침입신호 상관성을 이용한 침입 탐지 시스템☆

Intrusion Detection System Using the Correlation of Intrusion Signature

나 근 식*
Guen-Sik Na

요 약

본 논문에서는 네트워크 침입 탐지 시스템의 성능과 탐지 정확성을 높일 수 있는 침입 탐지 시스템의 구조를 제시한다. 네트워크를 통한 침입은 보통 여러 단계의 침입 동작으로 이루어 진다. 각 침입 동작은 특정 침입 신호로 탐지할 수 있다. 그러나 침입이 아닌 보통의 동작도 침입 행위와 같은 신호를 나타낼 수 있다. 따라서 특정 침입 신호로 침입을 판단하는 것은 잘못된 판단을 내릴 수 있게 된다. 제시하는 시스템은 침입을 구성하는 각 단계의 신호들 간의 신호 상관성을 이용한다. 따라서 제시하는 시스템의 침입에 대한 판단은 높은 신뢰성을 가질 수 있다. 또한 알려진 침입에 대한 변형도 잘 탐지할 수 있다.

Abstract

In this paper we present the architecture of intrusion detection system that enhances the performance of system and the correctness of intrusion detection. A network intrusion is usually composed of several steps of action taken by the attackers. Each action in the steps can be characterized by its signature. But normal and non-intrusive action can also include the same signature. It can result in incorrect detection. The presented system uses the correlation of series of signatures that consist of an intrusion. So its decision on an intrusion is highly reliable. And variations of known intrusions can easily be detected without any knowledge of the variations.

Keyword : Intrusion Detection, Intrusion Signature, IDS : Intrusion Detection System

1. 서 론

침입이란 네트워크나 컴퓨터 시스템에 대하여 불법 정보접근, 불법 정보조작, 시스템 무력화 등을 목적으로 행하는 모든 행위를 말한다. 예를들면 권한이 없는 사용자가 시스템에 접근하는 행위나, 접근 권한은 있지만 권한을 초과한 접근을 하는 행위, 네트워크나 컴퓨터 시스템의 자원을 고갈시켜 동작 불능 상태로 만드는 행위 등이 침입에 해당한다. 이러한 행위들은 악의적으로 의도된 행위일 수도 있고, 의도되지 않은 실수일 수도 있다. 의도된 행위이든 실수이든 시스템의 보안 정책(security policy)에 위배되는 모든 행위는 침입으로 취급된다. 침입 탐지 시스템 IDS(Intrusion Detection System)는 이러한 침입 행위를 신속히 탐지하고 대응하는 소프트웨어를 말한다.[1,2,3]

침입탐지 시스템은 침입 판단의 근거가 되는 자료의 수집 위치에 따라 호스트 기반의 침입탐지 시스템(host-based IDS)과 네트워크 기반의 침입탐지 시스템(network-based IDS)으로 분류된다.[1,3] 호스트 기반의 침입탐지 시스템은 시스템 로그나 감사 기록 등과 같이 시스템 내부에서 생성되는 자료를 분석해서 침입 여부를 판단하는 시스템으로, 보호되어야할 시스템마다 최소한 하나의 자료 수집 기능이 있어야 한다. 반면 네트워크 기반의 침입 탐지 시스템은 네트워크 트래픽량이나 네트워크를 지나다니는 패킷 헤더나 내용을 분석해서 침입 여부를 판단하게 된다. 네트워크 기반 침입탐지 방법은 하나의 네트워크 세그

* 정 회 원 : 한신대학교 컴퓨터학과 교수
ngs@hanshin.ac.kr(제 1저자)
☆ 이 논문은 2002년도 한신대학교 학술연구비 지원에 의하여 연구되었음.

먼트에 하나의 침입탐지 시스템만 있으면 될 뿐 아니라 호스트 시스템의 처리 능력을 저하시키지 않기 때문에 최근의 침입탐지 시스템들은 대부분 이 방법을 사용하고 있다.

최근의 침입 기술들은 여러 단계에 걸쳐 공격이 이루어지고 있고, 각 단계마다 탐지가 어려운 복잡한 방법을 사용한다. 여러 단계의 공격을 자동화하는 툴들까지 개발되어 인터넷을 통해 빠르게 전파되고 있다. 또한 쉽게 사용할 수 있는 공격 툴들의 보급으로 공격자의 수가 급격히 증가하고 있을 뿐 아니라 공격의 대상도 끊임없이 변하면서 다양화되고 있어 탐지를 더욱 어렵게 하고 있다. 그 외에도 IDS 자체에 대한 공격이나, 이동 코드를 이용한 침입, 네트워크 속도의 증가와 규모의 확대 등 다양한 환경의 변화에 따라 침입탐지 시스템들간의 상호 협력이나 분산형 침입탐지 시스템에 대한 관심도 높아지고 있다.

본 논문에서는 침입 탐지의 정확성을 높이면서 침입 기법의 부분적인 변화가 있어도 이를 쉽게 탐지할 수 있는 침입 탐지 시스템의 구조를 제시한다. 제시하는 시스템은 침입의 각 단계를 담당하는 탐지 에이전트를 두고, 각 에이전트는 분산 환경에서 독자적인 탐지 기능을 수행하도록 한다. 각 탐지 에이전트들이 탐지한 침입신호는 신호간의 상관성이 고려되어 종합적인 침입 여부가 판정되도록 한다.

