퍼지 이론을 이용한 효율적인 침입탐지 방법

김민수†, 노봉남‡
†‡ 전남대 전산학과

The Efficient Method of Intrusion Detection with Fuzzy Theory

Min-Soo Kim†, Bong-Nam Noh‡
†‡ Dept. of Computer Science, Chonnam National University

요약

본 논문에서는 Petri-net 형태로 침입탐지 규칙을 구성한다. 이것은 실시간 침입탐지가 가능하고 지연공격과 다중공격을 방어할 수 있다. 그리고, Petri-net의 플레이스에 퍼지값을 적용한다. 이 값은 침입의 진행에 따라 변경되며 침입을 판정하는 기준이 된다. 또한, 변형공격에 대응할 수 있도록 한다.

1. 서론

침입탐지 시스템은 불법적인 사용, 오용, 또는 불법적인 사용자나 외부 침입자에 의한 컴퓨터 시스템을 남용하는 침입을 알아내려는 시스템이다[2]. 이러한 시스템은 단일 컴퓨터나 또는 네트워크로 연결된 여러 컴퓨터를 감독한다. 침입탐지 시스템은 감사 기록, 시스템 테이블, 네트워크 부하(traffic) 기록의 자원으로부터 사용자 행위에 대한 정보를 분석하여 수행된다.

침입탐지 시스템의 목표는 크게 두 가지 방법으로 이루어진다. 하나는 침입자에 의한 불법적인 사용을 명시하는 것이고 다른 하나는 합법적인 사용자에 의한 오용이나 남용을 말한다[10]. 초기에 침입탐지 시스템은 후자를 주로 나타냈으나, 침입 방법이 복합적이고 다양화됨에 따라 전자와 후자 모두 처리하는 시스템이 개발되고 있다.

본 논문에서는 Petri-net 형태로 침입탐지 모듈을 구성하여 실시간 침입탐지가 이루어지도록 한다. 또한, 지연공격, 다중공격에 대응할 수 있도록 한다.

Petri-net에 퍼지값을 적용한다. 이 값은 침입이 진행됨에 따라 변화되며 종료 플레이스의 퍼지값으로 침입을 판정한다. 또한, 알려진 공격방법을 변형한 변형공격에 대응할 수 있도록 한다.

2장에서는 침입탐지에 대하여 설명한다. 3장에서는 침입탐지 모듈을 설명한다. 4장에서는 침입탐지 방법에 적용되는 이론을 설명하고, 5장에서 결론을 맺었다.

2. 침입탐지

침입(intrusion)이란 컴퓨터가 사용하는 자원의 무결성, 비밀성, 가용성을 저해하는 일련의 행위들의 집합을 말한다. 또 다른 정의에서는 컴퓨터 시스템의 보안 정책을 파괴하는 행위를 말한다. 그리고, 침입탐지 시스템 (Intrusion Detection System)이란 시스템의 비정상적인 사용, 오용, 남용 등을 알려주는 시스템이다[2].

2.1 침입탐지

침입탐지에 대한 현재 기술적 실제적 논점은 실제 침입이 아닌데 침입으로 판정하는 경우(false positives)와 실제 침입인데 탐지하지 못하는 경우(false negatives)를 어떻게 처리하는가에 있다[15]. 즉, 침입탐지 시스템의 성능이 좋다는 것은 false positive와 false negative가 최소화되는 것이다.

침입탐지 시스템에서는 감사 추적(audit trail) 분석이 필요하다. 보안 공격은 미리 알려지지 않는 경우가 많고 완벽한 보안을 보장하는 보안모델은 존재하지 않기 때문이다. 이러한 감사 자료는 다양한 침입을 기록하고 너무 방대해질 수 있는 자료를 축약(reduction)할 수 있어야 한다[14]. 또한, 감사 자료의 분석 도구는 재사용이 가능하여야 하고 보안 책임자가 질의하기 쉬운 인터페이스를 제공해야 한다.

2.2 침입탐지 시스템 분류

침입탐지 시스템은 다음 세 가지 방법으로 분류해 볼 수 있다.

(1) 침입탐지 내용에 따른 분류

침입탐지 내용에 따라 오용 탐지(misuse detection)와 이상 탐지(anomaly detection)로 분류할 수 있다.

