분산 침입 탐지 에이전트를 기반으로 한 지능형 침입탐지시스템 설계

이 종 성† · 채 수 환††

요 약

컴퓨터망의 확대 및 컴퓨터 이용의 급격한 증가에 따른 부작용으로 컴퓨터 보안 문제가 중요하게 대두되고 있다. 이런 추세에 따라 해커들로부터 침입을 줄이기 위한 침입탐지시스템에 관한 연구가 활발하다. 본 논문에서는 침입탐지시스템에 대해 고찰하고, 분산 침입 탐지 에이전트를 기반으로 한 새로운 침입탐지 시스템의 모델을 제시한다. 제안된 모델은 분산된 탐지 에이전트로부터 침입정보를 동적으로 수집하고 침입 시나리오 데이터 베이스에 의해 탐지 에이전트를 학습시키고, 학습된 탐지자 코드를 탐지자들에게 전달한다. 이렇게 함으로써 기존 분산에이전트 시스템에서 문제가 되고 있는 침입패턴 학습문제를 효과적으로 해결할 수 있다.

Design of Intelligent Intrusion Detection System Based on Distributed Intrusion Detecting Agents : DABIDS

Jong-Sung Lee† · Soo-Hoan Chae††

ABSTRACT

Rapid expansion of network and increment of computer system access cause computer security to be an important issue. Hence, the researches in intrusion detection system(IDS) are active to reduce the risk from hackers. Considering IDS, we propose a new IDS model(DABIDS : Distributed Agent Based Intelligent intrusion Detection System) based on distributed intrusion detecting agents. The DABIDS dynamically collects intrusion behavior knowledge from each agents when some doubtable behaviors of users are detected and make new agents codes using intrusion scenario data base, and broadcast the detector codes to the distributed intrusion detecting agent of all node. This will improve intrusion detection capabilities of detecting agents. The DABIDS can efficiently solve the problem to reduce the overhead for training detecting agent for intrusion behavior patterns.

1. 서 론

컴퓨터 및 네트워크 기술이 발전함에 따라 컴퓨터 간의 상호 연결성이 증가되고 이로 인해 컴퓨터 보안 문제가 중요하게 대두되었다. 1981년에 IP 프로토콜을 실험적으로 ARPANET에 적용할 때, 불과 210개의 호스트가 ARPANET에 연결되어 있었으나, 오늘날 전 세계적으로 7백만 대 이상의 컴퓨터 시스템이 동일한 IP 프로토콜을 사용하여 상호 연결됨에 따라 네트워크에 연결된 모든 컴퓨터가 해커들의 침입대상이 되고 있다.

이러한 위협에 대처하기 위해 정보보호를 필요로 하는 문서나 시스템에 대한 불법 침입을 분석하고 탐

† 준 회 원 : 한국항공대학교 대학원 컴퓨터공학과
†† 정 회 원 : 한국항공대학교 컴퓨터공학과 교수
논문접수 : 1998년 11월 16일, 심사완료 : 1999년 3월 4일

지하여 문제점을 사전에 방지하는 감사 기술의 발전적 형태인 침입 탐지 시스템(intrusion detection system)에 관한 연구가 활발히 진행되고 있다[1,2,3].

침입 탐지 시스템은 불법적인 침입으로부터 컴퓨터를 보호하기 위해 침입을 탐지하고, 이에 대한 적절한 조치를 취하는 역할을 수행한다. 일반적으로 시스템의 안전성과 사용 편리성은 서로 상반되는 개념이고, 안전한 시스템 설계는 엄청난 비용이 소요되므로 어떠한 공격에 대해서도 안전한 이상적인 시스템을 설계하는 것은 거의 불가능하다. 이와 같은 시스템에서 불법적 행위에 대한 대처 방법으로 모든 파일을 암호화하여 저장할 수 있지만 여기에는 암호 알고리즘 선정, 키 관리 문제, 시스템 관리자의 역할 조정 등의 새로운 문제를 발생시킨다.