이후 본 논문의 제2장에서는 기존 침입 탐지 시스템에서 사용하는 침입 탐지 방법들을 소개하고 문제점을 분석한다. 제3장에서는 제시하는 침입탐지 시스템의 기본 원리를 설명하고, 제4장에서는 시스템의 전체 구조와 각 에이전트의 역할, 탐지 에이전트와 판정 에이전트의 구조 등을 설명한다. 제5장에서는 앞에서 제시된 시스템의 구현에 관련된 사항들을 설명한다. 마지막으로 제6장에서는 향후 개선 방안과 함께 결론을 맺는다.

2. 침입 탐지 기술

침입탐지 방법은 침입 여부를 판단하는 방법에 따라 침입신호 탐지(Signature Detection) 방법과 비정상행위 탐지(Anomaly Detection) 방법으로 나누어 진다.[3]

침입신호 탐지 방법은 침입 행위의 특징이 되는 신호를 탐지하는 방법이다. 침입 신호는 패킷 헤더의 특정 필드 값이나, 데이터 필드내에 포함된 어떤 패턴으로 나타낼 수 있고, 알려진 침입 행위를 분석해서 알아낸다. 예를들면 ICMP를 이용한 스머프 스캔은 데이터 크기가 4 바이트이고, 타입은 8을 사용한다. 따라서 이러한 특징을 가진 패킷은 침입 패킷으로 의심할 수 있다. 또한 대부분의 버퍼 오버플로우 공격 패킷은 데이터 필드에 일정 패턴을 포함하므로 단순한 패턴 매칭으로도 탐지할 수 있다. 침입신호 탐지 방법은 탐지의 정확도는 높으나 침입신호가 이미 알려진 침입 행위만 탐지가 가능하다는 것이 문제점이다. 이러한 방법을 사용하는 시스템으로는 EMERALD [8], NetSTAT[9], Bro[10], NFR[11], NADIR[12] 등이 있다.

반면 비정상행위 탐지 방법은 정상적인 행위에서 크게 벗어나는 행위를 침입으로 판단하는 방법으로 인공 지능이나, 신경망 시술들을 사용한다. 정상적인 행위는 현재까지 관찰된 행위를 근거로 시스템이 자동으로 학습하게 된다. 이러한 방법은 새로운 침입 유형이나 기존 침입 유형의 변종도 탐지가 가능하다는 장점이 있지만, 정상적인 동작에 대한 정의가 애매해서 잘못된 판단을 내릴 가능성이 커지게 된다. 이러한 방법을 사용하는 시스템은 NIDES[13]이 있고 그 외에 많은 연구가 진행중이다.[14][15]

기존의 침입신호 탐지 방법이나 비정상 행위 탐지 방법은 정의된 침입 신호나 비정상 행위는 비교적 잘 탐지하지만 새로운 침입이나, 기존 침입의 변형은 탐지하기 어렵다. 또한 정상적인 행위를 침입으로 판단하는 경우(false positive)가 너무 많다. 예를들어 앞에서 말한 데이터 크기가 4 바이트이고 타입이 8 인 ICMP 패킷 모두가 침입

행위는 아니다. 또한 버퍼 오버플로우에서 나타나는 패턴이 포함된 패킷이라고 해서 모두 침입에 관련된 패킷은 아니다.

이러한 잘못된 판단으로 인한 침입 경보(alert)를 줄이며, 새로운 침입이나 기존 침입의 변형도 탐지하기 위한 여러가지 연구가 이루어 지고 있다. 이들은 전체적으로 탐지된 침입 경보들을 집단화해서 보고되는 침입 경보 수를 줄이며, 유사한 특성을 가진 새로운 침입을 찾아낸다. [16]에서는 침입 경보들 사이의 인과 관계를 기술해서 일련의 침입 경보가 하나의 침입 행위를 위한 것임을 찾아낼 수 있도록 하고 있다. [17]에서는 통계적 방법을 이용해 침입 경보들 사이의 유사성을 찾아 유사성이 있는 침입 경보들을 집단화 한다. 또한 [18] 에서는 일련의 침입 경보들 사이의 상관성을 찾아 하나의 시나리오에 의해 발생된 경보들임을 밝히기 위해 인공 지능 기법을 사용한다.

본 연구에서는 하나의 의도된 침입은 여러 단계의 침입 시도로 구성된다는 사실을 근거로 각 단계에 있는 침입 시도들을 하나의 침입 시나리오로 집단화 한다. 침입 판단을 위해 시나리오에 포함되는 각 단계 침입 경보의 가중치 합을 계산하고 이 합이 임계값을 초과하면 침입으로 판단한다. 이렇게 해서 잘못된 침입 경보나 위험하지 않은 개별 침입 경보에 대한 보고를 억제하고, 전체 시나리오에 대한 침입 경보만 보고하므로써 경보 발생 횟수를 줄일 수 있다. 또한 시나리오가 완성되기 전이라도 가중치 합이 임계값을 초과하면 경보를 발생시켜 더 치명적인 피해를 막을 수 있다.

