論文
5-2-8

계약망 프로토콜의 에이전트 선택을 위한 퍼지 컨트롤러 설계 (Fuzzy Controller Design for Selecting the Agent of Contract Net Protocol)

서 희 석(Hee-Suk Seo)[1], 김 희 완(Hee-Wan Kim)[2]

요 약

네트워크 보안의 중요성과 필요성이 증대됨에 따라 많은 조직들이 다양한 보안 시스템을 네트워크에 적용하고 있다. 침입 차단 시스템, 침입 탐지 시스템, 취약점 스캐너와 같은 보안 시스템들이 취약성 정보를 공유하게 되면 일관된 통합 보안 환경을 구축할 수 있다. 본 논문에서는 침입 탐지 에이전트와 방화벽 에이전트가 계약망 프로토콜(*Contract Net Protocol*)에 의해서 서로 연동할 수 있는 구조를 디자인하고 구축하였다. 계약망 프로토콜은 분산 시스템과 같은 이기종의 컴퓨터 시스템의 효과적인 연동을 위한 방법으로서 여러 에이전트들이 모여 서로 협력하며 하나의 문제를 해결하게 된다. 계약망 프로토콜의 선택 알고리즘과 퍼지 제어기를 사용하였을 경우 성능을 비교함으로써 에이전트 선택에 있어서 보다 효과적인 방법을 제시할 것이다.

ABSTRACT

As the importance and the need for network security is increased, many organization uses the various security systems. They enable to construct the consistent integrated security environment by sharing the vulnerable information among firewall, intrusion detection system, and vulnerable scanner.We construct the integrated security simulation environment that can be used by some security system model. In this paper, we have designed and constructed the general simulation environment of network security model composed of multiple IDSs agent and a firewall agent which coordinate by CNP (Contract Net Protocol). The CNP, the methodology for efficient integration of computer systems on heterogeneous environment such as distributed systems, is essentially a collection of agents, which cooperate to resolve a problem. We compare the selection algorithm in the CPN with the Fuzzy Controller for the effective method to select the agents.

논문접수 : 2004. 2. 11.
심사완료 : 2004. 2. 18.

1) 정회원 : 성균관대학교 정보통신공학부 박사
2) 정회원 : 삼육대학교 컴퓨터과학과 조교수

1. 서 론

컴퓨터 기술의 발달과 인터넷의 발전은 업무 효율을 향상시키고 생활의 질을 높여주는 등의 긍정적인 효과를 가져 온 반면 네트워크의 확장으로 외부에서의 시스템 불법 침입, 중요 정보의 유출 및 변경, 훼손, 불법적인 사용, 컴퓨터 바이러스 등의 역기능들이 증가되어 그 피해가 심각한 수준에 이르렀다[1]. 따라서 이러한 공격에 대한 대응책으로 여러 보안 시스템이 사용되고 있으며 본 연구에서는 침입 탐지 시스템(IDS)을 보안 요소로 도입하였다.

침입 탐지 시스템은 네트워크 시스템이나 컴퓨터의 불법적인 사용이나 잘못된 사용(misuse)을 감시하고 탐지하는 것으로 침입 차단 시스템처럼 단순히 네트워크를 통한 외부 침입을 차단하는 단계를 넘어 외부 침입에 의해 침입 차단 시스템이 해킹되는 순간 혹은 해킹 된 후 침입 사실을 탐지해 이에 대응하도록 하는 네트워크 보안 솔루션이다[2,3]. 침입 탐지 시스템은 데이터 소스의 종류에 따라 호스트 기반의 침입 탐지 시스템(Host-based IDS)과 네트워크 기반의 침입 탐지 시스템(Network-based IDS)으로 구분하는데, 시스템의 로그 정보과 특정 행위에 대한 감사 자료 등에 대한 분석을 통하여 침입을 탐지하는 호스트 기반 침입 탐지시스템은 설치되는 호스트의 성능에 따라 영향을 미치며 많은 네트워크를 관리 시 자원의 낭비와 시스템의 부하를 증가시키는 문제점이 있다. 이와 같은 문제점을 해결하기 위하여 네트워크상의 패킷을 분석하거나 트래픽량 등을 분석하는 네트워크를 기반으로 하는 다중 침입 탐지 시스템을 도입하였다. 분산 에이전트 기반의 침입 탐지 시스템은 감사 정보의 수입을 다수의 분산 에이전트들이 나누어 수행하므로 시스템의 부하를 감소시키고 침입 탐지의 속도를 향상시키며 발생된 침입에 적당한 에이전트를 선택하여 탐지의 성능을 향상시킬 수 있다. 이러한 다중 에이전트 환경에서 에이전트사이의 연동에 있어 효율적인 수행 능력을 위해서는 분산된 에이전트들에게 효과적인 작업의 할당이 이루어져야 하며[4], 본 연구에서는 이러한 에이전트 사이의 연동을 위하여 계약망 프로토콜(contract net protocol)을 적용하였다.

