韓國 컴퓨터情報學會 論文誌
第9卷 第1號, 2004. 3.

2004-9-1-1-8

이동에이전트를 이용한 침입탐지 모델의 제안

황 인 선*, 박 경 우**

Proposed of Intrusion detection model using the Mobile agent

In-Sun Hwang*, Kyung-Woo Park**

요 약

컴퓨터네트워크의 확대와 인터넷 이용자의 증가에 따른 부작용으로 컴퓨터 보안 문제가 중요하게 대두되고 있다. 따라서 침입자들로부터 위험을 줄이기 위한 침입 탐지 시스템에 관한 연구가 활발하다. 이동에이전트를 이용하는 제안된 컴퓨팅 패러다임의 잇점은 네트워크의 지연시간 극복, 네트워크 부하 감소, 비동기적이고 자율적인 실행, 동적인 적합성과 이기종 환경에서의 운영이다. 많은 정보 보호 모델들은 agent-to-agent, agent-to-platform, 그리고 platform-to-agent 위험한 요소들을 완화하기 위하여 제안되었다. 본 논문에서는 침입 탐지 시스템의 개발을 통해서 이동 에이전트의 성능을 분석하여 관리함으로써 데이터가 최상의 환경을 유지하도록 하였다.

Abstract

The computer security is considered important due to the side effect generated from the expansion of computer network and rapid increase of the use of internet. Therefore, Intrusion detection system has been an active research area to reduce the risk from intruders. A number of advantages of using mobile agent computing paradigms have been proposed. These advantages include: overcoming network latency, reducing network load, executing asynchronously and autonomously, adapting dynamically, and operating in heterogeneous environments. Many information security models have been proposed to mitigate agent-to-agent, agent-to-platform, and platform-to-agent element risks. In these paper, We have an object which is that through intrusion detection system development, the mobile agent is managed and through the analysis of performance data, the best environment is served.

▸ Keyword : mobile agent, intrusion detection, raw data, packet

* 광주보건대학 컴퓨터보안과 교수
** 목포대학교 컴퓨터 공학과 교수

Ⅰ. 서 론

최근 정보기술의 급속한 발전에 따라 정보시스템의 형태는 네트워크에 연결된 분산시스템으로 변화하고 있으며 보다 많은 사용자들이 네트워크를 통합하여 하나의 서브넷을 구성하여 원격제어, 공유의 방법으로 발전하고 있다[1].

또한 인터넷을 대상으로 한 사이버 공격의 경향은 분산 환경에서 다수 공격자의 대규모 분산 서비스 거부 공격(DDoS)의 출현과, 해외 해커들의 국내 전산망을 우회 루트로 활용하여 국제적인 사이버 공격 사례의 증가 등 고도화된 불법 행위가 점차 범죄의 강력한 주요 수단으로 이용되는 추세에 있다. 또한, 정부 부처 및 기업의 모든 정보 유통이 컴퓨터와 인터넷에 크게 의존함에 따라 국내의 해커, 불만 그룹, 적성적 내부자뿐만 아니라 외국 정보기관, 군사 조직, 테러리스트, 범죄자, 산업 경쟁 상대 등으로부터 사이버 공격의 위협이 증대되고 있다[2].

이러한 공격에 대응하기 위한 현재의 보안 기술은, 침입자에 대한 대응에 중점을 두기 보다는 자신의 서버를 어떻게 효율적으로 방어할 것인가에 주안점을 두고 발전하고 있는 상태이다. 따라서 해당 침입자의 공격을 탐지하였음에도 불구하고 침입자에 대한 대응이 자신의 서버 상에 그침으로써, 침입자는 인터넷을 자유로이 이용할 수 있게 되고 이로 인해 제2, 제3의 공격을 허용하게 된다. 또한 새로운 기술을 채택한 공격 방법이 등장하였을 경우, 이에 대한 탐지 및 대응 기술과 이를 수용한 시스템이 개발되기까지는 해당 공격에 대하여 몇 개월 동안 아무런 동작도 취할 수 없는 상태이다.

