있는 각각의 방화벽에 설정되어 있는 정책을 집중 관리할 수 있다. 즉, DBMS가 지원하는 각 테이블에 대한 동작들을 이용하여 특정 방화벽에 설정된 모든 정책을 검색할 수 있으며, 특정 사용자가 설정한 모든 정책내용을 볼 수도 있다.

정책 정보 추출의 예를 들면, 특정 에이전트에 대한 정책을 열람하고자 할 경우 다음과 같은 연산을 통해 얻어질 수 있다.(σ : SELECT Operation, ID : 원하는 에이전트의 ID):

Resulting Policy List = σ agent_id = ID(Policy table)

그리고, 특정 사용자가 설정한 정책들의 목록을 보기 위한 예는 다음과 같다.(ID : 원하는 사용자의 ID):

Resulting Policy List = σ user_id = ID(Policy table)



Policy table

index	...	agent_id	user_id
1	...	2	1
2	...		
3	...	1	2
⋮			

User table

user_id	...
1	...
2	...
⋮	

Agent table

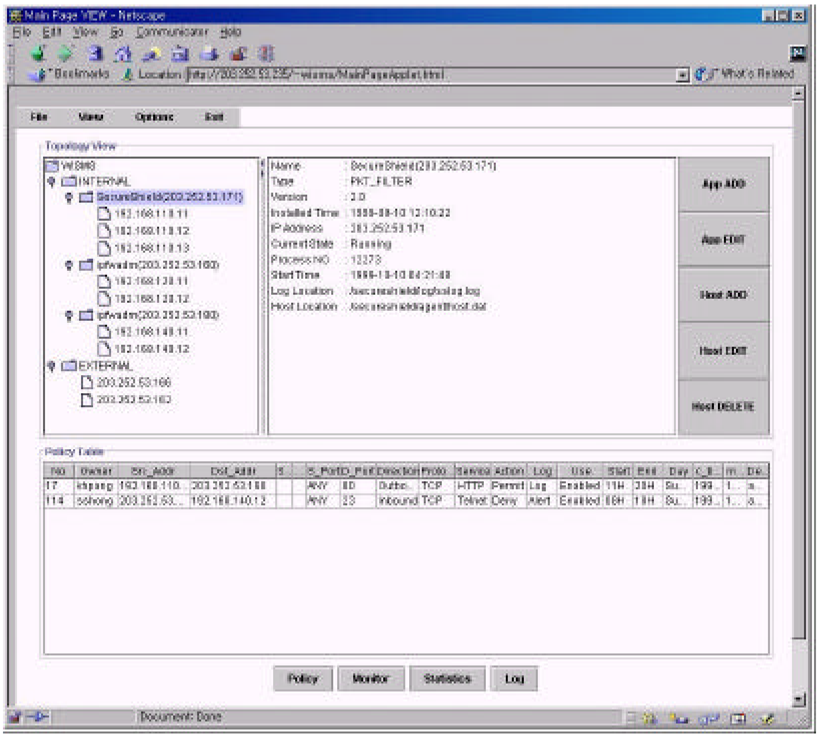
agent_id	...
1	...
2	...
⋮	

(그림 4) 각 테이블 간의 관계

정책의 중앙 관리를 위한 DBMS로 MySQL을 사용한 이유로 MySQL은 한 테이블 당 10000개 이하의 튜플(tuple)에 대한 동작 수행에 가장 좋은 성능을 보이며 자유롭게 이용 가능한 공개 DBMS라는 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 위해서 이다.

4. 관리 동작 및 정책 복구 과정

(그림 5)는 실제로 구현된 ISMS의 관리를 위해 사용되는 사용자 관리 인터페이스이다. 관리 인터페이스는 Java로 작성된 웹 기반 클라이언트 프로그램이며, 일반적인 웹 브라우저를 통해 방화벽 정책 관리를 할 수 있다. 사용자의 선택에 따라, 특정 방화벽의 정책만이 화면에 출력될 수도 있으며, 하나의 화면에 네트워크 전반에 걸쳐 설치된 방화벽의 정책들이 모두 보여질 수도 있다. 이 경우, 사용자는 마치 하나의 방화벽을 관리하는 것처럼 느낄 수 있다.



