면역 시스템 모델을 기반으로 한 침입 탐지 시스템 설계 및 성능 평가

Performance Evaluation and Design of Intrusion Detection System Based on Immune System Model

이종성*, 채수환*
Jong Sung, Lee, Soo Hoan, Chae

Abstract

Computer security is considered important due to the side effect generated from the expansion of computer network and rapid increase of the use of computers. Intrusion Detection System(IDS) has been an active research area to reduce the risk from intruders. We propose a new IDS model, which consists of several computers with IDS, based on the immune system model and describe the design of the IDS model and the prototype implementation of it for feasibility testing and evaluate the performance of the IDS in the aspect of detection time, detection accuracy, diversity which is feature of immune system, and system overhead. The IDSs are distributed and if any of distributed IDSs detect anomaly system call among system call sequences generated by a privilege process, the anomaly system call can be dynamically shared with other IDSs. This makes the IDSs improve the ability of immunity for new intruders.

* 한국항공대학교 컴퓨터공학과 분산/병렬처리 연구실

1. 서 론

컴퓨터 및 네트워크 기술이 발전하고 이에 대한 의존도가 증가함에 따라 컴퓨터의 결함은 인적 물적 손실뿐만 아니라 조직의 경쟁력을 약화시키는 결과를 초래하게 되어 정보사회의 역기능으로 컴퓨터 보안 문제가 중요하게 대두되고 있다. 이러한 컴퓨터 보안 문제에 대처하기 위해 정보보호를 요하는 시스템에 대한 불법 침입을 분석하고 탐지하는 감사 기술의 발전적 형태인 침입탐지시스템(Intrusion Detection System : IDS)에 관한 연구가 활발히 진행되고 있다[1,2,3,4]. 침입탐지시스템은 탐지 대상 시스템에 대한 정상적인 사용에 관한 프로파일과 시스템 상태를 유지하고 있는 동안 이 프로파일에서 벗어나는 행위들을 탐지하는 비정상적인 행위 탐지(anomaly detection) 방법과 시스템의 알려진 취약점들을 이용한 공격 행위들에 대한 공격 특징 정보를 통해 침입을 탐지하는 오용 침입탐지(misuse detection) 방법을 사용한다[4].

침입탐지시스템의 중요 요구사항은 시스템 관리자 없이도 지속적으로 수행되어야 하며, 컴퓨터 시스템에 최소한의 오버헤드를 부과해야 하고, 새로운 침입 유형의 변화에 대한 지속적인 학습 기능과, 어떤 침입탐지모듈에 결함이 발생되어도 전체 침입탐지시스템에 큰 영향을 주지 않는 결함 허용 관리 기능, 그리고 시스템의 정상상태를 침입이라고 탐지하는 긍정적 결함(false positive) 및 시스템의 침입상태를 정상상태로 판단하는 부정적 결함(false negative)과 같은 잘못된 침입 탐지를 방지해야 하고 실시간에 침입을 탐지해야 한다[3,4,6].

이와 같은 침입 탐지 서비스의 요구에 따라 다양한 기법과 모델들[7-13]이 개발되고 있으나 컴퓨터 통신망의 복잡성, 대상 시스템의 원초적 취약성, 정보 보호에 대한 이해 부족 및 새로운 불법 침입 기법의 개발 등으로 기존의 어떤 기법 또는 모델도 완전하지 못한 실정이다. 특히, 탐지대상에 대한 정상 개념이 시간이 지남에 따라 지속적으로 변화되므로 비정상적인 행위 탐지 방법에 의해 IDS를 구현하는 것이 오용탐지방법에 의해 IDS를 구현하는 것 보다 많은 어려운 점이 존재한다. 따라서 현재 상용화된 IDS의 대부분은 새로운 침입을 탐지하지 못하는 결점을 내재한 오용탐지방법에 의해 구현됨에 의해 오용탐지방법과 동일한 문제인 새로운 침입을 탐지하지 못하는 문제점을 안고 있다[23].

이에, 본 논문은 탐지 대상을 특권 프로세스로 하고, 특권 프로세스(privilege process)가 수행할 때 발생하는 시스템 호출 순서 중 비정상적인 시스템 호출을 탐지하여 이를 분산된 각각의 침입탐지 시스템들이 서로 동적으로 공유하여 침입에 대한 면역력을 향상시키는 면역 시스템 모델을 기반으로 한 침입탐지시스템을 설계하고 프로토타입을 구현하여, 이를 통해 제안한 침입탐지시스템의 접근 타당성을 고찰하고, 탐지시간, 탐지정확도, 면역 시스템의 특징인 다양성, 그리고 시스템 오버헤드 관점에서 제안한 침입탐지시스템의 성능을 평가한다.

2. 비정상적인 행위 탐지 방법에 대한 관련 연구

비정상적인 행위 탐지 방법은 시스템 또는 사용자의 행위가 정상 행위로부터 벗어나는 것을 탐지하는 것으로 이를 위해 시스템 또는 사용자의 정상 행위를 기록한 감사 데이터로부터 여러 가지 방법을 통해 정상 행위를 추출한 후, 수행되는 시스템의 행위가 정상 행위에 벗어나면 경고를 발생한다. 비정상적인 행위 탐지 방법은 예측하지 못한 시스템 취약점을 이용하려는 시도를 탐지할 수 있어 새로운 침입을 자동으로 탐지할 수 있으며, 어떤 보안 취약점을 직접적으로 이용하지는 않지만 특권을 오용하는 공격도 탐지할 수 있다. 그러나, 구성된 정상 행위 정보가 시스템의 모든 정상 행위를 포함하지 않기 때문에 긍정적 결함(false positive) 오류가 발생할 확률이 높으므로 이를 낮추는 방안이 모색되고 있다[25].

이하 대표적인 비정상적인 행위 탐지 방법을 간단하게 살펴보고 이를 통해 면역 시스템 모델을 기반으로 한 침입탐지시스템의 제안 배경을 설명한다.

• '통계적 접근 방법'은 침입탐지시스템에 가장 많이 사용되는 방법으로, 탐지과정은 먼저 사용자나 사용자가 실행시킨 프로세스의 행위를 관찰하

고, 각각의 행위에 대한 프로파일을 생성한 후, 사용자 및 시스템의 행위가 기설정된 정상 행위로부터 벗어나는지 감시한다. 이 접근 방법은 어떤 행위의 발생 순서를 고려하지 않고 단지 발생 빈도수만으로 정상 행위를 모델링하므로 동적으로 수행되는 시스템을 모델링하는데 제한적이라는 단점을 갖고 있으나, 현재 많은 침입탐지 시스템들과 프로토타입에 사용되고 있다[25].

• '전문가 시스템'은 if-then-action의 규칙 표현 방식에 따라 전문가의 지식을 표현하는 인공지능 기법으로 사용자의 정상 행위를 규칙을 사용하여 표현하였으며 이 방법은 새로운 사용 패턴을 추가하기 위해 규칙 집합을 생성하는 전문가가 필요하여 정확한 정상 행위를 지속적으로 구축하기가 어렵다는 문제점이 있다[25].

• '신경망'을 이용한 접근은 두 개 정보 집합간의 관계성을 학습하기 위해 사용하는 알고리즘적 기술로서, 이 방법은 명령어의 순서를 신경망으로 학습시켜서 다음에 수행될 명령어를 미리 예측할 수 있게 한다. 이 방법은 많은 연산량을 요구하는 기술이므로 침입탐지 시스템에 폭넓게 사용되고 있지 않다[25].

• '사용자 중심 접근 방법'은 시스템에서 사용자들이 수행할 태스크들의 집합에 의해 사용자들의 정상 행위를 모델링한 후, 각각의 사용자가 수행할 수 있는 태스크 집합을 관리하고, 발생된 어떤 행위가 태스크 패턴과 일치하지 않으면 경보를 발생한다. 이 방법의 단점은 새로운 사용자가 추가되면 이 사용자가 수행할 수 있는 정상 행위를 다시 모델링해야 하므로 침입탐지 시스템의 확장성에 원초적인 문제를 안고 있다[25].

• '컴퓨터 면역시스템'은 자연 면역시스템(natural immune system)을 모델링한 것으로서 1994년도에 S.Forrest 교수에 의해 면역 시스템과 컴퓨터 보안의 결합에 대해 소개된 후 지속적으로 연구되고 있으나, 실질적으로 어떻게 자연 면역시스템 아이디어를 컴퓨터 보안에 적용시킬 것인가에 대한 연구는 부족한 상태이다[11,12,13,22,23,25,26].