오용 침입이란 시스템이나 응용 프로그램의 약점을 통하여 시스템에 침입할 수 있는 잘 정의된 공격 형태를 말한다. 오용 탐지는 침입자의 행위에 대한 시나리오를 결정하여 저장한다. 시스템에 접속하여 사용하는 행위에서 이러한 시나리오와 같은 행위를 침입이라고 판정한다. 알려진 침입 패턴을 이용하며, 규칙 기반 침입 확인 방법이 가장 많이 사용되고 상태 변이 방법도 종종 사용된다.

이상 침입이란 컴퓨터 자원의 비정상적인 행위나 사용에 근거한 침입을 말한다. 예를 들면, 한 사용자의 컴퓨터 사용 시간이 오전 9시부터 오후 5시까지인데, 근무시간 이후에 컴퓨터를 사용하는 경우 올바른 ID와 패스워드를 사용한 정당한 사용일지라도 침입으로 간주하는 경우를 들 수 있다. 즉, 일반적인 행위 패턴으로부터 벗어남을 탐지하는 방법이

다. 이것은 알려진 침입 방법뿐만 아니라 알려지지 않은 침입 방법도 탐지 할 수 있다는 장점이 있으나, false positive가 증가하는 단점이 있다.

(2) 침입탐지 방법에 따른 분류

침입을 탐지하는 방법에 따라 통계적 변화 탐지(Statistical Anomaly Detection), 규칙 기반 침입 확인 (Rule-based Penetration Identification) 그리고 상태 변이 분석(State Transition Analysis) 방법으로 분류된다.

통계적 변화 탐지 방법은 특정한 사건을 정해진 시간동안 관찰하여 사건의 발생 횟수나 사건 간의 관계가 보통과 다를 때 침입으로 판정하는 방법이다. 이것은 이상 탐지에 가장 많이 사용하는 방법으로 과거의 경험적인 자료에서 침입을 탐지하기 때문에 자료의 양이 많을수록 정확하게 침입을 탐지할 수 있다.

규칙기반 침입 확인 방법은 시스템의 침입자가 사용하는 침입유형을 규칙으로 저장하여 이와 같은 사용 패턴을 침입으로 간주한다. 따라서, 오용 탐지는 규칙 기반 침입 확인 방법이 가장 많이 사용되고 있다.

상태 변이 분석 방법은 시스템 내부에 침입자가 시스템 관리자 권한을 얻기까지의 과정을 단계별로 기술하여 지식 베이스로 저장한 다음, 이를 감사 자료에 나타난 사용자의 사용패턴과 비교하여 침입자를 찾아낸다. 즉, 하나의 공격 시나리오에서 이 공격방법이 성공적으로 이루어지기 위해 반드시 필요한 행동들과 그러한 각각의 행동들로 인해 새로이 생긴 시스템의 상태를 규명하는 방식이다.

(3) 침입탐지 범위에 따른 분류

침입탐지 방법의 범위에 따른 분류 방법은 호스트 기반 감시(host-based monitoring) 방법, 다중 호스트 기반 감시(multi-host based monitoring) 방법과 네트워크 기반 감시(network-based monitoring) 방법으로 구분된다.

호스트 기반 감시 방법은 단일 호스트에서 감시와 분석을 모두 수행하는 것이다. 단일 호스트에서 감사 로그를 수집하고 내용을 분석한다. 따라서, 침입탐지는 호스트에 로그인 했을 때 시작한다.

다중 호스트 기반 감시 방법은 여러 호스트에서 감사 로그를 수집하여 서로 교환하는 방법이다. 이것은 여러 호스트를 통하여 침입이 이루어지는 것을 감시하기 위한 방법이다. Trust-host에 의한 NFS 접근에 IP spoofing을 이용하여 침입하는 경우를 감시하는 것이 대표적인 예이다. 따라서, 침입탐지는 LAN으로 연결된 여러 호스트에서 동시에 이루어진다.

네트워크 기반 감시 방법은 네트워크를 통해서 정보를 중앙 감시 시스템에 모아 분석한다. 최근의 연구 경향은 네트워크 기반 방법이 많다. 네트워크 상에 떠다니는 패킷(packet)을 잡아 내용을 분석함으로써 침입탐지가 이루어진다.