따라서, 침입자에 대해서는 현재의 보안 기술에 비해서 좀더 능동적(active)이고 공격적(aggressive)인 대응을 수행할 수 있는 보안 메커니즘을 지니며, 그 실행 구조에 있어서는 새로운 공격 기술에 대한 탐지 및 대응 기술을 손쉽게 수용할 수 있는 유연하고 적합성(adaptive)을 가지는 구조의 보안 프레임워크가 필요하다고 할 수 있다.

따라서 본 논문에서는 이동 에이전트를 이용하여 에이전트 관리자가 각각의 서버에 대해 이동 정보의 기록을 유지하여 코드를 이동하는 이동에이전트 기술에 대하여 2장에서 설명하고, 3장에서는 일련의 침입에 모델과 특성에 대하여 살펴봄으로서 이동에이전트의 기술이 침입 탐지에 운영되어지는 동작 과정을 4장에서 설명하고, 5장에서는 트래픽 근원지를 인접한 링 네트워크로 규정하여 서버에 침입이 발생하면 에이전트 관리자는 즉각적인 모니터링을 통하여서 침입을 탐지할 수 있도록 이동 에이전트를 이용하여 침입 탐지에 대한 프레임워크를 설계, 제안하였다.

Ⅱ. 이동에이전트기술

이동 에이전트 기술은, 에이전트 기술의 한 형태이다. 「Agent」라고 하는 말에 「대리인」이라고 하는 의미가 있도록, 에이전트 기술은 「유저로부터의 의뢰로, 유저에게 대신해, 자율적으로 작업을 실시하는 소프트웨어 프로그램」이라고 하는 특징이 있다[4].

네트워크의 발전과 함께 계산 시스템의 구성이나 원리, 그리고 그것을 실현하는 기술은 양상이 바뀌고 있으며 이동 에이전트는 차세대의 네트워크 대응 어플리케이션 개발운용 기술로서 주목을 받고 있다. 예를 들면, 기존의 네트워크에서는 컴퓨터간 통신으로 교환되는 것으로 데이터이지만, 이동 에이전트는 자율성을 가지는 프로그램이며, 그 프로그램 자신이 컴퓨터간을 이동할 수 있게 된다. 그 때 프로그램 상태, 예를 들면 프로그램의 변수 내용등도 함께 전송 되는 것으로부터, 각각의 호스트에서는 이동하기 직전 상태로부터 그대로 처리를 계속할 수가 있다. 즉 이동 에이전트는 네트워크상의 컴퓨터를 순회하면서 처리하는 자율적인 소프트웨어가 된다.

이동 에이전트 기술은 네트워크를 전제로 한 계산 처리나 소프트웨어 개발에 대해 유용하고, 특히 네트워크 트래픽의 삭감이나 통신 절단에의 대응 등 종래 수법에는 없는 수많은 이점을 가진다. 예를 들면 이동에이전트가 통신 상대의 서버측 컴퓨터로 이동하여 에이전트는 서버측 컴퓨터로 실행되어 서버와도 직접 통신할 수 있게 된다. 이것에 의해 컴퓨터간 통신은 에이전트의 이동으로 한정되어 서버와의 통신도 컴퓨터 내부의 통신이 되어, 네트워크상의 통신 회수와 통신 지연을 큰 폭으로 삭감할 수가 있다[4][5].

이동 에이전트는 Remote procedure calling(RPC)의 향상으로 발전하게 되었다. 이러한 이동 에이전트는 특정한 일을 수행하기 위해 이동성을 지원하는 기본 데이터

(mobility meta data)를 기초로 해서 동작하고 독립적으로 목적지 호스트로 언제 그리고 어디로 이동할 지 스스로 결정할 수 있다. 이동 에이전트는 실행코드, 프로세스 상태정보 그리고 다른 데이터를 제공함으로써 분산처리 기술에 향상을 가져왔다.

이동에이전트는 (그림 1)에서 보는 바와 같이 RPC 기법보다 네트워크 트래픽을 감소시킨다. 이동에이전트는 통신할 많은 대화를 한 묶음으로 묶어 그것을 목적지 호스트로 전송한다. 목적지에 도착하면 그 시스템 내에서 국부적으로 상호대화를 수행하고 결과를 되돌려 보낸다[6][7].