Name : SecureShield(203.252.53.171)
Type : PKT_FILTER
Version : 2.0
Installed Time : 1999-08-10 12:10:22
IP Address : 203.252.53.171
Current State : Running
Process NO : 12273
Start Time : 1999-10-10 04:21:48
Log Location : /secureshield/log/tsolog.log
Host Location : /secureshield/agent/host.dat

No	Owner	Src_Addr	Dst_Addr	S		S_Port	D_Port	Direction	Proto	Service	Action	Log	Use	Start	End	Day	C_B...	m...	De...
17	khpang	192.168.110...	203.252.53.168			ANY	80	Outbo...	TCP	HTTP	Permit	Log	Enabled	11H	20H	Su...	199...	1...	a...
114	sohong	203.252.53...	192.168.140.12			ANY	23	Inbound	TCP	Telnet	Deny	Alert	Enabled	08H	18H	Su...	199...	1...	a...

(그림 5) ISMS 의 사용자 인터페이스

관리 클라이언트 프로그램에서 전달된 사용자의 정책 설정 요구를 받은 관리 엔진은 다음과 같은 순서로 정책 설정 요구를 처리한다. 사용자가 관리 시스템에 로그인하면 관리 클라이언트를 종료할 때까지 엔진은 해당 클라이언트와의 세션 정보를 관리한다.

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.

③ 에이전트 테이블에서 해당 호스트/네트워크 접근 정책에 대한 책임이 있는 에이전트를 검색하여 찾아낸다.

④ 정책 설정의 대상이 되는 방화벽에 동일한 정책이 존재하는지를 검사한다.

⑤ 사용자의 정책 설정 요구 메시지 내의 정책 설정에 필요한 인자들을 추출한다.

⑥ 추출된 각 인자들에 해당하는 OID 와 인자들을 커플링 하여 SNMP SetRequest 메시지를 구성한다.

⑦ 에이전트에게 SNMP 메시지를 전송하여 성공적으로 응답을 수신한 경우 해당 정책을 데이터베이스에 저장한다.

위의 과정을 정보 흐름에 따라 사용자의 요구가 SNMP 메시지 형태로 변형되는 과정을 순서적으로 나타내면 (그림 6)과 같다.

방화벽 중앙 관리 시스템에서 각 방화벽의 정책을 새로이 설정하거나 갱신하는 과정에서 정책 무결성을 위반하는 조건들이 발생하였을 경우를 정책 충돌(policy collision)이라 정의한다. 방화벽 정책 중앙 관리의 일차적인 목적은 바로 정책 충돌로 인해 정책의 일관성이 깨어지는 것을 막는 것이다. 정책 충돌이 발생하는 경우는 다음과 같은 네 가지 상황이 있다. 여기서 Pe(existing policy)는 기존에 존재하는 정책이며 Pr(requested policy)은 새로이 설정이 요구되거나 갱신되도록 요구된 정책이다.

■ 동일한 정책

$\{Pe = Pr\}$

사용자가 요구한 정책과 동일한 정책이 이미 존재하는 경우이다. 즉, Pe 와 Pr 에 포함된 정책 설정 인자가 완전히 동일한 경우에 발생하는 정책 충돌 상황이다.

■ 포함관계 정책

$\{Pe \subset Pr \text{ OR } Pe \supset Pr\}$

방화벽의 정책 적용 범위가 포함관계에 있는 경우에 발생하는 정책 충돌 상황이다. 포함 관계에 있는 정책은 정책 적용 대상이 되는 주소를 기준으로 판단한다. 네트워크에 대한 정책 설정 요구는 호스트에 대한 정책 설정 요구를 포함하는 관계에 있게 된다. 즉, 네트워크 주소는 호스트 주소를 포함한다. 이러한 상황에서 일반적으로 포함하는 정책이 네트워크를 대상으로 하는 정책이고, 포함되는 정책이 네트워크에 속하는 영역(호스트나 서브 네트워크)을 대상으로 하는 경우이다.