2.3 여러 침입탐지 시스템

<표 1> 침입탐지 시스템 비교

침입탐지 시스템 종류	탐지 내용		탐지 범위			탐지 방법		
	이상 탐지	오용 탐지	호스트 기반	멀티호스트 기반	네트워크 기반	통계적	규칙 기반	상태 변이
ASAX		✓		✓			✓	
CMDS	✓	✓		✓		✓	✓	
Computer Watch		✓	✓				✓	
CSM		✓			✓			
DIDS				✓			✓	
EMERALD		✓			✓		✓	
IDES	✓		✓			✓		
IDIOT		✓	✓					✓
STAT		✓	✓				✓	✓
W&S	✓		✓			✓		

기존의 침입탐지 시스템은 각기 다른 특징을 가지고 있다. 이상과 오용 탐지를 하는가, 어떠한 방법으로 침입탐지를 수행하는가, 탐지 범위는 어디까지 하겠는가에 따라 여러 종류로 나뉘어짐을 표 1을 통해서 알 수 있다[1,3,4,5,6,8,9,13,16,17].

2.4 침입 방법

다음은 알려진 방법으로 최근에 가장 많이 이용되는 침입형태이다[18,19].

(1) 임의의 파일 생성

UNIX 시스템에서 rlogin이 가능한 호스트를 기록하는 hosts.equiv와 .rhosts 파일을 원하는 곳에 생성하거나 링크시키는 방법으로 접근할 수 있다. 또한, /tmp와 같은 누구나 쓸 수 있는 곳에 임의의 파일을 생성하여 해킹에 이용할 수 있다.

(2) 버퍼 오버플로우(buffer overflow)

UNIX내의 프로그램을 수행하는데 명령어와 매개변수를 전달할 때, 매개변수 뒤에 다른 코드를 집어넣어 버퍼 오버플로우를 유발시킨다. 이렇게 오버플로우가 발생했을 때 쉘 프로그램이 실행되도록 하여 관리자 권한을 얻을 수 있다.

(3) 서비스 방해 공격

네트워크를 통하여 대량의 메일을 보내거나 쓸모 없는 문자를 계속 목적 호스트에 보내어 시스템의 작동을 방해하는 공격이다. 이 방법은 최근에 늘어나는 공격방법으로 방화벽(firewall)이나 패킷 필터링을 통하여 대응할 수 있다.

(4) 스니핑(sniffing)

스니핑은 네트워크의 한 호스트에서 실행되어 그 주위를 지나다니는 패킷을 엿보는 방법으로 ID와 패스워드를 알아내기 위하여 침입자들에 의해 자주 사용된다. 네트워크 보안에 신경을 쓴 호스트라도 주변의 호스트가 공격당해서 스니핑을 위해 사용된다면 무력해질 수밖에 없다.

(5) 스푸핑(spoofing)

스푸핑이란 자신을 타인이나 다른 시스템에게 속이는 행위를 의미한다. 예를 들어, 특정 호스트에게만 접근권한을 준다고 가정했을 경우 해커는 당연히 자신이 특정 호스트로부터 접근하려는 것처럼 속이려 할 것이며, 이를 가리켜 바로 스푸핑이라고 할 수 있는 것이다.

3. 탐지 모듈

침입을 탐지하기 위해서는 침입 패턴을 정의하고 구현하는 방법에 대한 연구가 필요하다. 본 논문에서는 침입 패턴을 Petri-net으로 구성하여 침입 상황을 파악하여 알릴 수 있도록 설계하였다.

3.1 탐지 모듈

침입 유형에 따라 탐지 모듈을 구성할 수 있다. 이것은 침입 패턴을 구분할 수 있을 뿐만 아니라 실시간 침입탐지를 할 수 있도록 한다.

침입 패턴은 탐지 모듈을 생성하기 위한 중요 자료가 된다. 침입 패턴을 분석하여 탐지 패턴을 구하고, 이러한 패턴은 침입탐지 모듈을 구성하기 위한 입력 자료가 된다.