3. 침입신호 상관성을 이용한 침입탐지 기법

본 장에서는 제시하는 침입 탐지 시스템에서 사용하는 침입신호 상관성을 이용한 침입탐지 방법을 설명한다.

3.1 기본 원리

침입의 목적은 시스템을 장악해서 불법으로 사용하거나, 시스템에 저장된 정보를 변조 혹은 유출하거나, 시스템을 사용 불능 상태로 만드는데 있다. 침입자는 침입의 목적을 달성하기 위해 네트워크 정찰 단계에서부터 시작해서 여러 단계에 걸쳐 각 단계마다 다양한 기법들을 사용해서 침입을 시도한다. 침입의 각 단계를 구분하는 합의된 원칙은 없지만 대략적으로 정찰 단계, 취약점 분석 단계, 침투 단계, 제어 프로그램 설치 단계, 데이터 유출 단계, 공격 전이 단계 등의 단계로 구분할 수 있다.

침입의 각 단계는 다음 단계를 위해 필요한 정보를 얻거나 혹은 시스템 설정을 변경하거나 제어 프로그램을 설치하는 등과 같은 준비 과정이다. 침입의 초기 단계 행위들은 시스템에 직접적인 피해를 주지는 않지만 다음 단계들을 위해 반드시 필요한 단계이다. 그러나 초기 단계 동작들이 시도되더라도 완전히 성공하지 못하면 다음 단계의 침입을 시도하기가 어렵다. 또한 초기 단계 침입 시도가 성공하더라도 침입자가 자발적으로 혹은 다른 어떤 이유에서 침입을 포기하거나, 또는 침입이 저지될 수 있어 다음 단계 침입 시도로 바로 이어지지는 않는다. 일반적인 침입탐지 시스템들은 침입의 어떤 단계에서 나타나는 특징들을 탐지하여 침입 여부를 판정하기 때문에 정상적인 동작을 침입으로 판정하거나 교묘하게 위장된 침입 행위를 정상적인 행위로 판단하는 경우가 많다.

본 연구에서는 침입의 각 단계에서 나타나는 특징들을 종합적으로 평가해서 침입 여부를 판정하는 침입탐지 방법을 제시하고, 그 개념을 이용한 침입탐지 시스템의 구조를 제시한다. 침입의 각 단계에서 사용되는 기법들은 특징적인 침입 신호로 탐지할 수 있는데, 제시하는 시스템에서는 모든 알려진 침입 신호에 대해 신호 위험도를 정의한다. 신호 위험도값은 신호의 중요성이나 탐지

빈도수 등에 따라 정해진다. 또한 하나의 침입이 성공하려면 각 단계별로 하나 이상의 기법들이 사용되며, 하나의 침입에 관련된 각 단계별 기법들은 연관성이 있다고 가정하고, 이러한 연관성을 나타내는 신호 상관도를 정의한다. 신호 상관도는 하위 단계의 신호가 현재 단계의 신호와 얼마나 밀접한 관련이 있는가를 나타내는 값이다. 각 단계에서는 특정 침입신호가 탐지될 때마다 그 신호가 실재적인 침입인가 아닌가를 나타내는 침입 위험도를 계산한다. 이때 각 단계에서 계산하는 특정 침입신호의 침입 위험도는 그 신호의 신호 위험도값과 하위 단계의 모든 신호의 침입 위험도에 신호 상관도를 반영해서 합한 것으로 정의한다. 즉, 특정 신호가 실재적인 침입인가 아닌가는 하위 각 단계에서 탐지된 침입신호들과 현재 단계에서 탐지된 침입신호가 얼마나 연관성이 있고 또 위험한가에 따라 판단된다.

이러한 침입 탐지 방법은 여러 가지 장점이 있다. 우선 여러 단계에 걸쳐 수집된 침입 징후들을 종합해서 침입 판단을 하므로 잘못된 판단을 할 가능성이 적다. 또한 전체적인 침입위험도값으로 침입 여부를 판정하므로 전체 단계 중 하나 혹은 두 단계 정도에서 새로운 기술을 사용하는 새로운 침입 시도를 쉽게 탐지할 수 있다. 각 단계 에이전트가 공격을 받거나 부분적인 장애가 발생되더라도 비록 탐지 능력의 손실은 있지만 전체 시스템의 탐지 기능이 마비되지는 않는다. 그리고 에이전트들간의 정보 교환이 최소화되므로 불필요한 네트워크 트래픽을 유발하지 않는다. 그러나 각 단계에서 신호 위험도나 신호 상관도값, 침입 위험도 임계치 등에 따라 전체 탐지 능력이 민감하게 반응할 수 있으므로 미세한 조정을 필요로 한다.

3.2 신호 위험도

어떤 종류의 침입신호는 한번만 탐지되어도 침입이라고 추정해야 하는 것이 있는 반면 또 다른 종류의 침입신호는 한 두 번은 문제가 되지 않지만 여러번 반복되어 나타나면 침입으로 판단해야 하는 것들도 있다. 각 침입신호별로 신호 위험도를 적절히 계산할 수 있도록 각 침입 신호에 대해 다음 값들이 정의된다.

