실시간 침입 탐지를 위한 에이전트 모델의 설계

이 문 구[†] · 전 문 석[††]

요 약

기존의 침입 탐지 방법은 침입이 발생했을 때 침입을 실시간으로 탐지하지 못한다. 이유는 침입 패턴에 감사 데이터를 처리하는데 많은 시간이 소모되기 때문이다. 이러한 문제점을 해결하기 위해 실시간 침입탐지에 대한 연구가 많이 진행되고 있다. 따라서, 본 논문은 실시간 침입 탐지를 위하여 다단계 경고레벨을 사용하는 에이전트 모델을 제안한다. 이 실시간 침입 탐지 모델은 에이전트의 확장성과 에이전트간의 통신 메커니즘을 이용하여 분산 환경에 적용하여 사용할 수 있으며, 기존의 침입탐지시스템의 이식성이나 확장성, 비밀성들을 제공한다.

A Design of Agent Model for Real-time Intrusion Detection

Moon-Ku Lee[†] · Moon-Seog Jun[††]

ABSTRACT

The most of intrusion detection methods do not detect intrusion on real-time because it takes a long time to analyze an auditing data for intrusions. To solve the problem, we are studying a real-time intrusion detection. Therefore, this paper proposes an agent model using multi warning level for real-time intrusion detection. It applies to distributed environment using an extensibility and communication mechanism among agents, supports a portability, an extensibility and a confidentiality of IDS.

1. 서 론

기존의 방화벽이나 인증, 암호화 방법은 허가 받지 않은 사용자의 불법 침입을 방어해주는 차원에서 제공되지만, 현재의 해커들은 이러한 방법들을 피해서 계속적으로 지능적인 침입 패턴으로 해킹을 시도하고 있다. 이러한 지능적인 내부망에서의 침입을 막기 위해서는 불법적으로 컴퓨터 시스템에 침입하여 중요한 정보들을 손상시키는 행위들을 실시간에 탐지하고 중지시킬 수 있는 실시간 침입탐지 시스템 개발이 필요하다. 본 논문에서는 실시간 침입탐지를 위한 에이전트 모델을 제안하였다. 제안한 실시간 침입 탐지 에이전트 모델은 각 호스트 상에서 동작하는 에이전트들이 침입 패턴에 따라 다단계의 침입 위험도와 침입의 전송도에 따라 침입의 경고 레벨을 두고 이를 탐지하여 실시간에 침입의 정도를 알리도록 한다. 본 논문의 구성은 다음과 같다. 2장에서는 침입 탐지와 에이전트의 관계에 대해서 기술하고, 3장에서는 실시간 침입 탐지시 고려해야 할 사항들을 살펴보고, 4장에서는 본 논문에서 제안한 실시간 침입 탐지를 위한 에이전트 모델의 구조와 침입 탐지를 실시간에 처리하기 위해 사용되는 침입의 위험도와 전송도 및 경고레벨을 설명하며, 5장에서는 최근의 해킹 기법중 세 가지 침입패턴만 에이전트 모델에서 탐지되는 것을 제시하였다.

† 준 회 원 : 숭실대학교 대학원 전산과
†† 종신회원 : 숭실대학교 컴퓨터학부 교수

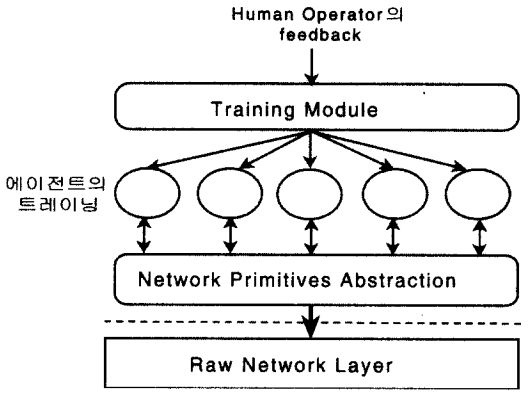
논문접수 : 1999년 7월 14일, 심사완료 : 1999년 10월 21일

2. 침입 탐지와 에이전트

2.1 에이전트

에이전트들은 서로 독립적으로 실행하는 개체이기

때문에 자율적이며, 동적으로 시스템에 추가되거나 삭제 또한 가능하다.



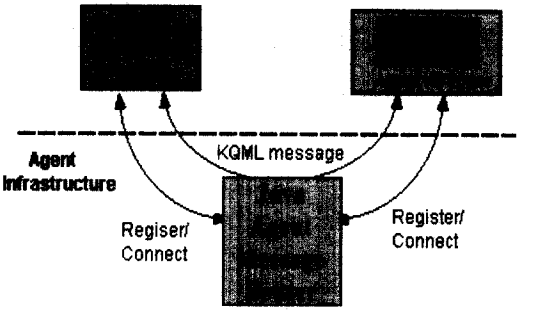
(그림 1) 에이전트를 이용한 침입탐지

에이전트를 이용한 침입 탐지 방법은 단일 침입 탐지 모듈에 비해 많은 장점을 지니는데, 다수의 에이전트를 이용하여 복잡한 침입 탐지 시스템 구축이 가능하며, 에이전트의 추가 및 삭제가 용이하여 큰 시스템으로의 확장과 구축이 쉽다.

