자기확장 모니터링 기반의 침입자동대응 시스템*

(Automatic Intrusion Response System based on a Self -Extension Monitoring)

요 약 차세대 정보전에서는 자신의 정보 시스템에 대한 침해방지, 복구 등의 수동적인 형태의 보호뿐만 아니라 상대방의 정보 기반구조(Information Infrastructure)에 대한 공격과 같은 적극적인 형태의 보호가 요구된다. 침입이 발생함과 동시에 시스템에 대한 피해를 최소화하고 침입자 추적 등의 즉각적인 대응을 하기 위해 정보보호시스템이 인간의 개입없이 자동적으로 대응하는 기능을 제공할 필요가 있다. 본 논문에서는 자기확장 모니터링 기법과 이를 기반으로 설계된 침입자동대응 모델을 제시한다. 또한 제안된 모델에 의해 설계, 구현된 침입자동대응 시스템인 ARTEMIS(Advanced Realtime Emergency Management Identification System)를 소개한다. 자기보호 기능을 가진 모니터링과 복제를 이용한 자기확장 모니터링은 모니터링 정보수집과 침입자 추적에 대한 공간적인 제약을 최소화하여 침입탐지와 침입자 추적의 정확도를 높인다.

ABSTRACT In the coming age of information warfare, information security patterns take on a more offensive than defensive stance. It is necessary to develop an active form of offensive approach to security protection in order to guard vital information infrastructures and thwart hackers. Information security products need to support an automatic response facility without human intervention in order to minimize damage to the attacked system and cope with the intrusion immediately. This paper presents an automatic intrusion response model which is developed on a Self-Extension Monitoring. It also proposes an ARTEMIS(Advanced Realtime Emergency Management and Intruder Identification System), which is designed and implemented based on the suggested model. The Self-Extension Monitoring using self-protection and replication minimizes spatial limitations on collection of monitoring information and intruder tracing. It enhances the accuracy of intrusion detection and tracing.

1. 서 론

네트워크를 통한 침입을 탐지하기 위해 상당한 연구가 이루어졌다. 하지만 탐지만으로는 급속도로 성장하는 네트워크를 위협하는 모든 요소를 막을 수 없다. 침입에 의한 피해를 최소화하기 위해 모니터링을 통해 보안관련 정보를 수집하고 침입을 실시간으로 탐지, 분석하여 자동적으로 대응할 수 있는 메커니즘이 요구된다[1].

침입탐지 및 대응 기능을 제공하는 여러 제품들이 개발되었다. EMERALD[2], Active Security Products[3], IDIP[1]등이 대표적인 시스템들이다. 이들은 다음과 같은 요구사항을 만족시키지 못한다. 첫째, 모두 지역적으로

* 본 연구는 한국소프트웨어진흥원 대학 정보통신 연구센터 육성, 지원사업과 BK21 지역대학육성사업 정보기술인력양성사업단의 일부 지원을 받아 수행되었습니다.

제한된 범위 내에서 정보를 수집하고 침입을 탐지하며 그에 대한 대응을 수행한다. 그러므로 침입자가 여러 네트워크를 거쳐서 침입한 경우 지역적인 정보의 수집과 대응으로는 침입의 근원지를 밝히기 어렵다. 둘째, 침입에 대해 자동적으로 대응하지 못하므로 침입자에게 네트워크에서 그들의 영역을 넓힐 시간을 제공한다. 셋째, 공격을 시작한 호스트를 확인하지 못하므로 침입에 대한 적절한 대응을 지원할 수 없다.

위에서 기술한 요구조건을 만족하는 새로운 침입자동대응 구조가 요구된다. 본 논문에서는 침입자동대응을 위한 메커니즘으로서 자기확장 모니터링 방안[4]을 제시한다. 자기확장 모니터링은 복제와 자기보호 기능을 기반으로 부정행위자에 대한 행위정보와 이동경로를 수집하여 최종적으로 신분을 확인한다. 모니터링 행위 자체를 보호하면서 부정행위자에 대한 추적영역을 동적으로 확장하므로 폭넓은 정보를 수집하는 충분한 시간과 공간을 확보할 수 있다. 제안하는 침입대응모델은 이러한 메커니즘을 기반으로 부정행위자에 대한 정보수집, 침입탐지, 추적과 같은 대응을 수행한다. 그러므로 모든 네트워크 컴포넌트를 같은 보안 스킴의 일부로 만드는 것이 가능하여 자동대응에 대한 시스템 기능을 향상시킨다. 이는 또한 공격받은 네트워크를 통한 침입자에 대한 추적, 타겟상의 대응, 침입자에 대한 대응, 공격에 대한 네트워크/호스트 증거 수집과 같은 침입탐지기능을 향상시킨다. 자기확장 모니터링의 범위를 단일 관리자 영역 또는 서로 신뢰관계에 있는 여러 관리자 영역으로 제한함으로써 관리 권한이 없는 호스트에 보안관리를 위한 프로그램을 복제하는 과정을 합법화한다. 본 논문에서 언급하는 모든 행위는 자기확장을 위한 영역 내에서 발생한다고 가정한다.

본 논문에서는 제 2 절에서 자기확장 모니터링 메커니즘을 소개한다. 제 3절에서는 소개한 메커니즘을 기반으로 한 침입자동대응 모델을 제시한다. 제 4 절에서는 자기확장 모니터링 메커니즘을 기반으로 설계, 구현된 침입자동대응 시스템인 ARTEMIS[5]의 구조와 동작에 대해 설명한다. 또한 구현 예와 성능비교 결과를 소개하고 제 5 절에서 결론을 맺는다.

2. 자기확장 모니터링

자기확장 모니터링은 침입자의 행위를 모니터링하며 그러한 모니터링 행위 자체를 보호한다. 또한 침입자의 이동 경로 상에 있는 호스트들에 자신을 복제함으로써 모니터링과 침입탐지, 침입대응을 위한 영역을 확장해 간다. 공간적인 제약을 극복하여 지역적인 보안정보 수집과 분석에서 벗어나 많은 보안정보를 공유할 수 있다.