S_0 : 최초 탐지시 신호 위험도 값
S_n : 두 번째 이후 탐지시 신호 위험도 값
$S_{\max}$: 신호 위험도 최대값

신호 탐지 횟수가 중요한 침입신호의 경우에는 $S_0 = S_n$이고 탐지 횟수가 아니라 탐지 자체가 중요한 침입신호의 경우에는 $S_n = 0$이다. 정의된 값들을 이용해서 침입신호가 탐지될 때마다 다음과 같이 신호 위험도가 계산된다.

$$S = \max(S_{\max}, S_0 + (n-1) \times S_n) \quad (1)$$

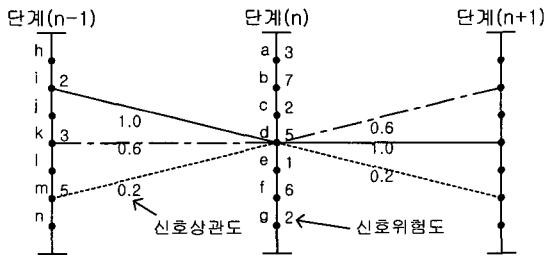
여기서 max(a, b)는 a와 b 중에서 큰 값을 선택하는 함수이고, n은 침입신호 탐지 횟수이다.

3.3 침입 위험도

각 단계에서의 침입 여부에 대한 판단은 탐지된 침입신호의 침입 위험도로 판단한다. 어떤 단계 n에서 침입 위험도는 다음과 같이 결정된다.

$$I_{ni} = S_{ni} + \sum_{j=1}^{k} I_{(n-1)j} \times C_{nij} \quad (2)$$

여기서
I_{ni} : 단계 n에서 신호 i의 침입 위험도
S_{ni} : 단계 n에서 신호 i의 신호 위험도
C_{nij} : 단계 n의 신호 i와 단계 (n-1)의 신호 j 간의 신호 상관도
k : 단계 (n-1)에서의 침입신호 수
이다. 단, 단계 1에서는 침입위험도가 신호위험도와 같아서 $I_{1i} = S_{1i}$이다.



〈그림 1〉 단계별 침입신호의 연관성

(그림1)에서 단계(n)에는 7개의 침입 신호가 정의되어 있고 각 신호의 위험도값이 부여되어 있다. 단계(n)의 신호 d는 단계(n-1)의 신호 i, k, m과 관련되어 있다. 즉, 단계(n-1)의 신호 i, k, m과 관련된 공격 기술을 사용하여 얻은 정보로 단계(n)의 신호 d와 관련된 침입시도를 하는 침입이 있을 수 있음을 나타낸다. 따라서 단계(n-1)에서 신호 i가 탐지되고, 단계(n)에서 신호 d가 탐지되면 이러한 징후는 실재 침입과 관련된 것일 가능성이 크다.

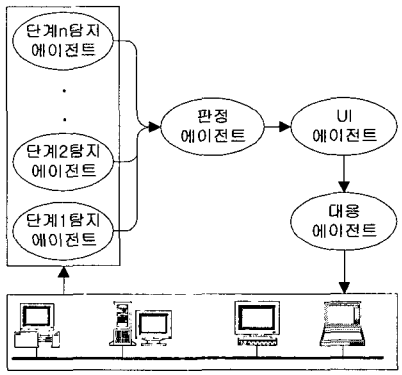
n이 2 일때 단계(n-1)이 1 단계 이므로 이때의 신호 d의 침입 위험도는 2 * 1.0+3 * 0.6+5 * 0.2+5=8.0이 된다. 8.0이 침입에 해당하는지 아닌지는 각 신호의 침입 위험도 임계치에 의해 결정될 수 있다.

4. 시스템 구조

제시하는 침입 탐지 시스템은 에이전트를 기반으로 설계되었다. 시스템 내의 각 에이전트들은 여러 네트워크 노드에 분산되어 있을 수도 있고 여러 에이전트가 같은 노드에 존재할 수도 있다.

4.1 시스템 전체 구조

전체 시스템은 (그림2)와 같이 각 침입 단계의 침입 신호를 탐지하는 탐지 에이전트들과 탐지 결과를 종합하여 침입 여부를 판정하는 판정 에이전트, 그리고 침입 판정 결과를 사용자에게 전



〈그림 2〉 침입탐지 시스템 구조

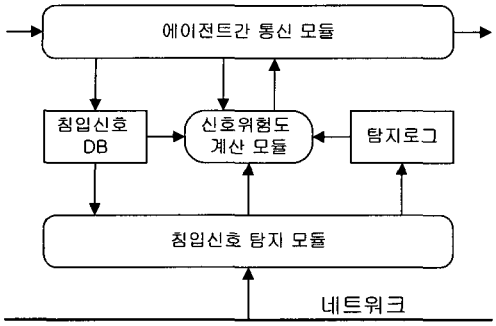
달하는 사용자 인터페이스 UI (User Interface)에이전트, 그리고 발견된 침입에 대해 적절한 조치를 취하는 대응에이전트로 구성된다.

