論文
2-8-1

다중 에이전트를 이용한 통합 보안 관리시스템 (Integrated Security Management System with Multi Agent)

이 문 구*
(Moon-Ku Lee)

요 약

인터넷 환경이 급속도로 변화하면서, 관리해야하는 내부 네트워크의 규모도 커지게 되었다. 이에 따라서 보안의 중요성에 대한 인식이 확대되고, 내부 정보의 보호를 위해 많은 보안 시스템이 도입되었다. 그러나 분산된 보안 시스템들은 각각이 다른 사용자 인터페이스로 인하여 효율적인 보안관리가 어려울 뿐만 아니라 장애 발생 시 일괄적이고 즉각적인 대처가 어렵고 관리 인력의 비대화를 야기 시킨다. 따라서 본 논문에서는 시스템의 일관적이고 통합적인 관리를 위해서 중앙에서 각 시스템의 상태를 파악하고 관리하는 통합보안관리시스템을 제안한다. 제안하는 통합보안관리시스템은 다중 에이전트를 이용함으로써 장애 대처가 빠르며, 각각의 보안 솔루션들에 대한 취약점을 최소화할 수 있고, 분산된 보안 시스템들을 일괄적으로 통제 및 관리할 수 있다.

ABSTRACT

As the internet environment has been rapidly changed, the scale of internet network that needs to be managed has been magnified. In this way, the recognition for the importance of security became extensive, and numerous security systems for the protection of internal information were introduced. But decentralized security systems because of there use of different user interfaces undergo difficulties in effective security management as well as prompt coping when an obstacle happens causing a corpulence of in the security management part.

In this paper, I propose an integrated security management system which can grasp the situation of each system and manage every system in the center so that we can consistently and integrally manage every system. Integrated security management system with multi agents has the advantages of prompt coping with obstacles, and the minimization of weaknesses that different security solutions have, and of consistent control and management for decentralized security systems.

* 정회원 : 김포대학 컴퓨터계열 인터넷정보 전공 전임강사

논문접수 : 2001. 7. 27.
심사완료 : 2001. 8. 6.

1. 서론

통신망 기술의 고도화 및 지능화 추세로 발전함에 따른 편리함과 동시에 정보전, 해킹 등 보안의 역기능이 출현하게 되었다. 그러므로 보안의 중요성에 대한 인식의 확산과 함께 내부 정보의 보호를 위해 많은 보안시스템이 도입되었다. 그러나 효율적인 보안관리를 위해서 관리자는 보안제품들이 설치된 네트워크 환경에 대한 사전에 전문적인 보안 지식을 갖고 있어야 하며, 개방형 네트워크 환경의 경우 새로운 보안제품이 추가되면 새로운 보안정책과 기술이 적용되어야 한다. 이로 인하여 전산망 운영자의 보안관리 업무의 비능률적 수행과 전산망 운영 기관의 보안 관리비용을 가중시키며 체계적이고 일관적인 보안 정책 및 기술 구현을 불가능하게 하여 오히려 보안 문제를 야기 시키는 역기능을 초래 할 수도 있다. 이처럼 각각 다른 사용자 인터페이스 관리로 관리자의 부담이 가중되고, 효율적인 보안관리가 어려워짐에 따라 통합 보안관리의 필요성이 증대되었다. 따라서 본 논문에서는 다양한 보안 제품들이 독립적으로 운용되던 보안 관리를 중앙에서 체계적이고 일괄적으로 보안 시스템들을 제어할 수 있도록 유연성, 확장성 그리고 안정성을 갖춘 다중 에이전트를 이용한 통합 보안관리시스템(ISMS : Integrated Security Management System)을 제안한다.

본 논문의 구성은 다음과 같다. 2.에서는 통합보안관리시스템의 필요성과 에이전트의 도입에 대해서 기술하고, 3.에서는 다중 에이전트를 이용한 ISMS의 구조와 다중 에이전트 모듈, ISMS의 동작과정에 대하여 기술하였다. 그리고 4장에서는 결론과 향후의 연구 방향을 제시하였다.

2. 통합보안관리시스템의 개요

2.1 통합보안관리시스템의 필요성

보안을 위한 솔루션은 접근 통제 및 접근 제어의 침입차단시스템, 해킹 탐지 및 차단의 침입탐지시스템 그리고 데이터의 비밀성을 보장하는 암호화 및 사용자의 신분을 확인하는 인증시스템들이 있다.