이와 같은 시스템들은 초기 학습 기능만 존재하는 정적인 시스템이고 각 시스템들마다 시스템에 대한 동일한 정상행위 지식을 가지고 있으므로 어떤 하나의 시스템이 침입 받으면 동일한 방법으로 다른 시스템이 침입을 받을 확률이 증가하므로 전체 침입탐지 시스템의 결함 허용 수준이 낮아진다. 따라서, 침입탐지시스템의 중요 요구사항인 새로운 침입 유형의 변화에 대한 자체 학습 기능과 결함 허용 관리에 적합하지 못하다.

3. 모델

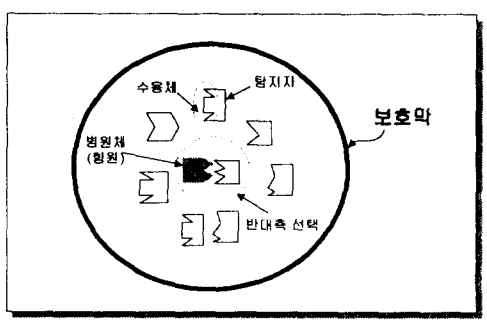
본 장에서는 자연면역시스템의 특징을 고찰하고, 이를 만족시키는 본 논문에서 제안한 침입탐지 시스템에 대한 이론적 모델에 대하여 논한다.

3.1 자연면역시스템의 특징

자연면역시스템은 다음과 같은 특징들을 가지고 있다[11,12,22,23,25,26].

○ 다계층 방어 특징

동물의 육체는 <그림 1>에 도시된 바와 같이 피부 및 점액막과 같은 정적인 보호막, pH 및 온도와 같은 생리적 조건, 일반적인 염증 대응, 그리고 B cell과 T cell 메커니즘에 의한 적응형 대응을 통해 외부로부터의 병원체 침입에 대해 다계층으로 방어하는 특징이 있다.



<그림 1> 자연면역시스템

○ 분산탐지 특징

면역시스템은 탐지자들을 동물의 육체에 분산시킨 후 침입한 병원체에 대해 대응한다. 면역시스템은 탐지자들간의 상호작용과 다양한 셀 분할 그리고 일정시간 작동 후 소멸되는 탐지자의 소멸률을 조절하여 육체에서 탐지자들이 필요한 곳에 탐지자들을 할당할 수 있고, 어떤 탐지자가 손상되어도 전체 면역시스템에 영향을 주지 않는 결함 허용 능력을 제공한다.

○ 다양성 특징

개체군에 존재하는 각 개체들은 서로 다른 면역시스템을 가지고 있으므로 어떤 개체가 병원체에 침입을 받아 감염되었다 하더라도 다른 개체의 경우 동일한 병원체에 의해 감염될 확률이 동일하지 않다.

○ 새로운 외래 패턴에 민감하게 대응하는 특징

면역시스템은 병원체에 대해 1차·2차 대응 메커니즘이 존재한다. 즉, 1차 대응(primary response)을 통해 새로운 감염이 발생한 경우에 새로운 감염 병원체에 대응하는 새로운 탐지자들을 진화시켜 대응하고, 2차 대응(secondary response)을 통해 전에 감염된 정보와 동일한 병원체의 침입이 발생하면 이를 즉각적으로 대응한다.

○ 불완전한 탐지 특징

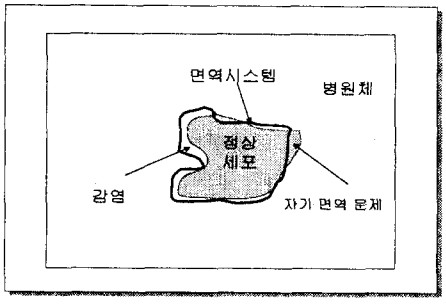
모든 병원체는 기존에 존재하는 탐지자에 의해 정확하게 일치되지 않는다. 이런 문제를 해결하기 위해 면역시스템은 1차 대응을 통한 학습과 분산탐지를 사용한다.

○ 병원체 탐지 및 대응

면역시스템은 물리적 화학적 작용에 의해 항원(antigen)에 탐지자(T cell, B cell, 그리고 항체로 구성됨)가 결합되는 증상에 의해 침입을 탐지(이를 부정적 탐지(negative detection)라 칭함)하고, 살균세포에 의해 항원을 공격하여 소멸시킨다.

○ 면역시스템의 탐지 범위

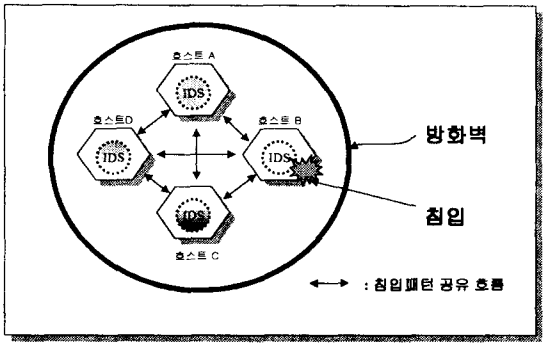
면역시스템은 탐지자들을 통해 병원체를 탐지하는데 병원체와 정상세포 모두 단백질로 구성되어 있어 <그림 2>와 같이 면역시스템이 병원체를 정상세포로 간주할 경우 병원체에 감염될 수 있고 정상세포를 병원체로 판단한 경우 자기면역문제(autoimmune disease)가 발생한다.



<그림 2> 면역시스템의 탐지 범위

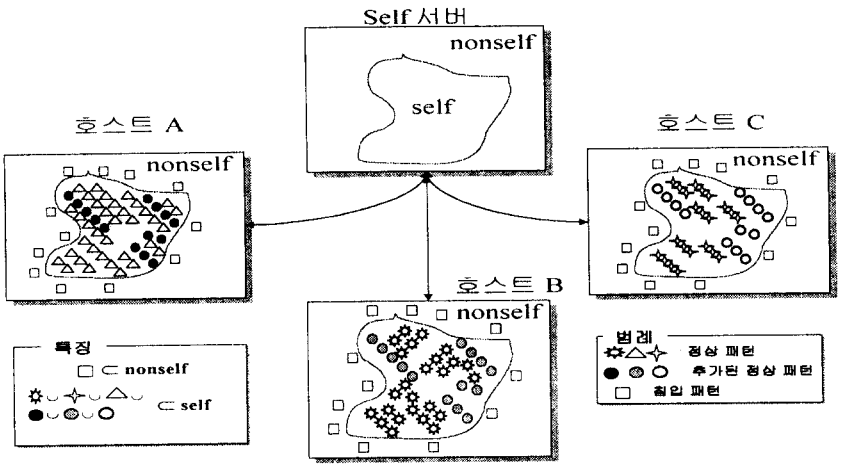
3.2 제안한 침입탐지시스템 모델

제안한 면역시스템 모델을 기반으로 한 침입탐지시스템은 <그림 3>에 도시된 바와 같이 네트워크를 통해 동질형의 여러 호스트에 분산된 침입탐지시스템(IDS)을 포함하고, 각각의 호스트는 자신의 침입탐지시스템을 통해 호스트에서 발생하는 이벤트들을 모니터링하면서 기설정된 정상 이벤트 패턴 정보에 따라 침입여부를 판단한다. 이때, 각각의 호스트에서 감시하는 대상은 모든 호스트에 존재하는 특권프로세스이며, 각 호스트는 상기 특권프로세스에 대한 비정상 이벤트를 공유하면서 새로운 침입으로부터 전체 시스템 면역력을 향상시킨다.



<그림 3> 제안한 침입탐지시스템

<그림 3>을 통해 이 과정을 살펴보면 침입자가 방화벽(내부공격인 경우 액세스 제어 시스템 또는 암호시스템)을 뚫고 호스트 B에 침입한 경우



<그림 4> 제안한 침입탐지 시스템에서 침입 패턴을 공유하는 개념

호스트 B에 존재하는 침입탐지시스템에 의해 침입이 탐지(즉, 자연면역시스템인 경우 1차 대응 과정)되고 탐지된 침입패턴을 인접한 호스트 A, C, D에 전달하여 호스트 A, C, D의 각각의 침입탐지시스템에 추가하여 추후에 추가된 침입패턴을 통해 동일한 침입을 즉각적으로 탐지(즉, 자연면역시스템인 경우 2차 대응 과정)할 수 있게 하여 침입으로부터 방화벽 내의 모든 호스트의 면역력을 향상시킨다.

