SOM을 이용한 멀티 에이전트 기반의 침입자 역 추적 시스템

(Multiagent based on Attacker Traceback System using SOM)

최 진 우 † 우 종 우 †† 박 재 우 †††
(Jinwoo Choi) (Chong-Woo Woo) (Jaewoo Park)

요 약 네트워크 기술의 발달에 힘입어 인터넷이 국가와 사회의 중요한 기반 시설로 자리잡고 있으며, 이를 통한 범죄적 동기를 갖는 해킹 사고의 증가 추세로 인해, 우리 사회 전반적으로 정보에 대한 보호가 시급한 문제로 대두되고 있다. 최근 침입의 형태는 기존 공격자의 직접적인 시스템 침입 및 악의적 행위들의 행사와는 달리 침입 자동화 도구들을 사용하는 형태로 변모해 가고 있다. 알려지지 않은 공격의 유형 또한 변형된 이들 도구들의 사용이 대부분이다. 이들 공격 도구들 대부분은 기존 형태에서 크게 벗어나지 않으며, 침입 도구의 산출물 또한 공통적인 형태로 존재한다. 본 논문에서는 멀티 에이전트 기반의 침입자 역 추적 시스템의 설계 및 구현에 관하여 기술하였다. 본 연구의 시스템은 우선, 알려지지 않은 다양한 공격을 기존 유사한 공격군으로 분류하기 위하여 SOM(Self-Organizing Maps)을 적용하였고, 침입 분석 단계에서는 공격 도구들의 패턴을 지식베이스로 정형화하였다. 이를 기반으로 에이전트 기반의 역 추적 시스템이 활성화 되고, 피 침입 시스템으로부터의 침입 흔적을 발견하여 이전 침입 경유 시스템으로의 경로를 자동으로 역 추적 하게 된다.

키워드 : 역 추적 시스템, 멀티 에이전트 시스템, 자가조직화 맵

Abstract The rapid development of computer network technology has brought the Internet as the major infrastructure to our society. But the rapid increase in malicious computer intrusions using such technology causes urgent problems of protecting our information society. The recent trends of the intrusions reflect that the intruders do not break into victim host directly and do some malicious behaviors. Rather, they tend to use some automated intrusion tools to penetrate systems. Most of the unknown types of the intrusions are caused by using such tools, with some minor modifications. These tools are mostly similar to the previous ones, and the results of using such tools remain the same as in common patterns. In this paper, we are describing design and implementation of attacker-traceback system, which traces the intruder based on the multi-agent architecture. The system first applied SOM to classify the unknown types of the intrusion into previous similar intrusion classes. And during the intrusion analysis stage, we formalized the patterns of the tools as a knowledge base. Based on the patterns, the agent system gets activated, and the automatic tracing of the intrusion routes begins through the previous attacked host, by finding some intrusion evidences on the attacked system.

Key words : Intruder Trace-back System, Multi-agent System, Self-Organizing Maps

1. 서 론

최근 ITU(International Telecommunication Union)의 조사에 따르면 우리나라는 광대역 사용의 선두 국가로 보고되고 있다. 이는 다른 나라로부터의 공격 시 상당한 매력을 느끼게 하는 요인으로서 대부분의 공격 시작점을 제공하게 된다는 점이다. 이러한 광대역 기반 사용자들의 빠른 증가는 다른 나라들에 의한 광대역 사용을 용이하게 함으로써, 우리나라는 10,000명당 가장 많은 공격 수치를 포함하고 있는 공격 랭킹 1위라는 오명을 남기고 있다[1]. 이러한 수치는 순수 우리나라에서 발생된 공격 개시뿐만이 아니라, 침입 경유지로 이용된 사실 모두를 포함하고 있다. 즉, 우리나라의 기관 및 개인 호스트들이 어떠한 형태로든 공격의 중간 경유지로

† 비 회 원 : 한국과학기술원 POST-DOC
Jwchoi@cd.kookmin.ac.kr
†† 정 회 원 : 국민대학교 컴퓨터학부 교수
cwwoo@kookmin.ac.kr
††† 비 회 원 : 국가보안기술연구소 선임연구원
guru@etri.re.kr
논문접수 : 2004년 4월 16일
심사완료 : 2005년 1월 20일

서 활용되고 있음을 의미한다[2]. 위의 사실과 같이 공격 회수의 급속한 증가는 인터넷에서 쉽게 획득할 수 있는 공개된 공격 도구의 사용 및 이들로부터 변조된 공격 도구들에 의해 이루어진 것들이 대부분이다. 따라서 최근에는 공격자로부터 공격이 감행된 근원지를 역 추적해 나가는 공격 근원지 식별이 이러한 침입들에 대응하기 위한 연구의 중심이 되어 오고 있다.

현재까지 보고 되었거나 진행 중인 공격자 근원지 식별을 위한 역 추적 시스템은 개념적인 측면에서 합리적이며, 그 타당성을 가지고 있지만, 실제 세계로의 적용 시 현재의 기술로는 많은 제약을 가지고 있으며, 완전한 자동화 시스템이 개발되기에는 여전히 해결해야 할 많은 문제점들이 존재하고 있다.

본 논문에서는 이러한 문제점을 해결하기 위하여 인공지능 분야에서 연구된 멀티 에이전트의 개념을 도입하여 자동적인 침입 탐지, 분석 및 능동적 대응을 위한 멀티 에이전트 기반 침입자 역 추적 시스템을 제안한다. 본 연구의 시스템은 효율적인 침입탐지를 위하여 자가조직화 신경망(SOM: Self-Organizing Maps)을 이용한 침입탐지시스템과 침입을 탐지한 후 침입의 근원지를 추적하는 멀티에이전트 기반의 침입자 역 추적 시스템의 두 부분으로 구성하였다. 본 시스템은 우선 SOM을 이용하여 몇몇 구분된 공격군으로부터 사용된 공격 도구들의 패턴을 파악하여 지식베이스로 구축하고, 이를 기반으로 침입의 특성을 분석한 후 멀티 에이전트시스템에 의해 침입자를 역 추적 하게 된다. 또한 실험을 위해서 신뢰성 있는 KddCup'99의 데이타 집합을 이용하였다.

본 논문의 구성은 먼저 기존 연구의 쟁점이 되고 있는 공격 근원지 식별에 관한 문제점, 그리고 침입분석에 적용되는 SOM에 관하여 제 2장 관련 연구에서 기술하고, 제 3장에서는 본 논문에서 제안하는 시스템의 설계에 관하여 기술한다. 그리고 제 4장에서 시스템의 구현과 제 5장에서의 결론으로 구성된다.

2. 관련연구

에이전트 기반의 공격자 역 추적 시스템의 개발을 위한 기반 연구로서는 우선 다음과 같은 선행 연구가 필수적이다. 첫째, 침입의 형태를 분석하고 이에 대한 적절한 대응을 위해서는 현재의 침입 에 대한 동향 분석[3] 및 침입탐지 시스템에 대한 전반적인 지식이 필요하다. 둘째, 역 추적 시스템의 기능적 모델에 관한 연구 및 분석이 필요하며, 셋째, 멀티 에이전트 시스템에 관한 구조 및 통신 프로토콜에 관한 연구가 요구된다. 그 밖에 본 연구에서는 침입 형태의 효과적인 분석을 위해 SOM을 적용하였기에 이에 따른 연구도 함께 수행되었다. 본 장에서는 침입행위에 관한 경향 및 SOM에 관하여 기술하고, 그 외의 연구는 제 3 장 시스템 설계에서 기술 한다.