2.1. 자기보호 기능을 가진 모니터링

네트워크로 연결된 여러 호스트가 보안관리의 대상이 된다. 자기확장 모니터링은 기본적으로 사용자 단위로 수행된다. 모니터링 결과 획득되는 정보는 그 성질에 따라 동적 모니터링 정보와 정적 모니터링 정보로 나뉜다. 사용자가 수행하는 명령, 그에 의해 발생하는 이벤트와 같이 계속적으로 변화하고 실시간 감시가 요구되는 것을 동적 모니터링 정보라 한다. 정적 모니터링 정보는 호스트에서의 사용자 디렉토리 정보, 사용자가 설치한 백도어, 사

용자에 대한 계정 정보, 로그 등으로 구성된다. 동적 모니터링 정보에서 이벤트는 사용자의 행위정보, 즉 사용자 명령어와 그들을 구성하는 프로세스들 또는 필요에 따라 시스템 콜의 집합도 포함한다. 발생하는 이벤트는 시간적 순서와 공간적 위치를 가진다.

[정의 1] 사용자 A에 대한 모니터링 정보 M 은 여러 이벤트 $\{ m_1, m_2, \ldots, m_i, \ldots \}$ 로 구성된다. 각 이벤트는 $t(m_i)$ 라는 이벤트 발생시각을 가진다. 이러한 이벤트들은 전체적으로 순서화되어 있으므로 $i \geq 1$ 인 모든 i 에 대해 $t(m_i) \leq t(m_{i+1})$ 이다.

[정의 2] 모니터링 정보 M 을 구성하는 정보 중에서 호스트 H_1 상에서의 모니터링 정보 M^{H_1} 과 호스트 H_2 상에서의 모니터링 정보 M^{H_2} 가 주어질 때, 이러한 두 호스트 상에서의 모니터링 정보를 통합한 결과는 $M^{H_1} \cup M^{H_2}$ 으로 나타내고 이는 M 의 부분정보이다.

[정의 3] 필터 함수 F_p 는 모니터링 정보 $M = m_1, m_2, \ldots, m_n$ 을 M 의 부분 정보인 $\subseteq M$ 으로 매핑하는 함수이다. $\subseteq M$ 는 M 에서 유용하지 않은 이벤트들을 삭제한 결과이다. p 는 필요없는 이벤트들을 삭제하기 위한 필터링 규칙이다.

F_{p_i} 는 침입자 이동경로 및 신분정보 필터 함수이고 p_i 는 수집된 정보 중 이동경로와 신분정보를 걸러내기 위한 필터링 규칙이다. 이러한 필터링 결과 수집되는 모니터링 정보 $\subseteq M$ 는 침입자가 이동한 호스트, 호스트 접근 시 사용한 ID, 패스워드 등의 신분정보를 가진다. F_{p_a} 는 침입자 행위정보 필터함수이고 p_a 는 수집된 정보 중 행위정보를 걸러내기 위한 필터링 규칙이다. 모니터링 정보 $\subseteq M$ 는 침입자가 호스트의 콘솔 상에서 수행한 명령뿐만 아니라 명령수행을 위한 프로세스, 그에 의한 함수호출 등의 정보를 가진다.

[정리 1] $M_N (1 \leq N)$ 을 사용자 N 에 대한 모니터링 정보로 정의하면 사용자에 A 대한 모니터링 정보 M_A 를 $M_A = M_A^{H_1} \cup M_A^{H_2} \cup \ldots \cup M_A^{H_n}$ 으로 나타낼 수 있다.

사용자 A에 의해 서로 다른 공간적 위치에서 발생하는 이벤트의 집합 M_A 는 $M_A = \{ m_1, m_2, \ldots, m_n \}$, $n \geq 1$ 로 나타낼 수 있다. 사용자 A 가 호스트 H_1 에서 $M_A^{H_1} = m_{H_11}, m_{H_12}, m_{H_13}, \ldots, m_{H_1p}$ 과 같은 이벤트, 객체, 값의 쌍들을 생성하고 다시 호스트 H_2 로 옮겨 $M_A^{H_n} = m_{H_21}, m_{H_22}, m_{H_23}, \ldots, m_{H_2q}$ 를, 결국 H_n 으로 옮겨 $M_A^{H_n} = m_{H_n1}, m_{H_n2}, m_{H_n3}, \ldots, m_{H_nr}$ 를 생성하였다고 가정한다. 또한 $H_11, H_12, \ldots, H_1k, H_21, \ldots, H_2p, \ldots H_n1, \ldots, H_nq$ 이 $1,2,\ldots,p+q+r$ 의 부분 순서열이라고 가정한다. 미행 메커니즘은 사용자의 이동경로를 따라 A가 수행하는 이벤트들을 모두 모니터링할 수 있으므로 결과적으로 생성되는 M_A 는 $M_A = M_A^{H_1} \cup M_A^{H_2} \cup \ldots \cup M_A^{H_n} = m_1, m_2, \ldots, m_{p+q+r} = M_A$ 이다. 침입경로상의 모든 호스트에서의 사용자 A의 행위는 미행 메커니즘에 의해 사용자 단위로 각 호스트에서 수집한 동적 모니터링 정보의 합집합과 같

다. 그러므로 단일 관리자 영역의 네트워크 내의 하나의 호스트에 침입자가 공격을 하였고 그 호스트에 미행 메커니즘 기반의 보안관리 스킴이 존재한다면 보안관리 결과는 네트워크 내의 침입자 공격경로상의 모든 호스트에 보안관리를 위한 시스템을 설치하고 수행한 결과와 같다.

자기확장 모니터링 메커니즘에 있어서 자기 보호는 부정 행위자에 대한 모니터링 행위 자체를 보호하는 것이다. 이는 부정 행위자 자신이 모니터링 되고있다는 사실을 인식하는데 걸리는 시간을 지연시킴으로써 부정 행위자를 감시할 수 있는 충분한 시간을 확보하도록 한다. 자기확장 모니터링 메커니즘에서의 자기 보호를 위해 흔적 지우기와 위장과 같은 두 가지 방법이 사용된다. 흔적 지우기에는 다음과 같은 방법이 사용된다. 프로세스 상태 명령이 프로그램에 의해 호출된 방법을 보호하는데 중요한 역할을 한다. 여주지 않기 위해 인자를 처리한 후 인자리스트를 지운다. 실행 바이너리를 삭제함으로써 링크 연결이 삭제되어 단지 프로그램 실행에 의해서만 참조된다. 또한 프로그램상의 버그가 흔적 남기는 것을 방지하기 위해 자원사용제한 함수를 사용하여 코어 덤프(core dump)를 막는다. /var/adm/utmp(x) 파일은 현재 시스템에 로그온 중인 모든 사용자 기록을 나타낸다. 이들 기록으로 부정 행위자가 모니터링 행위를 알아차릴 수 있으므로 이 기록을 삭제한다. 자기 보호의 또 다른 방법인 자신을 숨기기 위한 위장을 위해 다음과 같은 방법이 사용된다. 부정 행위자를 모니터링 중인 프로세스를 가장 흔한 프로세스인 쉘로 위장함으로써 모니터링 행위를 보호한다. 프로세스를 포크(fork)하여 부모와 자식으로 만든 후 부모는 빠져나오고 자식은 그대로 남아 수행함으로써 프로세스 재생성의 효과를 가져온다. 그 외의 부가적인 방법으로 트로이 목마와 같이 원래 프로그램을 바꿈으로써 자신을 숨길 수 있다. 프로그램 복제 시 프로세스를 숨기기 위한 ps, top, pidof 프로그램 또는 파일을 숨기기 위해 find, ls, du 프로그램을 함께 보내어 사용이 가능하다.