<표 1>은 이러한 보안 솔루션을 보안 대상에 따라 분류하였다. 일반적으로 외부의 침입으로부터 내부의 네트워크를 보안하기 위한 보안 솔루션으로 제기되는 것이 침입차단시스템(일명, 방화벽)이다[2,3,8]. 침입차단시스템이 일반적인 네트워크에서 보안 솔루션으로 많은 장점을 갖는 반면에 몇 가지의 한계점을 갖고 있다.

예를 들면 보안의 위협은 외부의 침입에 의해서보다는 내부의 역기능적인 행위에 의한 경우가 전체의 65%정도를 차지하고 있으며, 침입차단시스템의 주목적이 인증되지 않은 사용자(즉, 접근이 허용되지 않는 IP)에 대해서는 차단이 일차적인 방어이다.

그렇기 때문에 자주 사용하는 FTP, TELNET 그리고 RLOGIN 등에 대하여 완벽한 보안이 어렵다. 다시 말해서 침입차단시스템은 외부 네트워크와 내부 네트워크 사이의 패킷들을 필터링 기능은 있지만 그 내용은 검사하지 않기 때문이다. 그러므로 내부의 네트워크에 대한 보안 솔루션으로 침입탐지시스템(IDS : Intrusion Detection Systems)을 두어 사용자의 오용 또는 남용으로 인한 오류 등을 방지하고자하였다[9]. 그러나 침입탐지시스템을 크게 두 가지로 분류할 수 있는데[10], 먼저 네트워크 기반의 침입탐지시스템(NIDS : Network based on IDS)는 차후의 침입을 완벽하게 차단할 수 없으며, 호스트기반의 침입탐지시스템(HIDS : Host based on IDS)의 경우는 만약에 호스트가 침입자에게 점령당하면 무용지물이 되고 만다.

이처럼 기존의 보안 솔루션의 대명사처럼 주목되고 있는 침입차단시스템 과 침입탐지시스템들은 보안 시스템 자체만으로 보안에 대한 완벽한 솔루션을 제공하지 못하고 있다. 그렇기 때문에 일반적으로 외부적인 보안을 위해 침입차단시스템 그리고 내부적인 보안을 위한 침입탐지시스템을 설치하고 또한 파일 관리를 위한 서버, 웹서버, 메일서버 등과 같이 서버들을 관리하기 위한 서버관리자를 두게 된다.

관리자의 입장에서는 이러한 모든 보안 시스템들이 하나로 통합 관리되지 않으므로 일관적인 보안관리가 결여될 뿐만 아니라 지속적인 보안정책을 수립하고자 할 때도 일관성이 결여된다.

<표 1> 보안 솔루션의 분류
<Table 1> Classification of Security Solution

보안 대상	접근 통제 및 제어	해킹 탐지 및 차단	바이러스
네트워크	침입 차단 시스템 (Firewall)	네트워크기반 침입 탐지 시스템 (NIDS)	네트워크용 백신 (Virus Wall)
서버	서버 보안 시스템 (File Access Control)	호스트기반 서버 침입 탐지 시스템 (HIDS)	서버용 백신
PC	PC보안 시스템 (부팅제어, 암호화저장)	PC 해킹 탐지 시스템 (Back Orifice 차단)	PC용 백신
인증	신분확인과 권한 부여 및 제어		
암호화	네트워크와 네트워크간 암호화, 서버와 PC 간 암호화		
전자 메일	전자메일을 통한 정보 유출 방지 및 시스템(e-mail 모니터링 시스템)		
웹 접속	비 업무용 웹 사이트 접속 차단 시스템(Web 차단 시스템)		
스캐너	시스템 및 네트워크 취약점 분석 시스템		

따라서 본 논문에서는 침입차단시스템과 침입탐지 시스템 그리고 서버관리시스템 등과 같은 네트워크 장비 등을 통합 모니터링 및 관리할 수 있는 다중 에이전트를 이용한 통합보안관리시스템(ISMS : Integrated Security Management System)을 제안한다.

2.2 에이전트의 도입

본 논문에서 통합적인 보안관리를 위하여 제안하는 다중 에이전트를 이용한 통합보안관리시스템(ISMS : Integrated Security Management System)은 분산 배치된 각 보안 시스템들의 보안 솔루션을 위해서 에이전트를 이용하여 상호 인터페이스 하도록 한다. 에이전트란 사용자를 대신하여 사용자가 원하는 어떤 일을 수행해주는 프로그램이라고 할 수 있는데, 일반적으로 자율적(autonomous)이고 지능적(intelligent)인 특징을 갖는다. 즉, 에이전트는 서로 독립적으로 실행하는 개체이기 때문에 자율적이며, 동적으로 시스템에 추가 또는 삭제가 가능하다. 에이전트의 자율성이란 사용자를 대신하여 기능을 수행하는 것을 의미한다. 이는 에이전트가 상호적인 통신이 아닌 독립적으로 수행할 수 있다는 능력이다.