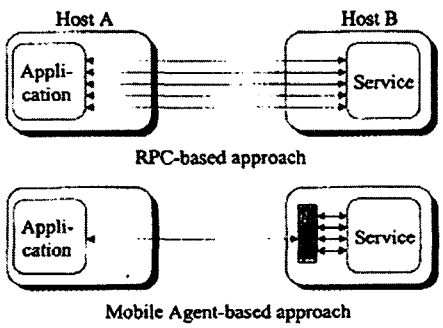

그림 1. 이동에이전트의 네트워크 패킷 흐름
Fig 1. Network packet flow of Mobile agent

III. 침입 탐지

초기의 침입 탐지에 대한 연구는 하나의 호스트에서 출발하였으나, 인터넷의 발전은 이의 영역을 네트워크로 확장시켰다. 그러나 이러한 침입탐지시스템이나 프로토타입들은 개개의 환경에 적합하게 설계되고 적용됨으로써, 대규모 네트워크로의 확장에 어려움을 가지게 되었다.

또한 침입탐지시스템의 중요성이 증가하고 여러 가지 제품이 나오면서 하나의 사건에 대해서도 여러 가지 다른 양상으로 다른 시스템에 적용되었다. 전체적으로 통합된 탐지 전략을 세우기 위해서는 침입탐지기법의 확장 및 서로 다른 방식으로 얻어지는 정보들에 대한 통합이 요구되며, 이에 따른 고수준의 통신 프로토콜에 대한 정의가 필요하다. 이러한 일련의 작업들을 IETF(Internet Engineering Task Force)내에 IDWG(Intrusion Detection Working Group)에서 수행하고 있다[1].

그러나 IDWG에서 추진되는 침입탐지시스템에서의 통신 프로토콜과 데이터 포맷 등을 표준화하기 위한 노력은 그 이전부터 시작되었다.

미 국방성은 DoD 시스템에 대한 어떤 공격이 성공할 지라도 군 정보시스템의 중요 서비스 및 기능에 대한 최소한의 성능을 지속시키기 위하여 1996년 DARPA/ITO(Defense Advanced Research Projects Agency/Information Technology Office)에서는 "정보 생존(Information Survivability)" 프로그램을 시작하였다.

이 프로그램은 고 신뢰 네트워킹(High Confidence Networking), 고 신뢰 컴퓨팅 시스템(High Confidence Computing), wrapper와 구성(Wrappers and Composition), 대규모 시스템의 생존(Survivability of Large Scale Systems) 등 네 분야로 나누어진다. 이 프로그램은 COST 제품 사용을 통하여 비용을 절감하고 이 프로그램에서 개발되어지는 기술들이 상용제품에서 다루어지지 않는 방어 요구를 만족하도록 하며 여기서 개발된 기술들이 상용 제품화로 전환될 수 있도록 여러 가지 기술 이전 전략을 세웠다. 네 가지 프로그램 중 침입탐지와 대응 기술을 연구하는 프로그램인 "대규모 시스템의 생존력" 프로그램의 기술 이전 계획 중 하나의 전략이 "모니터링 및 탐침(probe), 침입탐지, 대응 프로토콜 등에 대한 표준 인터페이스를 정의하고 이를 산업표준으로 채택"이다. 이를 수행하기 위한 과제가 CIDF(Common Intrusion Detection Framework)이며 1996년부터 1998년까지 수행되었다. CIDF 과제의 연구결과를 표준으로 제정하기 위해서 IETF와의 공동으로 작업하여 그 해 12월 IDWG을 조직하였다. 현재 IDWG은 침입탐지시스템에 대한 요구사항 기술과 공통 언어 정의, 그리고 침입탐지시스템에서의 통신 프로토콜과 데이터 포맷 등을 표준화하기 위해 노력하고 있다.

1. 침입탐지의 일반적인 모델 (Generic Model of Intrusion Detection Process)

ISO/IEC에서 정의하는 침입 탐지의 기본 모델은 (그림 1)과 같이, 원시 데이터(raw data source), 사건 탐지(event detection), 분석(analysis), 대응(response), 데이터 저장소(data storage)의 5 가지 요소로 구성된다.