계약망 프로토콜은 분산된 에이전트들 중에 입찰(bidding)을 통해 최상의 에이전트를 선택하고 선택된 에이전트는 서비스를 제공하게 된다[5,6,7]. 이러한 작업의 분산을 통해 각 에이전트들은 효과적으로 주어진 작업을 처리하며 자신이 처리하지 못하는 작업은 다른 에이전트에게 의뢰하여 처리할 수 있다. 이를 침입 탐지에 적용하면 기존의 침입 탐지에 비해 신속하고 정확한 탐지가 가능하다. 본 연구에서는 각 에이전트들이 입찰을 했을 때, 에이전트를 선택하는 Command Console에 단순히 선택 알고리즘을 이용하여 에이전트를 선택했을 때와 비교하여 퍼지 규칙 기반 시스템(fuzzy rule-based system)을 적용하여 에이전트를 선택했을 때의 보안 시스템을 모델링 하여 시뮬레이션 환경을 구축하고 그 결과를 비교하였다.

2. 배경 이론

2.1. 침입 탐지 시스템(IDS)

침입 탐지 시스템은 외부의 침입에 대해 능동적으로 대처 하는 시스템으로 방화벽의 앞 또는 뒤에서 침입 사실을 탐지해 침입자의 공격에 대응하기 위한 솔루션이다. 침입 탐지란 허가 되지 않은 외부 사용자의 침입 시도와 내부 사용자의 권한 남용을 탐지하는 시스템으로 침입 탐지 접근 방법에는 오용 탐지(misuse detection)와 비정상 행위 탐지(anomaly detection) 방법이 있다. 오용 탐지 방법은 일반적으로 침입이라고 알려져 있는 행위 또는 비정상적인 행위를 패턴으로 저장해 놓고, 이에 일치 또는 유사한 사용자의 행위가 나타났을 때 이를 탐지하는 것이다. 오용 탐지에는 주로 패턴 비교(pattern

matching)기술이나 전문가 시스템(expert system)이 사용되며, 전문가 시스템은 사용자의 작업 절차와 저장되어 있는 침입 특성을 가지고 추론을 하게 된다. 비정상 행위 탐지 방법은 시스템이나 네트워크에서 일어나는 행위들 중 일반적이지 않고, 발생 빈도가 매우 낮은 행위의 발생을 탐지하는 방법이다. 이러한 탐지 방법은 침입자의 행위가 일반 사용자의 행위와 주목할 만큼 다르다 가정을 기반으로 한다. 이를 위해서 정상적인 행위의 특성 정보가 필요하며, 침입 탐지 시스템은 사용자의 행동의 특성을 분석하는데 이 정보를 사용하게 된다. 일반적으로 비정상 행위 탐지 기법은 통계적 기법이나 뉴럴 네트워크(neural network) 등을 사용한다. 최근 개발되고 있는 침입 탐지 시스템은 이 두 가지 탐지 기법을 모두 사용하여 좀 더 효과적인 침입 탐지를 하고자 한다.