3.2 Petri-net

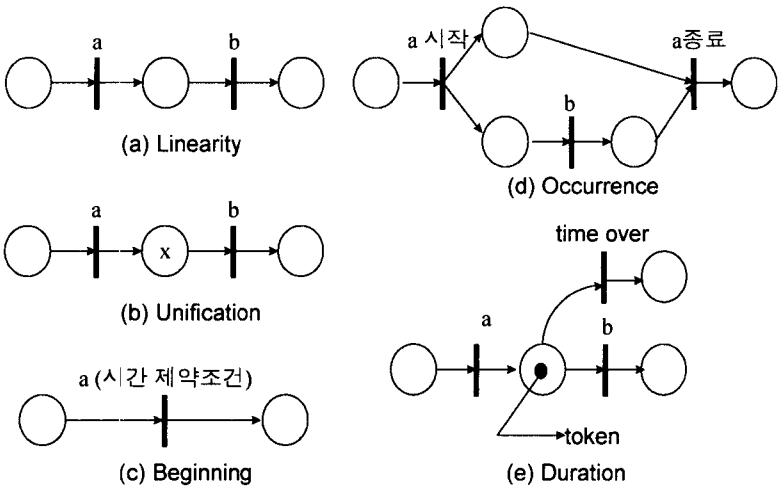
본 연구에서는 Petri-net을 응용하여 침입탐지 모듈을 구현하였다. Petri-net은 여러가지 구조나 상황을 모델링하고 검증하는 도구로 많이 사용되고 있으며, 침입탐지 분야에서도 침입규칙을 표현하는데 사용하고 있다[11,12].

Petri-net의 플레이스(place)는 침입 단계를 의미하고 트랜지션(transition)은 침입 패턴에서 나타나는 사건을 의미하며 토큰(token)의 진행 여부를 결정한다. 현 플레이스에서 해당되는 사건이 발생하면 다음 플레이스로 토큰을 진행시킨다.

3.3 침입패턴의 특징

침입 패턴은 여러 가지 유형을 가질 수 있다[7]. 여기서는 그 유형에 따라 어떠한 방법으로 Petri-net을 구성하는지 보여준다.

(a) 선형(Linearity)



(그림 1) 침입 패턴의 유형

사건 발생이 순차적으로 이루어진다. 즉, 그림 1-(a)처럼 사건이 a, b, c 순서로 진행되는 경우이다. 이것은 순서에 의미를 두고 있으며 순서에 의미를 두지 않는 비선형(nonlinearity)도 있다.

(b) 공통(Unification)

두 사건의 공통 요소가 존재한다. 이전 사건에 사용되었던 매개변수(argument)가 다음 사건에 이용되는 경우이다. 대개 파일을 생성하여 링크시키거나 자료를 편집하여 다른 행위를 취하는 경우가 해당된다. Petri-net으로 구성할 때는 플레이스에서 토큰의 속성에 매개변수를 저장하여 처리한다.

(c) 시작(Beginning)

사건의 시작 시간에 중점을 두고 있다. 일과시간 이후에 시스템에 접속하는 경우를 예로 들 수 있다. Petri-net에서는 트랜지션에서 사건의 시작 시간을 조사할 수 있도록 처리한다.

(d) 발생(Occurrence)

사건이 진행하는 도중 다른 사건이 발생하는 경우이다. 예를 들어, 어떤 무한루틴 프로그램을 수행시켜놓고 다른 침입 작업을 할 수 있다. 사건은 시작 시간과 종료 시간이 있다. 따라서, 그림 1-(d)에서처럼 사건이 시작해서 종료하기 전까지 다른 사건이 발생할 수 있는 경우를 표현할 수 있다.

(e) 기간(Duration)

사건의 진행 시간의 범위를 지정할 수 있다. 예를 들어, 어떠한 프로그램은 항상 5분 이내에 작업이 끝나야 한다고 하자. 그런데, 그 프로그램이 5분이 경과하여도 끝나지 않을 경우 문제가 될 수 있다. 그림 1-(e)처럼 time over라는 트랜지션을 두어 일정 시간이 지날 경우 이 트랜지션이 처리되도록 정의한다.

3.4 다양화된 공격방법

공격자는 침입탐지 시스템의 감시를 회피하기 위하여 공격방법을 다양화할 것이다.

지연 공격은 침입 작업을 나누어 조금씩 진행하는 방법이다. 이것은 통계적인 방법을 사용하거나 상태를 보존하지 않는 침입탐지 시스템의 감시를 벗어나기 위해 사용된다.

대부분의 침입탐지 시스템은 사용자별로 정보를 수집한다. 이러한 경우 여러 사람의 ID를 사용하여 터미널 창을 여러개 열어 놓고 접근하는 방법에 대응할 수 없게 된다. 이러한 공격 방법을 다중공격이라고 한다.

알려진 침입 방법을 연구하여 조금 변경된 방법으로 접근하는 경우가 있다. 이러한 변형 공격은 이처럼 침입 유형과 유사한 형태의 공격방법이다.