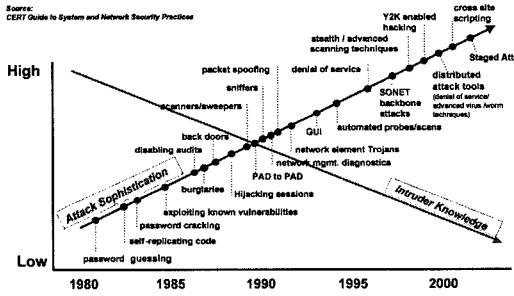

그림 1 최근 침입 행위의 경향

2.1 침입 행위의 경향

그림 1과 같이 1980년대 초 공격 형태는 공격자들의 수작업에 의한 직접적인 명령어들이 사용됨으로써 전문지식이 요구되는 반면, 그 정교함은 낮은 수준이었다. 그러나 이후 자동화된 공격 도구의 사용, 그리고 시스템 취약점 발견을 위한 빠른 스캐닝 도구들이 사용됨으로써 공격은 손쉽게 감행되어 왔다. 따라서 이전과는 상이하게 공격을 위한 전문 지식은 낮아진 반면 공격의 정교함은 매우 높은 수준이 되어 왔다[4].

중요한 점은 이러한 공격 도구들에 의해 수행되는 절차들은 대부분 공통된 패턴을 가지고 있다는 점이며, 이러한 도구들의 사용으로 인한 로그등과 같은 산출물들도 비교적 유사한 형태로 시스템 내부에 존재하게 된다. 따라서, 이들 도구분석의 목적은 공격의 유형 및 공격 기법에 대한 절차적 지식을 획득하기 위함이다.

2.2 공격 근원지 식별 모델

현재 침입탐지 연구의 방향은 수동적, 방어적이기보다 능동적, 공격적으로 변모하고 있으며, 그 결과 최근 연구의 중심은 네트워크 기반에서의 공격 근원지 파악을 목표로 공격이 감행된 근원지를 추적해 나가는 방향으로 진행되고 있다. 본 절에서는 이들 연구들에서 제시하는 해결방안들이 속하는 식별 모델과 그 문제점들에 대하여 기술한다.

네트워크 기반의 공격은 대부분의 경우 공격이 발생하는 시작 지점인 공격자의 호스트로부터 공격을 유도하는 패킷을 발생시킴으로써 공격의 개시가 이루어진다. 예를 들면, 공격자의 호스트는 일련의 패킷들을 생성하고, 생성된 패킷들은 최종적으로 피해자 호스트에게 도달하게 된다. 중요한 점은 이러한 일반적인 절차에서 공격자는 자신의 신원을 노출시킬 수 있는 정보들이 존재할 수 있다는 점이다. 예를 들면, 인터넷 프로토콜 주소(Internet protocol address) 및 사용자 계정(User account)

의 두 가지로 구분할 수 있다.

인터넷 프로토콜을 이용하여 해결하려는 문제는 대부분 네트워크를 구성하는 엔터티들 사이의 스트림에 관하여 해결하려는 패킷 근원지 식별 방안으로서, 네트워크를 경유하는 패킷의 정상적인 흐름의 역방향으로 거슬러 추적해나가는 것이다. 이러한 방법을 이용하는 대표적인 해결 방안들로는 IP 패킷 마킹 스킴(IP packet marking scheme)[5,6], iTrace(ICMP traceback message)[7] 등이 있다. IP 패킷 마킹 스킴은 라우터로부터 패킷 전송 시, 부분적인 경로 정보를 가지고 예측되는 패킷을 표식화하는 방법이며, iTrace는 라우터로부터의 패킷 전송 시, 패킷의 복사본을 저장하고, 역 추적을 위한 ICMP 메시지를 생성한 뒤, 해당 목적지로 전송하는 방법이다. 이들 방안들에서의 공통적인 문제점들로는 다음과 같다.

① 네트워크를 구성하는 모든 엔터티들의 소프트웨어/하드웨어 측면에서의 변경이 필요하다.
② 1라우터와 같은 네트워크 엔터티들의 기능적 오버헤드가 존재한다.
③ iTrace에서 제안하는 알고리즘에 의한 네트워크 프로토콜의 변경이 필요하다.
④ IP marking에서 제안하는 알고리즘에 의한 패킷 최소 사이즈의 변경이 필요하다.

사용자 계정에 관한 문제는 감사 증적(Audit trail)을 이용하여 해결하려는 고전적인 방안으로서, 대부분 시스템에 의해 제어되는 로그들에 대한 세밀한 분석을 통해 이루어진다. 감사 증적으로는 네트워크를 포함하는 시스템의 사용에 관한 로그들과 시스템 사용자들의 행위에 관련된 히스토리(history)들이 존재한다. 이러한 방법에 대한 문제점은 다음과 같다.

① 감사 증적을 위해 수집되는 로그와 같은 데이타의 크기는 매우 방대하다.
② 로그들의 분석 시 시스템 보안 전문가들에 의한 전문적인 지식에 의존되고 있다.
③ 대부분의 경우 전문가들의 수작업에 의한 분석 위주이며, 자동화 되어 있지 않다.

2.3 자가 조직화 신경망(SOM)

초기의 경쟁 학습 알고리즘은 단순 경쟁 학습 알고리즘으로서 항상 승자 뉴런만을 학습 시키므로 초기 가중치 벡터들의 분포에 따라 전혀 학습이 이루어지지 않는 출력 뉴런들이 생기는 문제점이 발생한다. 이를 해결하기 위한 다양한 접근들이 있었으며, 이 중 가장 대표적인 경쟁 학습이 Kohonen의 SOM이다[8]. SOM 알고리즘은 승자 뉴런뿐만 아니라 위상적으로 이웃한 뉴런들도 함께 학습시킨다. 이는 그림 2와 같이 십자형의 외부를 표현하는 2차원의 입력을 2000번 학습한 뒤에도 서로 인접한 비슷한 입력 패턴들이 인접한 출력 뉴런들의 기하학적인 관계로써 형성됨을 의미한다.

그림 2와 같이 일반적인 X, Y 좌표계로 이루어진 2차원 입력에 대한 2 차원 맵은 그 표현이 용이하지만, 고차원의 입력의 경우 2 차원의 맵으로 표현하기는 쉽지 않다. 이를 위한 방법으로 점증적 색상 대비를 이용하는 U-matrix 가시화 방법이 존재한다. 다음과 같은 다차원의 입력 그림 3에 대하여 U-matrix 가시화는 그림 4와 같다. 그림 4의 좌측 하단의 그림에서 상단으로는 날짐승이 위치하고, 하단으로는 그 외의 짐승들이 군집화됨을 볼 수 있으며, 우측화면은 각 특징들에 대한 분포를 나타내고 있다.

위 예제에서 군집화를 위한 SOM의 학습과정은 다음과 같이 수행된다.