2.2 자기복제

자기복제는 부정행위자가 획득한 관리자 권한을 이용하여 모니터링을 위한 모듈을 타겟 호스트로 복사, 실행하여 모니터링 영역을 확대하는 과정이다. 자기확장 모니터링 메커니즘에서 복제 프로토콜은 그림 1과 같다.

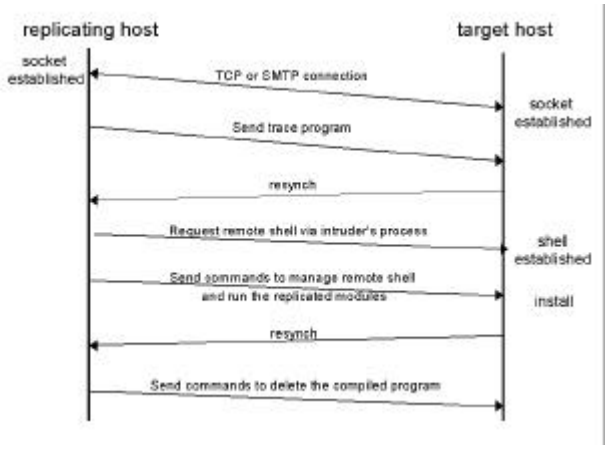

그림 1 복제 프로토콜

호스트 레벨에서 모니터링이 수행되다가 부정행위자를 발견한 경우 부정행위자에 대한 모니터링이 수행되고 그에 대한 행위정보와 인증정보를 수집한다. 부정 행위자가 다른 호스트로 이동하는 경우 이를 추적하기 위해 네트워크 레벨의 감시가 수행된다. 이로써 획득된 인증정보로 부정행위자가 이동한 호스트에 모니터링, 탐지, 추적, 대응을 위한 모듈을 복사한다. 부정 행위자가 이동한 호스트에서 관리자의 권한을 획득하는 경우 관리자 권한을 가진 쉘을 이용하여 명령을 전송한다. 그리고 복사된 모듈을 컴파일하고 데몬의 형태로 백 그라운드 실행함으로써 이동한 호스트에서의 부정행위자에 대한 추적이 계속된다. 타겟 호스트가 전송하는 resynch 메시지는 타겟 호스트와의 동기화를 위해 사용된다. resynch 메시지를 보냄으로써 복제가 완료되었음을 알리면 복제하는 호스트는 타겟 호스트에서의 모니터링 상태를 보호하기 위해 컴파일된 프로그램을 삭제하는 명령을 타겟 호스트에 전송한다.

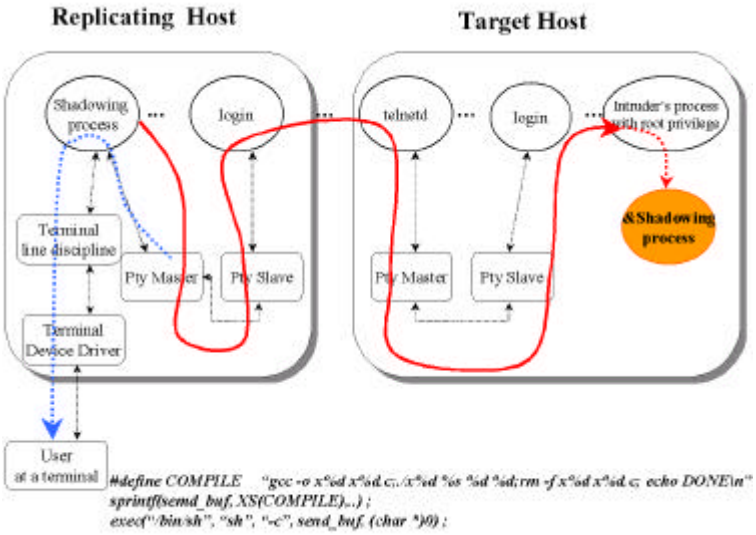

그림 2 모니터링과 복제

그림 2는 자기확장 모니터링에서의 복제 과정을 포함한 모니터링을 나타낸다. 침입자가 공격을 시도하

였을 때 가상 터미널을 할당함으로써 침입자의 행위를 감시할 수 있을 뿐 아니라 관리자가 침입자의 프로세스로 명령을 보내는 것이 가능하다. 가상 터미널을 통해 침입자를 감시하면서 telnet, ftp 등의 이동 명령 또는 침입자의 ID, 패스워드 등의 신분정보가 발견되면 즉시 복제 프로토콜을 구동하여 보안관리를 위한 모듈의 복사를 실행한다. 그림 2의 하단의 내용은 복제된 모듈의 컴파일을 위해 수행되어야하는 명령을 구현한 예이다.

3. 침입자동대응 모델

모니터링 수행 중 침입이 발생함과 동시에 시스템에 대한 피해를 최소화하고 침입자 추적 등의 즉각적인 대응을 하기 위해 인간의 개입없이 자동적으로 대응하는 기능을 침입자동대응[1]이라 한다. 본 논문에서 제시하는 침입자동대응 모델은 그림 3과 같이 데이터 계층, 전송 계층, 평가 계층, 제어 계층으로 구성된다. 데이터 계층은 사용자 명령어 레벨 또는 프로세스 레벨에서 동작하고 모니터링과 같은 데이터 조작을 위한 서비스들을 제공한다. 전송계층은 데이터 또는 모듈 전송 서비스와 메시지 전송 서비스들을 제공한다. 평가 계층은 수집된 데이터를 분석하고 그들로부터 연관된 의미정보를 추출한다. 제어 계층은 평가 계층으로부터 전송된 결과를 미리 설정된 정책에 따라 관리자에게 보고하거나 자동으로 적절한 대응을 취한다. 논리적 계층을 구성하는 모든 컴포넌트는 독립적으로 존재할 수 있다. 데이터 계층과 전송 계층은 네트워크상의 모든 영역에 설정될 수 있다. 반면 평가 계층과 제어 계층은 관리하는 영역의 범위에 따라 기능을 분리하여 분산적으로 배치된다.