제안한 침입탐지시스템에서 침입패턴을 공유하는 과정을 <그림 4>를 통해 상세하게 살펴보면 다음과 같다. self 서버는 모니터링하는 특권프로세스에 대한 정상행위 정보(self)를 가지고 있으며, 각 호스트의 침입탐지시스템은 상기 특권프로세스에 대해 호스트에서 발생하는 정상 행위를 수집하여 침입을 탐지한다. 이때, 상기 특권프로세스에 대한 각 호스트의 정상 행위 정보는 서로 다르므로 면역시스템의 다양성 성질을 제공한다. 각 호스트에서

<표 1> 자연면역시스템과 제안한 침입탐지시스템간의 대응 관계

구분	자연면역시스템	제안한 침입탐지시스템
보호막	피부, 점액막	방화벽, 액세스 제어 시스템, 암호시스템
Self	정상적인 세포	합법적인 사용자, 허가된 행동
Nonself	병원체(박테리아, 바이러스, 그리고 기생충)	침입자, 컴퓨터 바이러스, 트로이목마, 스푸핑
대응방법	병원체 소멸	프로세스 강제종료
Self/Nonself 구별자	펩티드	시스템 호출 패턴
대응(Ⅰ)	1차 대응	정상행위패턴을 이용한 긍정적 탐지를 통해 탐지자를 생성한 후 모든 IDS에 공유
대응(Ⅱ)	2차 대응	탐지자를 이용하여 부정적 탐지
탐지오류(Ⅰ)	감염	부정적 결함
탐지오류(Ⅱ)	자기면역문제 발생	긍정적 결함

수집된 상기 특권프로세스에 대한 정상 행위는 self 서버에 존재하는 정상행위 정보의 부분집합이다. 호스트에서 정상 행위를 수집한 후 호스트에 침입이 발생하면 침입탐지시스템은 자신의 정상행위 정보를 통해 침입여부를 판정(이와 같은 판정과정을 긍정적 탐지(positive detection)라 함)한 후 침입이라 판정되면 self 서버에 침입패턴을 전송한다. Self 서버는 전송된 침입패턴이 self의 정상행위 정보에 존재하지 않은 경우 모든 침입탐지시스템에 이를 전달하여 모든 침입탐지시스템이 침입패턴을 이용하여 부정적 탐지(negative detection)를 통해 침입을 탐지할 수 있게 한다. 제안한 침입탐지 시스템은 침입을 탐지한 경우 침입 행위를 계속할 수 없게 침입에 사용하는 프로세스를 강제로 종료시킨다.

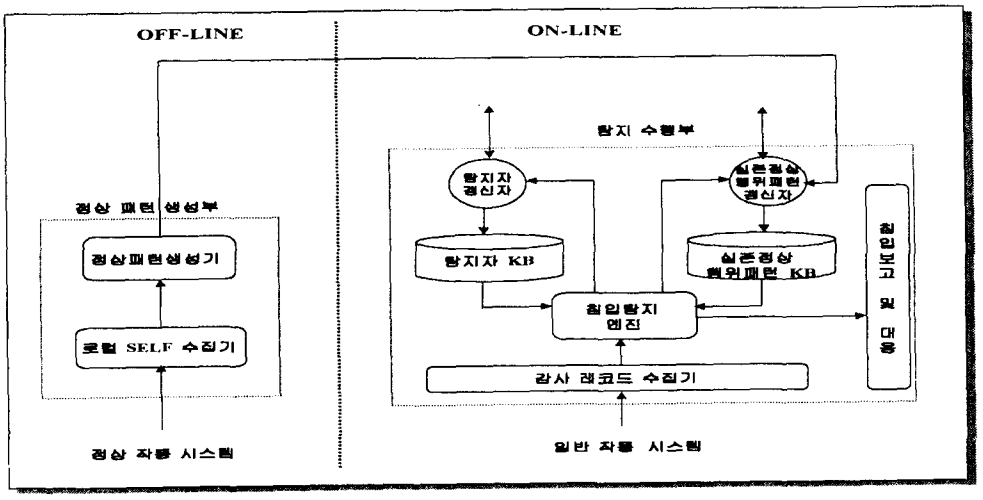
한편, 전술한 자연 면역 시스템과 제안한 침입탐지시스템간의 관계는 <표 1>과 같다.

4. 제안한 침입탐지시스템의 설계

제안한 시스템에서 각 컴퓨터의 침입탐지시스템은 공지된 침입에 대한 비정상행위 시스템 호출 패턴으로 구성된 탐지자KB와, 특권 프로세스에 의해 일정시간동안 발생한 시스템 호출 패턴을 수집하여 구성한 실존정상행위패턴KB를 포함하고, self 서버[1]는 동질형 시스템들로부터 침입탐지대상 프로세스가 정상적으로 수행하면서 생성한 시스템 호출 패턴을 문헌[26]에서 제시한 방법으로 수집하여 구성한 합성정상행위패턴KB을 포함한다.

4.1 시스템 구성

제안한 시스템에서 분산된 각각의 컴퓨터에 설치되는 침입탐지시스템의 구성 요소를 <그림 5>를 참조하여 살펴보면, 각 컴퓨터의 침입탐지시스템은 크게 오프라인으로 수행되는 정상패턴생성부와 온라인으로 수행되는 탐지수행부로 대별된다.



<그림 5> 각각의 컴퓨터에 대한 제안한 침입탐지시스템

1) Self 서버는 각각의 침입탐지시스템과 독립되어 존재하며 각각의 침입탐지시스템의 실존정상행위패턴KB의 합집합인 합성정상행위패턴KB를 포함한다.

4.1.1 정상패턴생성부

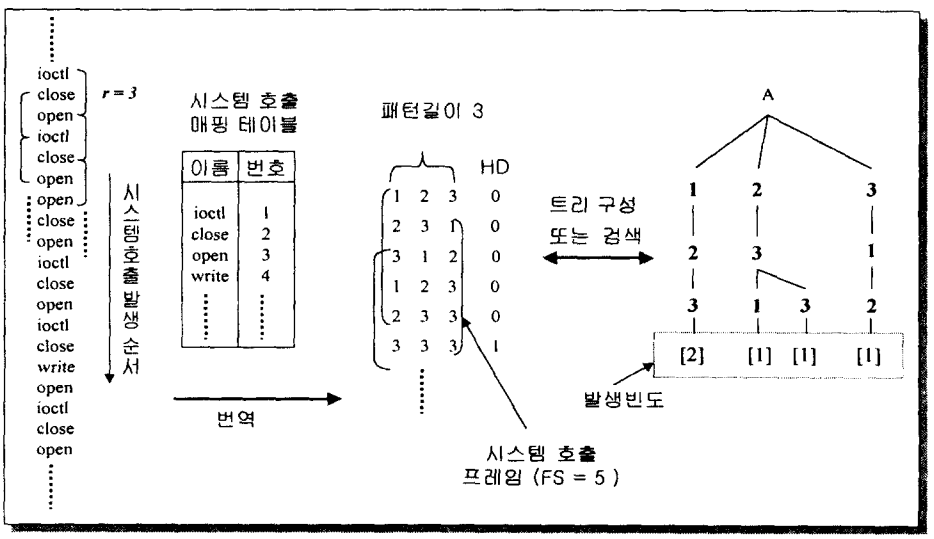
정상패턴생성부는 각 컴퓨터가 정상 상태, 즉 정상 사용자가 정상적인 수행을 할 때 발생하는 프로세스의 시스템 호출 순서를 로컬 self수집기를 통해 수집한 후, 패턴생성기를 통해 구성한다. 프로세스에 대한 시스템 호출 순서는 Solaris 2.6 BSM(Basic Security Module)의 감사서브시스템(audit subsystem)을 통해 구한다[19].

한편, 패턴생성기는 입력된 프로세스의 시스템 호출 순서를 r-contiguous-bits 방식[20]에 따라 r 크기단위로 분리하여 시스템 호출 패턴들을 트리로 표현한다. 프로세스 A에 의해 생성되는 시스템 호출에 대해 r을 3으로 하여 정상 시스템 호출 패턴 트리를 구성하는 것을 <그림 6>을 참조하여 살펴보면 다음과 같다. 먼저, 프로세스 A가 수행하면서 연속해서 시스템 호출을 생성하면, 생성되는 순서에 따라 시스템 호출 매핑 테이블에 시스템 호출 이름을 등록하고 번호를 부여한다. 계속해서 발생하는 시스템 호출들에 대해 시스템 호출 매핑 테이블을 통해 등록번호로 변경한 후 3개의 시스템 호출 이름 단위로 시간 축으로 시프팅하면서 시스템 호출 패턴들을 생성한다. 생성한 시스템 호출 패턴들은 자주 발생하는 패턴에 대해 추후 검색을 빠르게 하기 위해 패턴 발생 빈도에 따라 트리를 구성한다. 프로세스의 행위를 감시하는 과정도 <그림 6>과 같은 과정으로 수행하며 이에 대해서는 후술한다.