일반적으로 분산환경에서의 작업 수행은 두 개체 사이의 상호 작용을 통하여 이루어 졌지만, 에이전트는 한 번의 파견과 한번의 귀환만을 통해 다수의 네트워크 작용을 감소 시켰다. 이러한 점은 대역폭이 낮고 높은 지체를 일으키는 고비용 환경에서 특히 유용한 기능이다[4].

에이전트끼리 대화한다는 것은 정해진 언어 규약에 따라 메시지를 주고받음을 의미한다. 따라서, 에이전트는 다른 에이전트에게 서비스를 요청하기 위해 정해진 언어 규약에 따라 요구사항을 메시지 형태로 바꾼 후 해당 에이전트에게 전달한다. 다른 에이전트로부터 서비스 요청을 받은 에이전트는 그 메시지를 분석해 내부에서 처리할 수 있는 형태로 변환해서 이를 처리한다. 에이전트는 그 결과를 다시 메시지 형태로 바꾸어서 요청한 에이전트에게 전달한다. 이처럼 다중 에이전트 시스템(multi-agent system)의 장점은 독립적인 응용 프로그램의 집합으로는 해결할 수 없는 보다 복잡한 서비스를 다른 에이전트와의 협력을 통해 제공 할 수 있다는 것이다. 이밖에도 자신이 필요로 하는 에이전트를 시스템에 붙임으로서 새로운 서비스에 대한 시스템의 확장이 용이하다는 장점이 있다.

에이전트는 스크립트 언어로 쓰여진 프로그램이다. 이는 서로 다른 분산환경에서 사용자의 역할을

대신하여 사용자가 요구하는 작업을 수행할 수 있다는 것이다. 에이전트를 작성하는데 사용되는 언어는 Perl, Java, Tcl/Tk 등이 존재한다. 여기서 Perl의 경우 패턴 매칭(pattern matching)작업이 수월하고 Java의 경우는 스레드(thread)사용이 용이하다[5].

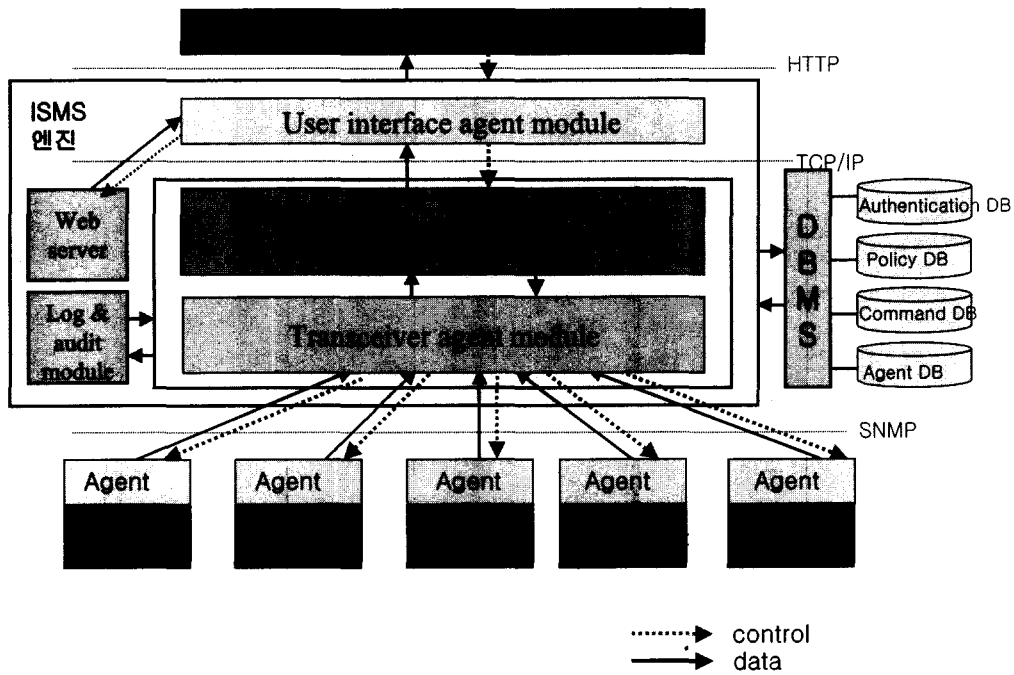
본 논문에서는 각 보안 시스템으로부터 발생하는 패킷이나 시스템 콜 등의 이벤트에 의하여 생성된 에이전트들을 패턴 매칭 방법으로 시스템의 보안상태를 실시간에 관리할 수 있도록 Perl을 사용하였다.