2.2 자바를 이용한 에이전트 모델

자바 에이전트는 플랫폼에 독립적이므로 이 기종간에 에이전트 구성이 가능하고, 주어진 에이전트의 성격에 따라 필요한 클래스만으로 구성이 가능하다. 에이전트간에 KQML(Knowledge Query and Manipulation Language)를 이용한 메시지 교환을 통하여 효율성을 증가시킬 수 있도록 하였다. 자바를 이용한 에이전트 모델 시스템의 예로 JATLite(Java Agent Template, Lite)가 있다. JATLite는 Java로 작성된 프로그램 패키지로서 인터넷상에서 안정적으로 데이터를 주고받는 에이전트를 빠르게 생성 가능하다. 이러한 JATLite는 에이전트 등록 기능, 연결 및 해제 기능, 메시지 전송 및 수신 기능, 파일 전송 기능 그리고 다른 컴퓨터의 프로그램이나 활동 실행 기능 등이 있다[1].

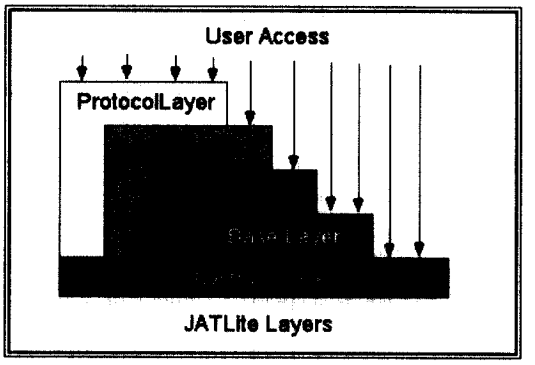
기존의 소프트웨어를 처리하는 JATLite의 접근 방법에서 에이전트 메시지 라우터는 에이전트에 관한 정보를 관리하며, 에이전트의 연결 및 해제, 메시지 전송 및 수신에 관여하게 된다. JATLite의 계층 구조는 (그림 3)과 같다.



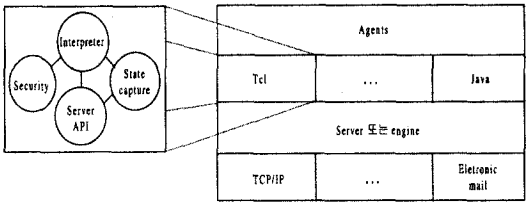
(그림 2) 자바 에이전트 라우터의 동작의 예

2.3 Aegent/Tcl을 이용한 에이전트 모델

AgentTcl은 Unix 워크 스테이션에서 실행되는 간단한 에이전트 시스템으로 복잡한 에이전트들을 빨리 개발하도록 허용한다[3], [4]. AgentTcl은 현재 Telescript와 같은 상업적인 시스템의 특징이 부족하지만[5], 작거나 중간 정도 크기의 어플리케이션을 개발하고 에이전트들의 실험을 위한 효율적인 플랫폼이다. AgentTcl 에이전트는 Tool Command Language(Tcl)의 확장된 버전으로 작성되었다. Tcl은 강력하고 배우기 쉬운 상위레벨의 스크립트 언어로, "기존의 어플리케이션과 도구들을 제어하고 확장하도록" 설계되었다[6]. AgentTcl의 설계 구조는 Telescript[7]의 서버 모델과 ARA[8]와 Dixie[9]의 다중 언어, Dartmouth의 두 개의 이전에 존재하던 시스템의 전송 메커니즘[10]을 기반으로 설계되었다. 구조는 네 가지 단계로 이루어져 있는데, 가장 하위 레벨은 유용한 전송 메커니즘 API를 나타내고, 두 번째 레벨은 에이전트가 전송할 수 있는 각 네트워크에서 실행하는 서버를 나타낸다.



(그림 3) JATLite 계층구조



(그림 4) AgentTcl의 구조

Agent Tcl 구조의 세 번째 레벨은 각 이용 가능한 언어에 대해 하나의 인터프리터로 구성된다. 각 인터프리터는 인터프리터 및 악의 있는 행위로부터 에이전트를 보호하는 보안 모듈, 실행되는 에이전트의 내부 상태를 캡쳐 하는 상태 모듈, 이동, 통신, 체크를 취급하는 서버와 의 대화를 위한 API 등의 네 가지 구성요소를 갖는다. Tcl 에이전트를 위한 보안 모듈은 Safe Tcl 확장 버전으로 존재하며, Tcl 인터프리터가 위험한 명령어를 접속체크를 수행하는 안전한 명령어로 변경되는 것을 허용한다[9].

2.4 Perl을 이용한 에이전트 모델

Perl을 이용한 에이전트 모델중 AAFID(Autonomous Agents for Intrusion Detection)2 시스템은 퍼듀 대학에서 에이전트를 이용하여 개발한 침입 탐지 시스템이다. AAFID2 시스템은 에이전트, 트랜시버, 모니터의 세 가지 구성요소들로 구성된다[2]. 이 구성요소들은 AAFID 개체(entities)나 단순히 개체라고 불리고, 이 개체들로 구성된 전체 침입 탐지 시스템은 AAFID 시스템이라고 불린다. AAFID 시스템은 네트워크상의 많은 호스트들에 널리 분포된다. 각각의 호스트는 많은 에이전트들을 가지고 있으며, 에이전트는 호스트에서 발생하는 이벤트들을 감시한다. 하나의 호스트에서 모든 에이전트들은 트랜시버에게 그들의 발견사항을 보고한다. 트랜시버는 호스트 당 하나씩 존재하며 호스트에서 동작중인 모든 에이전트들의 행위를 감독한다. 트랜시버는 호스트에서 동작하는 에이전트들을 제어하며 에이전트에게 구성(configuration) 명령어들을 시작하거나 중단하고 전송할 수 있다.