4.1.2 탐지수행부

탐지수행부는 공지된 침입 패턴 정보를 저장한 탐지자KB, 이를 관리하는 탐지자갱신자, 정상패턴생성부로부터 전달된 특권 프로세스의 정상행위 패턴을 저장한 실존정상행위패턴KB, 이를 관리하는 실존정상행위패턴갱신자, 감사서브시스템에서 제공하는 감사 레코드를 수집하는 감사레코드수집기, 수집된 감사레코드로부터 시스템 호출을 분리하여 해당되는 침입탐지부를 기동시켜 침입을 탐지하는 침입탐지엔진, 그리고 침입 발생을 알리고 해당 프로세스를 강제로 종료시키는 침입보고 및 대응부로 구성된다. 한편, 탐지자KB와 실존정상행위패턴KB에 저장된 시스템 호출 패턴은 <그림 6>과 같은 트리구조로 각각 저장된다.

침입탐지엔진을 보다 상세히 살펴보면, 감사레코드수집기를 통해 감사서브시스템에서 제공하는 감사 레코드를 입력받은 후, 할당자가 수집된 감사레코드에 특권 프로그램을 수행시키기 위한

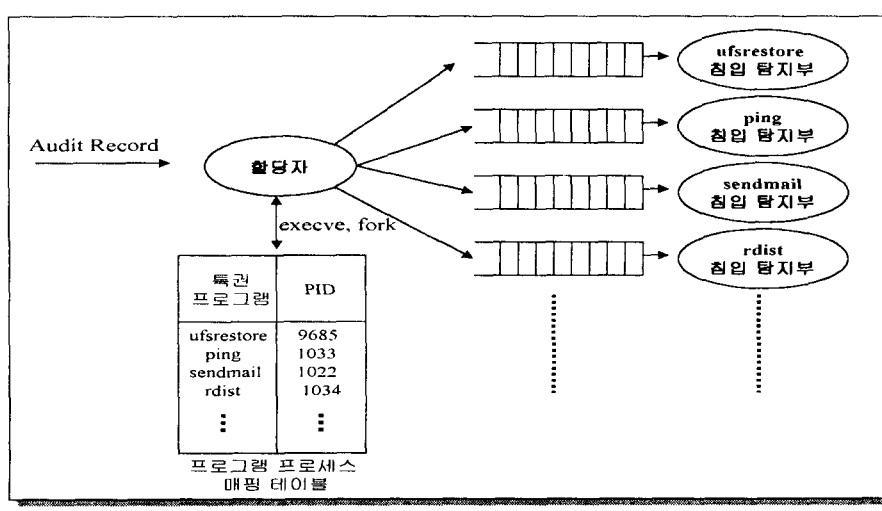


<그림 6> 프로세스 A에 대한 시스템 호출 패턴 생성 과정 또는 검색 과정

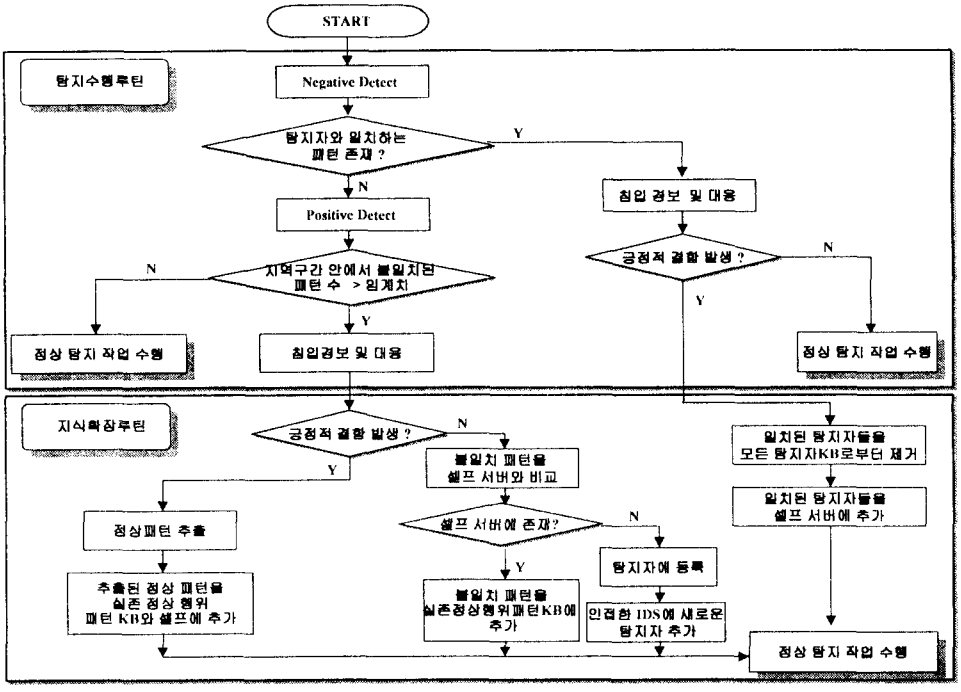
execve 시스템 호출이 있는지 조사하여, 존재하는 경우 프로그램 프로세스 매핑 테이블에 등록시키고 해당되는 침입탐지부를 기동시킨다. 한편, 수집된 감사레코드 중 fork 시스템 호출이 있는 경우, 부모 프로세스의 프로그램과 생성된 프로세스의 PID를 매핑시켜 해당 프로그램의 침입탐지부를 기동시킨다. 할당자는 입력되는 감사 레코드 중 <그림 7>과 같이 프로그램 프로세스 매핑 테이블에 등록된 PID에 해당하는 감사 레코드의 시스템 호출부분을 분리하여 침입탐지부에 전달하여 탐지를 수행한다.

전술한 각각의 침입탐지부는 <그림 8>의 동작 알고리즘에 도시된 바와 같이 탐지자KB를 이용한 반대측 탐지(negative detect)와 실존정상행위패턴KB를 이용한 긍정적 탐지(positive detect)를 수행하며, 임계치에 의해 경보를 발생하는 경우 탐지자 갱신자를 통해 불일치 패턴들을 self 서버에 전달 및 비교하여 현재 탐지된 패턴이 침입패턴인 경우 분산된 모든 탐지자KB를 갱신시켜 침입으로부터 분산된 모든 침입탐지 시스템의 면역력을 증가시킨다. 이때, 탐지자KB를 이용한 반대측 탐지(negative detect)는 자연면역시스템의 2차 대응(secondary response)과정을 모델링한 것이고, 실존정상행위패턴KB를 이용한 긍정적 탐지(positive detect)를 통해 침입을 탐지하여 침입 정보를 다른 호스트와 공유하는 것은 자연면역시스템의 1차 대응(primary response)을 모델링한 것이다.

이를 상세히 살펴보면, 외부로부터 전달된 침입패턴 정보 즉 탐지자를 이용하여 반대측 선택 방식(negative selection)에 따라 현재 프로세스가 발생하는 시스템 호출들을 감시하여 탐지자KB에 존재하는 시스템호출 패턴(즉, 탐지자)과 일치하는 시스템호출이 존재하면 이를 침입으로 간주하여 침입보고 및 대응부를 통해 시스템관리자에게 침입을 알리고 현재 침입에 사용된 프로세스를 종료시킨다. 이때, 만일 침입탐지부가 오판하여 긍정적 결함이 발생한 경우, 현재 수행중인 프로세스의 시스템 호출과 일치된 탐지자(들)를 분산된 모든 침입탐지시스템의 탐지자KB로부터 제거하고, 이를 self서버의 합성정상행위패턴KB에 추가한다. 침입탐지엔진은 반대측 탐지를 통해 프로세스가 발생하는 시스템 호출을 감시한 후, 상기 프로세스에 의해 발생된 시스템 호출과 실존정상행위패턴KB의 내용과 비교하여 상기 프로세스에 의해 발생된 시스템 호출 중 실존정상행위패턴KB에 존재하지 않는 패턴



<그림 7> 침입탐지 엔진



<그림 8> 제안한 침입탐지 시스템의 동작 알고리즘

(hamming distance[2])가 1이상)이 시스템 호출 프레임[3] 크기(FS) 안에서 임계치보다 많이 존재하면 이를 침입으로 간주하여 침입보고 및 대응부를 통해 시스템관리자에게 알리고, 현재 침입에 사용된 프로세스를 종료시킨다. 이때, HD 값과 FS 크기값 선정에 따라 탐지 시간과 탐지 정확도에 차이가 있으며 이에 대해서는 실험을 통해 후술한다. 임계치는 보안정책에 따라 조절할 수 있어, 임계치를 낮추면 보안강도가 높아지는 반면 긍정적 결함 발생확률이 증가하며, 임계치를 높이면 부정적 결함 발생확률이 증가한다.