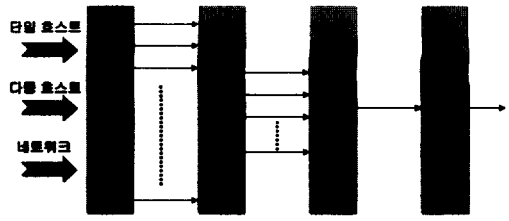
이와 같은 이유로 인해 불안전한 컴퓨터 시스템에 대한 침입 탐지를 수행하는 침입탐지시스템이 요구되며, 침입탐지시스템은 감사 추적(audit trail)을 위해서 사용자에 의해 발생되는 각 사건을 기록하고, 필요시 언제, 누가, 어떤 일을 수행했는지 추적할 수 있어야 한다. 일반적으로 불법 침입을 예방하거나 침입 발생시 그 사실을 탐지하여 손실을 최소화하기 위해서는 시스템 내의 모든 활동들을 면밀히 조사·분석해야 한다. 그러나 시스템 내에서 발생하는 로그 데이터는 유닉스의 경우 시간 당 수 메가바이트로 생성되므로 수작업에 의한 자료의 수집 및 분석은 불가능하며 자동화된 추적 방법이 필수적이다. 자동화된 감사 추적 기법의 발전적 형태인 침입 탐지 시스템에서도 방대한 양의 감사 자료를 필터링 등의 방법으로 축소하여 자료의 저장 및 분석에 따른 오버 헤드를 최소화시킬 필요가 있다.

이와 같은 침입 탐지 서비스의 요구에 따라 최근에 다양한 기법과 모델들이 개발되어 왔으나, 컴퓨터 통신망의 복잡성, 대상 시스템의 원초적 취약성, 정보 보호에 대한 이해 부족 및 새로운 불법 침입 기법의 개발 등으로 기존의 어떤 기법 또는 모델도 완전하지 못한 실정이다.

본 논문은 침입탐지시스템에 관해 고찰하고, 분산 탐지 에이전트를 기반으로 하여 분산된 각각의 탐지 에이전트들로부터 해커의 침입 정보를 동적으로 수집하여 이를 통해 탐지 에이전트들에게 침입 패턴을 학습시켜 학습된 탐지자 코드를 분배하여 침입을 탐지하는 분산 탐지 에이전트를 기반으로 한 새로운 개념의 지능형 침입탐지시스템을 제안한다.

2. 침입탐지 시스템 기술의 배경

침입 탐지 시스템은 크게 데이터의 소스(source)를 기반으로 하는 분류 방법과 침입의 모델을 기반으로 하는 분류 방법으로 나눌 수 있으며, 데이터 소스를 기반으로 하는 분류 방법은 단일 호스트로부터 생성되고 모아진 감사(audit) 데이터를 침입 탐지에 사용하는 단일호스트 기반(host based)과, 여러 호스트들로부터 생성되고 모아진 감사 데이터를 침입 탐지에 사용하는 다중호스트 기반(multihost based), 그리고 네트워크의 패킷 데이터를 모아 침입을 탐지하는데 사용하는 네트워크 기반(network based)으로 구분할 수 있다. 또한 침입 모델을 기반으로 하는 침입탐지시스템의 일반적인 분류 방법은 정상적인 시스템 사용에 관한 프로파일과 시스템 상태를 유지하고 있는 동안 이 프로파일에서 벗어나는 행위들을 탐지하는 비정상적인 행위 탐지(anomaly detection) 방법과, 시스템의 알려진 취약점들을 이용한 공격 행위들에 대한 공격 특징 정보를 통해 침입을 탐지하는 오용 침입탐지(misuse detection) 방법으로 분류할 수 있다[4].



(그림 1) 침입탐지 시스템의 기술적 구성요소

한편, 침입탐지 시스템은 (그림 1)과 같이 크게 정보수집 단계, 정보의 가공 및 축약 단계, 침입 분석 및 탐지 단계, 그리고 보고 및 조치 단계의 4 단계 구성요소를 갖으며, 정보수집(raw data collection) 단계는 침입탐지시스템이 대상 시스템에서 제공하는 시스템 사용 내역, 컴퓨터 통신에 사용되는 패킷 등과 같은 정보들을 수집하는 감사 데이터(audit data) 수집 단계로서 단일 호스트 기반에서는 호스트의 사용내역이 기록되어지는 자체의 로그 파일이 있으므로 이 파일들로부터 관련 정보들을 모으며, 다중호스트 기반에서는 여러

호스트의 로그 정보를 호스트간의 통신을 통해 취합한다. 수집된 일련의 감사 데이터들은 정보 가공 및 축약(data reduction and filtering) 단계에서 침입 판정이 가능할 수 있도록 의미 있는 정보로 전환시키며, 분석 및 침입탐지 단계에서 이를 분석하여 침입 여부를 판정한다. 이 단계는 침입탐지 시스템의 핵심 단계이며, 시스템의 비정상적인 사용에 대한 탐지를 목적으로 하는지, 시스템의 취약점이나 응용 프로그램의 버그를 이용한 침입 탐지를 목적으로 하는지에 따라 비정상적 행위와 오용 탐지 기술로 나뉘어진다. 이러한 목적을 가지고 접근하는 기술적 방법에 따라 여러 가지 기술들이 현재까지 적용 및 시도되어지고 있다[5~11]. 보고 및 조치(reporting and response) 단계에서는 침입탐지 시스템이 침입 여부를 판정하여 침입으로 판단된 경우, 보안관리자에게 침입 사실을 보고하며 보안관리자는 이에 대하여 적절한 조치를 취한다[4].