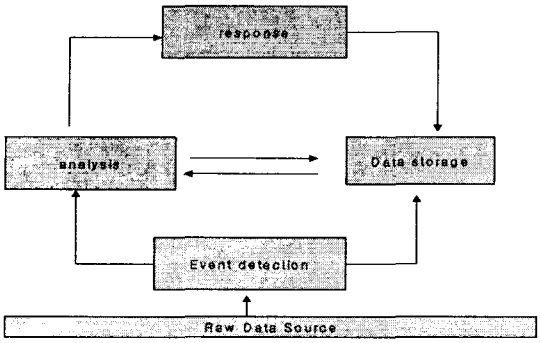

그림 2. ISO/IEC 침입 탐지 기본 모델
fig 2. ISO/IEC Intrusion Detection Basic model

'원시데이터'는 여러 곳의 시스템 자원으로부터 얻어진 감사 자료들과, 시스템 자원의 사용내용(CPU 사용도, 메모리 사용도, 시스템 자원의 고갈도 등), 네트워크 관련 정보 등을 말하며, '사건탐지'는 실제 사건을 탐지하는 방안으로 여기서의 사건이란 특정 데이터, 환경, 행동 등의 발생 상황을 말한다. 이 사건들은 간단한 사건들과 복잡한 사건의 두 부분으로 나눌 수 있다. 사건들을 분석하여 실제 침입이 발생할 확률을 결정하는 것이 '분석' 기능이다. 분석시 이용하는 정보들은 '사건탐지' 결과와 이전의 분석으로부터 얻어진 결과들, 각 사용자들의 행동양식에 대한 정보, 각 개체 및 시스템의 수행 양식 정보, 기타 위험하다고 알려진 사이트 및 개인에 대한 정보 등이다. '대응'은 침입이 발생했다고 판단된 경우 이를 관리자에게 알려주는 방안에 대한 것으로, 결과는 보통 콘솔에 GUI로 나타나며, 기타 이메일, 메신저 서비스 등이 이용될 수 있다. '데이터 저장소'에는 탐지된 사건 결과, 분석에 필요한 모든 데이터들, 알려진 침입에 대한 프로파일들, 세부적인 원시 데이터들(추후 추적 등을 위하여 사용될 수 있다)이 저장된다. 데이터 저장소는 저장된 데이터를 보호할 수 있는 정책 하에 관리되어야 한다.

2. 침입탐지의 특성들 (Characteristics of Intrusion Detection)

침입 탐지 시스템이 가져야 할 일반적인 특성들은 크게 기능적인 것과 비 기능적인 유형 두 가지로 나눌 수 있다. 기능적인 특성들은 다시, 원시데이터의 근원지가 어디인지, 탐지 분석 방법이 무엇인지, 대응 행동이 어떠한 지의 세 부류로 나누어진다.

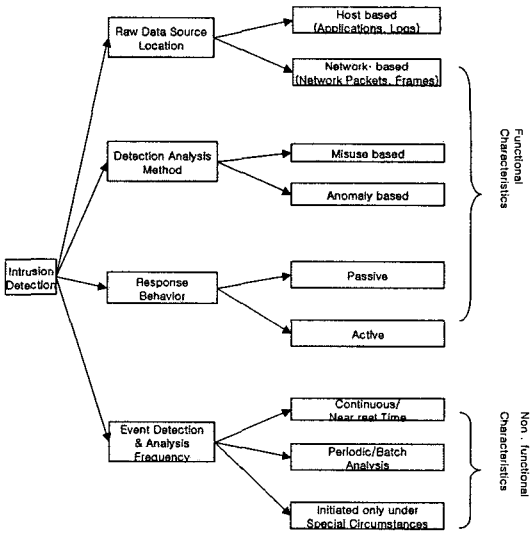

그림 3. 침입탐지유형
Fig 3. Intrusion Detection Types

원시데이터의 근원지가 어디인지는 호스트 기반 방식(사용자 감사 자료와 시스템 로그 등)과 네트워크 기반 방식(네트워크 패킷과 네트워크 프레임 등)으로 나누어진다.