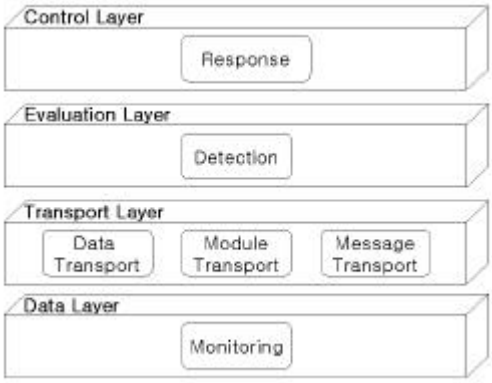

그림 3 침입자동대응모델의 논리적 계층

3.1 데이터 계층

데이터 계층은 데이터들과 데이터들을 조작하기 위한 모니터링 서비스로 구성된다. 일반적으로 데이터는 시스템에서 발생하는 이벤트들과 같은 동적 정보와 사용자 디렉토리, 로그, 계정 등과 같은 정적 정보를 포함한다. 이벤트는 단일 프로세스에 의해 내부적으로 발생하는 이벤트와 외부적으로 관찰될 수 있는 이벤트로 나누어진다. 프로

세스에 의해 수행되는 함수호출이 내부적으로 발생하는 이벤트의 예이다. 반면 외부적으로 관찰될 수 있는 이벤트는 시스템간 전송되는 메시지, 사용자 명령어 등을 포함한다. 사용되는 메시지는 3.2절의 전송계층에서 설명한다.

본 침입자동대응모델에서는 자기확장 모니터링 메커니즘을 기반으로 모니터링 서비스를 제공한다. 자기 보호 기능을 가진 모듈로써 침입자의 행위를 모니터링하여 호스트 레벨에서의 감시를 수행한다. 침입자가 다른 호스트로 이동하는 경우 침입자동대응을 위해 필요한 모듈을 이동한 타겟 호스트로 복제하여 네트워크 레벨에서의 감시가 수행된다. 일단 타겟 시스템이 설정되면 호스트 레벨의 행위감시를 수행하므로 정보수집을 위한 영역에 대한 제약이 적어지고 암호화된 데이터를 읽을 필요도 없다. 이로써 침입자에 대한 모니터링 영역을 넓혀가며 감시할 수 있다. 사용자 명령에 의해 생성되는 프로세스와 함수호출에 대한 모니터링은 그림 4와 같이 이루어진다. 여러 호스트에 걸쳐 사용자가 n 개의 명령어를 사용하였다. n 개의 명령어는 k 개의 프로세스로 이루어지고 각 명령어는 하나 이상의 프로세스로 구성되므로 $n \leq k$이다. 하나의 프로세스는 또 다시 하나 이상의 시스템 호출로 구성된다. 모니터링의 중심역할을 하는 자기확장 모니터(Self-Extension Monitor)는 부정행위자에 이동경로를 추적하여 계속적으로 모니터링을 수행하는 역할을 한다. 사용자 명령어, 프로세스, 시스템 호출들이 자기확장 모니터 SM_x에 의해 감시된다. 여기서 x 는 각 호스트를 구분하기 위한 인덱스이고 그림에서의 표현된 SM_1, ..., SM_p 는 자기확장 모니터가 모두 p 개의 호스트에 설치되었다는 것을 나타낸다.

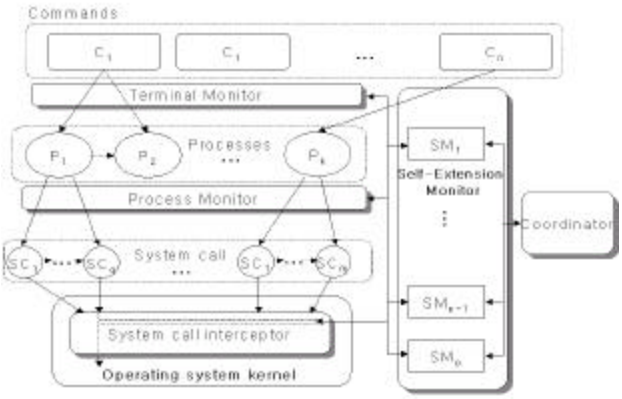

그림 4 사용자 명령어와 프로세스 레벨 모니터링

3.2 전송 계층

전송 계층은 데이터 계층의 모니터링 서비스에 의해 수집된 데이터와 메시지에 대한 전송 서비스를 제공한다. 이들은 메시지 통신을 위한 프로토콜과 복제를 위한 프로토콜을 포함한다. 네트워크상의 침입추적과 탐지, 대응 등의 보안관리를 위해서는 호스트들간의 정보공유가 중요한 역할을 한다. 이러한 정보들은 추적(Trace), 요청(Request), 지시(Indication) 메시지를 통하여 전달된다. 그림 5는 보안관리를 위한 메시지 교환을 나타낸다. 추적 실행기(Trace Dispatcher)는 마스터 시스템에 존재하고 추적 수집기(Trace Collector)는 각 에이전트 시스템에 배치된다. 추적 메시지는 하나의 에이전트에서 시스템과 네트워크에 피해를 주는 공격이 될 수 있는 이벤트들이 발견되는 경우 주위의 에이전트의 추적 수집기에 전송되어 이 공격이 각 에이전트를 거쳤는지 확인하는데 사용된다.

이 메시지는 발생한 이벤트, 침입자에 의해 사용된 연결의 종류, 요구되는 침입대응 등을 포함한다. 요청 메시지는 추적 메시지의 복사본으로 추적 수집기로부터 마스터에 있는 추적 실행기로 전송된다. 이 메시지는 추적 실행기에 의해 공격의 이동경로 분석, 침입 추적에 사용된다. 또한 추적 실행기는 지시 메시지를 통해 공격에 대해 각 에이전트가 취해야할 적절한 대응을 지시한다.