3. 통합보안관리시스템

3.1 ISMS의 구조

통합적인 보안관리를 위하여 본 논문에서 제안하는 다중 에이전트를 이용한 통합보안관리 시스템(ISMS)의 구조는 [그림 1]과 같다.

전체적인 구조는 보안관리자, ISMS 엔진 그리고 침입차단시스템(Firewall), 침입탐지시스템(IDS), 가상사설망(VPN), 서버관리시스템 그리고 라우터와 같은 네트워크 장비 보안 시스템 등과 같은 보안 시스템에서 동작하는 에이전트들로 구성된다. ISMS 엔진에는 트랜시버 에이전트 모듈(transceiver agent module), 모니터 에이전트 모듈(monitor agent module) 그리고 사용자 인터페이스 에이전트 모듈(user interface agent module)로 구성된다. 이밖에도 로그 및 감사기록을 위한 로그 및 감사 모듈(log & audit module), 웹 서버 그리고 데이터베이스 관리 시스템(Data Base Management System)으로 구성된다. 데이터베이스 관리시스템은 인증관리를 위한 인증데이터베이스(authentication database), 보안 정책 데이터베이스(policy data base), 명령어 데이터베이스(command database) 그리고 에이전트 데이터베이스(agent database)로 구성된다.



[그림 1] 통합보안관리시스템(ISMS)의 구조

[Fig. 1] The Structure of ISMS

3.2 ISMS 에이전트 모듈의 구성

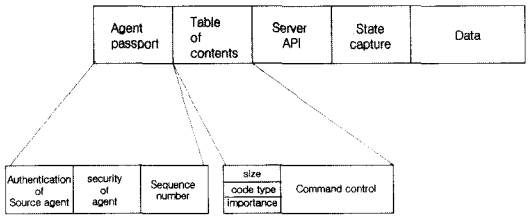
각각의 보안 시스템에는 많은 에이전트들을 가지고 있으며, 에이전트들은 보안 시스템에서 발생하는 이벤트를 감시한다. 하나의 보안 시스템에서 모든 에이전트들은 트랜시버 에이전트 모듈에 그들의 발견사항을 보고한다[6].

트랜시버 에이전트 모듈에서는 분산되어있는 보안 시스템에서 동작하는 에이전트들을 제어하며, 에이전트에게 구성(configuration) 명령어들을 시작하거나 중단하고 전송할 수 있다. 즉, 트랜시버 에이전트 모듈에서는 에이전트의 시작과 중지 그리고 전송명령 등으로 에이전트를 제어하는 모듈이다. 트랜시버 에이전트 모듈에서는 또한 에이전트들에게서 받은 데이터들로부터 데이터들을 정리하여 모니터 에이전트 모듈에게 그 결과를 보고한다.

모니터 에이전트 모듈은 트랜시버 에이전트 모듈의 행위를 감독하고 사용자 인터페이스 에이전트 모듈로 정보를 제공한다. 즉, 모니터 에이전트 모듈은 궁극적으로 사용자 인터페이스 에이전트 모듈에게 정보를 제공하고 사용자 인터페이스 에이전트 모듈로부터 제어 명령어를 얻을 수 있다.

3.3 ISMS 에이전트의 일반적인 구조

[그림 3]은 에이전트의 프로토콜 데이터 단위(PDU : Protocol Data Unit)의 일반적인 구조이다.