UI 에이전트는 탐지 에이전트의 침입 탐지 보고를 사용자에게 제시하며, 사용자의 지시를 대응 에이전트에게 전달한다. 필요한 경우에는 사용자를 대신해서 침입 탐지 보고에 대한 처리를 하여 대응 방법을 대응 에이전트에게 직접 지시하기도 한다. 또한 침입탐지 에이전트 구성을 위한 화면과 로그 분석을 위한 화면도 제공한다. 대응 에이전트는 UI 에이전트의 지시에 따라 침입에 대응한다. UI 에이전트와 대응 에이전트는 본 논문의 주제가 아니므로 여기서는 더 이상 언급하는 않는다.

4.2 탐지 에이전트

탐지 에이전트는 침입 단계별로 하나씩 존재한다. 각 단계의 탐지 에이전트는 지정된 침입신호를 탐지하여 각 침입신호별로 신호 위험도를 계산해 결과 값을 판정 에이전트에게 보고한다. 탐지 에이전트의 구조는 (그림3)과 같이 에이전트간 통신 모듈, 침입 신호 탐지 모듈, 신호 위험도 계산 모듈, 침입 신호 DB, 탐지 로그로 구성된다.

침입 신호 DB 에는 탐지 에이전트가 침입 신호 탐지, 신호 위험도 계산에 사용할 정보가 들어



〈그림 3〉 탐지 에이전트의 구조

있다. 침입 신호 DB의 각 엔트리는 다음과 같은 구조를 가진다.

```
struct SigRecord {
  int SigId;          // 신호 식별자
  char *SigPattern;   // 신호 패턴
  float S0, Sn, Smax;
  float LifeTime;     // 신호 유효 기간
  float Threshold;    // 보고 임계치
};
struct SigRecord SignalDB[];
```

신호 식별자 SigId 는 침입 신호들을 구분하는 식별자로 숫자로 표현된다. 각 침입 단계의 침입 신호마다 그 단계 내에서 유일한 식별자를 가진다. 신호 패턴 SigPattern 은 침입 신호로 프로토콜 헤더 필드 값이나 메시지 길이, 메시지 내용의 값 등으로 표현된다. 침입 신호의 표현 방법은 탐지 모듈에 의존적인데, 이에 대해서는 4 장 구현 부분에서 추가로 다룬다. 신호 위험도 파라메타는 신호 위험도 계산을 위한 S_0, S_n, S_{max} 값이 있다. 신호 유효 기간 LifeTime은 탐지된 침입 신호의 유효 기간으로 이 기간이 경과한 침입 신호는 신호 위험도 계산에서 제외된다. 그러나 탐지 로그에는 그대로 남아 있다. 보고 임계치 값 Threshold는 판정 에이전트에게 보고할 신호 위험도 임계치로 탐지 에이전트는 계산된 신호 위험도값이 이 값을 초과하면 신호 위험도 값을 판정 에이전트에게 보고해서 전체적인 침입 위험도가 다시 계산될 수 있도록 한다.

탐지 에이전트는 침입 신호 DB 의 각 침입 신호에 대해 침입 신호 탐지 모듈을 이용해 신호를 탐지한다. 탐지된 침입 신호는 다음에 신호 위험도 계산을 위해 탐지 로그에 기록된다. 탐지 로그는 각 침입 신호 종류별로 분리되어 저장되며 로그의 각 엔트리는 다음과 같은 구조를 가진다.

```
struct SigLog {
    int SigId;     // 신호 식별자
    long Time;     // 탐지 시각
    char *msg;     // 탐지 내용
};
```

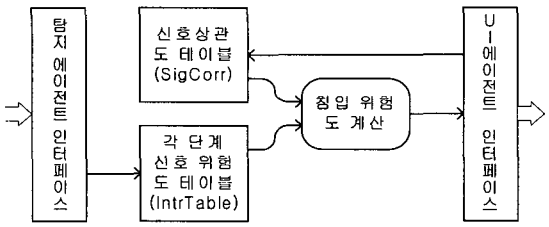
신호 식별자는 침입 신호 DB 에 있는 것과 같은 값으로 침입 신호를 구분하는 식별자이다. 침입 신호가 탐지되면 탐지 시각 Time과 함께 침입 신호를 포함하는 메시지 내용 msg가 저장된다.

침입 신호가 탐지될 때마다 탐지된 신호에 대한 신호 위험도가 2.2 절의 식(1)에 따라 계산된다. 이때 신호 유효 기간이 경과한 침입 신호는 신호 위험도 계산에서 제외된다. 계산된 신호 위험도가 보고 임계치 값보다 크면 그 결과가 판정 에이전트에게 보고된다.

판정 에이전트는 침입 판정을 위해 특정 탐지 에이전트에게 해당 단계의 침입 신호들에 대한 신호 위험도 계산을 요청하기도 한다. 이 때는 탐지 에이전트는 계산된 신호 위험도가 보고 임계치를 초과하지 않더라도 그 결과를 보고한다.