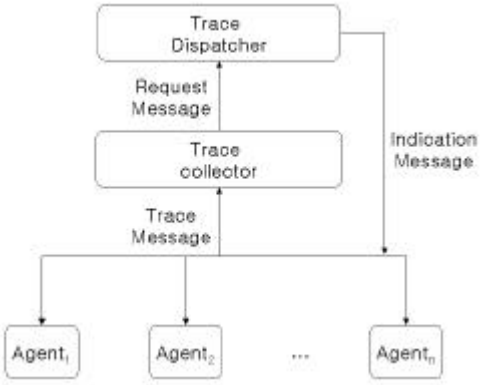

그림 5 정보공유를 위한 메시지 교환

침입자동대응을 위해 침입자의 이동경로를 따라 2.2절에 제시된 복제 프로토콜을 통해 필요한 모듈이 전송, 실행된다. 전송계층 역시 데이터 계층에 따라 동적으로 배치될 수 있다.

3.3 평가 계층

데이터와 전송 계층으로부터의 정보를 분석하여 침입을 탐지한다. 자기확장 모니터링 메커니즘을 통해 획득한 사용자 명령어 정보와 그에 의한 프로세스 상태를 입력으로 받아들여 의미정보를 추출한다. 침입 판정은 프로세스가 시스템 함수 호출을 호출하는 시점에서의 프로세스의 권한 상태 즉, 프로세스의 UID, EUID, GID, EGID를 이용하여 유한상태기계(FSM: Finite State Machine) 모델을 구축하고 상태전이에 의해 침입을 탐지한다. 또한 각 사용자의 터미널을 감시하여 수행되는 사용자 명령어들을 파싱하여 얻은 분석 결과와 시스템 호출의 부가적인 정보를 종합적으로 이용하여 오용탐지방법의 문제인 오류탐지(False Negative)를 해결한다.

프로세스의 상태를 반영한 침입탐지를 위한 유한상태기계 모델은 $M = (S, I, \delta, q_0, F)$와 같이 정의된다. 모델 내의 정의된 상태는 다음과 같다.

$S = \{N, SP, ABN, SSG, I\}$

N : 정상 상태(Normal State)

SP : 특별 권한 상태(Special Privileged State)

ABN : 이상 상태(Abnormal State)

SSG : 수퍼유저 또는 시스템 그룹 상태(Superuser 또는 System Group State)

I : 침입 상태(Intrusion State)

상태 전이를 발생시키는 이벤트의 집합은 다음과 같다.

$I = \{pc, cmp, gne, spe, sue, exe, timeout\}$ 이다.

pc : 프로세스 생성 이벤트

cmp : 프로세스 권한에 변화를 준 프로그램 또는 시스템 호출 종료 이벤트

gne : 일반적인 명령 또는 프로그램 수행 이벤트

spe : setuid(), setreuid(), setgid(), setregid()와 같은 시스템 호출 이벤트

sue : 수퍼유저에 의해 발생한 이벤트

exe : execve(), exec()등의 실행을 위한 프로그램 시스템 호출 이벤트

timeout : 설정된 시간 만료 이벤트

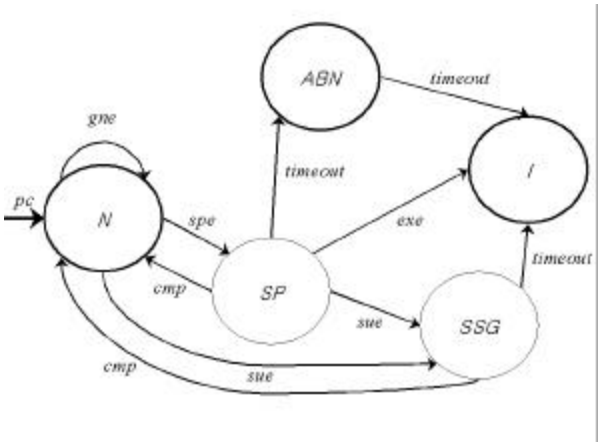

그림 6 침입탐지를 위한 상태 전이 다이어그램

그림 6에서의 상태전이를 표현하는 함수 δ : $S \times I \rightarrow S$ 는 다음과 같다.

$\delta(N, gne) = N$, $\delta(N, spe) = SP$, $\delta(N, sue) = SSG$

$\delta(SP, sue) = SSG$, $\delta(SP, cmp) = N$, $\delta(SP, exe) = I$, $\delta(SP, timeout) = ABN$

$\delta(ABN, timeout) = I$

$\delta(SSG, cmp) = N$, $\delta(SSG, timeout) = I$

초기 상태는 $q_0 = N$, $q_0 \in S$ 이고 상태 전이를 완료하는 시점에서의 상태의 집합은 $F = \{N, ABN, I\}$이다.

■ 정상상태(N) : 최초 프로세스가 생성되면 프로세스의 UID와 EUID는 해당 프로세스를 실행시킨 사용자의 UID로 설정된다. 프로세스 권한 상태가 프로세스를 초기화한 사용자의 소유값과 같다. 그러므로 프로세스 권한 상태값은 (uid, uid, ugid, ugid) 이다.

■ 특별 권한 상태(SP) : 프로세스 권한 상태가 특권 그룹에 속해있지만 UID, EUID값 중 하나는 특권이 없는 원래 프로세스 소유자의 ID로 설정된 경우이다. 즉, 이 두 개의 속성은 같은 특권 ID를 공유하지 않는다. 이것은 그룹 ID에도 적용된다. 이 특별 권한 상태로의 전이는 setuid(), seteuid(), setgid(), setegid()와 같은 시스템 호출의 결과 발생된다.

■ 수퍼유저 또는 시스템 그룹 상태(SSG) : UID, EUID가 모두 root, daemon, operator, bin, news 등의 높은 권한을 가진 사용자 ID로 설정되는 경우 수퍼유저 상태로 전이되고 GID, EGID가 모두 wheel,

daemon, kmem, sys, tty 등의 높은 권한을 가진 그룹 ID로 설정될 때 시스템 그룹 상태로 변화한다. 이러한 상태 변화는 일반적으로 시스템 수퍼유저 또는 수퍼유저가 수행하는 프로그램에 의해 발생한다.

■ 이상 상태(*ABN*) : 정상 상태에서 프로세스의 특권 상태가 변하거나, 사용자가 침입과 관련이 있는 명령어를 수행하여 그 프로세스의 상태에 영향을 주고자 시도한다면 이상 상태로 전이한다. setuid 프로그램의 경우에는 EUID의 값이 프로그램 파일 소유자의 UID값을 지나치게 오래 가지고 있을 경우에도 이 상태로 전이한다.