침입 탐지 시스템은 침입의 탐지에 사용되는 데이터의 원천에 따라 시스템 데이터 기반 침입 탐지 시스템과 네트워크 데이터 기반의 침입 탐지 시스템으로 나눈다. 시스템 데이터 기반 침입 탐지 시스템은 단일 호스트 내에서 발생하는 침입을 탐지하는 시스템으로 주로 시스템 내부의 로그 파일이나 감사 파일을 사용한다. 네트워크 데이터 기반의 침입 탐지 시스템은 네트워크 트래픽에서 침입의 패턴을 찾아내는 방법으로 침입을 탐지한다. 시스템 데이터 기반 침입 탐지 시스템은 단일 시스템 대상 공격은 탐지할 수 있지만 다중 시스템 대상 공격은 탐지가 어렵다. 반면 네트워크 기반 침입 탐지 시스템은 다중 시스템 대상 공격의 탐지에는 적당하지만 시스템 내부의 오용이나 남용에 대한 탐지가 어렵다. 최근의 침입 탐지 시스템은 이 두 가지 침입 탐지 시스템의 조합한 형태로 구성되어 다양한 침입을 탐지 할 수 있는 능력을 소유하도록 구성된다.

2.2 계약망 프로토콜(Contract Net Protocol)

계약망 프로토콜은 분산된 문제(distributed problem)를 해결하는데 있어 에이전트들 사이의 통신을 하기 위한 도구로서 제안되었다[6]. 계약망 프로토콜의 사용은 분산 감지 시스템과 분산 전달 시스템을 위해서 시도 되었다. 계약망 프로토콜은 에이전트들이 계약(contract)에 의하여 분산된 문제를 해결하기 위하여 협상하고 통신하는 메커니즘을 제공한다[5]. 에이전트들은 수행될 필요가 있는 작업을 알리고 다른 에이전트들에게 공지된 작업들을 수행하기 위해 bid를 만들어서 보내면 커맨드 콘솔(Command Console)은 각 에이전트들이 제출한 bid를 평가하여 최상의 에이전트를 선택하여 계약을 체결하게 된다[6]. 계약망 프로토콜의 적용은 다중 침입 탐지 시스템에 있어서 서로 보완하고 협력하여 탐지의 성능을 향상시키고 정확도를 높일 수 있다.

3. 기존 연구

3.1 계약망 프로토콜의 연동

계약망 프로토콜에서 모든 IDS모델의 에이전트를 제어하는 Command Console 모델은 크게 IDS에서 보낸 메시지를 주고 받는 Messenger 모듈과 각각의 IDS에서 보낸 bid를 선택하는 Selector 모듈, 그리고 내부 네트워크의 상태에 따라 IDS를 통제하는 Commander 모듈로 나눈다. Command Console에서 오는 패킷 데이터는 IDE (Intrusion Detection Engine) 모델에서 탐지하게 되고 컨트롤과 관련된 메시지는 IDE 모델을 통과하여 Agent 모델에서 처리하게 된다. 또한 IDE 모델에서 탐지하거나 처리된 메시지를 Agent 모델로 보내 처리하게 된다.

먼저 내부 네트워크로 패킷이 유입되면 Command Console은 모든 침입 탐지 에이전트에게 bid 메시지를 보내고 이 메시지를 받은 에이전트들은 bid 메시지를 보낸다. Command Console은 bid 메시지를 가지고 선택 알고리즘에 의해 침입을 탐지할 에이전트를 선택하게 되

고 선택된 에이전트에게 award 메시지를 보낸다. award 메시지를 받은 에이전트는 기다렸다가 Command Console에서 패킷 정보를 담은 데이터를 복사하여 선택된 에이전트에게 보내면 선택된 에이전트는 이 데이터를 가지고 침입을 탐지하게 된다.