4. 침입탐지 방법

4.1 기본 구성

실시간에 수집된 로깅 자료는 침입탐지 모듈에 적용된다. 침입 패턴에서 생성된 탐지 모듈은 실시간 처리를 위하여 데몬 프로세스로 작동되고 해쉬 검색 방법을 이용한다. 호스트 기반의 침입탐지 시스템에는 통계적 처리방법과 패턴 인식 방법 등이 주로 이용된다. 본 논문에서는 패턴 인식 방법을 주요 방법으로 하고 Petri-net을 이용하여 침입의 단계를 가시화 하였다.

각 플레이스의 상태를 보존하여 지연 공격에 대응할 수 있도록 하였다. 즉, 침입 단계 중 일부분을 수행한 후 몇 시간이 지난 다음에 다음 단계를 수행하더라도 이전 상태가 저장되어 있어서 계속 탐지가 가능하다. 탐지 자료를 보존하는 것은 저장 용량 문제가 있다. 그러나, 플레이스 정보는 고정된 작은 공간이기 때문에 지속적으로 보관한다.

Petri-net의 상태는 각 사용자별로 보관하는 것이 아니라 모든 사용자를 하나의 상태로 보관한다. 따라서, 다중 공격을 효율적으로 방어할 수 있다.

4.2 퍼지값을 이용한 대응방법

퍼지이론은 인간의 사고과장에서 내재된 모호성과 부정확함을 표현하는 개념으로 자데(Zedeh) 교수에 의해 처음 발표된 이래로 수학, 전자, 언어학 등에서 응용되어왔다.

변형공격은 일반적으로 알려진 공격방법을 조금 바꾸어 공격하는 형태이다. 이러한 공격을 대응하기 위해서는 침입 패턴의 유동성을 두어야 한다. 따라서, 본 논문에서는 침입 패턴에 대한 Petri-net의 각 플레이스에 퍼지값을 두어 변형공격에 대응할 수 있도록 한다.

(1) 퍼지값의 설정

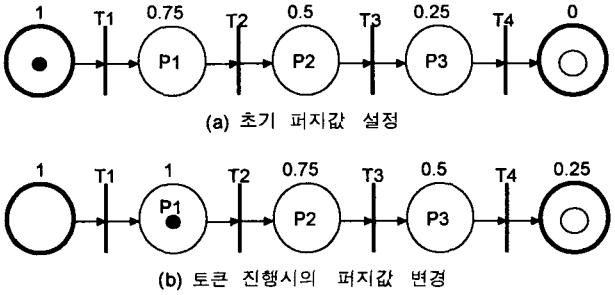
퍼지값은 또한 현재까지 침입이 어느 정도 이루어졌는지를 판단할 수 있는 근거가 된다. 퍼지값을 설정하는 것은 종료 플레이스에서의 상대적인 거리로 판단한다. 즉, 선형성을 띈 패턴이 그림 2-(a)와 같다면, 각 플레이스에서의 초기 퍼지값은 그림 2-(a)의 플레이스 위의 숫자이다. 퍼지값은 침입이 진행됨에 따라 값이 변경된다(그림 2-(b)). 따라서, 종료 플레이스의 퍼지값도 변경되므로, 이 값에 따라 침입의 진행상황을 판단할 수 있다.

초기 퍼지값은 시작 플레이스를 1로 종료 플레이스를 0으로 설정한다. 이하 플레이스

는 시작 플레이스와 종료 플레이스의 거리 비율로 계산된다. 플레이스 수에 따른 각 플레이스의 퍼지값 증가 비율(Δ)은 아래와 같다.

$$\Delta = \frac{F_{before} - F_{after}}{N-1}$$

F_{before}는 구하는 위치의 이전 플레이스 퍼지값이고, F_{after}는 이후 플레이스 퍼지값이다. N은 구하는 위치의 이전 플레이스와 이후 플레이스 사이의 플레이스 수이다.



(a) 초기 퍼지값 설정

(b) 토큰 진행시의 퍼지값 변경

(그림 2) 퍼지값 설정과 변화

트랜지션 분리가 있을 경우 분리에서 결합 사이의 플레이스 수(N)는 달라지는데 그 값은 아래와 같다(그림 3-(a)).