■ 상반되는 정책

$\{Pe \leftrightarrow Pr\}$

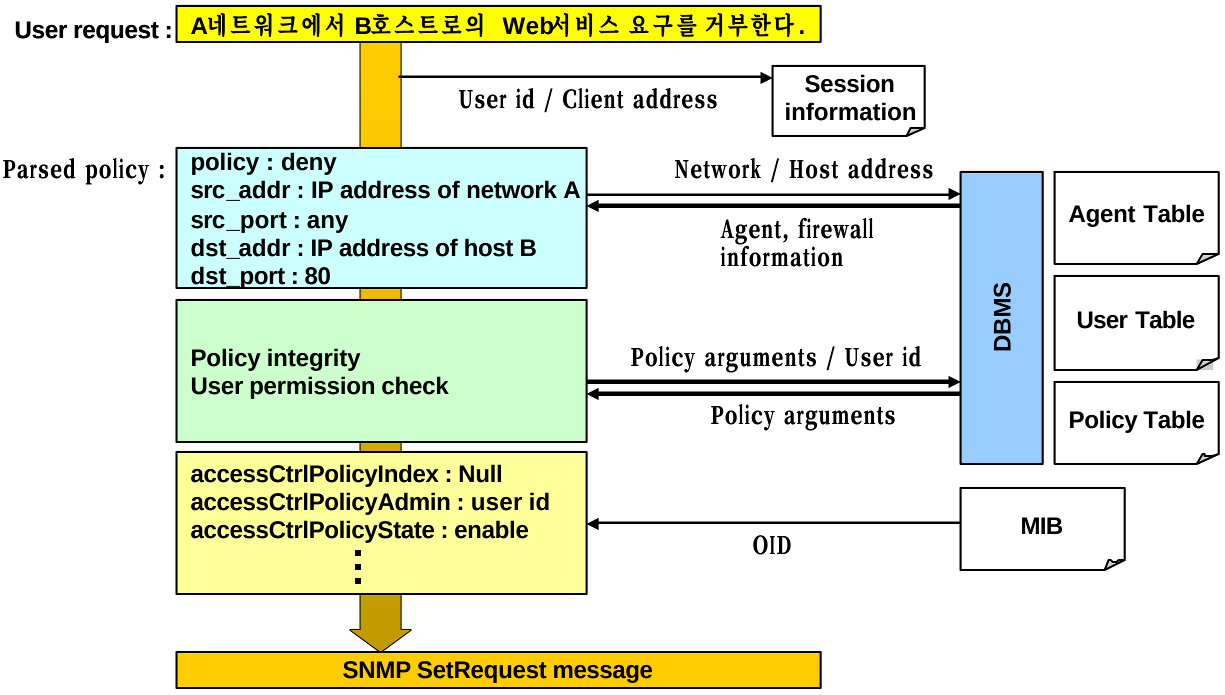
정책이 상반되는 경우는 다른 정책 설정 인자들은 동일하나, 정책 형태가 permit 과 deny 로 서로 상반되는 경우이다. 실제로 각 방화벽 제품에서는 이러한 상황이 발생하면 각 제품의 동작 방식에 따라 우선순위가 높은 정책이 적용된다. 그러나, 4 기능이 단순한 방화벽의 경우 정책이 불규칙하게 적용되는 경우도 있으며, 관리자에게는 혼란을 주게 된다.

■ 위반되는 정책

$\{(Pe \supset Pr \text{ OR } Pe \subset Pr) \text{ AND } (Pe \leftrightarrow Pr)\}$

위반되는 정책 관계는 비교 대상의 두 정책이 서로 포함관계를 가지면서 서로 상반되는 정책인 경우에 발생한다. 즉, 정책 적용 범위가 서로 다르며 한 정책의 적용 범위가 다른 정책의 적용 범위를 포함하고 정책 형태가 한 정책은 permit 이고 다른 정책이 deny 로 설정되어 서로 상반되는 경우이다.

정책 무결성 검사 과정은 하나의 방화벽에 대해 수행되는 것이 아니라 네트워크 상의 모든 방화벽에 설정되어있는 정책에 대한 검사를 수행하여 네트워크 전반에 걸친 방화벽 정책의 무결성을 보장할 수 있다.