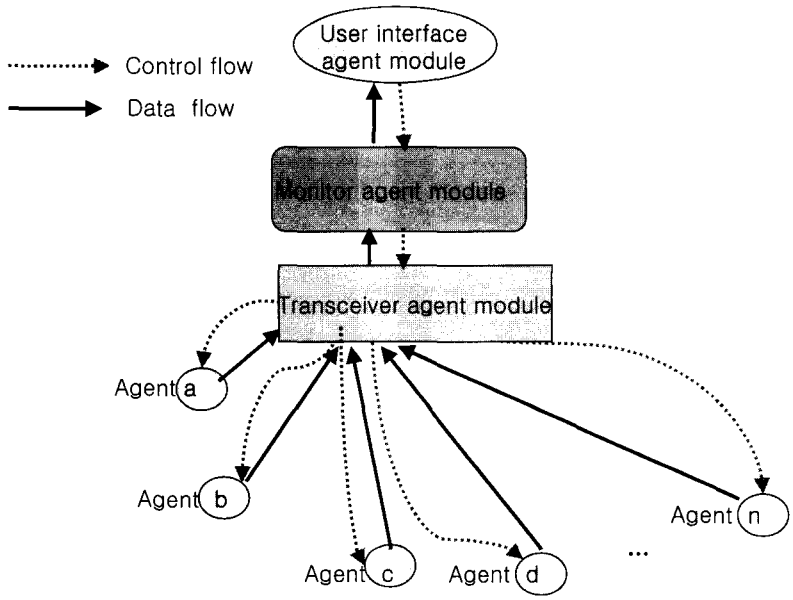


[그림 3] 에이전트 PDU 구조

[Fig. 3] The PDU Structure of agent

1) Agent passport :

- Authentication of source agent : 각 보안 시스템으로부터 발생된 에이전트 인증 필드.
- Security of agent : 이동성이 있는 에이전트의 보안을 위하여 암호화 및 복호화 필드.
- Sequence number : 에이전트의 실행 순서를 표시한 필드.



[그림 2] ISMS에서 에이전트의 물리적인 구성

[Fig. 2] The Physical Structure of agents in ISMS

2) Table of contents :
- size : agent의 필드 사이즈.
- code type : 서로 다른 보안 시스템의 이벤트에 의해서 생성된 멀티 에이전트들의 코드 유형을 나타내는 필드.
- importance : 각 보안 시스템에서 생성되는 이벤트의 중요도에 따른 등급을 설정하여 보안정책 데이터베이스(policy DB)에 등급 자료를 설정해 둔다. 즉, 보안시스템의 이벤트가 발생하면 보안정책 데이터베이스에서 보안의 중요도에 따른 등급이 설정된다.
- Command control : 명령어가 데이터 처리(data processing)또는 사용자(보안 관리자)의 요청에 따른 처리를 제어하는 필드.

3) Server API
서버와의 대화를 위한 응용 프로토콜 인터페이스 필드.

4) State
실행되는 에이전트의 내부 상태를 캡쳐하는 필드.

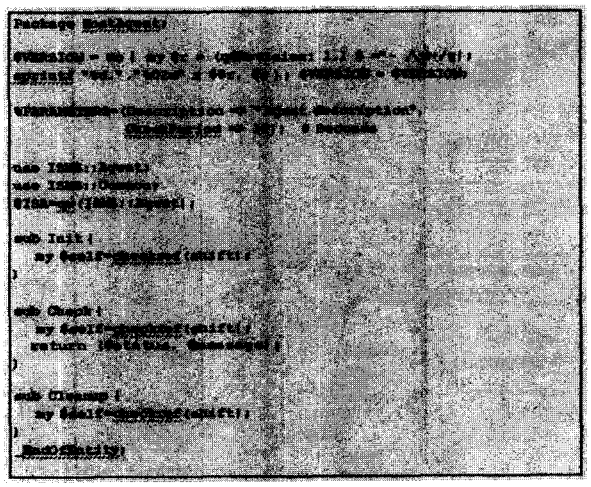
5) Data

다음 [그림 4]는 ISMS에서 사용되는 에이전트를 작성하는 가장 간단한 방법으로 가장 필수적인 템플리트(template)로 코드를 나타낸다.

템플리트 코드는 HostAgent.pm 파일에 저장되고, HostAgent는 package 문장에서 제공되는 이름이다. 설명을 위해 적절한 값을 입력하고 %PARAMETERS 해쉬에서 주기를 체크한다. 에이전트는 체크를 수행하는 Check 메소드와 에이전트를 생성하고 삭제하는 Int메소드 그리고 Cleanup 메소드들을 가지고 있어야 한다. 이 세 가지 메소드들은 에이전트 자체에 대한 참조(reference)인 하나의 인자(argument)만을 받게 된다. 이 참조를 통해 에이전트는 자신의 파라미터들과 다른 메쏘드들을 접속할 수 있다.

Check 메소드는 ($status, $message) 형태의 두 개의 엘리먼트 리스트이어야 한다. $status는 보안정책 데이터베이스(policy DB)에 정의되어있으며, 0에서 10사이의 정수의 값을 나타낸다. 0은 분산된 보안 시스템으로부터 아무런 문제가 없이 정상적인 통신이 이루어진 상태를 나타낸다. 반면에 10은 분산되어있는 보안 시스템으로부터 가장 심각한 상태 즉, 보안의 위험 요소들이 발생했던 것을 나타낸다. 현재 상태의 설명은 $message에 입력한다.