•1 단계: 연결 가중치를 초기화 한다. N개의 입력과

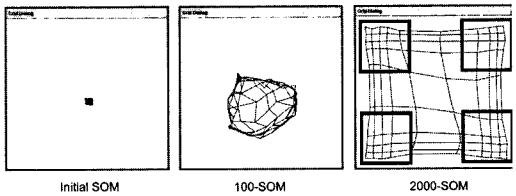

그림 2 SOM의 학습

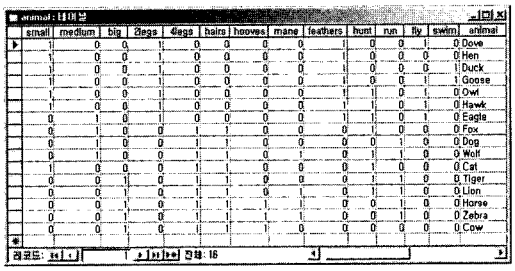

그림 3 학습 데이타로 사용되는 동물 이름과 속성

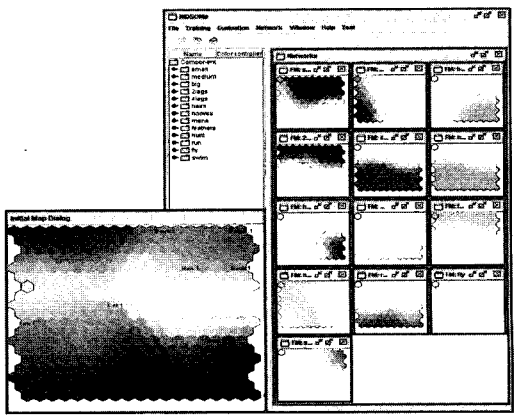
그림 4 U-matrix 기법을 이용한 SOM의 가시화

M개의 출력을 연결하는 가중치들은 아주 작은 임의의 값으로 설정한다.

- 2 단계: 새로운 입력 패턴을 입력 뉴런에 제시한다.
- 3 단계: 입력 벡터와 모든 출력 뉴런들과의 거리를 계산한다.

$$d_j = \sum_{i=0}^{N-1} [X_i(t) - W_{ij}(t)] \quad (1)$$

$X_i(t)$는 시간 t에서 뉴런 i로의 입력, $W_{ij}(t)$는 시간 t에서 입력 뉴런 i로부터의 출력 뉴런 j로의 가중치

- 4 단계: 최소 거리를 가지는 승자 뉴런 c를 구한다.

$$c = \min_j d_j \quad (2)$$

- 5 단계: 승자 뉴런 c와 이웃 출력 뉴런의 연결 가중치들을 갱신한다.

$$W_{ij}(t+1) = W_{ij}(t) + \alpha(t)[X_i(t) - W_{ij}(t)] \; for \; j \in NE_c(t) \quad (3)$$

$\alpha(t)$는 0과 1 사이의 값을 가지는 학습률로서 시간에 따라 감소하는 함수이며, $NE_c(t)$는 승자 뉴런 c로부터 일정한 범위 내에 있는 출력 뉴런을 포함한다.

- 6 단계: 2 단계의 새로운 입력 벡터를 처리한다.
- 7 단계: 지정된 학습 회수까지 2 단계부터 6 단계의 과정을 반복 수행한다.

침입탐지 알고리즘으로써 SOM의 적용시 얻게 되는 장점은 다음과 같다.

- 첫째, 신경망 학습에서 교사학습은 알려져 있는 문제에 대하여 학습 데이타들의 레이블링 작업이 수반된다. 특히 감사 시스템에 의해 생성되는 많은 양의 로그들을 분석하고, 레이블링 하는 작업은 많은 비용과 시간이 요구된다. 반면, 비교사 학습은 레이블이 필요치 않으며, 또한 알려져 있지 않은 문제에 대해서도 특징들간의 관계만을 가지고 학습이 가능한 장점을 제공한다.
- 둘째, 비교사 학습의 SOM은 패턴 분석 및 클러스터링, 양쪽 모두에게 적용성을 제공한다. 또한 U-matrix 기법을 이용한 클러스터 가시화가 가능함으로 비록 알려져 있지 않은 문제에서도 직관적인 이해가 가능하다는 장점을 제공한다.

3. 시스템 설계

본 연구에서 전체 시스템은 침입탐지 시스템과 역 추적 시스템의 두 부분으로 구성된다. 전반적인 시스템의 흐름은, 우선 공격자의 침입을 SOM 기반 침입 탐지 시스템이 판별하게 되고, 침입사실이 확인되면 공격자를 역 추적하기 위하여 첫 번째 경유 시스템으로 역 추적 에이전트가 파견되어 지속적으로 침입자를 추적하게 된다. 본 장에서는 이들 두 시스템의 설계에 대하여 기술한다.

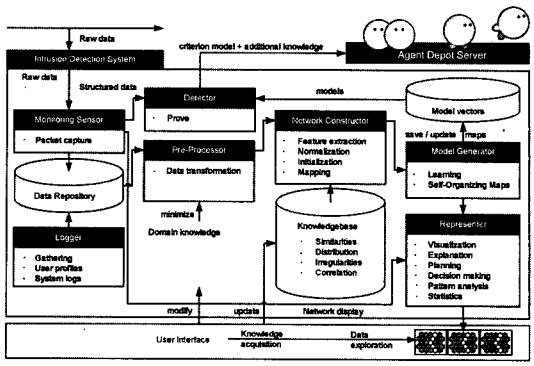

그림 5 침입 탐지 시스템의 설계

3.1 침입 탐지 시스템

침입탐지 시스템은 다양한 기능을 보유할 수 있지만, 본 연구에서는 침입탐지를 주 기능으로 설정하였다. 이러한 목적으로 본 연구의 침입 탐지 시스템은 6개의 핵심 모듈인 모니터링 센서(monitoring sensor), 탐지기(detector), 전처리기(pre-processor), 네트워크 생성기(network constructor), 모델 생성기(model generator), 모델 표현기(representer)로 구성하였다[그림 5]. 시스템의 일부를 구성하는 모듈인 로거(logger)는 감사 자료를 수집하고, 이를 데이타 저장소에 저장을 한다. 이는 오프라인 학습을 위한 부가적인 데이타로써 SOM의 재학습 시 이용하기 위함이다. 제안된 시스템의 구성 요소들과 이들의 수행과정을 살펴보면 다음과 같다.

- 네트워크 모니터링 센서는 네트워크를 경유하는 패킷들을 수집한다. 본 시스템에서는 이를 위해 tcpdump를 사용하여 산출물을 획득한 후 이를 구조화된 형태로 변환한다. 변환된 패킷 데이타들은 추후 오프라인 학습을 위한 입력으로 사용되기 위하여 데이타 저장소에 저장된다.
- 전처리기는 데이타 저장소로부터의 데이타의 선택 또는 정제를 위한 단계이다. 일반적으로 데이타 저장소에는 많은 부류의 데이타들이 저장되어 있으며, 이질적인 형태로 구성되어 있다. 이러한 이질적인 구조의 데이타들은 신경망의 입력으로 사용하기 위하여 알맞은 형태로 변환하거나 재배열하는 과정이 필요하다.
- 네트워크 생성기는 신경망의 속성에 관련된 변수의 결정과 같은 기능을 담당한다. 예를 들어 신경망을 구성하는 뉴런들의 개수, 뉴런을 대표하는 벡터들 간의 유사도 척도로 사용될 함수, 상관관계를 가지는 특징들의 선택 등이 지식베이스를 참조하여 결정된다. 그리고 맵의 초기화를 수행한 뒤 신경망의 입력으로 제

공한다.

