UNIX 환경에서 해킹방지를 위한 침입탐지시스템의 설계에 관한 연구

°정상수, 남길현
국 방 대 학 원

A Study on Design of Intrusion Detection System for hacking prevention under UNIX

Sang-Su Chung, Kil-Hyun Nam
National Defense University

요 약

본 논문은 UNIX 시스템에서 불법행위를 막을 수 있는 규칙 베이스 전문가 시스템의 설계에 관한 연구이다. 먼저 해킹의 유형과 예방 대책에 대한 연구와 아울러 침입 탐지 기법들을 소개하였다. 본 논문에서의 설계된 시스템은 오용 탐지 기법을 적용한 상태전이 분석 침입 탐지의 규칙 베이스 전문가 시스템이다. 기존의 규칙 베이스 시스템이 침입 시나로오에 대한 원시 감사 레코드를 패턴 매칭하는 반면 상태전이 분석 시스템은 상태 변화에 관한 분석에 초점을 둠으로써 시스템의 보안 침해를 받기 직전의 임박한(Impending) 위험에 신속히 대처할 수 있다. 따라서 이러한 분석 기법을 국내에서 개발된 주 전산기 타이컴Ⅱ의 감사 메카니즘과 연관시켜 규칙 베이스 침입 탐지 시스템을 설계하는 방안을 제시하고자 한다.

1. 서 론

컴퓨터 기술과 통신 기술의 발달은 대량의 정보를 신속 정확하게 처리하여 정보를 제공함으로써 정보의 효율성, 활용성, 편의성의 증대를 가져왔다. 그러나 이러한 장점의 이면에는 컴퓨터 통신망상에서의 도청뿐만 아니라 시스템내부 정보의 위조, 파괴, 개인의 프라이버시 침해와 같은 컴퓨터 범죄의 증가로 심각한 사회 문제를 일으키고 있다. 특히 정보통신기술의 급속한 발전이 가속화 되면서 인터넷은 연구 분야에서부터 광고, 정치, 예술, 교육, 비즈니스 등 전분야에 걸쳐서 현재 2000만명이상의 가입자와 250만대 이상의 컴퓨터가 연결되어 있으며, 향후 2001년경에는 인터네트를 통해 1억명 이상의 사용자가 천만대 이상의 컴퓨터와 연결될 것으로 예상된다[조원94]. 이와같이 TCP/IP를 근간으로하는 인터넷의 급속한 확산은 통신망을 통해서 컴퓨터 시스템의 취약점을 이용한 해커들의 공격에 무방비 상태에 놓이게 되어, 엄격한 보안이 요구되는 비밀자료나 시스템에 대해서 불법 침해 사례가 급증하고 있으며 이들에 대한 보안문제를 해결하기 위한 방안이 시급히 요구되고 있다.

현재까지 정보보안에 대한 대책으로서 전산망이나 컴퓨터 시스템에 있어서 신분확인이나 액세스 제어 기술이 적용되고 있으며 이러한 것은 외부의 침입을 막기위한 1차적인 방어 수단이 되지만 타협이나 공모에 의해 파괴 되었을 때의 피해는 매우 증폭되어진다. 그러므로 이러한 침입을 탐지하는 시스템의 개발을 요구하게 되었으며 기존의 auditing기법에 전문가 기법, 통계적 기법, 인공지능적 기법등 여러 첨단 기술을 적용한 시스템을 개발하게 되었다[신이93].

본 논문에서는 컴퓨터 시스템에서 일어날 수 있는 여러 가지 해킹에 관하여 고찰하고 기존의 침입 탐지 시스템에 대한 분석과 아울러 실시간 침입 탐지 시스템을 설계하는데 있어서 상태전이도를 이용한 분석기법과 이러한 분석기법을 국내에서 개발된 컴퓨터 타이컴Ⅱ의 감사 메카니즘과 연관시켜 규칙 베이스 침입 탐지 시스템을 설계하는 방안을 제시하고자 한다.

2. 해킹의 유형 및 방지대책

2.1 해킹(Hacking)

컴퓨터 시스템에 전산망을 통하여 액세스 권한 없이 무단 침입하여 부당 행위를 하는 것을 해킹이라고하며 이러한 행위를 하는 자를 해커라고 한다. 원래 순수한 의미에서 해커란 단순한 호기심과 모방행위, 그리고 자신의 능력에 대한 과시욕으로 액세스 권한을 얻어 보고자하는 영웅적 심리에서 출발했다. 그러나 정보보안의 중요성이 강조되는 현대 사회에서는 타인의 자료를 누출하고 변조와 파괴 등의 보복적인 행위를 일삼는 악의의 사용자도 함께 해커라고 일컫는다. 해커들은 컴퓨터에 대한 접근은 누구에 의해서도 방해받지 않고 자유스러워야 한다고 주장하면서 상업적인 소프트웨어의 저작권(Copyright) 개념을 전면부정하며 이의 반대 개념인 "Copyleft"개념으로 모든 소프트웨어는 소스코드(source code)와 함께 무료로 공개하여 누구나 사용하고 수정할 수 있어야 한다고 말한다. 그러나 그들은 허가를 받아야하는 경우에도 인가를 받지 않고 임의로 접근하는 그 자체가 바로 불법임을 간과하고 있기 때문에 그들의 행위는 명백히 범죄 행위라고 할 수 있다.

2.2 해킹의 유형

일반적으로 해킹은 시스템 사용자 계정을 이용하거나 또는 이미 알려진 시스템의 보안 취약점(security hole)을 이용하여 불법으로 그 시스템에 침입하여 시스템내의 정보를 탈취, 변조 또는 파괴하는데 목적이 있다. 해킹의 유형을 세분하면 크게 3가지로 나눌 수있다[연암94].

(1) 시스템 해킹

타인의 시스템에 불법으로 침입하여 시스템을 이용하거나 시스템내의 정보를 변조하거나 탈취하는 행위로서 트로이 목마(Trojan horse), 프로그램의 버그(bug:프로그램 상의 오류)나 약점 이용, 또는 특정 사용자 패스워드를 이용하는 방법을 말한다.

트로이 목마 프로그램을 이용한 시스템 해킹법은 주로 상대방 컴퓨터 사용자의 패스워드와 신상기록 등을 탐지하는 첩보활동이나 상대방 컴퓨터의 기능을 마비시킬 목적으로 하는 파괴활동에 이용되고 있으며 미국에서 몇몇 사설게시판을 개설한 사람들이 서로 상대방의 시스템에 피해를 주려는 목적으로 악용된 것이 시초이다.

또 프로그램의 버그나 약점을 이용하는 해킹법은 시스템 개발때 쓰인 언어의 특성이나 개발되어 사용하고 있는 프로그램내에 프로그래머의 고의 또는 실수에 의하여 지워지지 않은 약점 등을 최대한 활용하는 수법이다.

다음으로 패스워드 이용법은 가장 간단한 방법으로서 어떤 수단을 써서 특정사용자의 패스워드를 알아낸 뒤 그것을 이용해 시스템에 침입하는 방법이다.