■ 침입 상태(*I*) : 이상 상태에 있던 프로세스가 일정 시간 이상 그 상태를 유지하면 침입 상태로 전이한다. 이는 일반 사용자가 실행한 setuid 프로그램의 UID와 EUID값 모두가 프로그램 파일 소유자의 UID값을 가지게 되는 경우이다. 침입이 발생한 경우 그 소유자의 UID값은 거의 0이 된다. setuid 프로그램이 아닌 일반적인 프로그램의 경우는 정상 상태에서 직접 침입 상태로 전이할 수 있는데, 이 경우는 EUID 또는 UID가 0으로 되는 경우이다. 프로세스가 특별 권한 상태에 있을 때 다른 프로그램 또는 명령을 실행함으로써 execve()를 허용하는 것은 새로운 프로세스가 특권 상태를 계승받도록 하므로 침입 상태로 전이한다.

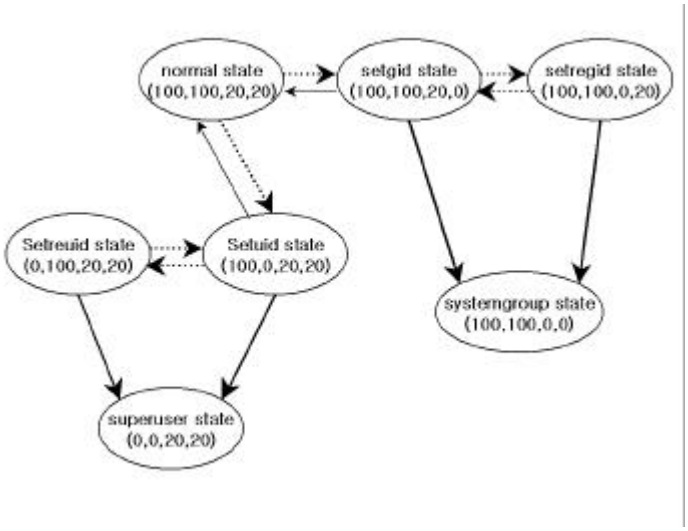

그림 7 상태 전이 다이어그램의 예

그림 7은 상태 전이 다이어그램의 예이다. 초기에는 사용자에 의해 생성된 프로세스의 상태 튜플 속성(UID, EUID, GID, EGID)이 입력된다. 선으로 표시된 상태전이는 시스템 호출에 의해 발생하며 이들은 상태 튜플 속성을 변경한다. 이들 속성은 숫자로 표현되며 각 상태에 대한 특정 시점에서 사용자 ID를 나타낸다. 이 예에서 프로세스를 생성한 사용자는 사용자 ID 100, group ID 20을 가진다고 가정한다. 사용자 ID 0 은 수퍼유저(superuser) ID를 나타낸다. 사용자 프로세스 실행의 초기에 프로세스는 항상 일반 상태에서 출발하고 이러한 튜플 속성은 사용자와 그룹 ID로 표현된다. 프로세스의 튜플 속성 중 하나가 특권 값을 가지게 되면 setuid, setgid, setreuid, setregid 상태로 변경된다. 가는 실선은 정상 상

태로의 전이이고 점선은 이상 상태로의 전이를, 굵은 실선은 침입 상태로의 전이를 나타낸다.

본 논문에서 제시하고 있는 침입 탐지 기법은 매우 단순한 상태 전이 다이어그램에 기반을 두고 있으나 사용자 명령어 정보와 프로세스, 한정된 시스템 함수 호출에 대한 정보만을 추출하여 이용함으로써 침입 탐지의 복잡성을 최소한으로 유지한다. 모든 프로세스의 상태를 추적하여 침입탐지를 수행하는 다른 기법[6,7]들에 비해 프로세스 모든 시스템 호출 정보를 수집하는 과부하를 최소화할 수 있다. 미리 정의된 명령어 패턴에 의지하지 않으므로 새로운 침입 형태 탐지가 용이하며 시스템 환경에 따른 확장도 쉽다.

3.4 제어 계층

평가 계층으로부터 전송된 결과를 기반으로 제어 계층에서 침입에 대한 대응을 수행한다. 일반적인 대응은 Fisch의 DC&A 분류법[8]에 의해 대응시기와 대응목적별로 분류된다. 대응시기에 따라 침입 진행 중 수행되는 대응인 손실 제어(Damage Control)와 침입이 완료한 후 대응인 손실 평가(Damage Assessment)로 나뉘어진다. 대응성격 또는 목적에 따라 제어는 적극적(Active Damage Control), 수동적(Passive Damage Control) 대응으로 나뉘어지며 평가는 단순 평가(Assessment)와 복구(Recovery)로 구성된다. 본 논문에서 제시하는 모델에서는 적극적 손실제어, 단순평가 등의 대응을 제공한다. 단순 평가는 시간 개념의 유무에 따라 시간제약평가와 요구평가로 나뉜다. 시간제약평가는 시간의 개념을 포함한 대응이다. 예를 들면, 보안상 위급한 상황 또는 미리 정의된 상황이 발생하였을 경우 즉시 이메일, 페이저, 콘솔 메시지를 이용하여 호스트에 대한 침입 또는 호스트의 현재 보안상태를 관리자에게 알리는 것이다. 요구평가는 관리자가 요청할 때 단순히 보안관리 현황등을 디스플레이하는 것을 포함한다. 적극적 손실제어는 침입자에 대한 추적과 타겟 호스트 또는 공격을 시작한 호스트에 대한 동적 설정 변경 등을 포함한다. 전송계층에서의 메시지 전송을 이용하여 침입추적이 이루어지고 그에 대한 지역적인 대응도 수행된다.

4. ARTEMIS : 침입자동대응 시스템

4.1 시스템 구조 및 동작 모델

ARTEMIS 는 하나의 마스터 시스템과 다수의 에이전트 시스템으로 구성된다. 자기확장 모니터링 메커니즘을 기반으로 하므로 필요한 곳에 에이전트 시스템을 동적으로 설치하는 것이 가능하다. 그림 8은 ARTEMIS의 전체 구조를 나타낸다.