3. 실시간 침입 탐지의 고려 사항

3.1 실시간 침입 탐지의 고려사항

실시간 침입 탐지 기능을 수행하기 위해서는 먼저 다중 호스트들을 대상으로 하는 네트워크 기반 침입탐지에 대해 분석해야 한다. 다중 호스트들의 침입을 탐지할 때의 문제점 중 첫째, 이 기종 호스트들의 환경에서 고려해야 할 사항으로 다중 호스트들은 서로 다른 환경에 존재하기 때문에 각 호스트의 구조나 운영체제, 감사 서브 시스템의 특징들을 고려해야 한다. 두 번째 문제점은 분산 침입 패턴이다. 침입 패턴 탐지는 각 호스트의 취약점 집합에 의해 결정되며, 각각의 호스트에서 개별적으로 처리되기 때문에 호스트 레벨에서는 정상이지만 네트워크 레벨에서 볼 때 보안상 취약한 사용자 행위들을 탐지하기 어렵다. 전체 사용자들의 행위 감시는 각 호스트에서 발생한 방대한 양의 감사 데이터의 분석을 필요로 한다. 세 번째 문제점은 다중 호스트들간의 이벤트의 순서를 나타내기 위한 전체 시간의 설정이다. 각 호스트간의 시간과 통신 지연은 감사 데이터의 타임 스탬프에서 불 일치성을 나타내고, 이벤트 횟수, 시간들과 같은 오용행위를 탐지하는데 사용하는 측정 방법들에 영향을 준다.

4. 제안한 에이전트 모델

4.1 공격 패턴의 분석

효율적이고 확장적인 침입 탐지를 수행하기 위해서는 침입 패턴을 계층적으로 분석할 필요가 있다. 다음은 계층적으로 침입 패턴을 분석하기 위한 요구사항들을 나타낸다:

- 분류의 상위 계층에서 공격 패턴들을 상세하게 추상화함으로써 침입 패턴의 일반화는 상위 레벨과 하위 레벨의 상세한 부분에서 적용 가능해야 한다. 다양한 플랫폼에서 특정 공격 행위의 상세한 부분은 하위 레벨에서 상세하게 남겨지기 때문에 이러한 공격 패턴의 분류는 다양한 플랫폼에 적용 가능하게 된다.
- 공격 패턴을 감시하는 기술보다 침입 패턴의 유형을 분류하는 것은 감사 레코드에서 이러한 공격 패턴을 효율적으로 인식하는 것을 허용한다.
- 공격 패턴의 공통적인 특성을 일반화하는 것은 공격 패턴의 유형을 규정하기 위한 효율적인 알고리즘의 기본이 된다.
- 침입 패턴의 분류는 침입 패턴들의 개념화와 침입 패턴들의 상호관계에 대한 이해를 허용한다.
- 침입 패턴의 분류는 침입 패턴의 계층에 새로운 침입 패턴의 형태 특히, 알려져 있는 공격 패턴의 유

〈표 1〉 위험도의 레벨과 설명

항목 / 레벨	위험도	설 명	예 제
1	최소 상태	- 일부의 알려진 공격 패턴의 프로파일의 초기상태 - 일부 비정상적인 행위의 초기 상태	- .rhosts, /etc/host.equiv 등의 변조
2	경계 상태	- 알려진 공격패턴의 다음 상태 - 알려진 공격 패턴의 일부분이라고 결정할 수 없는 상태	- ISS, SATAN 등
3	주의 상태	- 중간 정도의 잠재적인 위험을 가진 상태 - 비정상적인 행위 누적으로 적대행위 발생 가능 상태	- IP Spoofing 등
4	심각 상태	- 비정상적인 행위의 누적으로 인해 서비스 방해나 심각성 발생 가능 상태	- Mail Storm/Spam - Sync/Ping Flooding 등
5	최악 상태	- 알려진 침입 패턴을 나타내는 상태	- 패킷 스니퍼링 등 - 알려진 모든 침입패턴 들

사한 형태에 대해 추가가 용이하여 침입 탐지 시스템의 확장성을 제공한다. 이것은 기존에 정의된 정보가 손실되거나 혼동을 주지 않고 분류 계층에 추가할 수 있다.

4.2 경고 레벨의 계층구조

네트워크 상에서 잠재적인 침입 패턴은 각 공격패턴의 위험 수준에 근거한 단순한 계층보다 더 많은 사항들을 고려하게 한다. 잠재적인 공격 패턴이 발생한 호스트와 명백하게 관련이 있는 경우 네트워크 환경에서 다른 호스트와 아무 상관이 없을 수 있다. 경고 레벨의 계층 구조는 이러한 관련성을 측정하는데 사용될 수 있다. 경고 레벨은 임의의 네트워크상의 호스트에서 특정 침입 시나리오가 발생한 경우 같은 네트워크상의 다른 에이전트들에게 이 침입을 미리 알려주거나 대응할 수 있도록 하기 위해 에이전트들에게 침입의 여부를 알려주는 관련 중요성을 나타낸다. 이러한 방법으로 시스템의 에이전트들은 침입 패턴을 서로 협력하여 인식하거나 처리한다. 침입에서 특정 상태의 경고 레벨을 정량화 하는데 사용되는 두 가지 요인은 위험도와 침입의 전송도이다.