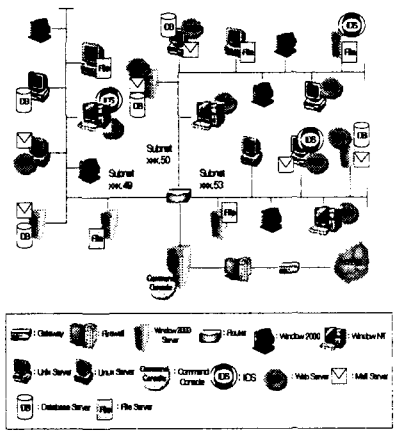
3.2 침입 탐지 에이전트 선택 알고리즘

위의 과정에서 어떤 에이전트를 선택할 것인가가 본 연구에서 중요한 부분인데 기존의 연구에서는 각 에이전트가 bid 데이터의 loading 필드의 값이 임계값을 넘지 않은 경우에 bid 메시지를 중앙 콘솔에 보내게 되는데 우선적으로 expertise 필드의 값을 기준으로 정렬하여 가장 큰 값을 갖는 에이전트를 선택한다. 만약 같은 값을 갖는 에이전트가 존재하면 그 에이전트 중에 experience 필드의 값을 기준으로 다시 정렬하여 역시 가장 큰 값을 가진 에이전트를 선택한다. 만약 experience 값마저 같은 에이전트가 존재한다면 마지막으로 loading 필드의 값을 기준으로 정렬하여 가장 작은 값을 가진 에이전트를 선택하게 된다. 다음은 에이전트 선택 알고리즘이다.

```
Let  bid_i  be bids
Set bid_list = empty set
Let bid_list = (  bid_1, bid2,... ,bid_n ) be a list
     of bids
for i = 1 to n
    if loading of bid_i  >=  threshold value
        Delete bid_i from bid_list
Sort bid_list by expertise in descending order
if the number of bid including the greatest
     value of expertise >= 2 then
{
    Delete bids from bid_list except bids
     including the greatest value of expertise
    Sort bid_list including bids of the same
     expertise by experience in descending
     order
    if the number of bid including the
     greatest value of experience >= 2 then
    {
        Delete bids from bid_list except bids
         including the greatest value of
         experience
        Sort bid_list including bids of the
         same experience by loading in
         ascending order
    }
}
Select Agent from bid_list(the first element)
```

4. 대상 네트워크의 설계

시뮬레이션 환경의 구축을 위해서 네트워크 및 보안 요소에 대한 모델링가 시뮬레이션을 수행할 네트워크의 설계가 선행되어야 한다. 실 시스템 수준의 시뮬레이션 환경 구축에 있어서 대상 네트워크의 설계는 시뮬레이션의 결과가 실 시스템에 반영될 수 있는지를 판단하는 기준이 된다.



<그림 1> 대상 네트워크 구조

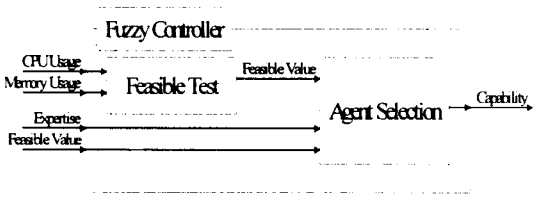
<그림 1>은 3개의 서브넷을 갖는 대상 네트워크의 구성도이다. 내부 네트워크에는 웹 서버, 메일 서버, 데이터베이스 서버 및 파일 서버를 설치한 호스트들이 있고 각 서브넷에 하나의 IDS가 장착되어 있다. 그리고 네트워크 구성요소로 라우터, 게이트웨이, 방화벽 그리고

Command Console이 구성되어 있다.

5. 시뮬레이션 및 실험 결과

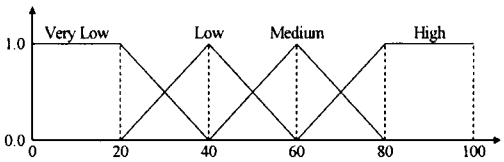
5.1 퍼지 컨트롤러의 설계 및 구성

에이전트 선택을 위한 퍼지 컨트롤러는 적합성 테스트(Feasible Test) 시스템과 에이전트 선택(Agent Selection) 시스템의 두 퍼지 규칙 기반 시스템으로 구성된다. 적합성 테스트 시스템은 입찰한 에이전트의 CPU 사용율과 메모리 사용률을 입력으로 퍼지 추론을 수행하여 적합성 값(Feasible Value)을 출력한다. 에이전트 선택 시스템은 적합성 테스트 시스템의 출력인 적합성 값과 입찰한 에이전트의 전문성과 침입 상태를 입력으로, 퍼지 추론을 수행하여, 능력(Capability)을 출력한다.