탐지 분석 방법의 일반적인 분류방법은 오용 기반(misuse-based)과 이상 기반(anomaly-based)으로 나눌 수 있다. 오용기반 침입 탐지 방법은 미리 잘못된 행동들을 지정하고 이 행동들이 발생하는지를 검사하는 것으로, 시그너처 분석 방법(signature analysis) 등이 이에 속한다. 이상기반 침입 탐지 방법은 정상적인 경우의 시스템 및 사용자들의 궤적을 기록해 놓고, 이를 벗어나는 행동들을 침입으로 간주하고 탐지하는 것으로, 통계적인 방법(statistical approach) 등이 이에 속한다. 침입이 탐지되었을 때의 대응행동은 수동적인 것과 능동적인 것으로 나눌 수 있다. 수동적인 대응방법은 시스템 자체의 수정은 수행하지 않는 반면에, 능동적인 대응방법은 수정이 필요한 경우, 시스템 자체의 수정도 수행한다. 비 기능적인 특성이란, 사건 탐지/분석을 수행하는 주기에 관계된 것으로, 준 실시간으로 수행하기, 주기적/배치로 수행하기, 특정 조건에서만 수행하기의 세 가지 부류로 나눌 수 있다.

3. 침입탐지 구성론(Architecture Considerations)

작은 회사나 단체의 경우에는 하나의 IDS로 운영을 해도 충분하나, 복잡하고 큰 환경에서는 하나의 IDS로는 필요한 모든 요구를 충족하지 못할 수 있다.

이 경우, 다수의 IDS가 모여서 전체 침입 탐지 기능을 수행해야 할 필요가 있으며, 이 때 각 IDS는 전체 시스템의 일부 요소 기능을 수행한다. 다수 IDS가 상호 동작하는 방법은 계층적(hierarchical) 방법과 중앙 집중적(centralized) 방법으로 나눌 수 있다. 계층적 방법에서는 각 IDS 요소들이 원시 데이터에서 사건을 탐지하고 분석하는 작업을 수행하고 그 결과를 상위에서 모아 최종 침입 탐지 여부를 결정한다. 중앙 집중적 방법에서는 각 IDS 요소들은 원시 데이터를 모으는 작업만을 수행하고 중앙관리 노드에서 이 원시 데이터들을 모아서 사건탐지/분석 등의 작업을 수행하게 된다. 중앙 집중적 방법은 간단하다는 장점은 있기는 하나, 어느 정도 작은 시스템에서 적합하지만 큰 시스템에서는 처리 시간의 증가와 기억공간이 커지게 됨으로서 적합하지 않다. 중앙 집중적 방법이 좀 더 큰 시스템에서도 유용하도록 하기 위하여 원시 데이터가 모아지는 대로 양을 줄이는 방법을 사용할 수도 있다.

Ⅳ. 침입탐지를 위한 이동에이전트 동작

이렇게 이동 에이전트가 서버에이전트들을 방문하여 침입을 탐지하는 방법은 (그림 4)와 같다. 이동 에이전트 동작 방법은 중앙의 에이전트 관리자에서 각각의 서버에이전트로 에이전트를 파견하고 파견된 에이전트들은 중앙의 관리자 노드로 복귀하여 이상의 유무를 전달하는 방법이다.

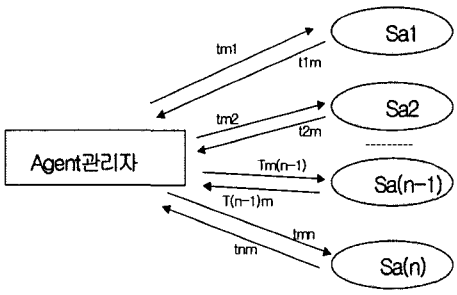

그림 4. 이동 에이전트의 파견방법(1:1)
Fig 4. Dispatch method of Mobile agent(1:1)

(그림 4)의 1:1 에이전트 파견 방법의 경우 관리자 노드와 관리 대상 노드들 사이의 네트워크 지연시간이 같다고 가정하였을 때 응답시간은 다음과 같다.

R(1:1) = 2nt + na + b
a = 에이전트관리자에서의 지역 계산 시간
b = 에이전트들이 소요한 작업 시간
n = 관리해야 할 전체 서버에이전트 수.

에이전트를 각각의 서버에이전트에 이동시킨 후 에이전트 관리자가 각각의 서버로부터 돌아온 작업시간을 계산해야하는 경우 보통 두 번의 값이 중복 체크되어진다.

실질적인 이동 에이전트를 이용한 응용에서는 모든 관리 대상 노드들의 작업시간이 동일하다고 가정하는 것은 적합하지 않으므로 관리 대상 노드들의 작업 시간이 다양할 수 있는 상황을 생각해 볼 수 있다.