위험도는 특정 상태에서 공격 패턴이 계속적으로 허용되는 경우 발생 가능한 잠재적인 위험으로 정의되며, 거짓으로 발생한 정상 상태의 빈도에 대한 공격의 종류와 상태를 비교하므로 써 측정된다. 침입의 전송도는 네트워크상의 다른 호스트에 공격의 정도에 의해 정의되며, 네트워크상의 유사한 운영 환경(침입이 발생 가능한 운영체제와 소프트웨어)이 존재하는 경우 공격의 종류를 비교하므로 평가된다.

위험도와 침입의 전송도의 수치는 특정 공격에서 각각 규정된 상태와는 독립적으로 정의된다. 이러한 수치들은 공격의 다양한 상태를 통해 변하거나 변하지 않을 수 있다. 이 결과는 네트워크 상에서 공격 패턴의 경고 레벨을 결정하는데 사용되고, 적절한 시스템의 응답들은 이러한 경고 레벨을 통해 배포되고 분산된다. 이 응답들 중의 하나는 공격당한 네트워크상의 다른 호스트들이 적절하게 대응하게 하기 위해 이 호스트들에게 알려준다. 다른 응답들은 관리자에게 알리거나 특정 행위의 로그를 시작하고 침입 행위를 막기 위한 행위를 수행한다.

경고 레벨의 수치를 결정하는 공식은 다음과 같다:

경고 레벨 := 위험도 * 침입의 전송도

위험도의 수치는 다음과 같다 :

1) 최소 상태 : 이 상태의 이벤트는 알려진 공격 패턴과 결정적으로 결합될 수 없지만, 일부의 알려진 공격 패턴의 프로파일의 초기 상태이거나 일부의 비정상적인 행위의 초기 상태를 나타낸다.
2) 경계 상태 : 이 상태의 이벤트는 알려진 공격 패턴의 두 번째 또는 다음 상태를 가리키지만 알려진 공격 패턴의 일부분이라고 결정할 수 없다. 비정상적인 행위의 누적 결과는 최소 상태에서 발전되지만 공격 패턴의 잠재적인 위험도는 최소이기 때문에 이 이벤트가 알려진 공격 패턴을 결정하거나 심각한 관심사로 생각하기에는 충분하지 않다.
3) 주의 상태 : 이 상태의 이벤트는 중간 정도의 잠재적인 위험을 가지거나 비정상적인 행위의 누적으로 적대 행위가 발생할 수 있는 상태까지 도달한 알려진 침입 패턴을 규정한다.
4) 심각한 상태 : 이 상태의 이벤트는 높은 정도의 잠재적인 위험을 가지거나 비정상적인 행위의 누적으로 서비스를 방해하거나 심각성이 발생할 수

〈표 2〉 전송도의 레벨과 설명

항목 / 레벨	전 송 도	설 명	예 제
1	침입이 전파되지 않은 경우		- 패스워드 크래킹 등
2	침입의 일부분이 전파된 경우	- 침입이 유사한 운영 환경이나 유사한 소프트웨어에서 발생하지만, 이러한 종류의 침입이 충분히 피해를 줄 수 있는 지의 여부는 의문인 경우	- sendmail - tftp - NFS - elm 등
3	침입이 완전히 전파된 경우	- 침입이 네트워크를 통해 운영체제나 소프트웨어의 일부분에서 발생하는 상태	- rootkit - Worm - AutoHack - ISS - SATAN 등

있을 정도의 알려진 침입 패턴을 나타낸다.

5) 최악의 상태 : 이 상태의 이벤트는 알려진 침입 패턴을 나타내며, 이 이벤트가 계속되면 최악의 손실 결과가 발생한다.

침입의 전송도 수치는 다음과 같다 :

1) 침입이 전파되지 않는 경우 : 네트워크 상에서 실질적인 공격이나 잠재적인 공격이 발생하는 환경이 유일한 경우를 말한다. 유사한 공격이나 같은 공격의 일부분이 발생 가능한 환경이 존재하지 않는다. 즉, 침입이 발생한 소프트웨어나 침입이 발생한 유사한 운영체제가 존재하지 않거나 이들의 조합이 존재하지 않는 경우를 말한다.
2) 침입이 일부분 전파되는 경우 : 침입이 유사한 운영 환경이나 유사한 소프트웨어에서 발생하지만, 이러한 종류의 침입이 충분히 피해를 줄 수 있는 지의 여부는 의문이다. 특정 침입에 대한 취약성은 특정 버전의 운영체제나 소프트웨어의 보안에 대한 기능의 부족에 기인한 것이다.
3) 침입이 완전히 전파되는 경우 : 침입이 네트워크를 통해 운영체제나 소프트웨어의 일부분에서 발생하는 경우를 나타낸다. 네트워크상의 다른 호스트들은 같은 공격에 취약하다.