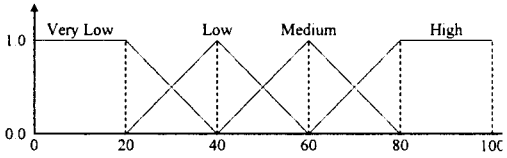


<그림 2> 퍼지 컨트롤러 모델

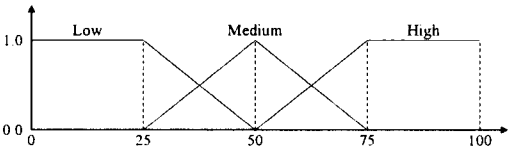
적합성 테스트 시스템의 입·출력에 대한 멤버십 함수(Membership Function)는 <그림 3, 4, 5>와 같다.



<그림 3> CPU의 멤버십 함수

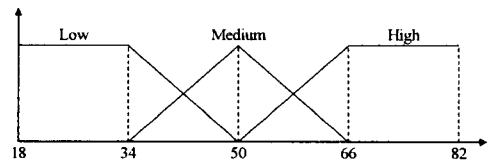


<그림 4> Memory의 멤버십 함수

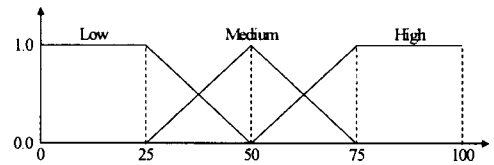


<그림 5> FV (Feasible Value)의 멤버십 함수

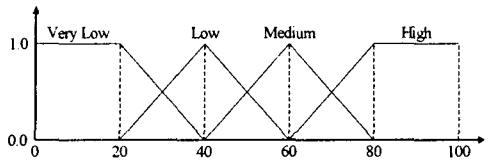
에이전트 선택 시스템의 입·출력에 대한 멤버십 함수(Membership Function)는 <그림 6, 7, 8, 9>와 같다.



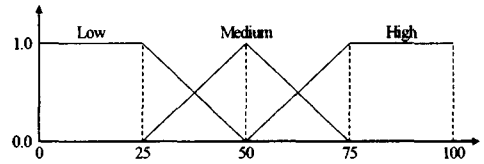
<그림 6> FV (Feasible Value)의 멤버십 함수



<그림 7> Expertise의 멤버십 함수



<그림 8> IS(Intrusion State)의 멤버십 함수



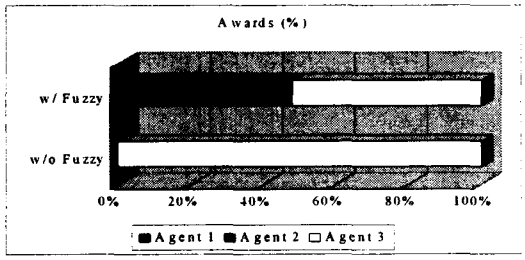
<그림 9> Capability의 멤버십 함수

적합성 테스트 시스템은 총 16개의 퍼지 IF THEN 규칙으로 퍼지 추론을 수행하며, 에이전트 선택 시스템은 총 36개의 퍼지 IF THEN 규칙을 사용한다.(Appendix 참조). 실험에서는 대상 네트워크에서와 같이 IDS 에이전트의 수는 3개로 한정하였으며 동일 시드에서 10,000개의 패킷을 발생시켜 퍼지 컨트롤러를 적용한 경우와 그렇지 않은 경우의 결과 비교하였다. 각 에이전트의 CPU 사용률, 메모리 사용률, 침입 상태는 IID U(0, 100)를 사용하고, 에이전트의 전문성(Expertise)은 보안 관리자가 사전 설정하는 값으로 시뮬레이션 중에는 불변하며 Low (25), Medium (50), High (75) 의 3 단계로 구분하였다. 본 연구에서는 에이전트의 전문성을 달리하며, 모든 조합에 대하여 시뮬레이션 수행하였다.