Ⅴ. 이동 에이전트 침입 탐지시스템 설계

1. 가정

에이전트가 생성자에 의해 생성된 뒤 에이전트관리자와 에이전트 서버(Sa)간을 이동하는 도중에 에이전트 자신 혹은 생성자 이외의 에이전트 서버는 코드를 변경할 수 없다.

이동에이전트 코드는 모든 호스트에서 동일한 형식으로 동작되어야 하고, 직접 해석되거나 다시 컴파일 되지 않고 수행될 수 있는 이식성이 있는 중간 언어이어야 한다. 인터프리터는 에이전트의 호스트자원에 대한 접근을 제어하면서 에이전트를 실행시킨다. 에이전트관리자로부터 받은 에이전트의 코드와 상태에 의해 인터프리터는 에이전트를 실행시킨다. 보안 구조 시스템은 에이전트를 구동시키는 인터프리터가 신뢰성 있게 구현되어 에이전트 관리 호스트에서 구동되고 있다고 가정한다.

에이전트의 실행 코드는 에이전트 생성 시 만들어지고 소멸할 때까지 활동하고 생성자의 감독을 통해서 코드가 변경되었는지에 대한 확인을 할 수 있다. 에이전트 실행상태는 에이전트가 실행되면서 계속 변화하기 때문에 각 에이전트 관리자는 에이전트 실행 상태에 대한 조정을 함으로써 에이전트 실행 상태 변경에 대한 부인 봉쇄를 이룰 수 있다.

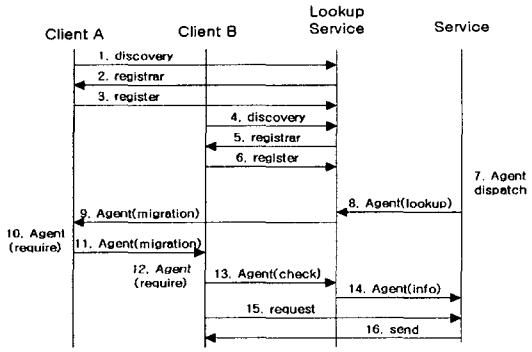

그림 5. 에이전트 상태도
Fig 5. Agent state diagram

2. 탐지 모델 구현

제안 모델인 링 순환 에이전트 파견의 경우 서버에이전트의 지역 작업시간에 따라서 새로운 에이전트의 파견을 결정하므로 서버 에이전트마다 같은 에이전트가 동일한 작업을 하되 상황에 따라서 지역 작업시간의 편차가 큰 경우에 적용할 수 있다. 즉, 초기에는 링 방식의 순환을 하지만 작업량이 많은 서버에이전트에서는 응답시간의 단축을 위하여 새로운 에이전트를 파견하는 것이다.

이렇게 작업량의 편차가 큰 경우는 동일한 작업을 할 때 작업의 자료 양이 많을 경우 발생 가능하다. 예를 들어, 서버에이전트들이 이동하고 있을 경우 시간에 따라서 에러 로그가 발생할 수 있는데 관리 대상 노드의 상황에 따라서 에러 로그는 발생하지 않을 수도 있거나 아주 많은 양이 발생할 수도 있다.

이처럼 링 순환을 하게 되면 한번의 에이전트 파견으로 전체 서버를 순회하여 이전의 에이전트와 방문한 에이전트를 서로 비교하게 됨으로써 다른 변경으로부터 보호할 수 있게 된다. 따라서 성능 평가를 위한 실험에서는 이러한 이동 호스트 에러 로그를 관리 대상 노드마다 정해진 양만큼이 발생한다고 가정한다. 에이전트의 작업은 관리할 모든 대상 노드를 방문하여 각 노드에서 에러 로그를 분석한 후 에러의 종류와 발생 시각, 횟수 등을 에이전트가 에이전트 관리자로 가져오게 된다. 성능 평가는 제안모델과 1:1모델과의 에이전트 관리자에서의 전체적인 응답시간과 네트워크 부하를 측정한다.