침입의 위험도와 전송도의 종류에 따라 발생 가능한 결과들은 1에서 15의 값을 가지는 경고 레벨을 나타낸다. 공격의 상태에서 특정 침입 패턴에 대한 에이전트 모델의 응답은 이 값과 직접적으로 연결되어 질 수 있다. 일반적으로 에이전트 모델의 응답은 <표 3>과 같이 위험도와 전송도의 경우에 따라 경고 메시지 전송의 범위와 세션 처리사항의 두 가지 형태로 나타난다.

4.3 제안한 에이전트 모델의 구조

새로운 침입 패턴은 발견되고 문서화되어야 하고 이 침입 패턴의 처리는 최소한의 피해를 주면서 추가되어야 한다. 이상적인 모델은 전혀 피해를 주지 않으면서 단지 관련된 부분만 갱신되거나 추가되어야 한다. 이러한 요구 사항들은 에이전트 모델이 여러 개의 구성요소 모듈로 구성될 수 있는 객체 지향의 설계가 이루어질 때 가능하다. 모듈간의 인터페이스들은 같지만 필요한 방법의 구현은 변경될 필요가 있다. 특정 침입 패턴의 처리가 특정 횟수 이상 변경되면 특정 침입 패턴 처리 모듈에서 침입 패턴을 처리하는 알고리즘은 모델의 나머지 부분에 영향을 주지 않고 변경될 필요가 있다.

에이전트 모델의 구성요소들은 다음과 같다 :

1) 초기화 모듈 : 이 모듈에서는 위험도와 전송도, 경고 레벨의 값을 0으로 초기화한다.
2) 통신 인터페이스 : 메시지를 수신하고 송신하는 두 가지 형태의 통신을 위한 추상적인 슈퍼클래스를 의미한다.

● 수신 모듈 : 통신 인터페이스의 서브클래스로서, 다른 에이전트로부터 메시지를 수신하거나 송신하는 소프트웨어의 일부분을 나타낸다.

● 송신 모듈 : 통신 인터페이스의 서브클래스로서, 다른 에이전트에게 메시지(경고)를 전송하는 소프트웨어의 일부분을 나타낸다. 경고를 패키지화 하고 다른 에이전트의 위치를 확인하고 메시지를 전송하는 네트워크 소프트웨어를 호출하는 방법을 제공한다.

〈표 3〉 경고레벨과 위험도, 전송도의 관계

항목 / 경고레벨	위험도	전송도	경고 메시지	세션 처리
1	최소 상태	미전파 상태	알리지 않음	없음
2	경계 상태	미전파 상태	알리지 않음	없음
3	경계 상태	일부 전파 상태	일부 호스트에게 알림	없음
4	경계 상태	완전 전파 상태	모든 호스트에게 알림	없음
5	주의 상태	미전파 상태	모든 호스트에게 알림	없음
6	주의 상태	일부 전파 상태	모든 호스트에게 알림	없음
7	주의 상태	완전 전파 상태	모든 호스트에게 알림	없음
8	심각 상태	미전파 상태	모든 호스트에게 알림	세션 종료
9	심각 상태	일부 전파 상태	모든 호스트에게 알림	세션 종료
10	심각 상태	완전 전파 상태	모든 호스트에게 알림	세션 종료
11	최악 상태	미전파 상태	모든 호스트에게 알림	세션 종료
12	최악 상태	일부 전파 상태	모든 호스트에게 알림	세션 종료
13	최악 상태	완전 전파 상태	모든 호스트에게 알림	세션 종료

3) 침입 패턴 비교 모듈 : 이 모듈에서는 수신 모듈을 통하여 들어오는 임의의 패턴과 에이전트에서 비교하는 침입 패턴을 비교하여 침입의 위험도와 전송도를 계산하여 그 결과인 경고 레벨을 송신 모듈로 보낸다.

4) 감사 자료 저장 모듈 : 이 모듈에서는 침입 패턴 비교 모듈에서 발생한 로그를 감사 데이터베이스에 저장하고 새로운 침입 패턴인 경우에는 재사용을 위해 침입 패턴 데이터베이스에 저장한다.

4.4 제안한 에이전트 모델의 침입탐지 알고리즘

에이전트 모델에서 임의의 패턴이 침입 패턴인지 아닌지를 탐지하여 경고 레벨을 산출하는 알고리즘은 다음과 같다:

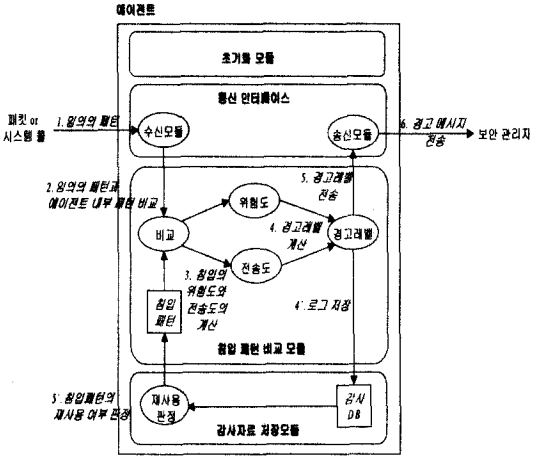
〈표 2〉 에이전트의 침입탐지 알고리즘

```
1. 초기화
2. loop
 2.1 임의의 패턴 입력 처리
 2.2 if (입력 패턴 == 내부 패턴)
  2.2.1 위험도 계산
   2.2.1.1 if (입력 패턴 == 최소상태) 위험도=1
   2.2.1.2 elseif (입력 패턴 == 경계상태) 위험도=2
   2.2.1.3 elseif (입력 패턴 == 주의상태) 위험도=3
   2.2.1.4 elseif (입력 패턴 == 심각상태) 위험도=4
   2.2.1.5 elseif (입력 패턴 == 최악상태) 위험도=5
  2.2.2 전송도 계산
   2.2.2.1 if (입력 패턴 == 미전파) 전송도=1
   2.2.2.2 elseif (입력 패턴 == 부분전파) 전송도=2
   2.2.2.3 elseif (입력 패턴 == 완전전파) 전송도=3
   2.2.3 경고 레벨:= 위험도 * 전송도
   2.2.4 경고 메시지 전송
 2.3 else Goto 2
```