(2) 컴퓨터 프로그램의 보호장치인 락(LOCK)을 풀어 공격하는 방법

실행 파일에 특정한 과정을 심어두고 프로그램 실행시 그 특정 과정에 해당하는 조건이 만족되지 않으면 프로그램 실행이 중단되게 하는 소프트웨어 방식과 프로그램상의 특정 과정이 시스템 자체의 하드웨어적인 특성을 검사함으로써 그 프로그램이 수행될 것인지 여부를 가리는 하드웨어 방식 등 2가지가 있다.

락의 해독방법에는 여러가지가 있으나 가장 기초적인 방법으로는 락이 걸려 있는 실행파일을 역어셈블하는 방법, 디스크나 파일의 내부코드를 볼 수 있는 프로그램을 이용하여 직접 해당코드를 조작하는 방법, 그리고 프로그래머의 실수로 생길 수있는 프로그램상의 오류를 검색하는 디버그(Debug)기능을 이용해 락을 해제하는 방법 등이 있다.

(3) 프로그램 변형

제작자의 허락없이 자신의 목적대로 프로그램을 변형하여 암호를 해독한 뒤 프로그램을 불법 변조하거나 자기 용도대로 프로그램을 바꾸는 방법이다. 쉐어웨어(shareware) 프로그램을 정식 버젼으로 고치거나 자신이 편리한대로 고쳐 쓴다든지하는 것도 이에 속한다.

2.3 해커 방지 대책

해커를 방지하기 위해서 제일 간편한 방법은 시스템을 네트워크에 연결하지 말고, 비밀 자료를 시스템내에 저장하지 않으면 된다. 그러나 업무의 편의성을 고려해 보면, 시스템은 항상 네트워크에 연결되어 있어야 하며, 또한 보안에 중요한 자료의 작업도 시스템내에서 이루어 지기 때문에 이러한 상황하에서의 방지 대책으로서는 외부의 네트워크로부터 들어오는 사용자에 대한 신분인증을 위해 Gateway를 이용하는 Firewall 시스템 같은 보안 프로그램의 설치와 보안에 적합한 패스워드의 관리(생성, 변경, 파기)를 철저히 해야하며, 파일 단위의 액세스제어 및 중요 자료에 대한 암호화와 시스템 모니터링 제도 및 사용자 로깅 파일의 관리를 철저히 해야 한다. 또한 시스템 감사 메카니즘의 개발로 시스템을 자동으로 분석 할 수 있는 도구가 필요하다.

3. 해킹 침입 탐지 기법

3.1 비정상 행위 탐지(Anomaly Detection)

침입 탐지 시스템의 개발에 있어서 사용되어지는 가장 널리 알려진 접근법 중의 하나로서 시스템상에서 생성되어진 감사자료의 양과 설정된 기준으로부터 변화를 측정하기 위해 통계적 분석 기법을 사용하는 기법이다. 통계적 비정상 행위 탐지에 의해서 침입을 탐지하는데는 경계 탐지(Threshold Detection)기법과 Profile-Based 비정상 행위 탐지(Profile-Based Anomaly Detection)기법 두가지가 있다[porr92].

(1) 경계 탐지(Threshold Detection)

경계 탐지는 가장 기본적인 침입 탐지의 형태이다. 경계 탐지의 목적은 특정한 사건 각각을 기록하는 것으로서, 사건 발생의 수치가 정상적인 시스템 작동 동안 일어날 수 있는 양을 능가 했을때 탐지된다. 여기서의 사건은 짧은 기간동안에 침입자의 존재를 알리는 이상하게 높은 발생 수치이다. 경계 분석 기법의 구현에서 어려운 점은 특정한 사건에 대한 경계값을 정하는 것이다.

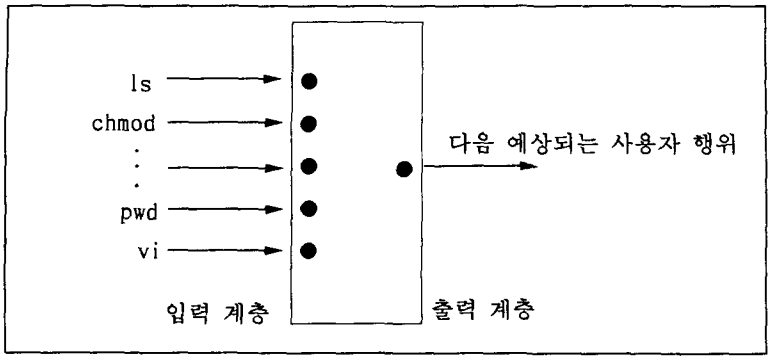
(2) 통계적 Profile-Based 비정상 행위 탐지

일반적으로 침입 탐지 기법은 Profile-Based 비정상 행위 탐지 기법이다. 여기서 profile이란 한 사용자의 예상되는 행위에 대한 기술로서 각각의 사용자의 모니터된 행위가 침입 탐지의 수단으로서 사용되기 위해 통계적으로 각 사용자의 profile에 기록되고 이러한 통계는 사용자의 역사적 profile을 형성한다. 따라서 통계적 Profile-Based 비정상 행위 탐지에서는 관찰된 측정값과 profile에 저장된 측정값을 비교하여 비정상 행위를 탐지해 내는 기법이다. Anderson이 설정된 사용 패턴으로부터 이탈된 사용을 모니터하므로써 침입 탐지가 될 수 있다는 연구보고서의 발표이후 Denning과 Newmann은 통계적 profile-based 비정상 행위 탐지를 강조한 더욱 완전한 기법을 제시했다. 이 기법이 SRI의 IDES(Intrusion Detection Expert System)모델에 적용되어서 실시간 감사자료 분석 도구로 활용되었는데, 물론 IDES는 Profile-Based 비정상 행위 탐지와 이미 알려진 침입을 탐지에 적용되는 규칙 베이스 기법등 두가지 기법을 다 적용한 시스템이다[Lunt93].

이 기법의 장점은 예상하지 못한 침입 행위들에 대해서도 탐지해 낼 수 있는 반면에 만약 사용자의 사용 패턴을 잘 알고 있는 내부 침입자가 사용자 패턴을 점차적으로 바꾸어서 침입할 경우에는 쉽게 모니터링이 불가능하다는 단점이 있다.

3.2 신경회로망(Neural Networks)을 이용한 침입 탐지

사용자의 관찰된 행위를 과거 사용자 행위의 모델과 매칭이 어려운 이유는 사용자 행위가 아주 복잡하기 때문이다. 감사자료로부터 사용자별 통계를 구성할 때 잘못된 기준은 거짓 경보(false alarms)를 발생할 수 있으며, 정상 행위로 부터 침입 행위를 구별 못할때 미 탐지(missed detection)가 나타난다. 이러한 경우 신경회로망 기법을 적용하면 쉽게 해결 할 수가 있다. 이 기법은 연속적인 정보(명령어)를 가지고 신경회로망을 학습시키는 것이다. 신경회로망에 입력은 현재 사용자 행위와 과거 다양한 사용자 행위들로 구성 된다. 한번 신경회로망이 사용자 행위에 대해서 학습이 되어지면, 신경회로망은 사용자의 profile을 형성하고 이러한 profile로부터 사용자 행위의 변화들을 측정한다. 신경회로망 사용을 나타내는 개념적인 다이어그램이 <그림 3-1>에 나타나 있다[Kuma95].