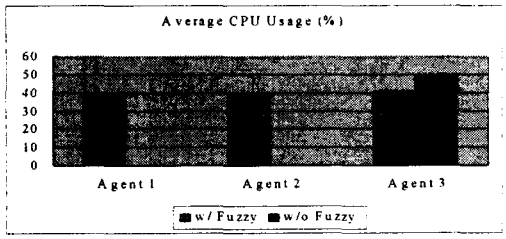
5.2 실험 결과

본 연구진은 계약망 프로토콜에서 에이전트 선택을 하기 위한 퍼지 컨트롤러를 구성하고 시뮬레이션 하여 퍼지를 적용했을 때와 적용하지 않았을 때를 비교 하였다. (Appendix 참조)

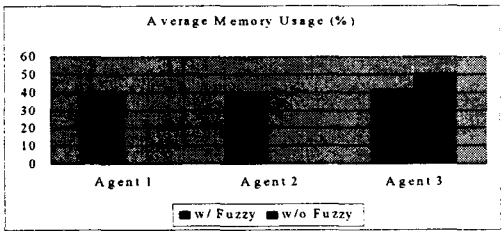
<그림 10, 11, 12>은 IDS 에이전트의 Expertise의 값이 각각 Medium, Medium, High 일 때 시뮬레이션을 한 결과이다. <그림 10>에서와 같이 퍼지를 적용하지 않았을 때는 한 에이전트에게 모두 작업이 할당되는 것과 달리 분산 퍼지를 적용하였을 때에는 분산된 에이전트들이 각각 작업을 할당 받은 것을 볼 수 있다. 또한 <그림 11, 12>와 같이 평균 CPU 사용율과 평균 메모리 사용률을 살펴보면, 퍼지를 적용하지 않았을 때 한 에이전트에서만 CPU와 메모리에 부하가 큰 반면 퍼지를 적용한 후에는 각각의 에이전트로 분산된 것을 볼 수 있다.



<그림 10> 낙찰된 에이전트의 비율



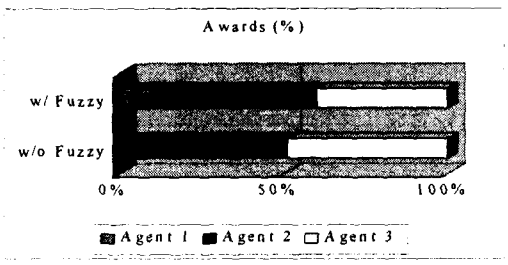
<그림 11> 평균 CPU 사용률



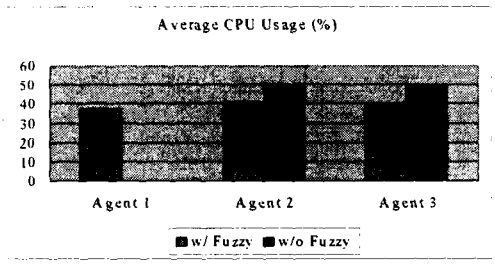
<그림 12> 평균 메모리 사용률

<그림 13, 14, 15>은 IDS 에이전트의 Expertise의 값이 각각 Medium, High, High 일 때 시뮬레이션을 한 결과이다. 한 개의 에이전

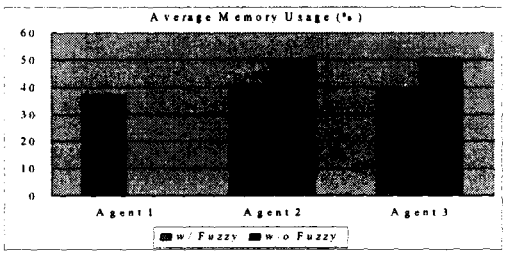
트만 Expertise 값이 High일 때와 달리 두개의 에이전트가 high값을 갖는다고 하더라도 퍼지를 적용하지 않았을 때는 두개의 에이전트가 똑같이 일을 나누어 갖는 반면에 퍼지를 적용한 후에는 세 개의 에이전트가 일을 분산해서 하는 것을 알 수 있다. CPU나 메모리의 평균 사용률 역시 퍼지를 적용 하였을 때 골고루 분산되는 것을 볼 수 있다.