이때 1:1 에이전트의 파견 방법은 매 파견마다 결과를 저장한 후 다음 파견 정보와 비교를 하게 됨으로써 중복계산과 상대정보가 반드시 기억되어야 하는 문제점을 가지고 있지만, 링 순환 파견 방법은 한번의 에이전트 이동에 의하여 최종 서버 에이전트에서 수집된 정보만을 취급하게 되어 계산시간이 단순하고 에이전트 정보를 기억하지 않아도 에이전트의 상태를 파악할 수 있다.

특정 서버에이전트만 과부하 되지 않고 부하가 균등하게 작동하도록 (그림 6)의 링 순환 에이전트 파견 방법의 경우 에이전트 관리자와 서버에이전트들 사이의 네트워크 지연시간이 같다고 가정하였을 때 응답시간은 다음과 같다.

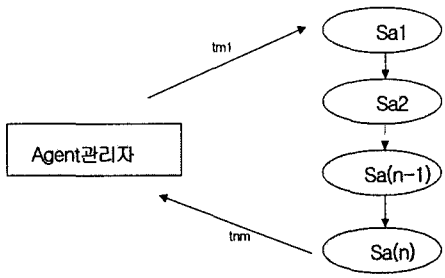

그림 6. 제안된 링 순환 에이전트 파견
Fig 6. Dispatch proposed ring cycle agent

$R(Ring) = (n+1)t + a + b$

a = 에이전트 관리자에서의 지역 계산 시간

b = 에이전트들이 소요한 작업 시간

n = 관리해야 할 전체 서버에이전트 수

3. 성능평가

성능 평가를 위한 실험환경은 10Mbps 이더넷을 기반으로 LAN을 구성하고 총 7대의 컴퓨터(에이전트 관리자 1대 +서버에이전트 6대)로 구성하였으며 이동 에이전트의 개발환경으로는 IBM사의 Aglets 소프트웨어 개발 키트(Aglets Software Development Kit : ASDK) 버전1.1Beta3를 사용하였고 자바 컴파일러와 자바 가상 기계는 JDK1.1.8 버전을 사용하였다.

실험에 사용한 매개변수는 다음과 같으며 실험 결과 값은 실험을 총 7번 반복하여 평균값을 이용한다.

서버 에이전트 수	6대
평균 메시지 전송시간	30(ms)
가상침입패킷	5(Kbytes)
에이전트 크기	7(Kbytes)
에이전트 평균 전송시간	165(ms)
네크워크 처리율	400Kbytes/sec
노드간 지연시간	15(ms)

응답시간의 측정 방법은 모두 생성되는 에이전트의 수에 상관없이 가장 처음 에이전트를 에이전트 관리자에서 만든 시간이 작업의 시작 시간이고 모든 관리 대상 노드의 작업을 마치고 온 에이전트가 관리자 노드에 결과를 보여주고 작업을 마치는 시간을 작업의 종료 시간으로 하여 측정, 비교한다. 네트워크 부하는 에이전트 관리자와 서버에이전트 간의 에이전트 및 메시지의 전송 그리고 에이전트의 작업 결과를 모두 합한 것으로 측정한다. 링 순환의 방법은 서버 에이전트에 침입 정보가 발생하면 관리에이전트에서 파견된 이전 서버에이전트 정보와 비교하여 침입을 탐지하게 된다.