<그림3-1> 침입 탐지에 있어서 신경회로망의 사용

입력 계층에 화살표들은 최근 사용자의 행위들이다. 이 단계에서 나타내어진 모든 입력은 각각 유일한 값으로 인코드(encode)되어서 최근 행위들과 정확하게 대응된다. 출력 계층에서는 예상되는 다음 사용자 행위를 하나로 나타낸다. 신경회로망의 단점으로는 망의 토폴로지(topology)나 가중치의 부여가 상당한 시행착오 후에 결정되고, 입력되는 명령어의 크기를 결정하는 문제가 어렵다.

장점으로는 자료에 관한 통계적인 가정에 종속되지 않으며, 신경회로망은 모호한 사용자 행위에 잘 대처할 수 있다. 그리고 하나의 측정 기준이 모든 사용자에게는 비효율적이지만 일부 특정한 사용자에게는 유익할 경우, 신경 회로망은 출력에 영향을 주는 다양한 측정기준들간에 자동적으로 조정이 가능하게 한다.

3.3 오용 침입 탐지(Misuse Intrusion Detection)

오용 침입 탐지란 미리 침입에 대해서 상세하게 정의해 놓은 다음, 침입의 발생을 예의주시하는 방법이다. 통계적 기법으로 모든 유형의 침입을 탐지한다는 것이 적합하지 않기 때문에 대부분의 침입 탐지 시스템에는 오용 (misuse) 탐지 구성요소들이 포함되어 있다. 오용 침입 탐지에서 침입 징후(Intrusion Signature)는 특징과 조건, 그리고 침입이나 또는 오용에 이르는 사건들간에 상호 관계를 나타낸다. 징후(signature)는 침입을 발견하는데 유용할 뿐만 아니라 침입을 시도하는데도 유용하다.

오용 침입 탐지기는 시스템이 위험한 상태(compromised state)를 야기하는 입력 사건들의 패턴을 토대로 침입들을 단순히 알려 주기만 한다. 따라서 최초 상태가 명시되지 않고 단순히 침입 징후들만 나타낸다는 것은 때로는 침입을 탐지하는데 부족하다. 오용 침입 탐지 시스템의 대표적인 것이 전문가 시스템이다. 전문가 시스템에서는 이미 알려진 시스템의 취약점들과 의심스런 행위에 관한 공격 시나리오의 정보를 규칙 베이스에 저장한다. 따라서 모니터 되어진 감사자료를 가지고 그 행위가 의심스러운지를 결정하기 위해 저장된 규칙들과 비교 된다. 통계적 분석 기법과 주요 차이점은 감사자료내의 사용 패턴을 식별하기 위해 통계적 공식을 사용하는 대신에 규칙베이스는 사용 패턴을 나타내고 저장하기 위해 규칙들의 집합을 사용한다는 점이다. 전문가 시스템 관점에서 감사자료는 사실(fact)로서 간주되고 이러한 사실들은 규칙베이스내의 규칙의 범위를 정한다. 만약 사실과 규칙이 일치되면 침입을 결정하기 위한 분석이 수행된다. 이 기법의 단점은 너무 감사자료에 의존적이며 이미 알려진 취약점과 공격들에 대해서 탐지가 가능하지만 아직 시도되어지지 않은 공격들에 대해서는 탐지가 불가능 하다는 것이다[Kuma95].

3.4 모델 베이스 추론(Model-Based Resoning)

이 접근법은 Garvey와 Lunt[GaLu91]에 의해서 제안되었으며 이것은 오용 모델과 오용의 발생에 대해서 결과를 입증할 수 있는 증거 추론(evidential reasoning)을 결합한 오용 침입 탐지의 변형이다. 공격 시나리오의 데이타베이스는 공격을 이루는 일련의 행위로 구성되어 있다. 어떤 순간에 이러한 공격 시나리오는 시스템을 공격하에 놓이게 한다. 이러한 공격 시나리오(가설)를 반박하거나 입증하기 위해서 감사자료내에 있는 정보를 찾아서 이러한 가설과 검사한다. 이러한 프로세스를 Garvey와 Lunt[GrLu91]는 anticipator라 불렀고, anticipator는 감사증적내에서 검사되어질 다음 행위의 집합을 생성해서 planner에게 넘겨 준다. planner는 가설된 행위를 감사자료의 특정한 형식으로 변환하여서 다음에 나타나는 감사자료와 비교한다. 가설된 행위와

활동(activity)의 mapping은 감사증적내에서 쉽게 인식되어야하고 감사증적내에서 출현 가능성

$$\frac{P(Activity|Behavior)}{P(Activity|\neg Behavior)}$$

이 높아야 한다. 즉, 가설되지 않은 행위로부터 감사증적내의 활동이 일어날 확률 P(Activity | ¬Behavior) 보다 가설된 행위로부터 감사증적내의 활동이 일어날 확률 P(Activity | Behavior)이 반드시 커야 한다. 가설된 시나리오에 대한 증거가 모아짐에 따라 현재 모델의 내용이 수정되어서 공격시나리오의 발생 확률이 시스템에 내장된 증거추론식(evidential reasoning calculus)에 의해서 수정되어 진다.

모델 베이스 침입 탐지의 장점으로는 많은 자료들중에서 관련있는 데이타에 한해서 범위를 좁혀서 선택적으로 검사를 함으로써 주어진 시간에 단지 필요한 작은 양의 자료만 검사가 가능하다. 또한 정의된 침입 모델을 기초로 해서 시스템이 다음 침입자의 행위가 어떤 것인가를 예측할 수 있다. 단점으로는 아직 이 모델은 프로토타입에 의해서 실행의 효율성이 검증되지 않았다.

4. 규칙 베이스 침입 탐지 시스템 설계 제안

4.1 상태전이를 이용한 침입의 표현

모든 컴퓨터 침입 사항이 갖는 두가지 기본 가정 사항은 다음과 같다.

① 공격자는 최소한의 접근 권한을 가지고 있다.

② 침입 완료후 공격자는 전에 갖지 못한 능력을 획득하게 한다.

본 논문에서 침입은 공격자가 시스템상의 최초상태에서 목표상태로 변화됨으로써 수행되는 행위의 순서로 정의되며, 최초 상태(initial state)는 침입 수행 바로 이전 시스템의 상태이며 위험한 상태(compromised state)는 침입 완료이후 나타나는 시스템의 상태를 말한다.

<표4-1>은 전자우편의 보안 취약점을 이용해서 불법적으로 root의 권한을 얻는 과정을 버클리 버전(BSD4.2)에서 보여 주는데 메일 유틸리티에서 메일은 메세지를 추가하고 소유자를 변경한 화일에 화일의 setuid 비트를 reset하지 못한다. 그 결과 공격자는 메일에 루트권한을 갖는 setuid 쉘 프로그램을 만드는 속임수를 쓸 수 있고 그것을 공개적으로 실행할 수 있다.