4.3 판정 에이전트

판정 에이전트는 침입 여부를 판단하기 위해 각 탐지 에이전트의 신호 위험도 보고를 근거로 종합적인 침입 위험도를 계산한다. 침입 위험도 계산은 UI 에이전트의 요청이나 탐지 에이전트의



〈그림 4〉 판정 에이전트 구조

신호 위험도 보고에 의해 시작되는데 결과는 UI 에이전트에게 보고된다. 이러한 판정 에이전트의 구조는 (그림4)와 같다.

탐지 에이전트가 보고한 신호 위험도 값은 다음과 같은 신호 위험도 테이블 IntrTable 에 저장된다. 여기서 MaxLevel은 침입 단계의 수를 나타내고, MaxSig는 한 탐지 레벨에서 가질 수 있는 침입 신호의 최대수를 나타낸다.

```
struct IntrValue {
    int SigId;        // 신호 식별자
    float Value;      // 신호 위험도 값
    long RTime;       // 보고 시간
};
struct IntrValue IntrTable[MaxLevel][MaxSig];
```

한편 판정 에이전트는 각 단계의 침입 신호에 대한 신호 상관도 테이블을 다음과 같은 3 차원 배열로 가지고 있다.

```
float SigCorr[MaxLevel][MaxSig][MaxSig];
```

SigCorr은 침입 단계 ILevel의 신호 0~MaxSig-1과 침입단계 ILevel-1의 신호 0~MaxSig-1에 대한 신호 상관도를 가지고 있다. 이 값은 사용자가 UI 에이전트를 통해 변경할 수 있다.

판정 에이전트는 UI 에이전트의 요청이 있거나 혹은 탐지 에이전트의 침입 신호 보고에 의해 어느 단계의 신호 위험도가 변경되면 다음과 같이 침입 위험도 계산을 다시 한다.

```
for (n = 1; n < MaxLevel; n++) {
    for (i = 0; i < MaxSig; i++) {
        k = 0;
        for (j = 1; j < MaxSig; j++) {
            k += IV[n-1][j]*SigCorr[n][i][j];
        }
        IV[n][i] = IntrTable[n][i] + k;
    }
}
```

여기서 각 침입 신호의 침입 위험도는 IV 에 저장되는데, 단계 0 의 침입 위험도 IV[0][] 는 모두 0 으로 초기화 되어 있다고 가정한다. 최종적으로 계산된 신호 i 의 침입 위험도는 IV[MaxLevel-1][i]에 있는데 이 값들이 UI 에이전트에 보고된다.

4.4 에이전트간 메세지

에이전트들은 같은 노드에 있을 수도 있고, 서로 다른 노드에 있을 수도 있다. 각 에이전트들은 메세지 교환 방식으로 통신하며 다음과 같은 메시지들이 정의되어 있다.

```
ADD_IS <SigId, SigPattern, S0, Sn, Smax,
        LifeTime, SigThr>
DEL_IS <SigId>
SET_CORRTBL <SigCorr[][][]>
REQ_IVAL < >
REP_IVAL <IVAL[]>
```

먼저 ADD_IS 메시지는 UI 에이전트가 탐지 에이전트에게 전송하는 메시지로 침입 신호 DB 에 새로운 침입 신호 SigId를 추가하려고 할 때 사용한다. 침입 신호가 이미 침입 신호 DB 에 있었다면 이를 변경한다. DEL_IS 메시지는 침입 신호 DB에서 침입 신호 SigId를 제거하기 위해 UI 에이전트가 탐지 에이전트에게 전송한다. UI 에이

전트는 또한 SET_CORRTBL 메시지를 이용해 판정 에이전트의 신호 상관도 테이블 값을 변경한다.

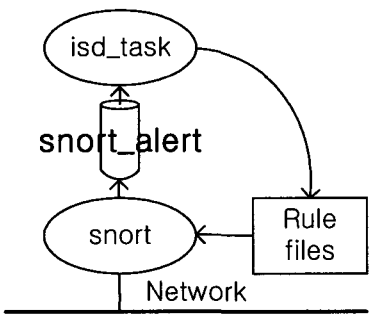
판정 에이전트는 탐지 에이전트에게 REQ_IVAL 메시지로 각 신호에 대한 신호 위험도 보고를 요청하고 탐지 에이전트는 이에 따라 신호 위험도를 재계산하여 그 결과를 REP_IVAL 메시지로 보고한다. 이 메시지들은 UI 에이전트와 판정 에이전트 사이에도 사용되는데 이 때는 UI 에이전트가 REQ_IVAL 메시지로 침입 위험도 보고를 요청하면 판정 에이전트가 침입 위험도를 계산해서 그 결과를 REP_IVAL 메시지로 UI 에이전트에게 보고한다. 탐지 에이전트는 탐지된 신호 위험도가 임계치를 넘으면 판정 에이전트의 요청이 없어도 REP_IVAL 메시지를 전송하고, 이 경우에 판정 에이전트는 침입 위험도를 다시 계산해서 그 결과를 UI 에이전트에게 보고한다.

5. 구현 및 실험 결과

본 연구에서 제안한 침입 탐지 시스템은 현재 리눅스 시스템에서 프로토타입 구현되었다.