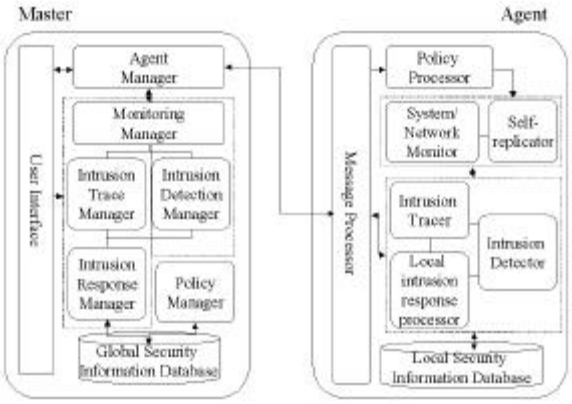

그림 8 ARTEMIS 구조

각 에이전트 시스템의 시스템/네트워크 모니터는 모니터링을 통해 보안관련 정보를 수집한다. 침입 탐지기는 수집된 정보로부터 의미적인 정보를 추출하여 지역적인 침입증상을 탐지한다. 빠른 대응이 요구되는 경우 지역침입대응처리기가 정책을 따라 적절한 대응을 취하고 침입자 추적이 가능한 경우 침입 추적기가 침입자의 이동경로와 행위정보에 의해 추적한다. 메시지 처리기를 통해 수집된 정보와 침입증상, 침입대응 등을 메시지의 형태로 마스터 시스템에 보고한다. 또한 에이전트 시스템은 특정 조건이 만족되는 경우 모니터링, 침입탐지, 대응을 위한 모듈을 자기복제기를 통해 다른 호스트로 전송, 실행함으로써 자신을 복제하여 침입탐지와 대응을 위한 영역을 넓힐 수 있다. 마스터 시스템은 에이전트 관리기를 통하여 네트워크에 분산된 모든 에이전트 시스템들을 관리한다. 각 에이전트 시스템들로부터 메시지의 형태로 전송되는 보안정보들을 모니터링 관리기로 관리 또는 조작한다. 침입탐지 관리기는 모니터링 관리기로부터 전송되는 보안정보들을 분석하여 전체 네트워크에서의 침입을 탐지한다. 침입대응 관리기는 마스터 시스템의 분석결과 대응전략에 따라 침입대응을 수행하며 또 다시 메시지의 형태로 각 에이전트에 적절한 대응을 지시한다. 또는 정책 관리기에게 명령하여 네트워크 전역에 적용되는 정책과 각 에이전트에 적용되는 지역정책을 동적으로 재설정한다. 적절한 대응을 전달받은 에이전트는 지역침입대응 처리기를 통해 침입에 대한 대응을 수행한다. 침입추적 관리기는 각 에이전트가 마스터의 모니터링 관리기에 전달한 정보로부터 추적에 관한 정보를 추출하여 전체 네트워크에서 침입자 추적을 담당한다. 에이전트 시스템의 시스템/네트워크 모니터와 마스터 시스템의 모니터링 관리기는 침입자동대응모델에서 데이터 계층에 속하고 메시지 처리기와 자기확장을 위한 자기복제기는 전송계층에 해당된다. 수집된 자료를 기반으로 탐지여부를 판단하는 침입탐지 관리기와 침입탐지기는 평가 계층에 배치된다. 대응을 담당하는 제어 계층에는 침입 추적과 같은 적극적 손실 제어를 위한 침입 추적기, 침입 추적 관리기가 구현된다. 관리자로의 통보, 보안현황 디스플레이 등의 단순 평가와 보안관리시스템의 재설정과 같은 적극적 손실 제어를 담당하는 침입대응 관리기, 지역 침입대응 처리기 또한 제어 계층에 포함된다.

ARTEMIS는 하나의 호스트에 마스터 시스템을 인스톨하면서 초기화된다. 마스터 시스템이 설치된 호

스트상의 사용자가 침입자로 추정되는 경우 침입자의 행위정보, 이동경로 등을 수집, 분석한다. 수동으로 에이전트 시스템을 설치하는 것이 가능하지만 침입자가 마스터 시스템을 거쳐 다른 호스트로 이동하는 경우 그 호스트가 단일 관리자 영역에 존재한다면 에이전트 시스템은 자기확장 모니터링의 복제과정을 통해 자동으로 타겟 호스트에 설치된다. 에이전트 시스템들은 침입자의 이동경로를 따라 필요에 의해 계속적으로 설치될 수 있다. 네트워크 상에서 침입이 탐지되는 경우 각 에이전트에 미리 설정된 정책에 따라 침입자 추적, 타겟 호스트 상에서의 대응, 침입을 시작한 호스트에 대한 대응, 설정 변경, 보고서 제공 등의 대응이 자동적으로 이루어질 수도 있고 마스터 시스템에 의해 전역적인 대응이 이루어지기도 한다.

4.2. 시스템 구현 및 성능 비교

ARTEMIS는 Solaris 2.x, Redhat 6.x Linux 운영체제에서 동작한다. 감사 데이터 수집 모듈은 GNU C/C++ 2.7.x.x, 시스템간의 인터페이스 및 사용자 인터페이스는 J2EE(Java 2 Platform, Enterprise Edition)를 이용하여 구현되었다. 감사 데이터 및 모니터링 정보를 저장하는 DBMS로는 MySQL 3.22.x를 사용하고 시스템간의 인증 및 암호를 위해서 JCE(Java Cryptography Enhancement)1.2 패키지를 사용한다.

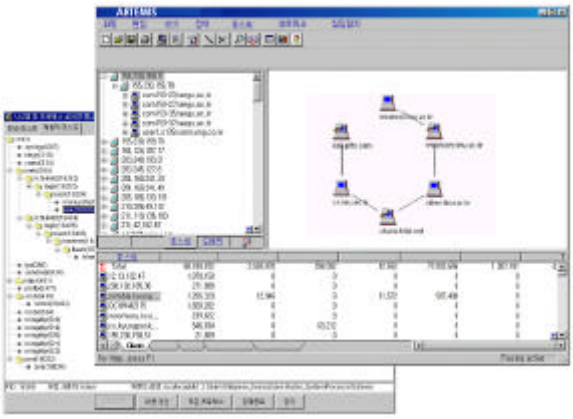

그림 9 자기확장 모니터링에 의한 침입추적 예

ARTEMIS 사용자는 초기 정책 설정, 관리 영역에 의한 에이전트 수동 설치 등을 입력할 수 있다. ARTEMIS가 설치된 호스트와 침입자의 이동 경로 상에 있는 호스트에 대한 침입탐지, 침입추적 등을 수행하고 각 호스트에 대한 모니터링 정보, 침입추적 결과, 침입자의 이동경로상의 호스트에 대한 정보, 침입자에 대한 정보를 보고서의 형태로 관리자에게 제공할 수 있다. 그림 9는 부정행위자의 침입사실을 확인한다. 또한 미행과 메시지 교환을 통해 확인한 부정행위자의 이동경로를 보여준다.