<표4-1> 침입 유형 A의 공격 시나리오

단계	명 령 어	내 용
1.	%cp /bin/csh /usr/spool/mail/root	- 메일 디렉토리내 root 메일 화일이 없는 것으로 가정
2.	% chmod 4755 /usr/spool/mail/root	- setuid 화일을 만듬
3.	% touch x	- 공(empty)화일을 생성
4.	% mail root < x	- 공(empty)화일을 root로 전송
5.	% /usr/spool/mail/root	- setuid가 enable된 root 쉘의 실행
6.	root%	

단계1에서 공격자는 C 쉘의 복사본을 만들고 그것을 root의 메일 화일로 이름을 부여 한다. 이단계가 성공적일려면, 공격자는 루트가 읽지 않은 메일이 없을 동안 기다려야만 한다. 그렇지 않으면 공격자는 가짜 메일 화일을 만들 수 없기 때문이다. 단계2에서 공격자는 가짜 메일 화일의 setuid 비트를 활성화 시킨다. 단계3과4에서 공격자는 공(empty)메세지를 만들어서 메일 유틸리티를 이용하여 루트에 보낸다. 보안 결함은 단계4에서 메일이 화일 소유자 속성을 루트로 set하기 전에 /usr/spool/mail/root 화일의 setuid 비트의 reset을 실패할때 일어난다. 단계5에서 공격자는 루트 권한을 가진 쉘에 액세스를 얻기위해 루트메일 화일을 실행시

키기만 하면 된다. 물론 여기서 공격자는 시스템 프로세스에 사용자 계정이나 원격 프로세스를 통해서 접근이 가능해야 한다는 가정(assertion)을 하고 시스템 침입은 표준 BSD4.2 유닉스 구성을 적용하였다. 침입 유형 A를 일련의 상태전이로 모델화하기 위해서는 반드시 최초 요구상태(initial requirement state)와 대상 시스템의 위험한 상태(compromised state)를 먼저 확인해야 된다. 위의 시나리오를 성공적으로 수행하기 위해서는 다음 가정(assertion) 사항들을 반드시 포함해야 한다.

1. 공격자는 반드시 메일 디렉토리에 대한 쓰기 권한을 가지고 있어야 한다.
2. 공격자는 반드시 cp(1), mail(1), touch(1), 그리고 chmod(1)의 실행권한을 가지고 있어야 한다.
3. 루트의 메일 화일은 반드시 존재하지 않거나 쓰여질 수 있어야 한다.
4. 공격자는 루트가 될 수 없다.

첫번째 가정은 전체 침입이 메일 유틸리티내의 결함에 기초를 두기 때문에 필요하다. 왜냐하면 특별히 메일은 /usr/spool/mail/ 디렉토리내의 메일 화일을 찾기 때문이며, 공격자는 이 디렉토리에 가짜 메일화일을 만들기 위해 쓰기 허가를 반드시 가져야 한다.

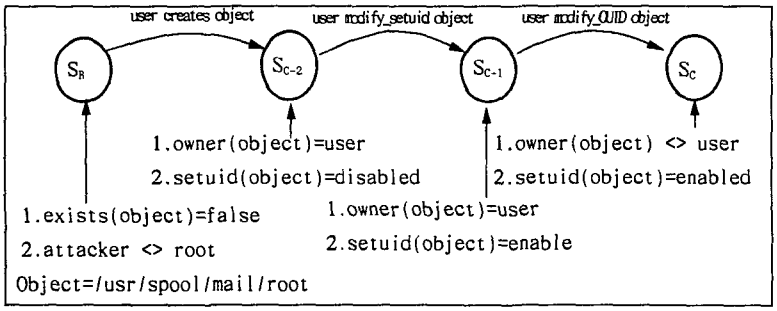
두번째 가정의 요구도 필수적이다. 나열된 어떤 화일에 대한 실행 권한이 없다면 시나리오 과정을 완성하지 못해서 공격이 이루어 지지 않는다.

세번째 가정 역시 루트의 합법적인 메일 화일이 존재하는한 공격자는 가짜 메일화일을 생성할 수 없기 때문에 필요하다.

마지막으로 네번째 가정은 하나의 프로세스가 가짜 메일화일 자신을 변조할 수 없다는 사실을 나타낸다.

이 예제는 BSD4.2 유닉스 구성을 채택하는 시스템에서 침입이 일어나는 것을 가정으로 했기 때문에 가정 1과 2는 자동적으로 표준 유닉스에서도 수행된다.

<표4-1>에서 단계5가 수행되지 않는다고 가정할 경우, 비록 공격자가 루트권한을 얻기 위해 가짜 메일 화일을 사용하지 않았더라도 루트가 생성하지 않은 setuid-to-root프로그램이 지금 존재하는 관계로 위협이 이루어졌다고 할 수 있다. <그림4-1>은 <표4-1>의 상태전이도를 나타낸다.



<그림4-1> 침입 유형 A의 상태전이도

위의 상태전이도를 간단히 설명하면, 상태 S_R은 침입 유형 A를 성공적으로 실행시키는데 필요한 최소한의 최초 요구에 대한 설명으로서 최초요구은 공격자의 가짜 화일의 생성 전에는 /usr/spool/mail 디렉토리에 메일 화일이 존재하지 않아야 함을 기술하고 그리고 공격자는 root 권한의 소유자가 될 수 없는 것을 나타내고 있다. S_R에서 S_{C-2}에 이르는 처음 선분은 공격자가 /usr/spool/mail/root 화일을 만드는 것을 의미한다. 이 징후행위의 실행결과로 침입의 상태가 최초시스템의 상태에서 S_{C-2}로 옮겨진다. 그다음 S_{C-1}의 상태로 되기위해, 공격자는 두번째 선분위에 있는 징후행위를 수행한다. 이 행위의 결과로서 /usr/spool/mail/root화일의 setuid 비트가 enable된다. 상태 S_{C-1}은 공격자가 /usr/spool/mail/root화일을 소유하고, enable된 setuid 비트를 갖는 것으로 나타난다.

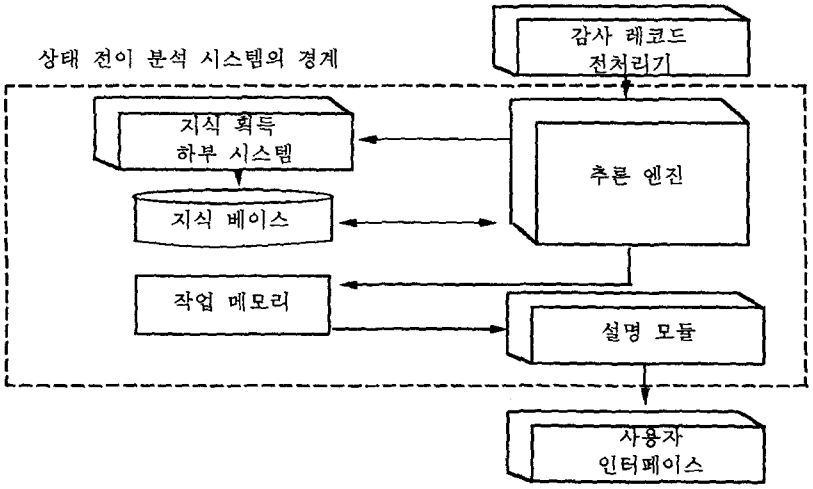
끝으로, 세번째 선분위에 표시된 징후행위의 실행으로 루트 메일화일은 수정된 소유자 속성을 갖는다. 마지막 징후행위로 인해 /usr/spool/mail/root 화일의 소유자는 바뀌었고, setuid 비트가 enable한 최종 위험한 상태 S_C가 된다.