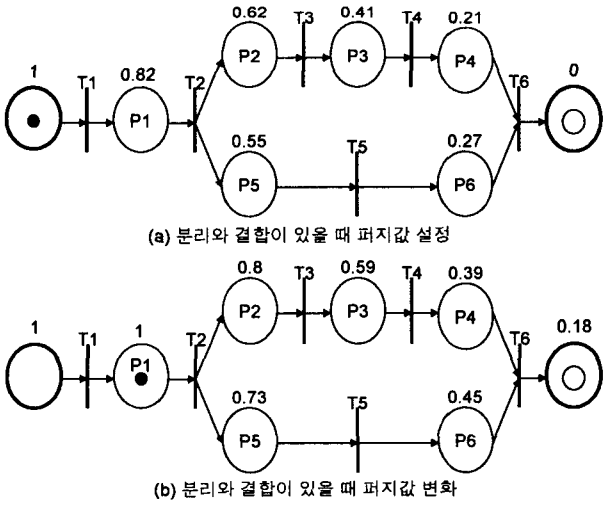
$$N = \frac{N_{place}}{N_{fork}}$$

여기서 N_{place}는 분리와 결합 사이의 플레이스 수이며 N_{fork}는 분리된 개수이다.

(2) 퍼지값의 변화

퍼지값의 변경은 역시 진행 토큰의 위치와 이웃 플레이스의 증가치에 따라 달라진다(그림 3-(b), 그림 3-(b)). 트랜지션이 처리될 때 다음 플레이스의 퍼지값 증가치(δ)는 이전 플레이스의 증가비율이다 ($\delta = F'_{before} - F_{before}$). 다음 플레이스가 트랜지션 분리일 때, 다음 플레이스의 퍼지값 증감치(δ_{fork})는 일반 증가치와 같다($\delta_{fork} = \delta$). 다음 플레이스가 트랜지션 결합일 때, 다음 플레이스의 퍼지값 증가치(δ_{join})는 결합 왼쪽 플레이스 증가치(δ')의 평균이다.

$$\delta_{join} = \frac{\sum_{i}^{N_{in}} \delta'_i}{N_{join}}$$



(a) 분리와 결합이 있을 때 퍼지값 설정

(b) 분리와 결합이 있을 때 퍼지값 변화

(그림 3) 분리와 결합에서의 변화

퍼지값의 변화는 토큰의 진행할 때뿐만 아니라 아직 진행되지 않은 임의의 트랜지션

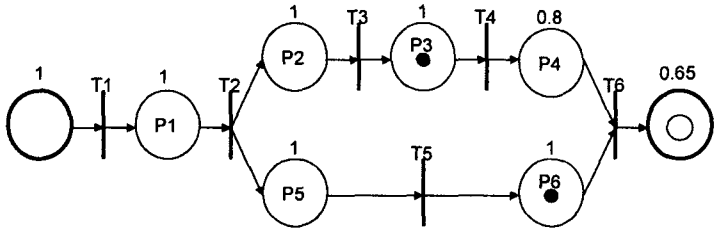
이 만족하였을 때도 처리될 수 있다. 이러한 경우는 토큰의 진행으로 보지 않고 각 플레이스의 퍼지값만을 증가시킨다. 임의의 트랜지션이 만족하였을 때 이웃 플레이스에 퍼지값 증감치(θ)는 아래와 같다.

$$\theta = (F_{before} - F_{after}) \times \varepsilon,\ \varepsilon < 1$$

침입의 판단은 종료 플레이스의 퍼지값을 가지고 한다. 퍼지값의 변화는 종료 플레이스의 퍼지값을 계속 상승시키게 된다. 이 값이 임계치 이상이 되면 침입이라고 판단할 수 있다.

(3) 변형 공격의 대응

위의 퍼지값의 변화를 보여주는 그림 4의 패턴을 예로 들어보자. 현재 침입의 진행은 P3과 P6까지 되었다. 따라서, P4의 퍼지값은 0.8이 되었고 종료 플레이스의 퍼지값은 0.65가 되었다. 만약 임계치가 0.75라면 P4와 P6의 퍼지값으로 판단하여 T6의 사건이 진행 가능하다. 다시 말하면, 침입의 진행에서 T4의 사건이 수행되지 않고 T6의 진행이 이루어지더라도 침입으로 판단할 수 있는 것이다.