표 1은 기존의 침입탐지와 대응을 위한 시스템들과 ARTEMIS를 여러 항목에 대해 비교한 것이다. 본 성능비교에서는 보안관리를 위한 영역 확장에 중점을 두었다. 자기확장 모니터링을 기반으로 설계되지 않은 기존의 보안관리제품들은 그들이 독립 시스템 또는 분산 시스템으로 동작하는가에 상관없이 보안관리를 위한 제품이 특정 지점에 미리 설치되어 있어야만 한다. 이것은 관리자의 관리 영역을 안전하게 보호하기 위해 되도록 많은 지점에 보안관리를 위한 제품을 손수 설치해야 된다는 것이며 이를 유

지, 보수하는데 많은 시간과 노력이 든다. 침입자의 이동 경로상에 있는 호스트에 보안관리 제품이 설치되어 있지 않은 경우 더 이상의 추적을 수행할 수 없고 침입자에게 네트워크에서 그들의 영역을 넓힐 충분한 시간을 제공한다. 그에 비해 ARTEMIS는 침입자의 이동경로를 따라 동적으로 에이전트 설치가 가능하다. 즉, 관리영역내의 모든 호스트에 미리 보안관리제품을 설치할 필요없이 외부에서 관리 영역으로 침입하는데 항상 거쳐야 하는 게이트웨이에만 본 시스템을 설치함으로써 침입자의 이동 경로를 따라 취약점이 있는 호스트에는 자동적으로 ARTEMIS 에이전트가 설치될 수 있도록 한다. 그러므로 기존 제품들에 비해 보안관리를 위한 공간적 제약이 최소화된다. 침입에 대해 자동적으로 대처하고 모니터링 행위 자체를 보호하므로 침입과 대응간의 시간차를 최소한으로 줄인다. 추적을 통해 공격을 시작한 호스트와 사용자의 신분을 확인하여 더 이상의 공격을 막을 수 있다.

표 1 침입대응시스템 성능비교

Name of System	Detection Principle	Time of Detection	Audit Source	Type of Response	Data Processing	Data Collection	**Domain for security management**
IDES[9]	anomaly	realtime	host	passive	centralized	distributed	**static**
NSM[10]	hybrid	realtime	network	passive	centralized	centralized	**static**
USTAT[11]	policy	realtime	host	passive	centralized	centralized	**static**
NIDES[12]	hybrid	realtime	host	passive	centralized	distributed	**static**
GrIDS[13]	hybrid	non-realtime	both	passive	distributed	distributed	**static**
EMERALD[2]	hybrid	realtime	both	active	distributed	distributed	**static**
ARTEMIS	hybrid	realtime	both	active	centralized	distributed	**extensible**

5. 결 론

본 논문에서는 자기확장 모니터링 메커니즘을 기반으로 한 새로운 침입자동대응 기법을 제시하였다. 모니터링, 침입탐지, 침입추적 및 대응에 자기확장 모니터링 메커니즘이 적용됨으로써 자동적으로 하나의 보안 스킴에 의해 보안관리가 수행된다. 또한 지역적으로 수집하는 보안정보에 대한 공유가 가능하게 되었다. 이로써 부정행위자 추적을 위한 공간적인 제약이 최소화되어 다양한 보안정보를 포괄적으로 분석하여 보다 정확한 침입탐지와 추적, 적절한 대응이 가능하다. 이러한 방법을 통해 수집되는 침입 경로, 악의적 행위와 같은 증거의 보존은 더 이상의 침입을 막고 오랜 기간동안 침입자의 습관과 경로를 분석하는 것은 해킹 시나리오 데이터베이스 생성을 용이하게 한다. 침입자 경로상의 시스템 보안 관리자들은 함께 공조하여 수사할 수 있고 결국 침입자의 신분을 확인할 수 있다.

본 논문에서 제시한 침입자동대응모델을 기반으로 한 ARTEMIS는 단일 관리자 영역 또는 서로 신뢰관계에 있는 여러 관리자 영역 내의 다른 호스트에서 부정행위자가 관리자 권한을 획득하는 경우 복제를 통한 미행이 가능하다. 향후 이를 다수의 관리자 영역으로 확장하는 방법에 대한 연구가 필요하다.

참 고 문 헌

[1] D. Schnackenberg and K. Djahandari, “Infrastructure for Intrusion Detection and Response”, http://seclab.cs.ucdavis.edu/projects/idip.html

[2] P.A. Porras and P.G. Neumann, "EMERALD : Event Monitoring Enabling Responses to Anomalous Live Disturbance," *Proceedings of the National Information Systems Security Conference*, pp.353-365, October 1997

[3] Network Associates, Active Security, http://www.nai.com/asp_set/products/tns/activesecurity/acts_intro.asp/

[4] H. Jang and S. Kim, “"A Self-Extension Monitoring for Security Management,"” P*roceeding of the 16th Annual Computer Security Applications Conference*, pp. 196-203, December 2000

[5] **장희진, 박보석, 김상욱, "미행 메커니즘에 의한 침입자동대응", 한국통신정보보호학회 종합학술발표회논문집**, pp.514-522, 2000.11.

[6] S.Garfinkel, G.Spafford, *“Practical UNIX and Internet Security”*, 2nd Ed. OReilly & Associates Inc., pp.731-757, 1996.

[7] S.A.hofmeyr, S.Forrest, A.Somayaji, “"Intrusion Detection using Sequences of System Calls”," Dept. Of Computer Science, Univ. of New Mexico, 1998, http://www.cs.unm.edu/~steveah/publication/ids.ps

[8] E.A.Fisch, "Intrusion Damage Control and Assessment: A Taxonomy and Implementation of Automated Responses to Intrusive Behavior", Ph.D. Dissertion, Texax A&M University, College station, TX, 1996

[9] T.F.Lunt, R.Jagannathan, R.Lee et al., "IDES : The enhanced prototype, A Real-time Intrusion Detection System", Technical report SRI-CSL-88-12, Computer Science Laboratory, SRI International, USA, October 1988

[10] T.Hebelein, G. Dias, K.Levitt et al., "A Network Security Monitor," *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pp.296-304, 1990

[11] K.Ilgun, R.A.Kemmerer, and P.A.Porras, "State transition analysis: A rule based intrusion detection approach," *IEEE Transactions on Software Engineering*, vol.21, no.3, pp.181-199, March 1995

[12] D.Anderson, T.Frivold, and A.Valdes, "Next generation Intrusion Detection Expert System", Technical Report SRI-CSL-95-07, Computer Science Laboratory, SRI International, USA, May 1995

[13] S.S.Chen, S.Cheung, R.Crawford et al, "GrIDS-A Graph based Intrusion Detection System for large networks," *Proceedings of th 19th National Information Systems Security Conference*, 1996