4.2 상태전이를 이용한 침입 탐지 시스템 설계

4.2.1 실시간 침입 탐지 시스템의 주요 구성요소

전문가 시스템의 지식베이스 모듈, 추론엔진 모듈, 지식획득 모듈, 설명 모듈, 사용자 인터페이스 모듈로 구성된다[김화95].<그림4-1>은 상태전이 분석 전문가 시스템의 주요 구성을 나타내고 있다. 본 시스템은 지식베이스,추론 엔진, 설명 모듈로 구성되어 있으며, 또한 외부와 2개의 구성요소를 가지고 있다.

상태전이 분석 시스템의 추론 엔진은 대상 시스템내에서 발생하는 상태 변화와 지식 베이스내에 포함된 상태전이도를 모니터하고 비교하는데 사용되는 모든 알고리즘으로 구성되어 있다. 추론 엔진으로부터 나온 출력은 설명모듈이라는 다른 구성요소로 넘겨진다. 설명모듈은 전문가 시스템에 의해서 만들어지는 응답들을 사용자에게 알리거나 앞으로 발생 가능한 위협에 대해서 사용자에게 경고하는 기능을 갖는다.



<그림4-2> 상태전이 분석 시스템의 주요 구성요소

4.2.2 타이컴Ⅱ의 감사 메카니즘으로부터 감사레코드 추출

감사레코드는 상태전이 분석시스템의 추론 엔진이 추론하기전에 분석을 위해서 사전 처리된다. 상태전이 분석시스템의 감사레코드 전처리기는 상태전이 분석에 적합한 감사 정보를 분리해 낸다.

(1) 타이컴Ⅱ의 감사레코드 형식

감사레코드의 추출을 위해서는 먼저 타이컴Ⅱ의 감사레코드를 이해해야 한다. 타이컴Ⅰ인 볼러런트시스템은 감사기능(audit facility)을 제공하지 않지만 유닉스 시스템Ⅴ 릴리즈4.2를 채택하는 타이컴Ⅱ에서는 감사기능을 제공한다. 타이컴Ⅱ의 감사레코드는 <표4-2>와 같다[Modi93].

<표4-2> 타이컴Ⅱ의 감사레코드 형식

번 호	필 드(field)	내 용
1	time	사건(event)이 일어난 시간(시간:분:초:일:월:년도)
2	event	사건 유형
3	pid	첫문자가 P로 시작하는 프로세스 ID
4	LWP-id	사건을 활성화(trigger)시킨 LWP(lightweight) id의 숫자
5	outcome	사건 결과(성공:s, 실패:f)
6	user	real ID와 effective ID를 콜론으로 구분
7	group	real GID와 effective GID를 콜론으로 구분
8	session	첫문자가 S로 시작하는 세션 ID
9	object field	객체 정보를 나타내는 필드
10	pgm_prm	각 사건에 대한 상세 설명

(2) 시스템 감사레코드와의 mapping

(가) 시스템 감사레코드 형식

본논문에서 상태전이 분석을 위한 상태전이 분석시스템의 감사레코드는 <표4-3>과 같다.

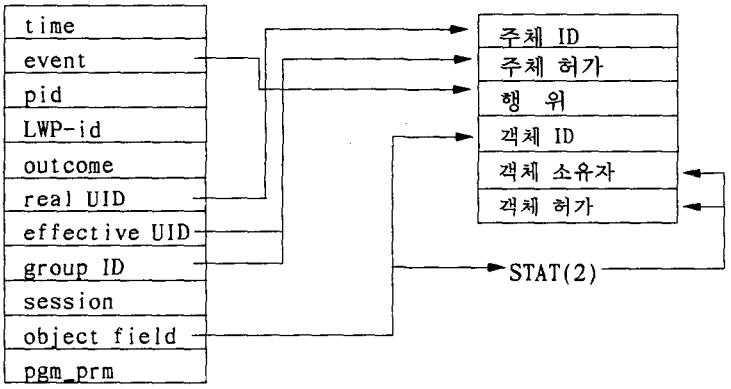
<표4-3> 상태전이 분석 감사레코드 형식

주체(Subject) ID	주체의 행위에 관한 유일한 인식자.
주체 허가	주체의 접근 권한.
행위	주체에 의해 수행되는 행위. 가능한 행위들은 객체 읽기,쓰기,생성,삭제,프로그램 실행,프로그램 종료,객체 소유자 수정, 객체 허가 수정,액
객체(object) ID	객체의 유일한 인식자나 이름. 만약 객체 액세스가 일어나지 않았다면 이 필드는 null이다.
객체 소유자	앞 필드내에 표시된 객체의 소유자. 만약 객체 액세스가 일어나지 않았다면 이 필드는 null이다.
객체 허가	객체에 관련된 접근 허용. 만약 객체 액세스가 일어나지 않았다면 이 필드는 null이다.

<표4-3>에서 주체 ID 필드는 사용자별 감사증적의 처리를 하기 위해 필요하며 이것은 침입이 탐지 되었을때 어떤 사용자인지 식별을 용이하게 한다. 주체 허가 필드는 사용자 접근 권한이 최초 상태와 공격후의 상태를 비교할때 사용되어진다. 행위 필드는 전체 상태전이 분석의 핵심으로서 주요 분석 대상이 된다. 마지막으로 객체 ID, 객체 소유자와 객체 허가 필드는 어느 객체가 주체에 의해 액세스될 것인지를 나타낸다.

(나) 감사레코드 mapping

<그림4-3>의 감사레코드 mapping에서 주체 ID는 su(1) 명령을 수행하더라도 사용자 ID가 변하지 않는 real UID로부터 직접 이동되고 주체권한은 effective UID와 group UID 필드로부터 직접 이동이 가능하다. 그리고 앞에서 언급한 행위 필드의 10개 행위들에 대해서는 타이컴II의 event 필드로부터 추출이 가능하다. 마지막으로 object 필드로부터 객체 ID가 추출 가능하지만 객체 소유자와 객체 허가는 주어진 객체 ID로부터 STAT(2) call 명령어를 사용함으로써 추출이 가능하다.



<그림4-3> 타이컴II의 감사레코드로부터 상태전이 분석 레코드의 mapping

(3) 감사레코드와 상태전이와의 대응

본 시스템은 지식베이스내에 침입를 나타내는 수단으로서 감사레코드를 사용하는 것이 아니라 시스템 내에서 일어나는 특정 침입과 관련된 상태변화들을 모니터하기 위해 감사레코드를 사용한다. 침입 유형에 대한 원시 감사레코드를 패턴 매칭하는것보다 오히려 상태 변화에 관한 분석에 초점을 둠으로써, 종래의

규칙 베이스 침입 인식 도구보다 침입내에 많은 변화를 탐지할 수 있다. 감사레코드로 부터 모니터된 상태 변화는 상태 전이도를 나타내기 위해서 사용되는 징후행위(signature action)와 대응된다. 침입 유형 A에 대한 상태전이도인 <그림4-1>에서 3개 징후행위는 다음과 같다.