User request :
A네트워크에서 B호스트로의 Web서비스 요구를 거부한다.
User id / Client address
Session information
Parsed policy :
policy : deny
src_addr : IP address of network A
src_port : any
dst_addr : IP address of host B
dst_port : 80
Network / Host address
Agent, firewall information
DBMS
Agent Table
User Table
Policy Table
Policy integrity
User permission check
Policy arguments / User id
Policy arguments
accessCtrlPolicyIndex : Null
accessCtrlPolicyAdmin : user id
accessCtrlPolicyState : enable
⋮
OID
MIB
SNMP SetRequest message

(그림 6) 정책 설정 요구 메시지의 처리 과정

실제적인 방화벽 운영 환경에서 갑작스러운 정전, 과도한 네트워크 부하, 방화벽 시스템의 자체적인 장애 등으로 인해 방화벽이 제 기능을 수행하지 못해 재시작 되거나 관리상의 목적으로 방화벽을 재시작 해야 될 필요가 있다. 이 경우, 방화벽이 동작을 정지한 동안 관리자의 조작에 의해 중앙 관리 시스템의 정책이 변경된 경우 등의 사건에 의해 을 대비하여 중앙 관리 DB 상의 정책과 일관성을 유지해야만 한다. 이를 위해서 각 방화벽의 정책 관리를 담당하는 에이전트는 방화벽의 이상 동작 및 시스템 재기동 시 이를 엔진에게 SNMP Trap 메시지를 이용하여 알리고 방화벽의 정책을 재설정한다. 이 절차를 정리하면 다음과 같다. 단, 에이전트는 엔진으로부터 정책을 새로이 받을 때까지 일단 방화벽의 정책을 기본 정책-모든 서비스 연결 요구 불허-으로 설정한다.

① 에이전트가 시스템 재기동이나 방화벽의 이상을 감지한다.

② 에이전트는 방화벽의 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 방화벽에 대해 DB 에서 최근에 설정된 정책 목록을 추출하고 이를 SNMP SetRequest 메시지를 이용하여 에이전트에게 전송한다.

⑤ 에이전트는 관리 엔진으로부터 전달된 정책을 방화벽에 설정한다.

⑥ 에이전트는 성공적으로 방화벽의 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 발생 시에 대비하여 각 방화벽의 정책을 쉽게 복구할 수 있으며 정책 일관성을 유지할 수 있다.

5. 결론 및 향후 계획

본 논문에서는 네트워크 상에 흩어진 각 방화벽의 중앙 관리가 갖는 장점에 대해 기술하고 실제로 다수의 이질적인 방화벽의 정책을 중앙에서 관리하기 위해 클라이언트-엔진-에이전트의 세 요소로 구현된 시스템에 대해 설명하였으며, 이 세 요소 중 엔진이 갖는 기능과 동작에 대해 설명하였다.

방화벽 정책 중앙 관리 시스템 엔진은 데이터베이스를 이용하여 다수의 이질적인 방화벽에 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 방화벽에 반영한다. 이러한 관리 구조를 통하여 다수의 방화벽을 위한 정책을 중앙 집중적으로 관리하게 된다. 그리고, 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, 방화벽에 대한 정보, 방화벽의 정책 및 방화벽의 관리 범위 내에 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 통해, 방화벽 정책 중앙 관리 시스템은 다수의 방화벽 정책 관리의 편의를 제공하며, 정책 충돌을 감지하여 정책 일관성을 유지하고, 유사시에 방화벽의 정책을 최신의 정책으로 빠르게 복구할 수 있다.

방화벽 정책을 중앙 관리 시스템의 엔진과 에이전트 사이의 프로토콜은 SNMP 를 이용하여 방화벽의 추가, 변경 및 삭제에 따른 관리 구조의 확장성을 향상시켰으며, 각 에이전트에 대해서는 타 망관리 시스템과 연동할 수 있는 가능성을 부여하였다. 또한, 관리의 보안 유지를 위해 실제 이 SNMP 메시지는 내부적인 비트 혼합을 거치게 된다.

앞으로는 방화벽 정책 중앙 관리 시스템의 각 구성요소 간 통신을 표준적인 보안 프로토콜로 전환, 현재의 메시지 보호 방식과 그 성능을 비교하

여 득실을 평가하고, 침입탐지 시스템, 호스트 접근제어 시스템과 같은 여타 보안 시스템의 통합 관리를 위한 시스템으로 확장할 계획이며, 이를 위해 타 보안 시스템의 정책 설정 내용의 일반화와 정책에 대한 SNMP MIB 정의 및 시스템 구현이 진행될 계획이다. 또한, 관리자의 신뢰성 있는 인증 메커니즘의 도입 및 적용이 필요하다.