소스 파일의 마지막 라인은 _EndOfEntity; 이어야 한다. 이것은 에이전트가 다른 개체(예, 트랜시버)나 stand-alone 프로그램에 의해 로드되는 것을 보장한다.



[그림 5] ISMS 에이전트의 템플리트(template)
[Fig. 4] Template of ISMS agents

3.4 ISMS의 동작과정

다음 [그림 5]는 ISMS의 동작 과정을 도식화 한 것이다. ISMS는 분산되어있는 각각의 보안 시스템으로부터 발생하는 모든 이벤트에 대하여 에이전트를 생성하고, 생성된 에이전트는 트랜시버 에이전트 모듈로 전송된다.

이처럼 분산된 에이전트들로부터 전송되어진 에이전트들은 보안에 대하여 매우 취약하다. 그렇기 때문에 암호화 된 에이전트는 트랜시버 모듈에서 암호화 및 복호화를 실행하게 된다[1].

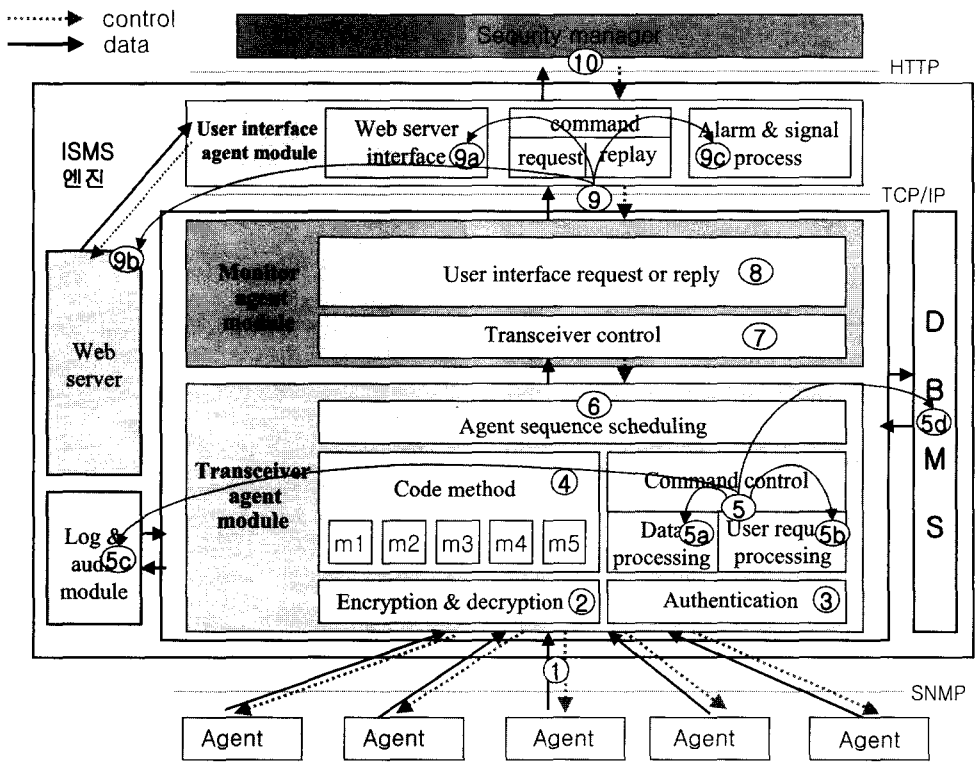
[그림 6]은 트랜시버 에이전트 모듈과 모니터 에이전트 모듈의 구조이다. 분산된 보안 시스템에서 발생되는 이벤트에 따라 생성되는 에이전트들은 이동성을 갖기 때문에 보안에 많은 취약점을 갖는다.

그러므로 암호화 되어있는 각 에이전트들은 트랜시버에서 암호화 및 복호화를 실행한다. 분산되어있는 각 보안 시스템의 에이전트로부터 ISMS엔진에게 정보를 보내거나 에이전트에게 요청 메시지를 보내고자 할 경우 이동성이 있는 에이전트에게는 무엇보다도 가장 큰 문제점이 보안일 것이다[1].

또한 수많은 에이전트 중에서 자신이 원하는 서비스를 제공하는 에이전트를 어떻게 효율적으로 찾는가하는 인증의 문제를 갖게 된다. 따라서 본 논문에서 제안하는 ISMS의 에이전트들의 보안을 위하려 해쉬알고리즘을 이용하여 암호화 및 인증기능을 갖도록 한다.

일반적으로 해쉬알고리즘은 다양한 길이의 입력을 고정된 짧은 길이의 출력으로 변환하는 함수이며, 해쉬알고리즘의 핵심적인 암호적 특성은 일 방향이고 충돌이 없다는 것이다.