한편, 침입탐지엔진은 침입 경보가 발생하고, 발생된 경보가 긍정적 결함이 아닌 경우 self 서버와 통신하면서 <그림 8>의 지식확장루틴을 수행한다. 이를 상세히 살펴보면, 침입탐지엔진은 FS 안에 HD값이 일정값 이상인 패턴들을 self 서버에 전송하여 합성정상행위패턴KB에 일치하는 패턴(들)의 존재 유무를 판단하여 패턴(들)이 존재하는 경우(즉, hamming distance가 0인 경우) 상기 패턴(들)을 실존정상행위패턴KB에 추가하여, 추후에 이와 동일한 패턴에 의해 긍정적 결함이 발생하는 것을 방지한다. 만일 합성정상행위패턴 KB에 일치하는 패턴(들)이 존재하지 않는 경우, hamming distance가 일정치 이상인 패턴(들)을 분산된 모든 컴퓨터의 탐지자KB에 추가하여 추후 이와 같은 시스템 호출 패턴을 발생하는 프로세스의 수행을 반대측 탐지 단계에서 빠르게 탐지할 수 있게 한다. 만일, 발생된 경보가 긍정적 결함인 경우 현재 수행 중인 프로세스의 로그 데이터를 분석하여 정상행위 패턴을 추출하여 추출된 정상행위 패턴을 실존정상행위패턴KB와 self서버의 합성정상행위패턴 KB에 추가한다. <그림 8>의 제안한 침입탐지 시

2) hamming distance(HD)는 문자열의 일치 여부를 판단하는 방법으로 예를 들어 "1 2 3 6 2"와 "7 2 3 5 2"의 HD는 2가 된다.

3) <그림 6>에 도시된 바와 같이 입력되는 시스템 호출 패턴들과 정상행위패턴KB와의 일치 정도를 비교하는 창.

스템의 동작 알고리즘 중 시스템 호출 프레임 크기(FS)안에서 불일치된 패턴 수와 임계치를 비교하는 것을 <그림 6>을 참조하여 살펴보면 시스템 호출 프레임 크기(FS) 만큼 입력된 각각의 시스템 호출 패턴의 hamming distance가 C보다 큰 패턴들의 개수가 임계치를 넘는지를 판단한다.

즉,

$$\text{IF} \quad \sum_{i=1}^{FS} \{ HD_{\min}(i) \geq C \} > \text{임계치} \quad \text{THEN}$$

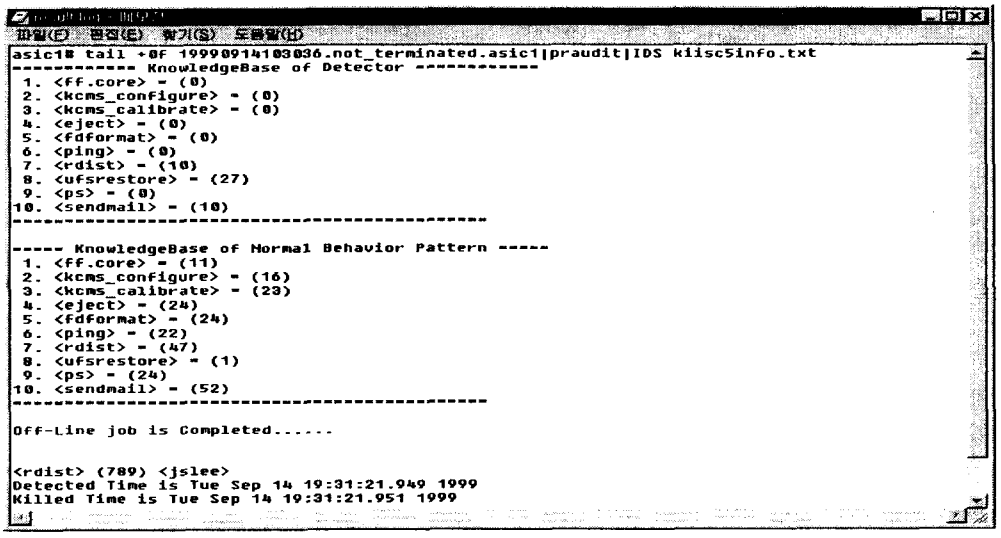
침입

5. 프로토타입 구현 및 성능 평가

5.1 구현 환경

제안한 모델에 대한 프로토타입은 Solaris 2.6환경에서 C++ 언어를 사용하여 구현하였고, SunOS BSM의 서브감사시스템을 사용하였으며, 탐지대상 프로그램으로는 다중 호스트 상에서 복사 파일들을 동일하게 유지하기 위한 유닉스 유틸리티인 rdist [21](Remote File Distribute Program)로 하였다. rdist프로그램을 탐지대상으로 사용한 이유는 setuid 프로그램이고 시스템 관리자와 일반사용자에 의해 많이 사용되고 있으며, rdist를 이용한 버퍼오버플로우공격 예[17]가 존재하기 때문이다.

<그림 9>를 통해 UltraSparc10 333㎒ Solaris 2.6에 설치된 제안된 모델에 대한 일부분의 프로토타입에 의해 rdist 프로그램을 이용한 버퍼오버플로우 공격[17]이 발생한 경우 이를 탐지 및 대응하는 것을 살펴보면 다음과 같다. 도시된 바와 같이, 제안한 모델의 프로토타입은 탐지대상 프로그램들에 대한 각각의 탐지자KB와 실존정상행위패턴KB를 구성한 후, 탐지를 수행하던 중 rdist 프로그램을 수행시키는 프로세스의 행위가 비정상인 경우, 이를 침입이라 판정하고 침입에 사용된 프로그램 이름과 이 프로그램을 기동시킨 프로세스 번호 및 프로세스 소유자 ID, 그리고 탐지 시간에 관한 정보를 출력한 후, 현재 침입에 사용된 프로세스를 강제로 종료시켜 침입에 대응한다. <그림 9>를 통해 프로토타입은 10개의 탐지대상 프로그램을 감시하고, self 서버로부터 rdist, ufsrestore, sendmail에 대한 새로운 침입 탐지자패턴을 전달받았음을 알 수 있다.



<그림 9> 프로토타입에서 rdist 프로그램을 이용한 공격 탐지 예

5.2 성능 평가

제안한 시스템의 성능을 평가하기 위해 <표 2>의 조건으로 rdist 프로그램을 이용한 버퍼오버플로우 공격을 예로 침입탐지시간과 탐지정확도에 대하여 살펴본다.

<표 2> 탐지대상 시스템 조건

조건 / 경우	패턴 길이	동시 수행 프로그램 수	rdist 무한정 수행	FS=10		FS=20		탐지자 FS=30
				HD	임계치	HD	임계치	
a	9	5	×	6	6	6	14	HD=7
b	5	5	×	3	6	3	14	HD=4
c	9	10	×	6	6	6	14	HD=7
d	5	10	×	3	6	3	14	HD=4
e	9	0	○					HD=7

※ ○ : rdist를 무한정 수행함,
× : rdist를 무한정 수행 안 함

성능평가에 사용되는 프로토타입은 rdist 프로그램을 포함한 67개의 setuid 프로그램의 행위를 감시한다. <표 2>의 조건을 바탕으로 시스템 성능 평가의도를 살펴보면, 먼저 패턴길이를 가변시켜 패턴길이에 따른 탐지시간의 의존도를 고찰하고, 표준 출력을 반복 수행하여 무한정으로 시스템 호출을 발생하는 I/O bound 프로그램(TestPGM I)들을 여러 개 동시에 수행시켜 이를 통해 컴퓨터의 부하에 따라 제안한 침입탐지시스템의 탐지 능력을 평가한다. 또한, rdist 프로그램을 무한정 기동시켜 제안한 모델에서 탐지 대상 프로그램이 무한정 발생할 때 제안한 알고리즘의 처리 부하에 따른 탐지 능력을 평가한다.