- 모델 생성기는 침입 탐지 시스템의 핵심 모듈로써, 생성된 모델은 신경망의 학습 후 생성되는 산출물이 된다. 본 설계에서는 비 교사 학습인 SOM을 사용한 모델 벡터들의 생성을 제안한다. 그리고 신경망 속성들을 지식베이스를 참조하게 함으로써, 다양한 학습 엔진의 탑재가 가능하도록 적응력을 고려한다.
- 모델 표현기는 학습 되어진 모델 벡터의 가시화를 담당한다. 가시화 작업은 SOM이 제공하는 대표적인 특징들 중의 하나로써, SOM에 의해 형성된 맵은 가시적인 형태로 표현된다. 이는 보안 전문가들과의 협력을 통한 새로운 해석이 가능하며, 보다 효과적인 새로운 모델의 생성이 가능하다.
- 탐지기는 모니터링 센서들로부터의 구조화된 데이타를 수신한 후, 이를 판별하기 위하여 모델 생성기에 의해 생성된 모델 벡터들과의 유사성 척도를 기준으로 평가하게 된다.

3.2 침입 탐지를 위한 자가 조직화 신경망(SOM)

SOM 프레임워크의 구조는 그림 5의 모델 생성기에 위치하며, 그림 6과 같이 6개의 모듈로 구성된다. SOM 모듈의 상위는 입력과 출력을 의미한다.

- 네트워크 관리자(Network Manager): 벡터 핸들러 에서 유입된 입력 데이타로부터 선택된 특징들에 대하여 맵을 구성한다. 이때 맵을 구성하는 뉴런들의 개수를 설정할 수 있으며, 또한 학습된 맵의 가시화를 수행하는 U-matrix 알고리즘을 내부적으로 포함하여 가시화를 위한 맵의 구성 에도 참여한다.
- SOM 계층(SOM Layer): 네트워크 매니저로부터 유입되는 입력을 받아 SOM 알고리즘을 이용하여 학습한다.
- ALPHA 관리자(ALPHA Manager): 0과 1 사이의 값을 가지는 학습률로서 시간에 따라 감소하는 함수이며, 이를 통해 현재 학습률(α)을 조절한다.

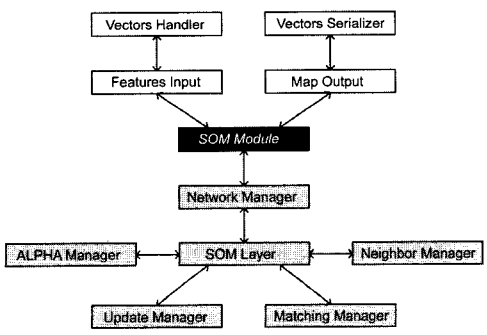

그림 6 SOM 프레임워크

- 이웃 관리자 (Neighbor Manager): 이웃함수를 가지고 적절한 반경 내의 이웃 뉴런들을 선택한다. 네트워크 평가를 위한 이웃함수로써 Bubble 또는 Gaussian 함수를 선택 할 수 있도록 구성한다.
- 매칭 관리자(Matching manager): 입력과 맵 상의 뉴런들의 BMU(best-matching unit)를 선택한다.
- 업데이트 관리자(Update Manager): 현재의 뉴런들의 가중치의 업데이트를 담당한다.

이러한 절차에 상응하는 결과들은 제 4장 구현부분에서 확인할 수 있다.

(1) 정규화

본 논문에서 사용하는 KddCup'99 데이타 집합[9]은 1998 DARPA Intrusion Detection Evaluation Program에 의해, 표준 데이타 집합을 얻기 위하여 미국 군사 네트워크(military network) 상에서 9주에 걸친 시뮬레이션을 통해 만들어진 데이타로써, 침입탐지 알고리즘을 시험하기 위해 제공되는 데이타 집합이다. 이들 데이타들은 수치형 데이타와 기호형 데이타들의 혼합으로 구성되어 있는데, 기호형 데이타들은 개별적 TCP 연결의 기본 특징들에 관한 부분으로 "*ptotocol_type*", "*service*", "*flag*" 등으로 표현된다. 또한 수치형 데이타로써 연속적인 값을 표현하는 "src_bytes", "dst_bytes"는 수치 분포의 범위가 매우 크기 때문에 각각의 벡터를 별도의 변환과정 없이 신경망의 학습에 사용한다면, 높은 분산을 가지는 차원이 맵의 조직에 다수를 점하는 경향으로 나타내게 될 것이다. 이러한 문제를 해결하기 위하여 특징 벡터들에 대하여 유클리드 거리의 개념을 내포하는 정규화를 수행한다. 총 길이 의 입력 데이타 에 대하여 차원의 연속형 실수값인 경우 다음을 만족한다.

$$X = \{X_i\}_{i=1}^{n},\ X_i^R \in R^d \tag{4}$$

위를 만족하는 경우 정규화는 다음과 같다.

$$\textit{the length of } X_i^R = l_i = \sqrt{\sum_{j=1}^{d} X_{ij}^2} \tag{5}$$

$$n_i = \frac{x_i}{l_i} \tag{6}$$

$\langle x_1, x_2, \cdots, x_d \rangle \rightarrow \langle n_1, n_2, \cdots, n_d \rangle$ n_i : *normalized vector; range[0,1]*

연속형 실수값, 뿐만 아니라 기호형 데이타들도 침입탐지를 위한 중요한 정보를 내포하고 있을 수 있으며, 또는 시험에 의해 클래스를 구성하는 새로운 특징들간의 관계를 획득할 수도 있다.

본 논문에서는 두 가지 타입의 데이타를 함께 학습에 참여시킴으로써 보다 효과적인 클러스터링의 성능을 기대할 수 있도록 설계하였다. 기호형 데이타의 값들은 임의 이산 수치값을 부여하고 위와 유사한 과정을 수행한 후 SOM 학습을 위한 입력으로 사용한다.

Algorithm SOM
Input: Set of N dimension vector, X
Output: Subsets of input data. (M subsets)

begin
Randomly initialize $W_j = (w_{j1}, w_{j2}, w_{j3}, \ldots, w_{jn})$ for each node
for ($t = 0$; unless a topping condition is reached; Increase t)
  for (for all input data)
    for ($i = 0$ to M)
    Compute $D_i = \|X_t - W_i^{(t)}\|$
    endfor
  Find the winner $j = i$ such that $D_i(t)$is minimum for overall i
  Update the winner j (and its neighbors)
  endfor
endfor
end

그림 7 SOM의 학습 알고리즘

(2) SOM 학습

그림 7의 알고리즘에 대한 단계적인 설명은 다음과 같이 4 단계로 구분된다.

① 연결 가중치 벡터들의 초기값을 임의로 할당한다.
② 임의 연결 가중치를 할당 후 입력 벡터와 유사성을 측정한다. 본 논문에서는 일반적인 유클리드 거리를 사용한다.
③ 승자 뉴런을 발견하면 연결 가중치 갱신을 위해 normalized vector sum을 사용한다. 또한 이웃 뉴런들의 가중치를 갱신하기 위하여 가우시안 함수를 사용한다.
④ 이를 모든 입력 벡터에 대하여 순환 적용하기 위해 ②-③을 반복한다.

침입 탐지 결과는 크게 4가지로 구분된다. 이러한 구분은 KddCup'99의 데이타가 4개의 공격군(DoS, R2L, U2R, Probing)으로 구성되기 때문이다. 이들 개별적인 공격군들에 대한 학습된 모델을 기준 모델(criterion model)로 결정하고, 이러한 정보에 기반한 에이전트들이 최초의 피 침입 시스템으로 파견된다.