	주 체	행위	객 체
1.	사용자	생성(creates)	/usr/spool/mail/root
2.	사용자	modify_setuid	/usr/spool/mail/root
3.	사용자	modify_OUID	/usr/spool/mail/root

<표4-4>는 상태전이 분석시스템의 전처리기에 의해서 포맷되어진 후 침입 유형 A의 감사정보를 나타내는 것으로서 위의 징후행위와의 대응을 나타낸다.

<표4-4> 침입 유형 A의 형식화된 감사증적

```
user% cp /bin/sh /usr/spool/mail/root
  1. <user,user,exec, "/bin/cp",root, "-rwzr-xr-x">
  2. <user,user,read, "/bin/sh",root, "-rwzr-xr-x">
  3. <user,user,create, "/usr/spool/mail/root",
      user,"-rw------">
  4. <user,user,exit, "/bin/cp",root, "-rwzr-xr-x">
```

```
user%chmod 4755 /usr/spool/mail/root
  1.<user,user,exec, "/bin/chmod",root, "-rwzr-xr-x">
  2.<user,user,modify_perm(-rwsrwxrwx),
    "/usr/spool/mail/root",user,"-rw-----">
  3.<user,user,exit, "/bin/chmod",root, "-rwzr-xr-x">
```

```
user% touch x
  1.<user,user,exec, "/bin/touch",root, "-rwzr-xr-x">
  2.<user,user,create, "x",user, "-rw-------">
  3.<user,user,exit, "/bin/touch",root, "-rwzr-xr-x">
```

```
user% mail root < x
  1.<user,root,exec, "/bin/mail",root, "-rwzr-xr-x">
  2.<user,root,read, "x",user, "-rw-------">
  3.<user,root,write,"/usr/spool/mail/root",user,"-rwsrwxrwx">
  4.<user,root,modify_OUID(root),"/usr/spool/mail/root",
     user,-rwsrwxrwx">
  5.<user,root,exit, "/bin/mail",root, "-rwzr-xr-x">
```

테이블내의 각 엔트리는 침입내에서 사용된 4개의 사용자 명령어를 나타내며, 각 명령어마다 첫번째와 마지막 감사레코드는 최초 실행과 최종적인 종료에 의해 생성된 것이다. 처음 사용자 명령어에 의해서 4개의 감사레코드가 만들어 진다. 두번째 감사레코드는 /bin/sh를 읽음으로서 생성되어지고, 세번째는 /usr/spool/mail /root화일의 생성이다. 여기서 세번째 감사레코드는 <그림4-1>의 3개 징후행위중 처음 징후행위에 일치하는 것을 나타내기 위해 반전으로 나타내었다. 두번째 엔트리는 chmod(1) 명령으로서 두번째 감사레코드는 /usr/spool/mail/root화일에 대한 액세스 허가 속성의 수정을 나타내기 위해 반전되었으며, <그림4-1>의 3개 징후행위중 침입 유형 A의 두번째 징후행위와 일치한다. 다음 엔트리는 침입 유형 A의 세번째 사용자 명령어 touch(1)의 감사레코드들로서 이 감사레코드에는 침입의 상태전이 표현에 대한 아무런 징후행위도 없다. 마지막 네번째 엔트리는 위험한 상태에 도달하는 침입 유형 A의 마지막 사용자 명령어이다. 네번째 엔트리내의 두번째 감사레코드는 표준 입력으로 부터 redirection을 통해서 화일 x가 참조되고, 세번째 레코드는 화일 x가 가짜 메일 화일에 추가되는 것을 나타낸다. 마지막으로 네번째, 반전된 감사레코드는 메일 프로그램이 setuid 비트가 enable된 상태에서 root의 메일 화일의 소유자 속성을 수정할때 생성되어 진다. 이 감사레코드는 <그림4-1>의 3개 징후행위중 마지막 위험한 상태로 되는 세번째 징후행위와 일치한다.

4.2.3 지식 베이스

(1) 사실 베이스

본 시스템에서의 사실 베이스는 4개의 화일 집합과 보안에 아주 중요하고 공격의 대상이 되기쉬운 2개의 중요화일 집합으로 구성 했다. <표4-5>는 이러한 화일 집합을 분류해 놓은 것으로서 특정한 침입으로부터 보안 침해를 받을 수 있는 화일의 특성들을 묶어 놓음으로써 침입 규칙의 일반성을 증가 시키는데 그 목적이 있다. 다시 말해서 추론 엔진에 의해 침입 유형을 판별해 낼 때 맨 처음 상태기술 테이블에서 화일 집합을 참조함으로써 이러한 침입이 어떤 유형인지를 탐지해 낼 수 있게 한다. <표4-6>은 중요화일의 집합을 나타내었는데 중요화일 집합은 읽기 제한과 쓰기 제한으로 구분 할 수 있다. 여기서 읽기 제한 화일의 경우 이 화일의 내용이 아주 민감한 정보를 가지고 있기 때문에 시스템관리자가 아닌 다른 사용자가 /dev/kmem 화일로부터 직접 정보를 읽을 경우 시스템의 보안에 위협을 받기 때문에 제한을 가했다. 그러나 합법적인 사용자가 보안과 관련 없는 정보를 ps(1) 유틸리티를 사용하여 /dev/kmem 화일을 액세스 할 경우에 액세스가 가능하도록 시스템의 구현시 검토되어야 한다.

쓰기 제한의 경우는 데이타의 화일과 공개적으로 실행가능한 시스템 유틸리티의 두가지로 분류할 수 있다. 데이타 화일로서 /etc/passwd와 같은 화일을 수정하거나 공유 실행화일들이 트로이목마나 바이러스에 감염된다면 많은 피해를 입기 때문에 쓰기 제한을 해야 할 것이다.

사실베이스에서 화일집합의 구성은 인터넷 공개 소프트웨어인 COPS나 유닉스 유틸리티 FIND(1)를 이용해서 쉽게 구성할 수가 있다.

화일 집합은 동등한 접근 권한,소유권을 공유하거나, 또는 시스템상에 동등한 기능을 제공하는 화일(예, 메일 화일)들에 의해서 주로 형성된다.

<표4-5> 화일 Set 요약

File Set #1	#!/bin/sh를 포함하는 모든 setuid/setgid가 가능한 스크립트
File Set #2	실행가능한 모든 setuid/setgid화일
File Set #3	root 소유권을 갖는 setuid/setgid화일
File Set #4	/var/spool/mail/ 디렉토리에 지정된 메일 화일
File Set #5	읽기가 제한된 File Set
File Set #6	쓰기가 제한된 File Set
File Set #7	File Set #5를 참조하는 인가된 유틸리티 집합
File Set #8	File Set #6을 참조하는 인가된 유틸리티 집합

<표4-6> 중요 화일의 예

읽기 제한 화일들	- /dev/kmem - /dev/mem
쓰기 제한 화일들	- /etc/* ; /bin/* - /.* ; /usr/etc/yp* - /usr/lib/cronab에 참조되는 화일

(2) 규칙 베이스

규칙 베이스는 추론엔진이 사실 베이스의 내용과 입력 데이타를 기초로 새로운 정보를 추론할 수 있도록 단계를 정의한다. <그림4-1>의 상태전이도를 규칙베이스내에서 구체화 될 수 있는 규칙으로 나타내기 위해서 각 침입 상태전이도의 상태기술을 저장하는 상태기술 테이블과 규칙 포맷을 설명하도록 한다.