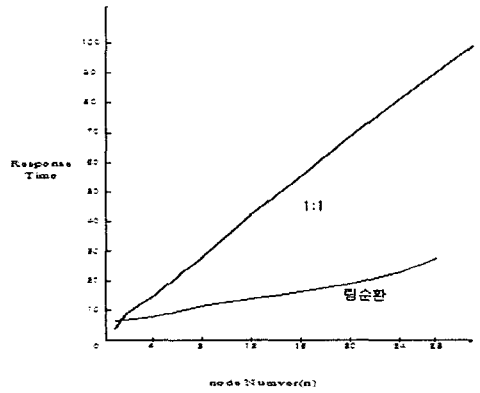

그림 7. 성능비교
Fig 7. Compare to performance

(그림 7)의 실험 결과 그래프에는 두 종류의 값이 존재하는데 제안 모델과의 비교를 위한 모델은 관리 대상 노드를 한 개의 에이전트가 방문한 후 다시 에이전트 관리자로 돌아오고 다시 새로운 서버 에이전트로 방문하는 방법인 1:1 클라이언트 서버방법과 모든 서버 에이전트의 수에 따라서 에이전트를 에이전트 관리자에서 한꺼번에 생성하고 모든 서버 에이전트에 멀티 캐스트 방법으로 파견한 후 파견된 모든 에이전트가 에이전트 관리자에서 작업을 마치고 돌아와서 결과를 출력하는 방법과 비교를 한다. 이 실험 결과로 볼 때 데이터 처리는 이동 에이전트들이 서버 에이전트를 링 방식으로 순환 시 각각의 서버 에이전트에서 작업한 결과가 누적이 되므로 에이전트 관리자로 복귀할 때까지의 결과 값의 크기가 변하게 되고 이 크기에 따라서 응답시간의 변화가 생기게 된다. 에이전트가 다음 번 방문할 노드로 이동하기 전에 매번 작업한 결과 값을 에이전트 관리자로 전송하는 방법과 결과 값의 임계값을 두어 임계값을 넘으면 전송하는 방법 중 링 순환의 방법이 노드를 연속적으로 반복 수행할 때마다 응답시간이 최소화됨을 알 수 있다.

V. 결론

침입 탐지시스템의 성능향상을 위해 이동에이전트를 이용한 침입 탐지 모델은 계층형 통합 탐지와 중앙 집중형 탐지를 유기적으로 결합한 형태이다. 각각의 서버에이전트에 침입된 데이터를 탐지하여 관리하여 탐지능력을 키우고 신뢰성을 향상시키기 위해 고려하였다.

또한 서로 다른 서버에이전트의 에이전트들을 처리하기 위하여 통지 유형을 명명트리에 등록하여 같은 유형의 통지를 에이전트 관리자가 하나의 이벤트채널을 통해 전달받을 수 있도록 하였다.

본 논문에서는 제안하는 모델이 지역 작업시간에 따라서 새로운 에이전트의 파견을 결정하므로 서버 에이전트마다 같은 에이전트가 동일한 작업을 하되 상황에 따라서 지역 작업시간의 편차가 큰 경우에 적용할 수 있도록 제안하였다.

향후 연구 방향으로는 다양한 컴퓨팅 환경에서 거짓 정보를 줄이면서 에이전트의 신뢰성을 확보하기위한 에이전트 서명절차와 지역 네트워크내의 트래픽을 다양하게 분석하여 작업 부하가 균형적인 상태에서 에이전트만을 보호하고 이동하는 동안 보안의 취약점을 찾아 좀 더 정확한 이동 정보를 암호화하여 보안 사고에 대비할 필요가 있을 것 같다.

참고문헌

[1] 컴퓨터네트워크 보안기술 연구, 한국전자통신연구소, pp.168~177, 1991.

[2] 유정준, 백주성, 박종열, 이동익, Java 기반 이동 에이전트 시스템 : X-MAS, 한국통신학회 춘계학술대회, 1998

[3] 조영상 외, "XML 기반의 지능 에이전트 시스템의 설계 및 구현", 정보과학회, 1999.

[4] 정의현, "Technology in Mobile Agent", KRnet, 99 발표자료집, pp.160~109, 1999.

[5] Bennet S.yee, "A Sanctuary for mobile Agents", DARPA Workshop on Foundations for Secure Mobile Code Workshop, pp.26~28, 1997.

[6] Dejan S. Milojicic, et al., "Mobile Objects and Agents", OSF Research Institute, 1996.

[7] G. Coulouris, J. Dollimore, T. Kindberg, Distributed Systems, Concept and Design, Addison-Wesley, 1994

[8] G. H. Forman, J. Zahorjan, "The Challenges of Mobile Computing," IEEE Computer, V 27, N 4, pp. 38-47. April 1994

[9] J. Gosling and H. McGilton, "The Jave Language Environment : A White Paper", Technical Report, Sun Microsystems, 1995

[10] M.J. Wooldridge, et al., "Software Engineering with Agents : Pitfalls and Pratfalls", IEEE Internet Computing, 1999

[11] Giovanni Vigna, "Protecting Mobile Agents through Tracing", ECOOP Workshop, 1997.

저 자 소 개

황 인 선
광주보건대학 컴퓨터보안과 교수

박 경 우
목포대학교 컴퓨터공학과 교수