5.1 구현

프로토타입은 탐지 에이전트와 판정 에이전트가 구현되었고, UI 에이전트는 우선 텍스트 인터페이스를 가지도록 구현되었다. 시스템 내의 각 에이전트들은 독립 프로세스로 구현되어 서로 다른 컴퓨터에 존재할 수 있도록 하였다. 에이전트 간의 통신은 인터넷 도메인의 스트림 소켓으로 이루어 진다.



〈그림 5〉 침입신호 탐지 모듈

현재 각 단계의 침입 신호 탐지 기능은 snort의 신호 탐지 기능을 활용한다. (그림5)와 같이 독립 프로세스로 수행되는 snort는 침입 신호 탐지 타스크(isd_task)와 유닉스 도메인 소켓에 의해 연결된다. isd_task는 침입 신호 DB 의 각 신호에 대한 신호 패턴으로 snort의 규칙 파일(rule file)을 생성한 후 출력 모드를 alert_unixsock 으로 하여 snort를 동작시킨다. snort는 주어진 패턴에 대한 신호를 탐지하면 이를 이름이 "/dev/snort_alert"인 유닉스 도메인 소켓에 전달한다. isd_task는 이 소켓을 통해 침입 신호 보고를 받아 이를 로그에 기록하고 침입 신호 DB의 정보를 이용해 신호 위험도를 계산한다.

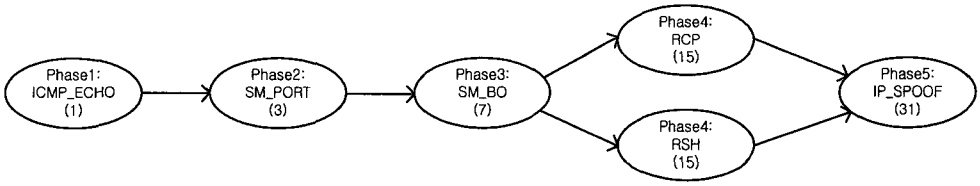
5.2 실험 및 결과

구현된 프로토타입에 대한 침입 탐지 능력 실험은 DARPA 2000 침입 탐지 시스템 평가 데이터 중에서 첫 번째 실험 데이타를 사용하여 수행하였다.[19] 이 데이터는 5 단계로 이루어진 침입에 대한 전체 데이터로 tcpdump 형식의 파일로 제공되고 있다. DARPA 2000 의 침입 시나리오는 다음과 같은 5 단계로 이루어진다.

단계1 : 1024 개의 연속된 IP 주소에 대해 ICMP를 이용해 IPsweep을 한다. 이 단계의 침입 신호는 ICMP Echo Request 패킷이다.

단계2 : UDP 포트매퍼를 이용해 앞 단계에서 발견된 호스트 중에서 sadmind 관리 툴이 존재하는 호스트를 파악한다. 이 단계의 침입 신호는 UDP 포트매퍼를 통한 RPC sadmind 포트 요청이다.

단계3 : 발견된 sadmind 툴에 원격 버퍼 오버플로우 공격을 해서 "hacket2"라는 이름의 관



〈그림 6〉 단계별 침입 위험도 값

리자 권한을 가진 계정을 만든다. 이 단계의 침입 신호는 RPC sadmind 포트를 통한 버퍼 오버플로우 패킷이다.

단계4 : telnet을 이용해 hacket2로 로그인 해서 rcp와 rsh를 이용해 DDoS 공격 툴을 설치한다. 이 단계의 침입 신호는 rcp 나 rsh 포트에 대한 접근이다.

단계5 : telnet을 이용해 다시 로그인 해서 DDoS 공격을 시작한다. DDoS 공격은 랜덤하게 생성한 목표 IP에 대해 최대 속도로 패킷을 전송하는 것이다. 이 단계의 침입 신호는 스푸핑된 IP 주소이다.

실험에서는 제공되는 데이터 파일을 탐지 에이전트의 오프라인 입력으로 주고 실험을 수행하였다. 침입 위험도가 하나의 의도된 침입을 찾아 낼 수 있는가를 알아보기 위해 각 침입 신호가 탐지될 때마다 판정 에이전트의 침입 위험도 값의 변화를 관찰하였다. 5단계의 모든 데이터 처리가 완료된 후의 침입 위험도 값 일부를 (그림6)에 나타내었다. 실험에서는 모든 침입 신호의 신호 위험도를 1로 하고, 시나리오에 포함되는 신호 상관도 값을 2로 하였다. (그림6)에서 보는 것처럼 모든 신호 위험도를 1로 하고 있지만 단계 3에서 침입 위험도는 7이 되고, 단계 5가 되면 침입 위험도가 31이 된다. 이 정도의 큰 값이면 충분히 침입이라고 판단할 수 있으므로 시나리오가 알려진 침입은 쉽게 탐지할 수 있다.