한편, 설정된 시스템 호출 프레임 크기(FS) 내의 시스템 호출 패턴들 중 실존정상행위패턴KB에 대한 hamming distance가 일정치(즉, HD) 이상인 패턴이 임계치 이상인 경우 침입으로 판정하는 긍정적 탐지(positive detect)방식에 따라 <표 2>와 같은 값으로 프로토타입에 적용한 경우에 침입 탐지시간 및 정확도를 평가[4]하고, 합성정상행위패턴KB와 hamming distance 값이 일정치(<표 2>의 경우 7 또는 4)인 탐지자를 사용한 부정적 탐지(negative detect)방식과 FS을 30으로 한 긍정적 탐지 방식을 혼용한 제안한 침입탐지방법을 프로토타입에 적용한 경우에 침입 탐지시간 및 정확도를 평가한다. 이를 통해 단지 긍정적 탐지를 수행하는 Forrest[11]과 Debar[24]의 침입탐지시스템 모델과 본 논문에서 제안한 모델과의 성능을 평가한다.

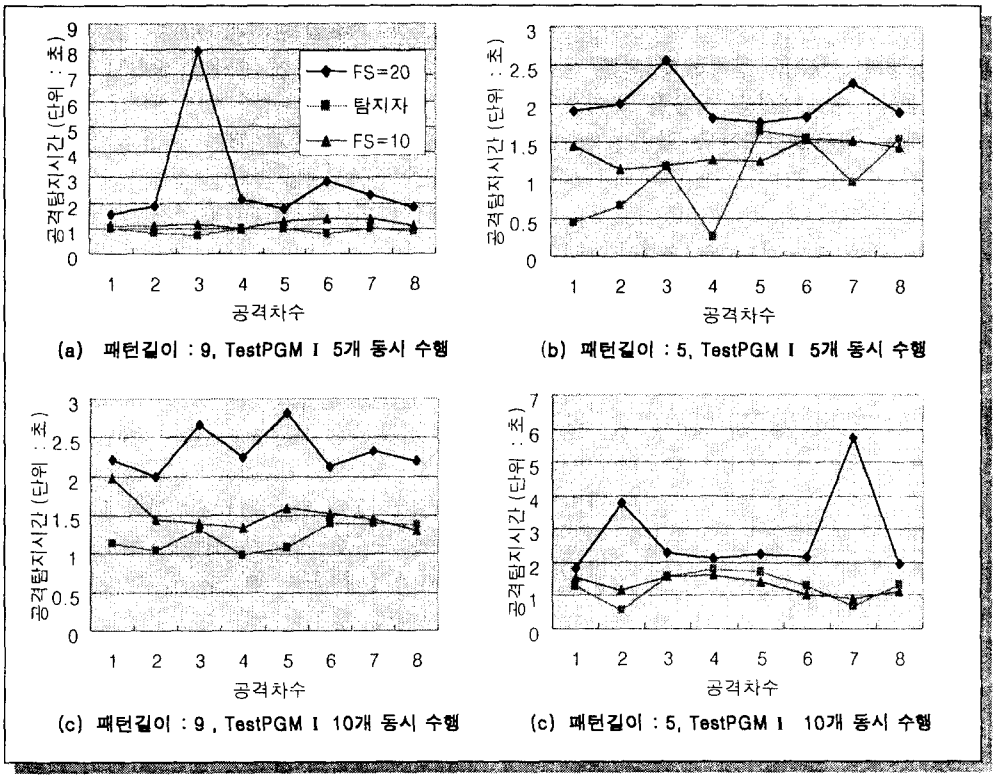
5.2.1 탐지시간 관점

<그림 10>(a)~(d)를 통해 <표 2>의 a~d 경우에 따른 탐지시간을 살펴보면, 시스템 호출 프레임 크기(FS)가 20인 경우 FS가 10인 경우와 탐지자를 사용하는 경우보다 탐지 판단시간이 많이 소비됨을 알 수 있으며, <그림 10>(a)과 <그림 10>(b)을 통해서 컴퓨터의 부하가 적게 걸려있을 경우에는 패턴 길이가 침입을 탐지하는데 큰 영향을 미치지 않음을 알 수 있고, 동시수행프로그램 수에 따라 탐지시간도 <그림 10>(a),(c)와 <그림 10>(b),(d)를 통해서 큰 영향이 없음을 알 수 있으며, 패턴 길이가 길고 동시 수행 프로그램 수가 많을 경우 <그림 10>(c)을 통해서 알 수 있듯이 탐지자를 이용하여 빠르게 침입을 탐지할 수 있다.

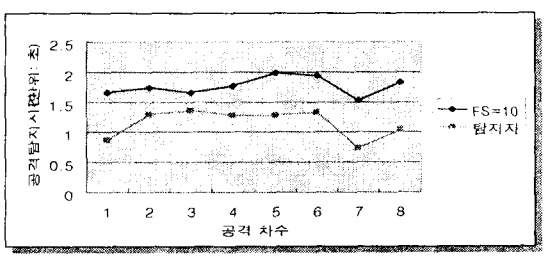
<그림 11>를 통해 <표 2>의 e 경우처럼 rdist 프로그램을 무한정 기동시키는 동안 rdist 버퍼오버플로우 공격을 수행할 때 탐지시간을 살펴보면, 이 경우 탐지자를 사용하는 것이 FS를 10으로 한 긍정적 탐지보다 탐지시간이 빠른 것을 알 수 있다.

결과적으로, 전체적인 탐지시간은 시스템 환경과 운영체제의 메모리 관리 방법 등 여러 가지 요인에 의해 공격 시도 차수에 따라 다르므로 큰 신뢰성을 갖기 힘드나, 자주 사용하는 탐지자 프로그램을 탐지할 경우와 패턴 길이가 길고 동시 수행 프로그램 수가 많을 경우 탐지자를 이용한 부정적 탐지를 수행하는 제안한 모델이 단지 긍정적 탐지를 수행하는 Forrest[11]과 Debar[24]의 침입탐지시스템 모델보다 탐지 시간 측면에서 빠르다는 것을 알 수 있다.

4) Forrest[11]과 Debar[24]의 침입탐지시스템 모델에 대한 성능 평가를 의미함.



<그림 10> 4가지 경우에 따른 침입탐지시간 측정 결과



<그림 11> <표 2>의 e 경우에 따른 침입탐지시간 측정 결과

5.2.2 탐지정확도 관점

본 절에서는 침입탐지시스템의 성능에 있어 중요한 요소 중 하나인 탐지 정확도를 측정한다. 이를 위해 정상적으로 수행되는 rdist 프로그램에 의해 생성된 시스템 호출을 패턴 길이 9로 나누어 실존정상행위패턴KB를 구성한다. 이때, rdist 명령에 대한 실존정상행위패턴 수는 59개이다. 이처럼 rdist 명령에 대한 실존정상행위패턴KB를 구성한 후 <표 3>과 같이 rdist 명령에 여러 옵션을 부가하여 각각 수행할 때 긍정적 결함(false positive) 발생 유무를 실험한다.

<표 3>을 통해 실험결과를 살펴보면, FS가 짧을 수록 오판할 확률이 높으며, 제안한 시스템에서 FS를 30으로 하고 탐지자를 사용할 경우 긍정적 결함(false positive)이 발생하지 않음을 알 수 있다. 한편, 탐지자를 사용하지 않고 패턴 길이와 FS를 각각 9, 30으로 하고, hamming distance 값이 8 이상인 패턴의 개수를 20으로 하여 침입판단 기준을 높인 경우 긍정적 결함(false positive)은 발생하지 않으나 침입 발생시 탐지를 못하는 부정적 결함(false negative)이 발생한다. 이때, 만일 상기 침입에 대한 시스템호출 패턴을 다른 침입탐지시스템으로부터 전달받아 탐지자KB를 구축하여 <표

3>과 같이 침입 탐지자를 사용할 경우 상기 침입을 정확하게 탐지할 수 있다. 즉, 다른 침입탐지시스템으로부터 침입 패턴 정보를 수신하여 침입에 대한 면역력을 향상시켜 이와 동일한 침입을 받으면 이를 정확하게 탐지할 수 있다. 따라서, 탐지자 정보를 공유하여 부정적 탐지를 수행하는 제안한 모델이 단지 긍정적 탐지를 수행하는 Forrest[11]과 Debar[24]의 침입탐지시스템 모델보다 탐지 정확도가 높음을 알 수 있다.