[참고문헌]

[1] N. Freed, S. Kille, "Network Services Monitoring MIB", RFC2248, January 1998.

[2] N. Freed, S. Kille, "Mail Monitoring MIB", RFC2249, January 1998.

[3] C. Krupczak, J. Saperia, "Definitions of System-Level Managed Objects for Applications", RFC2287, February 1998.

[4] C. Kalbfleisch, C. Krupczak, R. Presuhn, J. Saperia, "Application Management MIB", RFC2564, May 1999.

[5] H. Hazewinkel, C. Kalbfleisch, J. Schoenwaelder, "Definitions of Managed Objects for WWW Services", RFC2594, May 1999.

[6] An Introduction to Computer Security : The NIST Handbook, NIST Special Publication 800-12, January 1.

[7] A Study on the Development of Countermeasure Technologies against Hacking and Intrusion in Computer Network Systems, KISA final development report, January 1999.

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

[9] D. Brent Chapman, Elizabeth D. Zwicky, Building Internet Firewalls, O Reilly & Associations, Inc., Janyary 1996.

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

[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] 이동영, 김동수, 방기홍, 김홍선, 정태명, "SNMP 를 이용한 웹 기반의 통합 보안관리 시스템", KNOM Review Vol. 2, No. 1, pp.1167-1171 April 1999.

[15] D. Y. Lee, D. S. Kim, K. H. Pang, T. M. Chung, "A Design of Scalable SNMP Agent for Managing Heterogeneous Security Systems", APNOMS '99, pp. 469-479, 1-3 September 1999.

[16] 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", NOMS2000, 10-15 April 2000.

[17] 방기홍, 김홍선, 정태명, "이기종 환경의 침입차단시스템을 위한 웹기반 보안서비스 관리시스템의 클라이언트 개발", 한국정보처리학회 추계학술대회 논문집, Vol. 6, No. 2, pp. SEC 130 - SEC 136, Oct., 1999.

[18] 이동영, 방기홍, 홍승선, 김동수, "이종의 침입차단시스템 관리를 위한 웹기반의 통합 보안관리시스템 개발", 한국정보보호센타 정보보호 우수 논문 공모전 응용기술 분야, '99 정보보호 우수논문집, pp153-180, Dec. 1999.

[19] J. Case, K. McCloghrie, M. Rose, S. Waldbusser, "Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC1902, January 1996

[20] D. Harrington, R. Presuhn, B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC2271, January1998.

[21] J. Case, D. Harrington, R. Presuhn, B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol(SNMP)", RFC2272, January 1998.

[22] D. Levi, P. Meyer, B. Stewart, "SNMP v3 Applications", RFC2273, January 1998.

[23] "TIS Firewall Toolkit Overview", Trusted Information Systems Inc., June 1994.

[24] SecureShield Administrator s Guide Version 1.0, SecureSoft Inc.

[25] Jos Vos, Willy Konijnenberg, "Linux firewall facilities for kernel-level packet screening", X/OS Experts in Open Systems BV, November 1996.

[26] Iosif G. Ghetie, Networks and System Management : Platforms Analysis and Evaluation, Kluwer Academic Publishers, 1997

김동수

1998 성균관대학교 정보공학과 졸업 (학사)

2000 성균관대학교 정보공학과 졸업(석사)

현재 성균관대학교 전기전자 및 컴퓨터공학부 박사과정

관심분야 : 네트워크 관리, 보안 관리

정태명

1981 연세대학교 전기공학과 졸업 (학사)

1984 University of Illinois Chicago, 전자 계산학과 졸업 (학사)

1987 University of Illinois Chicago, 컴퓨터 공학과 졸업 (석사)

1985-1987 Waldner and Co., Systems Engineer.

1987-1990 Bolt Bernek and Newman Labs., Staff Scientist.

1995 Purdue University, 컴퓨터 공학과 졸업 (박사)

1995-현재 성균관대학교 교수

관심분야 : 실시간 시스템, 컴퓨터/네트워크 보안, 네트워크 관리