<그림 13> 낙찰된 에이전트의 비율

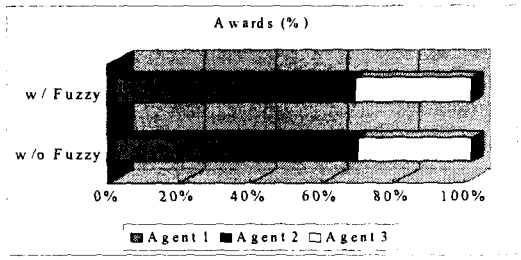


<그림 14> 평균 CPU 사용률

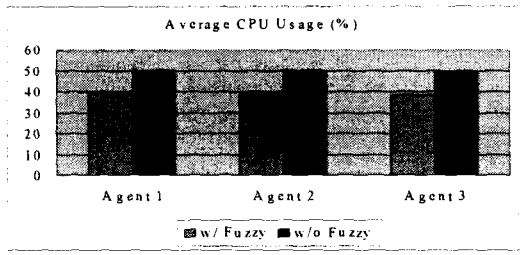


<그림 15> 평균 메모리 사용률

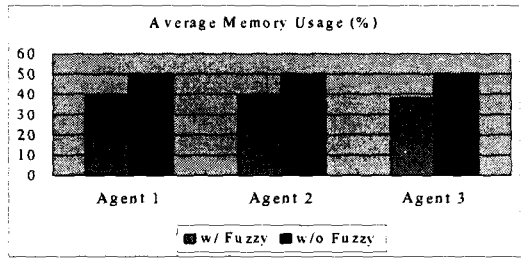
<그림 16, 17, 18>는 IDS 에이전트의 Expertise의 값이 각각 High, High, High 일 때 시뮬레이션을 한 결과이다. award를 받는 과정은 세 개의 에이전트 값이 모두 High이기 때문에 비슷하나, 평균 CPU 사용률이나 평균 Memory 사용률을 보면 그 값이 단순히 loading 값이나 experience 값만을 가지고 비교하여 선택하였을 때보다 현저히 낮은 것을 확인할 수 있었다.



<그림 16> 낙찰된 에이전트의 비율



<그림 17> 평균 CPU 사용률



<그림 18> 평균 메모리 사용률

6. 결 론

실험 결과에서 알 수 있듯이 커맨드 콘솔에서 에이전트를 선택할 때 단순히 선택 알고리즘을 통하여 에이전트를 선택하는 것보다 퍼지 룰 기반 시스템을 적용하였을 때 에이전트들 사이에 작업의 분산이 잘 이루어지며 기존 컨트롤러와는 달리 에이전트의 전문성, 침입 상태, CPU 사용률, 메모리 사용률을 전체적으로 고려하여 효율적인 에이전트를 선택하였다. 만약 전문성이 다른 에이전트로 구성되어있는 경우 부하가 전문성 높은 에이전트로 편향되는 현상을 방지할 수 있으며, 전문성이 동일한 에이전트로 구성되어있는 경우에도 기존보다 더 나은 로드 밸런싱 기능을 수행할 수 있다.

최근 몇몇의 사례를 보면 알 수 잇듯이 네트워크의 활용이 증대되는 만큼 정보 유출이나 침해 사고의 발생 또한 증가 할 것이다. 본 시뮬레이션을 통하여 여러 개의 침입 탐지 시스템이 계약망 프로토콜에 의해 연동되어 있을 때에 퍼지 룰을 적용하여 침입 탐지 에이전트들을 선택하게 되면 효과적으로 침입을 탐지할 수 있으며 시스템이나 네트워크 시스템들의 부하도 줄일 수 있는 것을 볼 수 있다.

참 고 문 헌

[1] 한국정보학회, "차세대 네트워크 보안 기술" 한국정보보호진흥원, 2002.

[2] R. Base. "Intrusion Detection", Macmillan Technical Publishing, 2000.

[3] E. Amoroso, "Intrusion Detection - An Introduction to Internet Surveillance, Correlation, Traps, Trace Back, and Response", Intrusion.Net Books, 1999.