3.3 멀티 에이전트 기반 역 추적 시스템

침입탐지 시스템에 의하여 침입사실이 확인되면, 침입자를 역추적하기 위한 에이전트 시스템이 활성화 된다. 본 논문에서는 역 추적 시스템을 3-계층의 멀티에이전트 시스템으로 설계 하였다[그림 8]. 첫 번째 계층은 SOM 기반의 IDS로 구성된다. 이는 모든 하위 네트워크를 구성하는 호스트들 상의 실시간 모니터링으로 인한 부담을 해결하기 위해서이다. 두 번째 계층은 에이전트 플랫폼과의 통신을 위한 에이전트 서버(Agent Depot Server)로 구성된다. 세 번째 계층은, 침입 추정 시에 에이전트 서버로부터의 호출되는 각 호스트상의 독립적인 멀티에이전트 시스템으로 구성된다.

본 절에서는 이러한 멀티 에이전트 시스템의 표현을 (1)에서 정의하였고, 멀티 에이전트 시스템의 기본요소들은 (2) 멀티 에이전트 시스템의 구조에서 기술하였다. 또한 에이전트의 통신 프로토콜 및 메시지 해석기 등은 사전 연구에서 수행하였기에, (3)에서는 에이전트간의 상호 협동에 관하여 간략히 기술한다.

(1) 에이전트 표현을 위한 액티브 오브젝트

에이전트는 일반적으로 하드웨어, 소프트웨어, 프로세스 또는 쓰레드 등 다양한 관점으로 해석되고 있다[10-12]. 그러나 에이전트의 설계 시 반드시 고려해야 할 점은 에이전트 자체의 특성을 반영하여야 하고, 이를 표현할 수 있는 개발 언어를 선택하여야 한다는 점이다. 본 논문에서는 에이전트의 표현을 액티브 오브젝트(Active Objects)로 정의하였다. 본 논문에서 정의하는 액티브 오브젝트의 구성 조건은 다음과 같다.

- 첫째, 액티브 오브젝트들은 프로그래밍 언어와 디스크립션 언어가 함께 설계되어야 한다. 이는 오브젝트 내에 제어 구조를 제공하는 API를 갖추어야 하고, 이들이 참조할 상위 레벨에서 인스트럭션 표현의 명시화가 가능한 디스크립션으로 설계되어야 함을 의미한다
- 둘째, 액티브 오브젝트들은 순환적 행위를 제공하도록 설계되어야 한다. 이는 액티브 오브젝트들이 작업을 만족 할 때까지 지속적으로 수행되도록 설계되어야 함을 의미한다.
- 셋째, 액티브 오브젝트들은 재사용과 복제가 가능해야 하며, 자신의 상태를 보존할 수 있는 언어로 표현 가능하여야 한다. 에이전트의 개념에서 복제란 액티브 오브젝트 실행 기간의 어느 시점에서 자신의 상태를

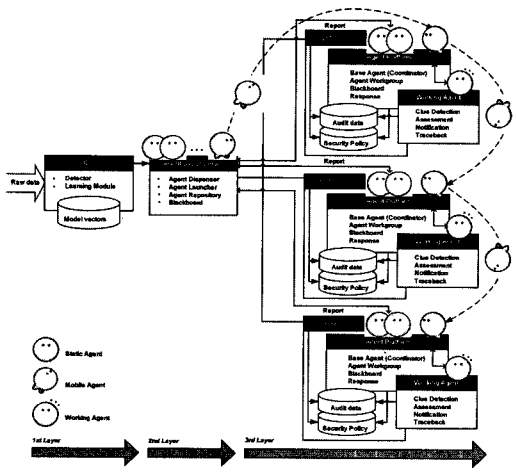

그림 8 멀티 에이전트 기반의 역 추적 시스템 구조

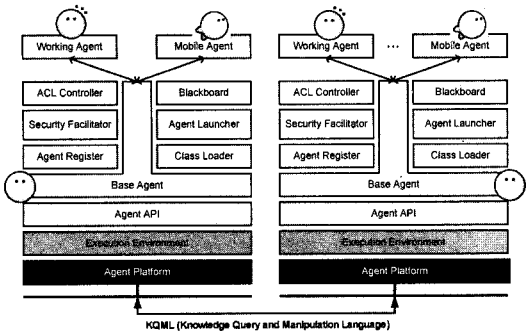

그림 9 멀티 에이전트 시스템의 기본 구조

보존하는 또 다른 객체의 생성이 가능함을 말한다.

- 넷째, 액티브 오브젝트들은 이벤트 위주의 방식(event-driven fashion)으로 설계되어야 한다. 일반적인 프로그램은 실행 도중 무엇이 발생 할 지 미리 예측할 수 있도록 설계될 수 없다. 이를 위해 설계 시 발생할 수 있는 이벤트들은 미리 명세화되고 표현되어야만 한다.

본 설계에서는 위의 구성 조건을 만족 시키기 위하여, 첫째, 절차적 지식 표현으로써 많은 장점을 제공하여온 "S-expression" 구조를 선택하였다. 둘째, 모든 액티브 오브젝트를 추상화 객체인 "Runnable object"로 부터 확장 가능하도록 설계하였다. 셋째, 재사용과 복제를 위해 Java 언어에서 기본적으로 제공하는 Cloneable과 Serializable 패러다임을 이용하여 설계하였다. 넷째, 이벤트의 표현 또한 "S-expression"에 기반하여 설계하고, 에이전트들간 상호 통신은 KQML[13,14]을 사용하여 설계하였다.

(2) 멀티 에이전트 시스템의 구조

본 연구의 멀티 에이전트 시스템 구조는 에이전트간 상호 협조적 관계를 위한 역할과 에이전트 자체의 존속성에 따라 구분된다. 우선 각 에이전트 역할을 기준으로 다음 세 가지 형태로 구성 하였다. 즉, 조정자 역할을 하는 기지 에이전트(Base Agent), 주어진 작업을 수행하는 작업 에이전트(Working Agent), 그리고 작업 에이전트로부터 유도되어 또 다른 에이전트 플랫폼으로 이동하여 실행되는 이동 에이전트(Mobile Agent)들로 구성하였다. 또한, 이들 에이전트들은 존속성에 따라 정적 에이전트(Static Agent)와 동적 에이전트(Dynamic Agent)로 구별된다. 정적 에이전트는 시스템 상에 항상 위치하여 데몬과 같은 서비스를 제공하는 기지 에이전트와 해당 호스트에서의 보안 관련 시스템 자원(ACL controller, Security Facilitator)에 대하여 직접적인 접근이 가능한 작업 에이전트들을 포함한다. 동적 에이전트는 에이전트의 존재 유무가 가변적인 에이전트로써, 보안 관련 시스템 자원으로의 직접적인 접근이 가능한