4.5 제안한 에이전트 모델의 침입 패턴 처리

네트워크상의 특정 호스트에 생성된 에이전트는 패킷 또는 시스템 콜을 수신모듈에서 받아들인다. 수신모듈은 입력된 임의의 패턴과 에이전트의 내부 패턴과 비교하여 침입의 위험도와 침입의 전송도의 분류에 따라 계산된 경고레벨을 송신 모듈에 전송한다. 송신 모듈에서는 산출된 경고 레벨에 따라 경고 메시지를 보안 관리자에게 알리게 된다

.



(그림 5) 에이전트 모델의 침입 패턴 처리 흐름도

5. 에이전트 모델을 이용한 침입 패턴의 표현

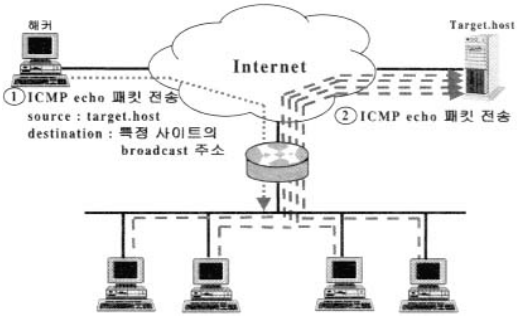
5.1 침입 패턴

최근의 해킹 기법중 smurf와 mscan, SYN flooding

등의 세 가지 침입 패턴만 에이전트 모델에서 탐지되는 것을 표현하고자 한다.

5.1.1 smurf

smurf는 공격 대상 주소로 소스 IP 주소를 만들고 임의의 브로드캐스트 주소(X.X.X.255)로 ICMP echo 패킷을 발송한다. 스푸핑된 IP를 가진 호스트는 ICMP reply 패킷들로 인해 시스템 부하가 증가한다.



(그림 6) smurf

smurf의 실행단계는 다음과 같다 :

1) ICMP echo 패킷의 source 주소가 공격할 호스트의 IP 주소로 설정한다.
2) ICMP echo 패킷의 destination 주소가 bounce site의 broadcast 주소를 설정한다.
3) ICMP echo 패킷을 destination 주소로 전송한다.
4) bounce site의 broadcast 주소에서 공격할 호스트로 ICMP echo reply 패킷들을 전송한다.

침입 패턴 smurf의 각 단계는 에이전트 모델에서 다음과 같이 표현된다 :

1단계는 해커가 ICMP echo 패킷의 source 주소를 공격할 호스트의 주소로 설정하기 때문에 잠재적인 위험을 갖는 상태로 위험도는 3이 되며, 아직까지는 패킷이 전송되지 않았으므로 전송도는 1이다. 2단계는 destination 주소가 broadcast 주소로 설정이 되므로 이 단계가 계속되면 최악의 손실 결과가 발생하므로 위험도는 5가 되고, 아직은 침입이 전송된 단계는 아니므로 전송도는 1이다. 3단계는 실제로 ICMP echo패킷을 destination 주소로 전송하므로 전송도는 2가 된다. 4단계는 공격할 호스트로 ICMP echo reply 패킷들을 전송하므로 침입이 완전히 전파되었으므로 전송도가 3이 된다.

5.1.2 mscan

mscan은 네트워크 블록 전체를 스캐닝하여 최근에 알려진 몇가지 보안 취약점을 원격으로 한번에 스캐닝하여 그 결과를 나타낸다.

mscan의 실행 단계는 다음과 같다 :

1) wingate를 스캐닝한다.
2) phf를 스캐닝한다.
3) handler를 스캐닝한다.
4) test-cgi를 스캐닝한다.
5) NFS export를 스캐닝한다.
6) statd를 스캐닝한다.
7) named를 스캐닝한다.
8) X server를 스캐닝한다.
9) ipopd를 스캐닝한다.
10) imapd를 스캐닝한다.

침입 패턴 mscan의 각 단계는 에이전트 모델에서 다음과 같이 표현된다 :

1단계는 wingate를 스캐닝한다. 네트워크 블록에서 하나의 부분만 스캐닝하기 때문에 위험도는 최소상태인 1을 나타낸다. 2단계는 wingate를 스캐닝한 후에 phf를 스캐닝하므로 네트워크 블록에서 취약한 부분을 연속해서 스캐닝하므로 위험도는 경계상태인 2를 나타낸다. 3단계는 세 번째로 handler를 스캐닝하므로 위험도는 주의 상태인 3을 나타낸다. 4단계는 네 번째로 test-cgi를 스캐닝하므로 위험도는 심각 상태인 4를 나타낸다. 5단계 이상은 계속해서 취약 부분을 스캐닝하기 때문에 위험도는 최악상태인 5를 나타낸다. 전송도는 특정 호스트만 대상으로 하기 때문에 전파되지 않아서 1의 값을 나타낸다.