(가) 상태기술 테이블

상태기술 테이블은 지식 베이스내에서 각 침입의 상태기술들을 저장한다. 상태기술 테이블의 행은 지식 베이스내 각 침입를 나타내고 열은 침입 상태전이도내의 각 상태와 일치한다. 상태기술 테이블은 규칙

베이스 내에서 표현되는 침입들의 개별적 상태기술에 대하여 필요한 정보를 제공한다.각 테이블의 엔트리는 시스템 규칙 베이스내에있는 하나의 규칙에 의해서 정확하게 참조되어 진다. 규칙은 추론 엔진에 의해서 모니터 되어지는 상태전이와 알려진 침입의 상태전이와 비교 할때 이러한 엔트리를 참조한다.

<표4-7>은 침입 유형A의 상태전이도를 토대로한 상태기술 테이블이다. 상태기술들은 추론 엔진에 의해 평가 되어지는 가정(assertion)의 형태로 되어 있으며, 추론엔진은 상태기술 테이블에서 파일 집합을 참조한다. 규칙 베이스내의 규칙은 상태기술 테이블내의 상태기술을 참조하고 상태기술들은 차례로 사실 베이스내의 객체 집합을 참조한다.

<표4-7> 상태기술 테이블 예

구분	상태S_1	상태S_2	상태S_3	상태S_4	상태S_5
침입 유형 P_A	1.exists(file)= false, file exists in File Set #4 2.attacker<>root	1.owner(file)= user 2.setuid(file) = disabled 3.name(file) exists_in File Set #4	1.owner(file) = user 2.setuid(file) = enable 3.name(file) exits_in File	1.owner(file) = user 2.setuid(file) = enable	

(나) 규칙 형식

규칙들은 추론 엔진에 의해서 모니터 되어지는 사용자 행위와 공격 시나리오 진행을 알리는 상태변화간의 관계를 설정하기 위해 사용되어 진다. 특별한 침입과 관련이 있는 모든 규칙들은 규칙 체인(rule-chain)으로 불린다. 상태전이도내의 각 상태전이는 상태전이 분석시스템 규칙 베이스내에서 서로 일치되는 규칙을 가지고 있다. 규칙은 세개의 필드로 나누어 진다.

규칙 형식: (상태기술,징후행위, 규칙 종속)

상태기술: (행,열)은 상태기술 테이블 엔트리와 동등하다.

징후행위: 상태기술 필드와 일치하는 상태에서 상태전이도내의 다음상태로 이르는 징후행위

규칙종속: 규칙 체인내에서 이 규칙이 다른 규칙에 종속을 설명하는 정규 표현식

각 필드의 기능을 이해하기 위해서는, 전문가 규칙에 적용 되어지는 상태전이 용어를 정의할 필요가 있다. 상태전이는 두개 항목의 결합 즉, 시스템의 상태와, 징후행위 그자체이다. 예를 들면, <표4-8>의 침입A에서 처음 상태전이도의 상태전이는 상태 S_1과 처음 징후행위의 결합으로서, S_2상태에 이른다. 그래서, 규칙의 처음 두개 필드는 하나의 상태전이를 나타낸다.

<표4-8> 침입 A의 규칙체인

구 분	상태기술	징후행위	규칙 종속
규칙 1	< SDT(P_A,S_1)>	<user creates file>	<0>
규칙 2	< SDT(P_A,S_2)>	<user modify_setuid file>	<규칙 1>
규칙 3	< SDT(P_A,S_3)>	<user modify_owner file>	<규칙 2>

<표4-8>에서 상태기술 필드는 상태 테이블의 열과 행을 참조한다. 예를 들면, 침입 A의 처음 규칙으로 부터 상태기술 필드는 SDT(P_A,S_1)와 같이 쓰여진다. 이 필드는 <표4-7>에 나타나 있는 상태기술 테이블의 1행, 1열을 참조한다. 침입 A의 첫번째 규칙에 대한 징후행위 필드는 상태 SDT(P_A,S_1)에서 SDT(P_A,S_2)에 이르는 징후행위와 일치한다. 규칙 종속 필드는 자신의 상태전이에 의존하는 다른 규칙들의 연결을 나타낸다.

예를 들면, 침입 A의 첫번째 규칙에 나타나 있는 상태전이는 그 이전 어떤 상태전이에도 종속되지 않는다.그래서 이 규칙 종속 필드의 값은 제로이다. 침입 A의 두번째 상태전이에서, 사용자는 처음 상태전이내에 생성된 화일을 실행 시킨다. 사용자가 존재하지 않는 화일을 실행시키는 것은 불가능하기 때문에 규칙2의 상태전이는 규칙1에 종속적이다. 그러므로 규칙 2의 규칙종속필드는 규칙 2가 오직 규칙 1이 먼저 수행하고

나서 반드시 수행되어야 한다는 것을 명시하고 있다.

4.2.4 추론엔진

본 시스템에서의 추론 형식은 사실들의 집합과 입력을 만족시키는 규칙을 찾아서 추론을 용이하게 하기 위해서 전향 추론 시스템의 적용이 바람직하다고 생각되며,전통적인 전향추론은 If-Then 의 구조로 쓰여진다.

If <선행부>
Then <결론부>

<표4-8>을 If-Then절 규칙 포멧을 사용해서 다음과 같이 표현되어 질 수 있다.

Rule 1:
If (Sig_Action(P_A,A_1) occurs and SDT(P_A,S_2) holds)
Then stat_Transition(S_1,A_1) occured

Rule 2:
If (Sig_Action(P_A,A_2) occurs and SDT(P_1,S_3) holds)
Then stat_Transition(S_2,A_2) occured

Rule 3:
If (Sig_Action(P_A,A_3) occurs and SDT(P_A,S_4) holds)
Then stat_Transition(S_3,A_3) occured

각 규칙의 선행부는 결론부가 추론 되어질수 있는 조건을 명시하는데, 규칙1에서 상태전이(S_1,A_1)(즉, 침입 A의 처음 상태전이)는 다음 두개의 조건하에서 추론 되어질 수 있다.

1. 침입1의 처음 징후행위가 목격되어졌고(즉, Sig_Action(P_A,A_1))
2. 상태기술 테이블의 엔트리 SDT(P_A,S_2)내에 정의된 상태기술 가정이 Sig_Action(P_A,A_1)의 결과로서 갖는다. 또한 규칙2는 침입 A의 두번째 징후행위가 목격되어 지고, 상태기술 테이블의 엔트리 (P_A,S_3)에서 정의되어진 상태기술 가정이 결과로서 가진다면 침입 A의 두번째 상태전이가 일어난다고 말할 수 있다. 규칙3 역시 세번째 징후행위가 목격되어지고, 상태기술 가정이 결과로서 갖는다면 세번째 상태전이가 일어난다. 그러나 위의 규칙에는 세개의 핵심 조건들이 생략 되었다. 첫번째 조건은 침입 A의 선행 요구 상태(즉, 상태기술 가정(P_A,S_1)을 지닐때 규칙1의 선행부가 평가 되어져야 한다는 것이다. 두번째로 규칙2의 선행부는 침입 A의 처음 상태전이(즉, 상태전이 (S_1,A_1))가 일어난 후에 평가 되어져야 하고, 세번째도 두번째의 상태전이(S_2, A_2)가 일어난 후에 평가 되어져야 한다는 것이다. 따라서 수정된 규칙은 다음과 같다.