신호 상관도를 2로 정한 것은 실험에 사용된 5단계 시나리오가 알려진 침입 시나리오이기 때문이다. 만약 시나리오가 정확히 알려져 있지 않은 기존 침입 방법의 변형 예를들어, 단계3과 단계4의 관계가 알려져 있지 않아 이 사이의 신호 상관도 값을 1로 두는 경우를 가정해 보자. 이 때의 단계 5에서의 침입 위험도는 17이 된다. 이 값 역시 개별 침입 신호의 위험도 값의 단순 합인 5에 비하면 상당히 큰 값이므로 침입으로 판단할 수 있다. 따라서 기존 침입 방법에 일부 수정이 가해진 침입의 경우에도 탐지 가능성이 높아진다.

6. 결 론

본 논문에서는 침입 신호 상관성을 이용한 침입 탐지 시스템의 구조를 소개하였다. 단일 침입 신호만으로 침입 여부를 판단하는 기존의 침입 탐지 시스템은 침입이 아닌 정상적인 동작을 침입으로 판단하는 경우가 많았다. 제시하는 시스템은 하나의 침입을 이루는 각 침입 단계의 특징적인 침입 신호들을 탐지하고 이들 사이의 상관성을 이용하여 전체적인 침입 여부를 판단한다. 이러한 구조의 시스템은 잘못된 판단을 할 가능성이 작고, 기존 침입 방법의 변형도 쉽게 탐지해 낼 수 있다. 뿐만 아니라 각 단계의 침입 신호 탐지를 독립적인 에이전트가 수행하도록 하여 침입 탐지 시스템의 부하 분산이 자연스럽게 이루어질 수 있다.

제시하는 시스템은 현재 리눅스 시스템 상에서 프로토타입 구현이 이루어져 침입 능력에 대한 확인을 했다. 본 연구와 관련되어 기존의 여러 침입 방법들에 사용된 시나리오를 표현할 수 있는 그래픽 도구와 신호 상관도 값 결정 방법에 대한 연구가 더 이루어져야 할 것이다.

참 고 문 헌

[1] 김병구, 정태명, "침입 탐지 기술의 현황과 전망", 정보과학회지 18권 호, 2000.1

[2] Dorothy E Denning, "An Intrusion-Detection Model", IEEE Transactions on Software Engineering, 13(2):222-232, Feb. 1987

[3] Biswanath Mukherjee, L Todd Heberlein, and Karl Levitt, "Network Intrusion Detection", IEEE Network, 8(3):26-41, May 1995

[4] Stephen NorthCutt, Judy Novak, "Network Intrusion Detection : An Analyst's Handbook, 2001, New Riders

[5] Koral Ilgun, Richard A Kemmerer, and Phillip A Porras, "State transition analysis: A rule-based intrusion detection approach", IEEE Transaction on Software Engineering, 21(3): 181-199, March 1995

[6] Martin Roesch, "SNORT-Lightweight Intrusion Detection for Networks", Proceedings of LISA '99, 1999.11

[7] Martin Roesch, "Snort Users Manual, Snort Release:1.9.x", 2002.4

[8] P.A. Porras and P.G. Neumann, "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances", Proceeding of the IEEE Symposium on Research in Security and Privacy, May 1997

[9] Kemmerer, Richard A. "NetSTAT : A Model-Based Real-Time Network Intrusion Detection System", <http://www.cs.ucsb.edu/~kemm/netstat.html/ducuments.html>

[10] Paxson, Vern, "Bro: A System for Detecting Network Intruders in Real-Time", Proceedings of 7th USENIX Security Symposium, Jan. 1998

[11] Ranum, Marcus J., et al. "Implementing a Generalized Tool for Network Monitoring", http://www.nfr.net/forum/publications/LISA-97.htm

[12] Kathleen A Jackson, David H DuBois, "NADIR : An expert system application for network intrusion detection", Proceeding of th 14th National Information Systems Security Conference, 1991

[13] Anderson, Debra, "Next-Generation Intrusion Detection Expert System", <http://www.sdl.sri.com/nides.index5.html>

[14] Stephanie Forrest, Steven A. Hofmeyr, et al, "A sense of self for Unix processes", Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy, 1996.

[15] C.Ko, M. Ruschitzka, K. Levitt, "Execution monitoring of security-critical programs in distributed systems : A specification based approach", Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy, 1997.

[16] Peng Ning, Douglas S. Reeces, Yun Cui, "Correlating Alert Using Prerequisites of Intrusion", <ftp://ftp.csc.ncsu.edu/pub/tech/2001/TR-2001-13.ps.Z

[17] A. Valdes, K. Skinner, "Probabilistic alert correlation", "Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection", 2001

[18] O. Dain, R.K. Cunningham, "Fusing a heterogeneous alert stream into scenario", Proceeding of the 2001 ACM Workshop on Data Mining for Security Application", 2001

[19] Lincon Lab MIT. "DARPA 2000 intrusion detection evaluation datasets", http://www.ll.mit.edu/IST/ideval/data/2000/2000_data_index.html

◐ 저 자 소 개 ◑

나 근 식

1986년 고려대학교 전자공학과 졸업(학사)
1988년 고려대학교 대학원 전자공학과 졸업(석사)
1992년 고려대학교 대학원 전자공학과 졸업(박사)
1992~현재 : 한신대학교 컴퓨터학과 교수
관심분야 : 컴퓨터 네트워크, 네트워크 모니터링, 정보보호
E-mail : ngs@hanshin.ac.kr