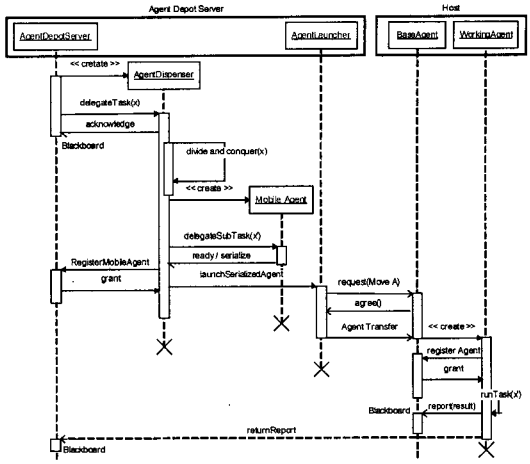

그림 10 멀티 에이전트 시스템의 순차 다이어그램

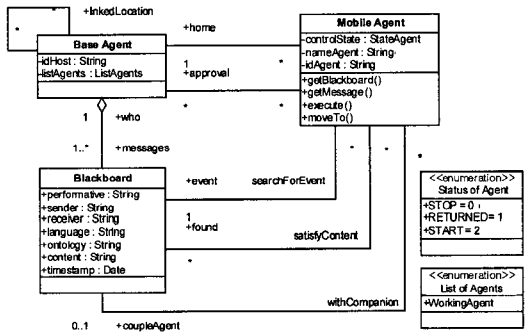

그림 11 멀티 에이전트 시스템의 클래스 다이어그램

에이전트를 제외한 모든 작업 에이전트, 즉 다른 에이전트에 의해 생성되거나, 기지 에이전트에 의해 동적으로 생성되고 소멸되는 에이전트들이 이에 속한다. 그리고 이러한 작업 에이전트로부터 유도된 이동 에이전트들이 정적 에이전트에 속한다[그림 9].

(3) 에이전트간의 상호작용 및 협동

멀티 에이전트 시스템에서의 협력은 에이전트들의 목표 수행 과정 중 하나의 문제를 해결하기 위하여 여러 다른 에이전트들과 상호 통신하여 해결해 나아감을 의미한다. 이를 위하여, 본 설계에서는 주어진 문제, 즉 역추적이라는 문제를 해결하기 위하여 네트워크를 구성하는 각 호스트들 상에 존재하는 에이전트들이 상호 협력하여 해결하도록 설계하였다.

그림 10은 전체 시스템 구조 내에서의 협력 관계를 도식화한 순차 다이어그램이다. 에이전트 서버(Agent Depot Server)로부터 특정 공격군의 형태를 인지하고, 이를 목표로 설정한 뒤, 이를 해결하기 위하여 에이전트들을 생성하고, 각 호스트 상의 에이전트 플랫폼으로 파

견한다. 파견된 에이전트(Mobile Agent)는 에이전트 플랫폼의 승인을 얻고, 기지 에이전트(Base Agent)에 등록한 후, 자신의 임무를 수행하게 된다. 그리고 최종 수행 결과를 기지 에이전트와 에이전트 서버로 통보한다.

에이전트들의 목표 수행 과정 중 문제를 해결하기 위하여 다른 에이전트들의 서비스를 요구할 수 있고, 또한 이들 에이전트들 사이의 정보 교환이 필요로 하게 된다. 이를 위하여 본 시스템에서는 블랙보드 구조를 사용하여 해결하도록 설계하였다. 이때의 블랙보드는 모든 에이전트들에게 개방되어 있는 전역 공간으로써 자신에게 지칭된 메시지들만을 수신하기 위해 사용된다[그림 11].

3.4 침입 분석 단계

역 추적 에이전트는 파견된 피 침입 시스템으로부터 침입흔적을 분석하여 지속적으로 추적할 수 있어야 한다. 본 절에서는 에이전트의 침입분석과정에 대하여 기술한다.

일반적으로 네트워크 공격인 경우, 예를 들어 스캐닝 공격, DoS 공격들은 네트워크 기반에서의 네트워크 통신량 또는 동일 근원지로부터의 패킷 전송 여부에 관련된 정보를 이용하여 위치 정보를 분석할 수 있다. 그러나 이러한 네트워크 공격은 근원지 IP 위장 기술로 인해 발견이 쉽지 않다. 따라서, 본 논문의 역 추적 관점은 호스트 기반으로 제한한다. 또한 사용된 KddCup'99 데이타 집합을 이용할 때 설계 상의 제약이 있으며, 이를 위한 해결 방안은 다음과 같다.

첫째, KddCup'99 학습 데이타 집합에는 TCP 기반의 자료와 Content 기반의 축약 자료가 혼용되어 있다. 즉, R2L(Remote to Local)과 U2R(User to Root)과 같은 공격군에 관한 정보는 TCP 기반의 자료와는 관련성이 희박하다는 것이다. KddCup'99 학습 데이타 에서 Content 기반의 축약 자료는 오직 피 침입 시스템 또는 침입 경유 시스템에서 사용되어야 타당할 것이다.

둘째, SOM의 학습을 위한 시험 데이타 집합에는 시스템들의 위치 정보는 존재하지 않는다. 그러므로 역 추적 시스템과의 연계는 불가능하다.

본 논문에서는 이러한 제약을 해결하기 위해 방안으로 다음과 같이 설계하였다.

- 첫째, KddCup'99 데이타 집합을 학습하여 생성된 모델과 위치 정보가 존재하지 않는 시험 집합을 수정없이 적용하기 위한 역 추적 시나리오를 선택한다. 이를 위해 본 논문에서는 R2L과 U2R 양쪽 모두를 고려하여 맵을 구성한다.
- 둘째, R2L과 U2R의 경우, 공격자는 자신들의 흔적을 제거하기 위해 로그 파일들을 제거한다. 그러나 실제 이들 공격군에는 일반적으로 알려진 루트킷/백도어(Rootkit/Backdoor)와 같은 악성 프로그램들이 사용되며, 이러한 프로그램들은 공통적으로 특수목적의 파일들을 생성한다. 이들 파일들의 분석을 통해 역 추적 과정을 수행한다.

(1) 침입에 사용된 도구의 분석

침입에 사용된 도구에 관한 구체적인 분석 작업이 필요한 이유는 대부분의 침입이 단일 공격에 의해서 일어나지 않기 때문이다. 즉 U2R만의 공격에 의해 침입이 이루어지는 것이 아니라, R2L 공격과 병행하여 시도된다. 예를 들어, R2L의 "guess_passwd", "dictionary" 공격들이 선행되어 사용자의 패스워드를 알아내고 적법한 사용자 로그인을 성공한 뒤, 루트킷/백도어를 다운로드하고, 이를 이용하여 루트의 권한을 획득하기 위한 U2R 공격이 이루어지는 것이다. 그러므로 이들을 구체적으로 분석할 수 있는 선행작업이 우선 요구된다.

(2) 침입 경유지/발원지 분석 단계

침입에 사용된 도구 분석 등 세부적인 분석이 완료되면 그 결과는 어떠한 공격군에 포함된 구체적인 침입 유형이 될 것이며, 이 정보를 이용하여 이전 침입 경유 시스템의 위치 정보를 발견하게 되고, 침입 발원 시스템에 도달할 때까지 이러한 과정을 지속적으로 순환한다. 예를 들어 U2R 공격군에서, 루트킷이 사용하는 특수 목적의 파일로부터 공격자의 접속 정보를 획득할 수 있다. 이때의 특수 목적이란 시스템 명령어들의 변조를 의미하며, 루트킷은 이러한 변조를 위하여 몇몇 파일들을 생성한다[그림 12]. 그러므로 이들의 발견을 위한 관련 정보를 지식화 하여 작업 에이전트들의 개별적인 고유 지식으로 활용하며, 이들을 분석할 수 있는 절차 지식들을 설계하였다.