Rule 1A:
If (SDT(P_A,S_1) held)
If (Sig_Action(P_A,A_1) occurs and SDT(P_A,S_2) holds)
Then stat_Transition(S_1,A_1) occured

Rule 2A:
If (stat_Transition(S_1,A_1) occured)
If (Sig_Action(P_A,A_2) occurs and SDT(P_A,S_3) holds)
Then stat_Transition(S_2,A_2) occured

Rule 3A:
If (stat_Transition(S_2,A_2) occured)
If (Sig_Action(P_A,A_3) occurs and SDT(P_A,S_4) holds)
Then stat_Transition(S_3,A_3) occured

If-Then 규칙의 첫번째 레벨은 대응되는 침입 규칙 체인의 규칙 종속 필드에 위치한 규칙 종속을 나타낸다. 다음 레벨의 선행부는 예상되는 징후행위를 나타내며, 이것으로 인한 상태 가정은 징후행위의 결과로서 반드시 유지해야 한다. 마지막으로 내부 결론부는 새로이 수행된 규칙을 나타낸다.

4.2.5 설명모듈

설명모듈은 추론엔진에 의해서 수행된 침입 규칙들을 근거로해서 상태전이 분석 시스템이 사용자 인터페이스에게 어떤 메세지를 보내야하는가를 결정하는 메카니즘이다. 설명모듈은 각 주체들이 어느정도까지 위협에 도달되었는지를 식별하고, 적절히 사용자 인터페이스에 대한 메세지를 생성한다. 설명모듈은 추론엔진에 의해서 유지되는 수행된 규칙을 모니터링 하므로써 각 주체의 진행상태를 식별한다. 설명 모듈은 수행된 규칙에 대한 응답을 하기 위해 설명 테이블을 가지고 있다. <표4-9>의 설명 테이블은 상태전이 분석시스템 규칙 베이스내의 모든 규칙에 대하여 열과 행으로 구성되어 있으며, 각 규칙에 대응되는 설명 테이블 엔트리는 규칙이 수행되어 질때 취해지는 적절한 응답들을 포함하고 있다.

<표4-9> 설명 테이블의 예

구분	상태S_1	상태S_2	상태S_3	상태S_4	상태S_5
침입 P_A	nil	nil	nil	위협 탐지!!!(root 권한에 비인가자의 액세스) 사용자가 root의 유효 uid를 획득 했슴.	

사용자 인터페이스에 어떤 응답들이 보내져야 할 것인가의 최종 결정은 시스템 의존적이거나 다음과 같은 몇개의 요소를 토대로 한다.

- o 침입의 위협에 대한 가혹성을 나타내는 경고 메세지
- o 다음 적절한 규칙이 수행하는 것을 방지하기 위해서 어떤 단계가 취해져야 하는가에 관한 제안들
- o 새로이 수행될 규칙에 의해서 나타내어지는 상태전이를 취소하기 위해 어떤 단계가 수행되어져야 하는가에 관한 제안들

이러한 경보 메세지들이 텍스트로 설명되어 테이블 엔트리에 저장될 수 있고 사용자 인터페이스에게 넘겨진다. 높은 등급의 보호를 요구하는 대상시스템에서, 상태 분석 시스템 설명 테이블은 최종 위험한 상태에 도달하기 전에 침입을 저지하기 위해 취해질 수 있는 간결한 명령어 집합을 저장 해서, 시스템 보안 담당자에게 어떠한 행위가 취해져야 할 것인가를 나타낼 수 있다. 여기에 대한 구체적인 행위들은 새로 생성된 객체를 제거하는 것과 의심스런 공격자의 세션을 종결 시키거나 또는 다음에 액세스 될 객체를 막는 것 등의 행위가 포함 될 것이다.따라서 이러한 구체적인 행위가 수행 가능하도록 상태전이 분석시스템의 구현이 이루어 져야 할 것이다.

5. 결 론

본 논문은 상태전이 분석 기법을 적용하여 침입을 표현하고 분석하는 방법과 국산 주전산기인 타이컴Ⅱ의 감사기능을 이용하여 침입 탐지 시스템의 설계 제안에 대하여 연구를 하였다. 상태전이 분석 시스템이란 시스템에서 알려진 결함을 이용한 침입에 대해서 상태전이도를 이용하므로써 침입의 표현이 가시적이고 이해가 용이할 뿐만 아니라 규칙의 생성 절차를 편리하게 할 수 있는 침입탐지시스템이다. 또한 기존의 규칙베이스 시스템이 침입 시나리오에 대한 원시 감사레코드를 패턴 매칭하는 것보다 상태전이 분석 시스템은상태 변화에 관한 분석에 초점을 둠으로써 침입내에 많은 변화를 탐지할 수 있다. 또한 시스템의 상태 변화에 기초를 둠으로써 시스템의 보안침해를 받기 직전의 임박한(Impending) 위협에 신속히 대처할 수 있을 것이다.

본 논문에서 침입 탐지 시스템의 설계에 적용한 규칙 베이스 전문가 시스템은 일반적인 시스템을 기준으로 하였으나 시스템의 특성에 따라 목적에 맞는 전문가 시스템 도구를 사용할 수 있다. 타이컴Ⅱ에서 보다 완전한 침입 탐지 시스템을 구현 하기 위해서는 본 논문에서 연구되어진 이미 알려진 시스템의 결함에 대해서 규칙 베이스 침입 탐지 시스템의 사용과 Profile-Based 비정상 행위 탐지 기법인 통계적 분석 기법을 혼합한 방법으로 침입 탐지 시스템이 구현 되어야 하며, 실시간에 감사자료의 분석을 위해서는 타이컴Ⅱ의 감사 메카니즘에 의해서 생성되어지는 감사자료가 디스크에 쓰여지기 전에 작업 메모리로부터 침입 탐지 시스템의 입력자료로 활용되어 질 수 있도록 시스템의 커널 수준에서 이러한 기능이 구현되어 져야 할 것이다.

참고 문헌

[김화95]김화수,조용범,최종욱, 전문가 시스템, 집문당, 1995

[신이93]신종태,이대기, "침입 감지 모델 설정과 시스템의 분석," 통신정보보호학회지, 제3권 제3호,pp23-28,1993

[연암94]김진식,장환용,김영민,김희상, 해커들의 해킹 기법,연암출판사,1994

[조원94]Richard J. Smith Mark Gibbs 원작,조원희 역, "인터넷의 모든것," 인포·북,1994

[GaLu91]Thomas D. Garvey & Teresa F. Lunt, "Model-Based Intrusion Detection," In Proceedings of the 14th National Computer Security Conference, pp 372-385,1991

[Kuma95]Sandeep Kumar, "Classification and Detection of Computer Intrusions," Purdue University, 1995

[Lunt93]Teresa F. Lunt "A survey of intrusion detection techniques", Computer & Security, Vol. 12, No. 4, 1993

[Modi93]Mark Modig, Audit Trail Administration, UNIX press,1993

[Porr92]Phillip Andrew Porras "A State Transition Analysis Tool For Intrusion Detection," University of California, 1992