5.1.3 SYN flooding

SYN flooding은 TCP의 Three-way Hand Shaking에서의 문제점을 이용한 침입 방법으로 계속적인 Half-Open 상태로 다른 TCP 요청을 거절한다. 많은 수의 half-open TCP 연결을 시도하여 상대 호스트의 listen queue를 가득 채워 TCP 서비스 연결을 거부한다. 이

공격 방법은 시스템이 피해를 입고 있는 인지하지 못할 수 있으며, 공격자가 출발지의 IP 주소를 속여서 보내기 때문에 공격자를 추적하기가 힘들다. 이미 만들어진 TCP 연결이나 나가는 패킷에는 영향을 주지 못하며, 인터넷 서비스 공급업체들에게 치명적인 공격 방법이다.

(그림 7) SYN flooding

SYN flooding의 실행 단계는 다음과 같다:

1) SYN 패킷이 연속적으로 2개 보낸다.
2) 다음에 또 하나의 SYN 패킷을 보낸다(전송한 SYN 패킷의 개수:3개).
3) 다음에 또 하나의 SYN 패킷을 보낸다(전송한 SYN 패킷의 개수:4개 이상).

침입 패턴 SYN flooding의 각 단계는 에이전트 모델에서 다음과 같이 표현된다 :

1단계는 해커가 두개의 SYN 패킷을 전송하는 경우로 위험도는 중의 상태인 3을 나타낸다. 2단계는 두 개의 SYN 패킷을 전송한 후 또 하나의 SYN 패킷을 보내는 경우로 침입 공격이 확실시되기 때문에 위험도는 최악 사상태인 5를 나타낸다. 3단계부터는 계속해서 SYN 패킷이 전송되므로 SYN flooding 침입으로 간주하여 위험도는 최악 상태인 5를 계속 나타낸다. 침입이 전파되지는 않기 때문에 전송도는 1의 값을 나타낸다.

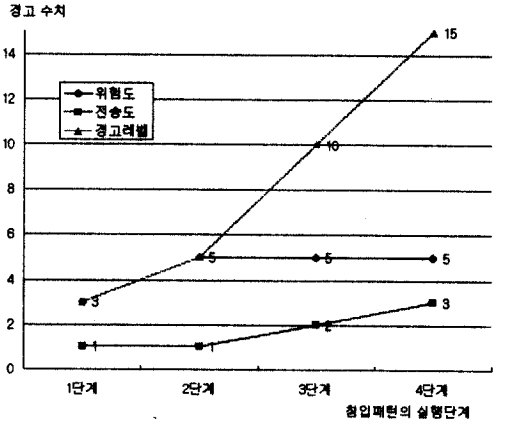
5.2 실시간 침입 탐지 에이전트 모델의 분석

5.2.1 분석을 위한 고려사항

실시간 침입 탐지 에이전트 모델을 분석하기 위한 모의실험에서 고려해야 할 침입 패턴은 다음과 같다 :

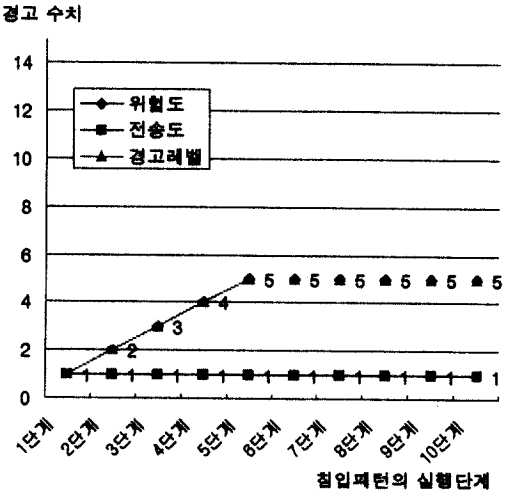
- smurf
- mscan
- SYN flooding

각 침입 패턴의 실행 단계에 대한 위험도와 전송도, 경고레벨 등은 다음과 같이 나타난다.



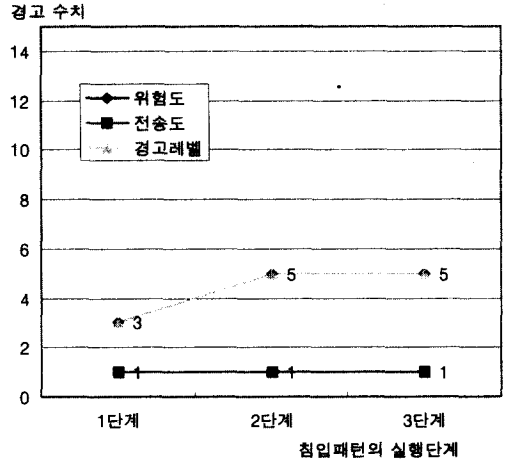
(그림 8) smurf 침입 패턴의 단계별 경고레벨과 위험도, 전송도

smurf 침입 패턴은 1단계와 2단계에서 다른 호스트로 전송되지 않기 때문에 전송도가 1이지만 3단계와 4단계에서는 전송도의 수치가 2와 3이 되기 때문에 경고 레벨은 10과 15의 수치를 나타내고 있다. 이것은 smurf 침입 패턴이 실행되면서 단계가 진행할수록 다른 호스트로 전송되어 경고 레벨의 수치가 높아진다는 것을 알 수 있다.