일반적으로 루트킷이 발견되면 시스템 운영에 관련하여 빈번도가 높은 명령어들은 이미 변조되었을 가능성이 있다. 예를 들면 *find, ps, top, ifconfig* 등이 그 가능성을 가지는 명령어들이다. 이러한 문제의 해결을 위해서 시스템 에이전트에 의해 제공되는 기능을 사용하도록 설계하였다.

Netstat	Modified to remove tcp/udp/sockets from or to specified addresses, uids and ports. The file is ROOTKIT_ADDRESS_FILE. ***default data file.*** /dev/ptyq ***type 0:*** hide uid ***type 1:*** hide local address ***type 2:*** hide remote address ***type 3:*** hide local port ***type 4:*** hide remote port ***type 5:*** hide socket path
Examples	***Explanations***
0 500	Hides all connections by uid 500
1 123.45	Hides all local connections from 123.45.X.X
2 123.45.67.89	Hides all remote connections to 123.45.67.89
3 8000	Hides all local connections from port 8000
4 6667	Hides all remote connections to port 6667
5 .term/socket	Hides all UNIX sockets including the path .term/socket

그림 12 루트킷에 의해 생성되는 특수 파일

4. 시스템 구현

본 논문에서는 신뢰성 있는 모의 실험을 위하여 훈련 데이타집합으로 KddCup'99데이타를 사용하였으며, 침입 탐지 분석 알고리즘으로는 비 교사 학습인 SOM을 이용하여 개발하였다. 본 시스템의 개발 언어로는 Java(JDK 1.4.2)를 사용하였고, KQML 프레임워크를 지원하는 라이브러리로 JATLite[15]를 사용하였다. 구현 플랫폼으로는 Linux 6.0(RedHat 6.0)을 선택하였다.

4.1 멀티 에이전트 시스템 구현

시스템 구현은 우선 사용자에게 시뮬레이션 인터페이스를 제공 함으로서 사용자에 의해 구현 결과를 단계별로 수정 및 진행해 볼 수 있으며, 이러한 검증이 종료된 이후에는 자동화된 단계로 진행된다. 멀티 에이전트 시스템을 위한 시뮬레이션은 다음 몇 가지 절차로 구성된다.

① 첫째, 시스템의 데몬과 같은 역할을 수행하는 기지 에이전트의 구성 단계이다. 에이전트 서버 및 하위 호스트들에는 기지 에이전트가 설치되어, 각 시스템 상의 위치 정보와 통신 정보를 유지하게 된다. 그림 13의 좌측 상단의 윈도우는 기지 에이전트의 생성을 나타내고 있다.

② 둘째, 기지 에이전트에 의해 작업 에이전트들이 동적으로 생성되는 단계이다. 시뮬레이션 인터페이스에서는 작업 에이전트의 속성을 확인하거나, 수정할 수 있게 된다. 그림 13의 좌측 하단의 윈도우는 생성된 작업 에이전트와의 커넥션을 나타내고 있다.

③ 셋째, 작업 에이전트와 기지 에이전트의 등록 단계로써, 작업 에이전트들을 생성한 상위 에이전트인 기지 에이전트와의 관계를 등록한다. 작업 에이전트들의 등록 다이얼로그를 통해 상위 에이전트의 아이디 및 이름을 확인할 수 있다. 이때의 위치 정보는 내부적으로만 해석되며, 상위 객체의 이름만을 기술하게 된다. 그림 13의 좌측 하단의 윈도우 내의 우측 패널에서 등록된 작업 에이전트로의 메시지 전송 부분을 나타내고 있다.

④ 넷째, 기지에이전트에 등록된 작업 에이전트들과의 통신 단계이다. 작업 에이전트는 자신의 목표에 따라 다른 작업 에이전트들 또는 시스템 에이전트들과의 통신을 기지 에이전트를 통한 자유로운 메시지 전달이 가능하다. 에이전트 상호간 메시지 교환을 위해 KQML로 구성된 스트림을 전송하게 되며, 이들 메시지들은 에이전트 내부의 인터프리터에 의해 해석된다. 그림 13의 우측 하단의 중앙 패널은 생성된 작업 에이전트의 수행 결과를 기지 에이전트로의 전송을 나타낸다.

4.2 침입 탐지 학습을 위한 시스템 구현

침입 탐지 시스템의 학습을 위한 시스템은 시스템 성능에 의해 리눅스 플랫폼이 아닌 Windows 2000에서 구현되었다. 그리고 훈련 및 시험 데이타 집합을 저장하기 위해 Microsoft Access를 사용하였다. KddCup'99 학습 데이타 집합을 이용하여 학습을 수행한 후, 평가된 최종 결과는 침입 모델로 구성되어, 에이전트 서버 내에 존재하게 된다.

(1) 네트워크의 생성

학습 데이타를 구성하는 필드들은 선택적으로 학습에 참여할 수 있도록 구성하였다. 이를 위해 그림 4-2의 좌측의 데이타베이스 다이얼로그를 통해 조정이 가능하며, 이는 학습 시의 소모적인 연산을 피하기 위하여 최종 결과에 영향력이 없는 특징들을 배제하기 위한 것이다. 학습에 참여할 필드들을 선택하면 전체 필드 중 선택된 필드의 개수와 선택되지 않은 필드의 개수가 기록된다. 그림 14 우측 상단의 맵 속성 다이얼로그를 통해 네트워크 구성에 관여하는 속성들을 설정할 수 있다. 네트워크 구성의 속성들로는 뉴런들의 개수와 위상 구조, 그리고 학습 회수를 설정할 수 있다. 선택된 특징들의 개수들은 이전 단계에서 선택된 필드들의 개수와 동일하다.

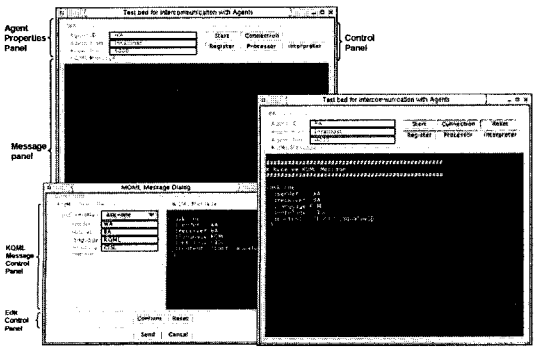

그림 13 기지 에이전트와 작업 에이전트들 간의 통신

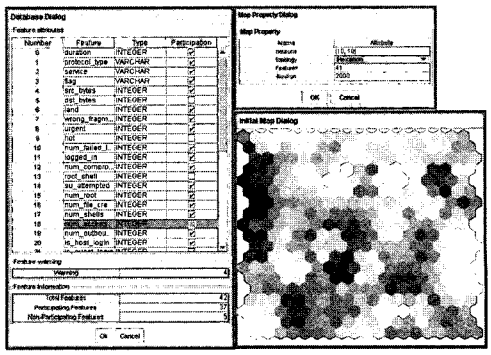

그림 14 SOM의 구성

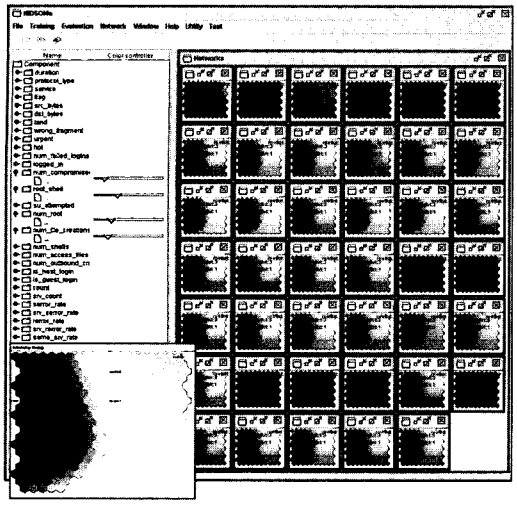

그림 15 SOM의 학습 결과와 특징들간의 관계

그림 14의 우측 하단은 맵의 초기화된 상태를 나타내고 있다.