또한 공개키 암호 알고리즘보다 속도가 빠르다. 즉, 두 개체 통신 객체간의 키 값이 프로그램 설치시에 교환 및 분배된다. 해쉬알고리즘의 처리과정은



[그림 6] ISMS의 에이전트 모듈과 동작

[Fig. 5] Module and Operation of ISMS agents

다음과 같이 기술 할 수 있다[7].

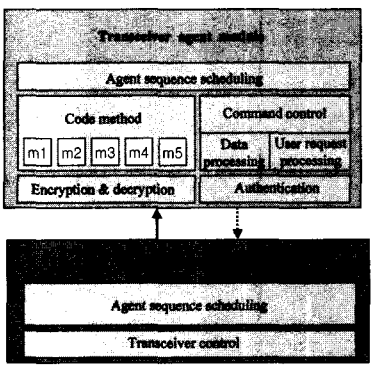
임의의 길이의 메시지 X를 입력단위의 배수가 되도록 덧붙이기 하여 t개의 입력 블록 (X_1, …, X_t) 로 분할한다. 해쉬 코드는 각 블록 X_i에 대해 연쇄 변수를 주어진 초기값(IV)으로 초기화 한 후 압축함수를 반복적으로 적용하여 계산된다.

$$H_0 = IV$$
$$H_i = f(H_{i-1}, H_i), 1 \le i \le t,$$
$$h(X) = H_t$$

여기서 f는 h의 압축 함수이며, H_i는 단계 i - 1과 i의 중간 계산 값이다.

트랜시버 에이전트 모듈에서는 에이전트의 근원지에 대한 인증과정을 실행하고, 멀티 에이전트들의 코드 방식에 따라 분류를 한다. 데이터 처리 또는 사용자의 요구에 대한 처리를 위한 명령어 제어를 수행한다. 그리고 멀티 에이전트들의 우선순위 또는 이벤트의 중요도에 따라 에이전트 실행 순서를 위한 시퀀스 스케줄링이 이루어진다.

트랜시버 에이전트 모듈에서 전송되어온 에이전트들은 모니터 에이전트로 전송된다. 모니터 에이전트 모듈에서는 이들 에이전트들을 제어하고, 사용자 인터페이스 에이전트모듈에 전송하거나, 사용자 인터페이스 에이전트 모듈의 응답을 취합하여 재전송을 하게된다.

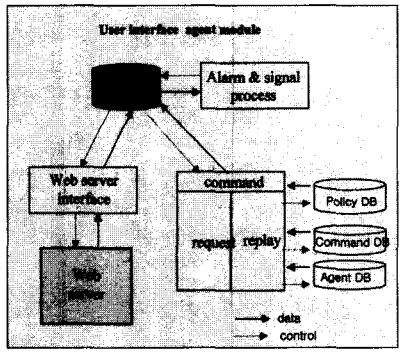


[그림 7] 트랜시버와 모니터 에이전트 모듈의 구조

[Fig. 6] Structure of Transceiver and Monitor agents module

모니터 에이전트모듈은 트랜시버를 감독 및 제어하고, 사용자인터페이스의 정보 요청 및 제공이 이루어진다.

[그림 7]은 사용자 인터페이스 에이전트 모듈의 구조이다. 사용자 인터페이스 에이전트 모듈은 사용자 즉, 통합 보안 관리자로부터 정보에 대한 요청과 응답을 웹서버로 인터페이스가 이루어지게 된다. 이때 모든 상태는 웹서버로부터 인터페이스가 이루어지고, 보안 등급의 정도에 따라 알람 및 신호를 보안 관리자에게 보내게된다. 사용자 인터페이스는 지속적으로 보안관리자와 에이전트로부터 전송되어온 메시지에서 보안 정책의 중요도를 보안정책 데이터베이스와 정보를 지속적으로 교류하게된다. 만약, 보안상의 위험상태나 보안에 대한 침입이 시도되었으나 실패한 경우 등에 대하여 사건의 중요도에 따라 신호(예, 알람) 처리 기능을 제공하도록 한다. 그리고 명령어 형식이 요청(request) 및 응답(reply)을 제어하고 실행한다.



[그림 7] 사용자 인터페이스 에이전트 모듈

[Fig. 7] User Interface Agent Module

명령어는 데이터를 처리 또는 사용자의 요청을 제어하게 되며, 발생된 에이전트에 대한 로그 및 감사 기록이 진행되고, 에이전트 명령어들의 중요도를 설정하기 위하여 데이터베이스의 보안정책(security policy)에 따라 0에서 10사이의 정수의 값을 얻게 된다.