(그림 4) 변형공격시 퍼지값

5. 결론

본 논문에서는 Petri-net 형태로 침입탐지 모듈을 구성하였다. Petri-net의 빠른 상태변화와 해쉬 검색으로 실시간 침입탐지가 가능하도록 하였다. 또한, Petri-net의 상태를 보존하여 지연공격에 대응하였고, 모든 사용자에 대한 공동 검색으로 다중공격에 대응할 수 있도록 하였다.

Petri-net에 퍼지값을 적용하였다. 이 값은 침입이 진행됨에 따라 변화되며 종료 플레이스의 퍼지값으로 침입을 판정한다. 또한, 알려진 공격방법을 변형한 변형공격에 대응할 수 있도록 퍼지값이 임계치 이상이면 트랜지션이 진행할 수 있도록 하였다.

퍼지이론은 오용공격 뿐만 아니라 이상공격에도 적용하여 처리할 수 있을 것이다. 앞으로 오용공격에 대한 대응방법과 멀티호스트 환경으로 확장할 수 있는 방법에 대한 연구가 필요하다.

참고문헌

[1] M. Crosbie, B. Dole, T. Ellis, I. Krsul, and E. Spafford, "IDIOT - Users Guide," Technical Report TR-96-050, COAST Lab., Sep., 1996.

[2] D. E. Denning, "An Intrusion-Detection Model," IEEE Trans. on Software Engineering, No. 2, Feb., 1987.

[3] C. Dowell and P. Ramstedt, "The ComputerWatch Data Reduction Tool," 13th National Computer Security Conference, Oct., 1990.

[4] N. Habra, B. L. Charlier, A. Mounji, and I. Mathieu, "ASAX: Software Architecture and Rule-Based Language for Universal Audit Trail Analysis," Proc. of ESORICS'92, Nov., 1992.

[5] K. Ilgun, "USTAT: A Real-time Intrusion Detection System for UNIX," Proc. IEEE Computer Society Symposium on Research in Security and Provacy, May, 1993.

[6] H. S. Javitz and A. Valdes, "The SRI IDES Statistical Anomaly Detector," Symposium on Research in Security and Privacy, May, 1991.

[7] S. Kumar and E. H. Spafford, "An Application of Pattern Matching in Intrusion Detection," Purdue University, Jun., 1994.

[8] G. E. Liepins and H. S. Vaccaro, "Intrusion Detection: Its Role and Validation," Computers & Security, Nov., 1992.

[9] T. F. Lunt, A. Tamaru, and F. Gilham, "IDES: A Progess Report," Proc. 6th Annual Computer Security Applications Conference, Dec., 1990.

[10] T. F. Lunt, "A Survery of Intrusion Detection Techniques," Computer & Security, Vol. 12, No. 4, Jun., 1993.

[11] J. L. Peterson, *Petri Net Theory and the Modeling of Systems*, Prentice Hall, 1981.

[12] P.A. Porras and R.A. Kemmerer, "Penetration state transition analysis: A ruel-based intrusion detection approach," Proc. 8th Annual Computer Security Applications Conference, Nov., 1992.

[13] P. A. Porras, and P. G. Neumann, "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances," 20th National Information Systems Security Conference, 1997

[14] P. Proctor, "Audit Reduction and Misuse Detection in Heterogeneous Environments: Framework and Application," Proc. 10th Annual Computer Security Applications Conference, Dec., 1994.

[15] Sandia National Lab., "Intrusion Detection and Response," National Info-Sec Technical Baseline, http://doe-is.llnl.gov/nitb/docs/nitb.html, Oct., 1996.

[16] S. R. Snapp, J. Brentano, G. V. Dias, T. L. Goan, L. T. Heberlein, C. Ho, K. N.

Levitt, B. Mukherjee, S. E. Smaha, T. Grance, D. M. Teal, and D. Mansur, "DIDS (Distributed Intrusion Detection System) - Motivation, Architecture, and An Early Prototype," 14th National Computer Security Conference, Oct., 1991.

[17] G. B. White and U. Pooch, "Cooperating Security Managers: Distributed Intrusion Detection Systems," Computer & Security, May, 1996.

[18] 노정석, 김휘강, 조용상, 최재철, "특집II - 해킹의 최신 형태와 방지 테크닉," http://www.chosun.com./internetmag/9605/sp2.html, 1996.

[19] 이희조, "인터넷 침입수법/대응방안," NETSEC-KR '97, 1997년 5월.