(2) 네트워크의 학습

그림 15은 구성된 네트워크를 2000번 학습한 결과와 클러스터링 된 결과를 나타낸다. 학습 결과로부터 새로운 지식을 도출하기 위해 "color controller" 인터페이스를 사용함으로써 각 특징들의 관계를 가시화할 수 있다.

5. 결 론

본 논문에서는 기존 공격으로부터 변형된 알려지지 않은 다양한 공격 유형을 기존의 유사한 공격군으로 분류하기 위하여, 침입 분석 알고리즘으로 SOM을 적용하여 침입의 특성을 분석하였다. 이러한 침입분석의 결과에 따라 피 침입 시스템으로부터의 침입 흔적을 발견하여, 이전 침입 경유 시스템으로의 경로를 역추적하는 멀티 에이전트 기반의 공격자 역 추적 시스템을 설계 및 구현하였다. 본 논문의 중요성은 다음과 같다.

첫째, 에이전트화를 위한 액티브 오브젝트를 정의하였다. 즉, 에이전트 설계 시 고려 해야 할 다양한 정의를 분석하여 이를 액티브 오브젝트로 정의 하였다. 둘째, 피 침입 시스템으로부터 침입 경유 시스템과 최종 침입 발원 시스템으로의 추적을 위한 구체적인 추적 단계를 상세히 설계하였다. 또한, 역 추적 과정의 분석을 바탕으로 하여 시험 환경을 구축하였으며, 실제 구현 함으로서 역 추적의 타당성을 검증하였다. 셋째, 이러한 연구를 기반으로 멀티 에이전트 시스템을 설계하였고, 에이전트들간의 협력을 위한 통신 프로토콜을 KQML기반으로 설계 하였다. 넷째, 침입 탐지를 위한 에이전트의 학습을 위해 SOM을 적용하였다.

본 논문의 결과는 다음과 같이 활용할 수 있을 것이다. 우선 급증하고 있는 해킹 피해 사례에 대하여 기존의 시스템들은 수동적인 탐지기능으로 일관하고 있으나, 본 논문의 결과는 해킹 사고에 적극적으로 대응할 수 있는 방향을 제시 할 수 있을 것이다. 또한 이러한 관점에서 차후의 에이전트 기반 역 추적 시스템 개발을 위한 기본 모델로도 활용 가능할 것이다.

본 연구에서의 제약 및 추후 연구 사항은 다음과 같다. 첫째, 본 시스템에서 사용된 학습 데이타 집합은 많은 양을 가지고 있어 모든 데이타에 대한 학습 시 상당히 많은 시간이 소비되었다. 또한 네트워크 전문가들에 의해 인위적으로 구조화된 데이타이므로, 실제 감사 데이타의 경우 그 크기가 훨씬 방대해 질 것이며, 모든 데이타에 따른 레이블링 작업은 학습을 위한 또 다른 부담을 제공할 것이다. 그러므로 본 논문에서 제안한 SOM 기반의 학습보다 더욱 효율이 높은 개선된 알고리즘에 대한 연구가 병행되어야 할 것이다. 둘째, 보다 많은 공격 자동화 도구들을 분석함으로써 이들이 가지는 내부적 패턴의 발견이 필요할 것이다. 이를 기반으로 각 특징에 주어진 문제 영역을 해결할 수 있는 에이전트의 절차적 지식을 확장하는 연구가 필요할 것이다.

참 고 문 헌

[1] Symantec Internet Security Threat Report, Vol. 3, 2002, available at http://www.symantec.com/region/hu/huresc/download/2003_02_03SymantecInternetSecurityThreatReport.pdf.

[2] Korea Computer Emergency Response Team Coordination Center, 2004. available at http://www.-certcc.or.kr/statistics/hack/hack.htm.

[3] CERT Coordination Center, "Overview of Attack Trends," Carnegie Mellon University, 2002. available at http://www.cert.org/archive/pdf/attacktrends.pdf.

[4] Allen, J. H., "CERT Guide to System and Network Security Practices," Addison-Wesley, 2001.

[5] Savage, S., Wetherall, D., Karlin, A. and Anderson, T., "Practical Network Support for IP Traceback," In Proc. of the 2000 ACM SIGCOMM Conference, pp 295-306, 2000.

[6] Song, D. X. and Perrig, A. "Advanced and Authenticated Marking Schemes for IP Traceback," In Proceedings of the IEEE Infocomm 2001, April 2001.

[7] Bellovin, S., Leech, M. and Taylor, T. "ICMP Traceback Messages," 2001. available at http://www.cse.ogi.-edu/~wuchang/cse581_winter2002/papers/draft-ietf-itrace-01.txt.

[8] Kohonen, T., Self-Organizing Maps, Springer-Verlag, Berlin, 1995.

[9] KddCup'99. available at http://kdd.ics.uci.edu/ databases-

/kddcup99/kddcup99.html
[10] Franklin, S. and Graesser, A., "Is it an agent, or just program?: A taxonomy for autonomous agents," Proc. Of the Third International Workshop on Agent Theories, Architectures, and Language, 1996.
[11] Wooldridge, M. and Jennings, N. "Intelligent Agents," Lecture Notes in Artificial Intelligence #890, Springer-Verlag, 1995.
[12] Nwana, H. S., "Software agents: An Overview," The Knowledge Engineering Review, 11(3), pp 205-244, 1996.
[13] Chalupsky, H., Finin, T., Fritzson, R. McKay, D., Shapiro, S. and Weiderhold, G. "An overview of KQML: A knowledge query and manipulation language," Technical report, KQML Advisory Group, April 1992. available at http://www.csee.-umbc.edu/kqml/papers/kqmloverview.ps.
[14] Finin, T., Fritzson, R., McKay, D. and McEntire, R. "KQML as an agent communication language," Proc. of CIKM'94, pp. 126-130, 1994.
[15] Petrie, C., "JATLite," Online Documentation, 1998. available at http://java.stanford.edu/.

최 진 우
1998년 한성대학교 전산학과 졸업(학사) 2000년 국민대학교 대학원 전산과학과 졸업(석사). 2004 국민대학교 대학원 전산과학과 전산과학과 졸업(박사). 2004년~현재 한국 과학 기술원 CAD/CAM 실, 지능형 로봇팀, POST-DOC. 관심분야는 인공지능, ITS, 에이전트, 정보 보호

우 종 우
1978년 서울대학교 농생물학과 졸업(학사). 1983년 Minnesota State University at Mankato 전산학과 졸업(석사). 1991년 Illinois Institute of Technology 전산학과 졸업(박사). 1994년~현재 국민대학 컴퓨터학부 교수. 관심분야는 인공지능, 에이전트, 정보보호

박 재 우
1999년 경북대학교 무기재료공학과(학사) 2001년 경북대학교 컴퓨터과학과(석사) 200년~현재 국가보안기술연구소 선임연구원