(그림 9) mscan 침입 패턴의 단계별 경고레벨과 위험도, 전송도

mscan 침입 패턴은 특정 네트워크 블록을 스캐닝하는 공격 패턴이기 때문에 앞의 smurf 침입 패턴과 같이 전송도의 수치가 증가하지 않는다. mscan 침입 패턴이 네트워크 블록의 취약 부분을 임계치 이상 스캐닝하면 위험도 최악 상태로 표시하여 침입으로 간주한다.



(그림 10) SYN flooding 침입 패턴의 단계별 경고레벨과 위험도, 전송도

SYN 패킷을 한번 받으면 정상적인 상태라고 할 지라도 관리자에게 주의 상태로 알리고 두 개 이상이 되면 SYN flooding 공격으로 간주하여 최악 상태를 나타낸다.

6. 결 론

본 논문에서 제안한 실시간 침입 탐지를 위한 에이전트 모델은 각 침입 패턴의 실행 단계를 분석하여 관리자에게 의심의 등급에 따라 미리 그 정도를 알려 줌으로써 침입을 사전에 알게 한다. 특히, 임의의 행위가 호스트에 영향을 주는 정도를 위험도로 나타내고, 침입이 전파되는 정도를 전송도로 나타내어 경고 레벨을 산출한다. 이 다양한 경고 레벨은 관리자에게 의심스러운 행위에 대해서 미리 알려 줌으로써 침입에 대처할 수 있도록 한다.

향후의 연구할 방향으로는 이 모델에 근거하여 모든 침입 패턴을 분석하고 각각의 침입패턴의 유형을 정립하여 각 유형에 따른 대처 방안을 실행하는 실시간 침입 탐지 시스템을 개발하는 것이다.

참 고 문 헌

[1] 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.

[2] 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.

[3] Robert S. Gray. Agent Tcl : A transportable agent system. In James Mayfield and Tim Finin, editors, *Proceedings of the CIKM Workshop on Intelligent and Information Agent, Fourth International Conference on Information and Knowledge Management (CIKM 95)*, Baltimore, Maryland, December 1995.

[4] Robert S. Gray. Agent Tcl : A flexible and secure mobile-agent system. In Mark Diekhans and Mark Roseman, editors *Proceedings of the Forth Annual Tcl/Tk Workshop (TCL '96)*, Monterey, California, July 1996.

[5] James E. White. Telescript technology : Scenes from the electronic marketplace. General Magic White Paper, General Magic, 1995.

[6] John K. Ousterhout. *Tcl and the Tk Toolkit.* Addition-Wesley Professional Computing Series. Addison-Wesley, Reading, Massachusetts, 1994.

[7] R. Stockton Gainess. Dixie language design and interpreter issues. In *Processing of the USENIX Symposium on Very High Level Languages (VHLL)*, Sante Fe New Mexico, October 1994.

[8] Kenneth E. Harker. TIAS : A Transportable Intelligent Agent System. Technical Report PCS-TR95-258, Department of Computer Science, Dartmouth College, 1995.

[9] Keith Kotay and David Kotz. Transportable agents. In Yannis Labrou and Tim Finin, editors *Proceedings of the CIKM Workshop on Intelligent*

Informations, Third International Conference on Information and Knowledge Management (CIKM 94), Gaithersburg, Maryland, December 1994.

[10] Dag Johansen, Robbert van Reness, and Fred B. Scheidner. Operating system support for mobil agents In *Proceeding of the 5th IEEE Workshop on Hot Topics in Operating System (HTOS)*.

[11] N.S. Borenstein and M. Rose. Safe Tcl. Available at pages 42-45, 1995. at available ftp://ftp.fv.com/pub/code/other/safe-tcl.tar.Z.

이 문 구

e-mail : yeon0330@hotmail.com

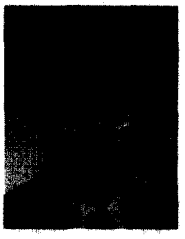
1984년 숭실대학교 전산과(학사)

1993년 이화여자대학교 교육대학원 전산학과(석사)

1996년~현재 숭실대학교 대학원 전산과(박사수료)

1997년~현재 명지전문대학 전산과 겸임교수

관심분야 : 정보 보안, 인터넷 보안, 침입 탐지 및 차단 시스템관련 분야



전 문 석

e-mail : mjun@computing.soongsil.ac.kr

1980년 숭실대학교 전자계산학과 (학사)

1986년 University of Maryland, Computer Science(석사)

1988년 University of Maryland, Computer Science(박사)

1989년 Morgan State Univ. 부설 Physical Science Lab. 책임 연구원

1991년~현재 숭실대학교 컴퓨터학부 부교수

관심분야 : 병렬 처리, 암호화 알고리즘 설계 및 분석 정보보안, 인터넷 보안, 침입 탐지 및 차단 시스템 보안