<표 3> rdist 명령 옵션에 따라 false positive 발생 유무 점검

탐지방법 / rdist 명령종류	FS = 10	FS = 20	탐지자 사용 FS = 30
rdist -b	○	○	×
rdist -D	×	×	×
rdist -h	○	×	×
rdist -i	○	×	×
rdist -n	×	×	×
rdist -q	○	×	×
rdist -R	○	×	×
rdist -v	○	×	×
rdist -w	○	×	×
rdist -y	○	×	×
rdist -f	○	×	×
rdist -m	×	×	×

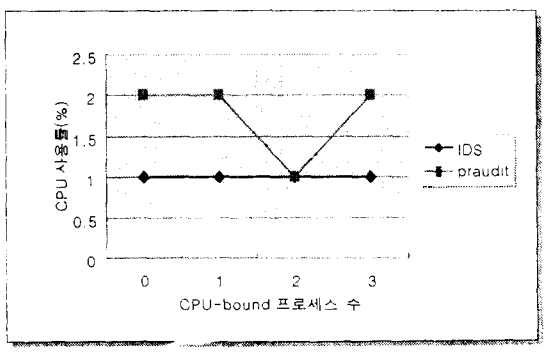
※ ○ : false positive 발생,
× : false positive 발생 무

5.2.3 다양성 관점

제안한 모델에서 분산된 각각의 침입탐지시스템들은 동일한 특권 프로그램에 대해 각기 다른 침입판단 기준을 갖고 있으므로 어떤 침입탐지시스템은 침입을 탐지하지 못하였지만, 다른 침입탐지시스템은 동일한 침입공격에 대해 침입을 탐지할 수 있다. 예를 들어, 어떤 하나의 IDS에서 긍정적 결함(false positive)을 줄이기 위해 침입판단 기준을 높인 경우 침입을 탐지하지 못할 수 있으나, 반면 다른 IDS에서 엄격한 침입탐지를 위해 긍정적 결함(false positive)을 감수하면서 침입판단 기준을 낮춘 경우 침입을 탐지할 수 있다. 따라서, 제안한 모델은 A 침입탐지시스템이 설치된 호스트에 침입이 성공한 경우, 동일한 방법을 적용하여 B 호스트를 침입할 수 있다고 볼 수 없는 면역시스템의 특징인 다양성 성질을 갖고 있다.

5.2.4 시스템 오버헤드 관점

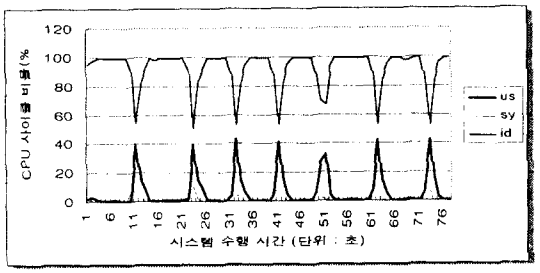
본 절에서는 제안한 모델의 프로토타입을 통해 제안한 침입탐지시스템에 의해 발생하는 시스템 오버헤드를 측정한다. 일반적으로 시스템 오버헤드 또는 시스템 성능을 측정할 때 가장 많이 고려되는 요소로 CPU 사용량을 사용하므로 여기서는 이를 통해 제안한 모델에 대한 프로토타입의 시스템 오버헤드를 측정한다. 이를 위해 Solaris 2.6에서 제공하는 ps 명령을 통해 제안한 침입탐지시스템에서 사용하는 프로세스들(IDS, praudit)에 대한 CPU 사용량을 측정하고, vmstat 명령을 통해 1초 간격으로 전체 프로세스들에 대한 전체 CPU 사용량을 체크하여 침입이 발생할 때 CPU 사용량의 증가 상태와 탐지작업을 수행할 때 전체 시스템 오버헤드를 측정한다. praudit 프로세스는 생성되는 감사 데이터를 감사 레코드로 출력하는 서브감사시스템을 구성하는 요소이다.



<그림 12> CPU bound한 프로그램에 따른 IDS와 praudit의 CPU 사용률

먼저, CPU bound한 프로그램에 따라 제안한 모델에 대한 시스템 오버헤드를 측정하면 <그림 12>과 같다. <그림 12>을 통해 알 수 있는 것은 제안 모델은 CPU bound 프로세스 수와 무관하게 일정량의 CPU 사용률을 가지며 praudit 프로세스가 IDS 프로세스 보다 1배 정도의 CPU 사용률을

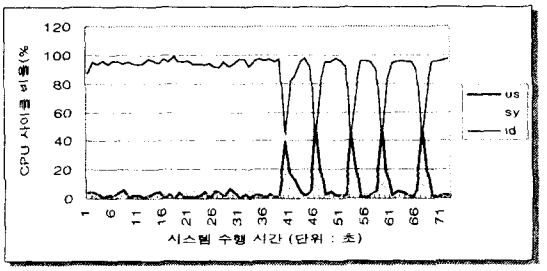
가지므로 IDS 프로세스의 CPU 오버헤드가 Solaris 2.6에서 제공하는 praudit 프로세스의 CPU 오버헤드보다 적음을 알 수 있다.



<그림 13> 제안한 침입탐지시스템의 CPU 사용률

시스템에서 단 하나의 사용자 프로세스도 수행하지 않고 제안한 침입탐지시스템을 수행할 때 CPU 사용량과 침입이 발생할 때 CPU 사용량의 증가 상태를 <그림 13>을 통해 살펴보면 다음과 같다. 이 측정을 위해 사용된 유틸리티는 vmstat 명령으로 이 명령이 보여주는 많은 시스템 통계치 중에 CPU 활동 상황 정보를 이용한다. 즉, us, sy, 그리고 id로, us는 사용자 프로세스를 위해 사용된 CPU 사이클의 비율을 의미하며, sy는 시스템 부하로 소비되는 총 CPU 사이클의 비율, 그리고 id는 사용되지 않는 CPU 사이클의 비율을 의미한다. <그림 13>를 통해 알 수 있는 것은 제안한 침입탐지시스템은 탐지 대상시스템으로부터 행위를 탐지하는 동안 CPU 사용률이 거의 없으며 침입 발생시 CPU 사용률이 증가하나 그 기간이 짧음을 알 수 있다.

한편, 시스템에서 I/O bound 프로그램 5개를 동시에 수행시켜 많은 양의 감사 데이터가 발생하는 동안 제안한 침입탐지시스템을 수행할 때 CPU 사용량과 침입이 발생할 때 CPU 사용량의 증가 상태를 <그림 14>를 통해 살펴보면 다음과 같다. 제안한 침입탐지시스템은 탐지 대상시스템으로부터 행위를 탐지하는 동안 CPU 사용률이 1내지 4 정도이며 침입 발생시 CPU 사용률이 증가하나 그 기간이 짧음을 알 수 있다.



<그림 14> I/O bound 프로그램을 5개 동시에 수행할 때 제안한 침입탐지시스템의 CPU 사용률

결론적으로, <그림 12>을 통해 제안한 침입탐지시스템이 CPU bound 프로세스 수와 무관하게 일정량의 CPU 사용률을 가지며 Solaris 2.6에서 제공하는 praudit 프로세스의 CPU 오버헤드보다 IDS 프로세스의 CPU 오버헤드가 적음을 알 수 있었다. 또한, <그림 13>와 <그림 14>를 통해 시스템에 제안한 침입탐지시스템을 설치할 경우 많은 오버헤드가 발생하지 않음을 알 수 있었다. 물론, 낮은 오버헤드가 빠른 침입탐지를 의미하지는 않는다. 이는 CPU의 이용 가능성이 전체 시스템 성능에 영향을 주는 단 하나의 요인이기 때문이다. 그러나 이와 같은 측정결과는 제안한 침입탐지시스템을 사용하더라도 눈으로 분간할 수 있을 정도로 시스템 부하가 생기지 않으므로 실제 사용에 대한 타당성을 입증한다.

6. 결론 및 향후 연구과제

본 논문에서는 탐지 대상을 특권 프로세스로 하고, 특권 프로세스(privilege process)가 수행될 때 발생하는 시스템 호출 순서 중 비정상적인 시스템 호출을 탐지하여 이를 분산된 각각의 침입탐지 시스템들이 서로 동적으로 공유하여 침입에 대한 면역력을 향상시키는 면역 시스템 모델을 기반으로 한 침입탐지시스템을 설계하고 프로토타입을 구현하여, 이를 통해 제안한 침입탐지시스템의 접근 타당성을 고찰하였으며, 탐지시간, 탐지정확도, 면역

시스템의 특징인 다양성, 그리고 시스템 오버헤드 관점에서 제안한 침입탐지시스템의 성능을 평가하였다.