[4] K. M. Sim, S. K. Shiu, and B. L. Martin, "Simulation of a Multi-agent Protocol for Task Allocation in Cooperative Design," IEEE SMC '99 Conference Proceedings. International Conference on, vol.3, pp. 95-100, 1999.

[5] Jihoon Yang, Raghu Havaldar, Vasant Honavar, Les Miller and Johny Wong, "Coordination of Distrbuted Knowledge Networks Using Contract Net Protocol," Information Technology Conference, IEEE, pp. 71-74, 1998.

[6] R. Smith, "The Contract Net Protocol: High-level Communication and Control in a distributed problem solver," IEEE Transactions on Computers, vol. C-29, no. 12, pp. 1104-1113, December. 1980.

[7] H. van Dyke Parunak. Manufacturing Experince with the Contract Net. In Research Notes in Artificial Intelligence: Distributed Artificial Intelligence, Vol. 1, pp. 285 - 310, Morgan Kaufmann Publishers, 1987.

[8] S. Northcutt, "Network Intrusion Detection : An Analysts Handbook", New Riders Publishing, 1999.

[9] H.S. Seo and T.H. Cho, "An application of blackboard architecture for the coordination among the security systems", Simulation Modelling Practice and Theory, Elsevier Science B.V., vol. 11, issues 3-4, pp. 269-284, Jul. 2003.

[10] B. P. Zeigler, H. Praehofer, T. G. Kim, Theory of Modeling and Simulation, 2nd Ed., Academic Press, 2000.

[11] T.H. Cho and Bernard P. Zeigler, "Simulation of Intelligent Hierarchical Flexible Manufacturing: Batch Job Routing in Operation Overlapping," IEEE trans. Syst. Man, Cyber. A, Vol. 27, pp. 116-126, Jan. 1997.

< Appendix >

Fuzzy Rule

1. Feasible Test

Input : CPU = Very_Low, Low, Medium, High
Memory = Very_Low, Low, Medium, High
Output : FV (Feasible Value) = Low, Medium, High

IF CPU=Very_Low ∧ Memory=Very_Low THEN FV=High
IF CPU=Low ∧ Memory=Very_Low THEN FV=High
IF CPU=Medium ∧ Memory=Very_Low THEN FV=Medium
IF CPU=High ∧ Memory=Very_Low THEN FV=Medium

IF CPU=Very_Low ∧ Memory=Low THEN FV=High
IF CPU=Low ∧ Memory=Low THEN FV=High
IF CPU=Medium ∧ Memory=Low THEN FV=Medium
IF CPU=High ∧ Memory=Low THEN FV=Low

IF CPU=Very_Low ∧ Memory=Medium THEN FV=Medium
IF CPU=Low ∧ Memory=Medium THEN FV=Medium
IF CPU=Medium ∧ Memory=Medium THEN FV=Medium
IF CPU=High ∧ Memory=Medium THEN FV=Low

IF CPU=Very_Low ∧ Memory=High THEN FV=Medium
IF CPU=Low ∧ Memory=High THEN FV=Low
IF CPU=Medium ∧ Memory=High THEN FV=Low
IF CPU=High ∧ Memory=High THEN FV=Low

서희석

2000. 2. 성균관대학교 산업공학과 졸업 (공학사).
2002. 2. 성균관대학교 전기전자 및 컴퓨터공학부 졸업 (공학석사).
2002. 3. ~ 현재 성균관대학교 정보통신공학부 박사과정 재학 중.
관심분야 : 네트워크 보안 시뮬레이션, 지능형 시스템, 분산 에이전트.

김희완

1987년 광운대학교 전자계산학과 졸업(이학사).
1988년 한국전력공사 정보처리처(DBA)
1995년 성균관대학교 정보공학과(공학석사)
1996년 정보처리 기술사(정보관리 부문) 취득
1999년 정보시스템 감리인(한국전산원) 자격 취득
2002년 성균관대학교 전기전자 및 컴퓨터공학부 (공학박사)
1996년 삼육의명대학교 전산정보과 조교수
2001년 삼육대학교 컴퓨터과학과 조교수
관심분야 : 컴퓨터 및 네트워크 보안, 동시성 제어, 분산 DB, 보안 시뮬레이션