그리고 이벤트의 중요도와 우선순위에 따라 에이전트 시컨스에 대한 스케쥴링이 이루어진다. 이로써 트랜시버 에이전트 모듈에서는 에이전트들의 동작을 지속적으로 감독할 뿐만 아니라, 전송 명령어를 실시간에 수행할 수 있도록 한다.

4. 결론

컴퓨터와 정보통신 기술의 발달 및 전송 속도의 고속화와, 지능화로 대용량의 데이터 전송 등으로 업무 효율을 향상시키게 되었고, 생활의 질을 높여 주며 국가 경쟁력을 강화 시켜주는 긍정적인 효과를 거두게 되었다. 그러나 개방된 네트워크 구조인 인터넷의 확산으로 생활의 편리함이 이루어지는 반면 컴퓨터 바이러스 및 해킹과 정보전과 같은 정보 자원에 대한 침입 가능성은 날로 증대되고 있다. 이에 따라 네트워크 및 시스템의 구성 및 환경에 따라 다양한 보안 시스템이 구축되었다. 그러나 이러한 보안 시스템들은 각각이 다른 사용자 인터페이스로 인하여 효율적인 보안관리가 어려울 뿐만 아니라 장애 발생 시 일괄적이고 즉각적인 대처가 어렵고 관리 인력의 비대화를 야기 시킨다. 그렇기 때문에 본 논문에서는 시스템의 일관적이고 통합적인 관리를 위해서 중앙에서 각 시스템의 상태를 파악하고 관리하는 통합보안관리시스템을 제안하였다. 제안하는 통합보안관리 시스템은 다중 에이전트를 이용함으로써 장애 대처가 빠르며, 각각의 보안 솔루션들에 대한 취약점을 최소화할 수 있고, 분산된 보안 시스템들을 일괄적으로 통제 및 관리할 수 있다. 차후에는 제안하는 통합보안관리시스템이 보다 포괄적인 호환성을 갖는 시스템의 구축이 더 연구되어야 한다.

※ 참고 문헌

[1] Antonio Corradi, Rebecca Montanari, Cesare Statefanelli, "Security Issues In Mobile Agent Technology", http://wwwlia.deisunibo.it/Software/SOMA.

[2] C. Hare and K. Siyan, Internet Firewalls and Network Security, 2nd Ed. New Riders, 1996.

[3] D. Chapman and E. Zuicky, Building Internet Firewalls , O'Relly & Associates. Inc., 1995.

[4] Hartmut Vogler, Thomas Kunkelmann, Marie-Louise Moschgath, "An Approach for Mobile Agent Security and Fault Tolerance using Distributed Transactions", Proceedings of the 1997 International Conference on Parallel and Distributed Systems, pp. 268-274, 1997 IEEE.

[5] H.R. Frost and M.R. Cutkosky, "Design for Manufacturability via Agent Interaction," Paper No. 96-DETC/DEM-1302, Proceeding of the 1996 ASME Computers in Engineering Coference, Irvine, CA, August 18-22, 19996, pp.1-8.

[6] Jai Sunder Balasubramaniyan, Jose Omar Garcia-Fernandez, Engene Spafford, and Diego Zamboni. An Archiecture for intrusion detection using autonomous agent. Technical Report 98-05, COAST Laboratory, Perdue University, West Lafayette, IN 47907-1398, mAY 1998.

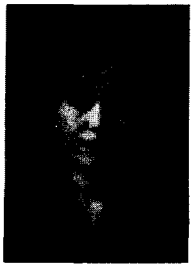
[7] Rivest, R., "The MD5 Message-Digest Algorithm." Internet report. RFC 1321, aPR 1992.

[8] W. R. Cheswick and S. M. Bellovin, Firewalls and Internet Security, Addison Wesley, 1994.

[9] 이 문구. "분산 환경을 위한 실시간 침입 탐지 모델의 설계" 통신 보호학회 논문지 제 9권 1호 1999. 3.

[10] 이 문구 저. "인터넷 보안" 도서출판 문영 2001.6.

이 문 구



1984년 숭실대학교
전자계산학과 (학사)
1993년 이화여자대학교
교육대학원 전산학과 (석사)
2000년 숭실대학교 대학원
전산과 (공학 박사)
1997년~1999년 명지
전문대학 전산과 겸임교수
2000년3월~현재 김포대학
컴퓨터계열 인터넷정보 전공
전임강사
관심분야 : 정보통신,
네트웍 프로그램, 암호 이론,
인터넷 보안,
침입차단 시스템