성능 평가 결과, 제안한 시스템을 사용할 경우 인접한 침입탐지시스템간에 탐지자 정보를 공유하므로 각 컴퓨터의 침입에 대해 단지 긍정적 탐지를 수행하는 Forrest[11]과 Debar[24]의 침입탐지시스템 모델보다 빠르고 정확하게 침입을 탐지 및 대응하여, 전체 컴퓨터 시스템들의 면역력을 향상시킴을 알 수 있으며, 제안한 모델이 면역시스템의 특징인 다양성을 제공함을 알 수 있었다. 따라서 제안한 IDS모델은 인접 IDS가 공격을 받으면 받을 수록 전체 IDS 시스템의 면역력이 향상되므로 새로운 침입을 효과적으로 방지할 수 있다.

그러나, 다른 침입탐지 접근과 같이 제안한 침입탐지 시스템도 모든 침입탐지 문제를 다 해결하지는 못한다. 예를 들어, 시스템 호출의 변화가 없는 공격인 경우 침입을 탐지하지 못한다. 또한, 감사서브시스템으로부터 제공되는 감사 데이터가 공격자에 의해 손상되었으면 제안한 침입탐지 시스템은 침입을 탐지할 수 없다. 한편, self 서버에 저장된 특권 프로세스에 대한 합성정상행위패턴KB를 구성할 때 특권 프로세스가 가질 수 있는 거의 모든 시스템 호출 패턴 정보를 구하여 구성해야 한다. 특권 프로세스에 대한 완벽한 실존정상행위패턴KB를 구성하는 문제는 비정상적 침입탐지 시스템 연구분야에 공통된 문제로 문헌[26]에서 제시한 합성정상행위패턴KB 구성 방법에 의해 이상적으로 특권 프로세스에 대한 거의 모든 정상 행위 패턴이 합성정상행위패턴KB에 구성된다고 가정하였다.

향후 연구과제는 특권 프로세스의 모든 정상 행위에 대한 시스템 호출 패턴들을 관리하는 self 서버의 효율적인 구축과, 현재 구현된 프로토타입을 전체 분산시스템으로 확장 구현하여 ftp와 같이 많은 사용자들이 사용하는 프로그램들을 탐지대상으로 적용하고, 현재 동질형 호스트간에 제안한 모델을 적용하였으나 이기종간의 감사 데이터 표준화를 통해 제안한 모델을 이기종 환경에 확장시키는 연구가 필요하다.

참고문헌

[1] James Cannady, Jay Harrell, "A Comparative Analysis of Current Intrusion Detection Technologies", http://iw.gtri.gatech.edu/Papers/ids_rev.html 1998.2.

[2] Mansour Esmaili, Rei Safavi-Naini, "Case-Based Reasoning for Intrusion Detection", *Computer Security Applications Conference,* 1996, PP.214-222.

[3] Jai Sundar B. Spafford E, "Software Agents for Intrusion Detection", Technical Report, Purdue University, Department of Computer Science, 1997.

[4] 이종성, 채수환, "분산 침입 탐지 에이전트를 기반으로 한 지능형 침입탐지시스템 설계", 「한국정보처리학회 논문지」 제6권 제5호, 1999년 5월, pp.1332-1341.

[5] 은유진, 박정호, "침입탐지 기술 분류 및 기술적 구성요소," 「정보보호센터 정보보호뉴스」, 통권 13호, 1998년 7월, pp.10-13.

[6] Crosbie M, Spafford E, "Applying Genetic Programming to Intrusion Detection", Technical Report, Purdue University, Department of Computer Science, 1996.

[7] Paul Helman and Gunar Liepins, "Statistical foundations of audit trail analysis for the detection of computer misuse", *IEEE Transactions on Software Engineering*, 19(9), September, 1993, PP. 886-901.

[8] H.S. Vaccaro and G.E. Liepins, "Detection of anomalous computer session activity", *In Proceedings of the 1989 IEEE Symposium on Research in Security and Privacy*, 1989, PP.280-289.

[9] Cheri Dowell and Paul Ramstedt, "The ComputerWatch data reduction tool", *In Proceedings of the 13th National Computer Security Conference*, Washington, DC,

October 1990, PP.99-108

[10] Paul Spirakis et al, "SECURENET : A network-oriented intelligent intrusion prevention and detection system", *Network Security Journal*, 1(1), November 1994.

[11] S. A. Hofmeyr, A. Somayaji, and S. Forrest. "Lightweight Intrusion Detection for Networked Operating Systems", Journal of Computer Security, Vol. 6, 1998, PP.151-180.

[12] A. Somayaji, S. Hofmeyr, and S. Forrest, "Principles of a Computer Immune System," *New Security Paradigms Workshop*, september, 1997

[13] Calvin Cheuk Wang Ko, *Execution Monitoring of security-critical programs in a distributed system : A specification-based approach*. PhD thesis, Department of Computer Science, University of California DAVIS, 1996.

[14] 정진욱, 안성진,「UNIX 프로그램밍 기술 -SVR4 시스템 프로그래밍의 이론과 실제-」, 컴퓨터출판, 1996.

[15] Sun Security Bulletin #00169, 1998/4/28 http://www.certcc.or.kr/advisory/ka98/ka98-65.txt

[16] http://www.rootshell.com/archive-j457nxiqi3gq59dv/199707/psrace.c.html

[17] http://161.53.42.3/~crv/security/bugs/SunOS/rdist6.html

[18] http://www.rootshell.com/archive-j457nxiqi3gq59dv/199807/solaris-sendmail-8.8.4.sh.html

[19] SunSoft, Mountain View, Californina, *SunSHIELD Basic Security Module Guide*, 1995

[20] Kosoresow AP, S. Hofmeyr, "Intrusion Detection via System Call Traces", *IEEE Software*, V.14 N.5, 1997.9, pp.35-42.

[21] Sun Microsystem, 「Man Pages: Rdist - remote file distribution program」, November 1993.

[22] 이종성, 채수환, "컴퓨터 면역 시스템을 기반으로 한 침입탐지 시스템 설계", 「1999 한국정보과학회 봄 학술발표논문집」 Vol. 26. No.1, 1999년 4월, pp.236-238.

[23] 이종성, 채수환, 박종서, 지승도, 이종근, 이장세, "침입탐지 기술 동향", 「한국통신학회 학회지」 Vo.16, No.11, 1999년 11월, pp.1320-1337.

[24] Debar, H., Dacier, M., Nassehi, M. and Wespi, A., "Fixed vs. Variable-Length Patterns for Detecting Suspicious Process Behavior", Research Report, RZ 3012 IBM Zurich Research Laboratory, 1998.

[25] 이종성, 채수환, "컴퓨터 면역 시스템을 기반으로 한 지능형 침입탐지시스템", 「한국정보처리학회 논문지」 제6권 제12호, 1999년 12월.

[26] 이종성, 채수환, "특권 프로세스의 시스템 호출 추적을 사용하는 침입탐지시스템 설계 : 면역 시스템 접근", 「한국정보보호센터 '99 정보보호 우수논문집」 1999년 12월, pp.181-206.

저자소개

이종성 (E-mail : jslee@kisa.or.kr)
1994년 한국 항공대학교 전자계산학과 졸업(이학사)
1996년 한국 항공대학교 전자계산학과 대학원 졸업(이학 석사)
1998년~1999년 : 국립 순천대학교 시간강사, 현대전자연수원 시간강사
2000년 한국 항공대학교 컴퓨터공학과 졸업예정(공학박사)
1999년~현재 : 한국정보보호센터 개발부 선임연구원
관심분야 : 컴퓨터보안, 침입탐지시스템, 병렬/분산처리, High Performance Computing, 등임

채수환
1973년 한국 항공대학교 항공전자공학과 졸업(공학사)
1985년 미국 Univ. of Alabama 전자계산학과 졸업(공학석사)
1988년 미국 Univ. of Alabama 전기공학과 졸업(공학박사)
1973년~1977년 공군교육사령부 통신학교 교관
1977년~1983년 금성통신 근무(연구원)
1996년 9월~1997년 8월 영국 Newcastle upon tyne 대학교 교환교수
1989년~현재 한국항공대학교 컴퓨터공학과 교수
관심분야 : 컴퓨터 구조, 병렬처리시스템, 컴퓨터 보안 등임