일반적인 침입탐지시스템의 중요 요구사항은 시스템 관리자 없이도 지속적으로 수행되어야 하며, 침입탐지시스템 자신이 파괴되지 않도록 자기 자신을 모니터링 해야하고, 컴퓨터 시스템에 최소한의 오버헤드를 부과해야 하며, 새로운 침입 유형의 변화에 대한 자체 학습 기능, 결함 허용, 시스템 환경 변경시 유지 및 관리와, 시스템의 정상상태를 침입이라고 탐지하는 긍정적 결함(false positive) 및 시스템의 침입상태를 정상상태로 판단하는 부정적 결함(false negative)과 같은 잘못된 침입 탐지를 방지해야 한다[3,11,12,13].

3. 관련연구

침입탐지시스템 연구는 1987년에 Dorothy E. Denning이 제시한 침입 탐지 모델[5]를 바탕으로 하여 현재 미국을 중심으로 활발하게 진행되고 있으며, 대표적인 연구 그룹, 또는 프로젝트를 살펴보면 다음과 같다[14].

- University of Southern California의 CRISIS
- SRI 인터내셔널의 NIDES
- 카네기멜론대학(CMU)의 INVICTUS
- 보잉연구소의 Dynamic Cooperating Boundary Controllers
- University of California(UC)-Davis의 Security Lab에서 진행되는 프로젝트
- George Mason University의 CSIS에서 진행되는 프로젝트

또한, 현재까지 알려져 있는 대표적인 침입탐지 시스템들을 침입모델 및 데이터 소스에 따라 분류하고, 어떤 방법으로 침입을 탐지하며 이를 위한 학습능력 존재유무와 자치적 학습능력 존재 유무에 따라 다시 분류하면 <표 1>과 같다[4,11,12,13,17].

〈표 1〉 침입탐지 시스템 비교

침입 탐지 시스템 종류	침입모델		데이터 소스			탐지 방법			학습능력	
	비정상 행위 탐지	오용 침입 탐지	호스트 기반	다중호스트 기반	네트워크 기반	패턴매칭	규칙기반	상태변이	존재	자치적
ASAX		●		●			●			
CMDS	●	●		●			●			
AID		●		●			●			
NADIR	●	●			●		●			
DIDS				●			●			
EMERALD		●			●		●			
IDES/NIDES	●	●		●			●			
OmniGuard		●	●				●			
MIDAS	●	●	●				●			
W&S	●						●			
GASSATA		●				●				
IDIOT		●				●				
STAT		●					●	●		
COAST	●	●	●						●	

그러나, 이와 같은 시스템들은 하나의 침입탐지프로세스를 통해 침입 탐지를 수행함에 따라 프로세스의 크기가 너무 커서 프로세스 한 부분의 결함이 전체 시스템의 성능을 떨어뜨리는 문제를 야기한다(단, COAST는 다중 에이전트를 사용). 또한 자체 학습 기능이 없는 정적인 시스템이기 때문에 시스템 환경의 변화와 새로운 기능의 추가나 삭제 등 침입탐지시스템의 유지 및 보수에 대한 비용이 매우 크며, 침입탐지시스템의 중요 요구사항인 새로운 침입 유형의 변화에 대한 자체 학습 기능, 결함 허용, 시스템 환경 변경시 유지 및 관리에 적합하지 못하다[12,13].

이러한 한계점을 해결하기 위한 접근으로서 동물의 면역시스템을 모델링한 컴퓨터 면역시스템에 관한 연구[15,16]가 New Mexico 대학에서 진행되고 있으며, 또 다른 접근으로 독립 에이전트를 이용하는 방법이 Purdue 대학의 COAST 연구실에서 진행되고 있다[11, 12,13].

그러나, 컴퓨터 면역시스템에 관한 연구는 현재 시작단계이므로 구체적인 침입탐지모델이 제시되지 않은 상태이고, 독립 에이전트를 이용하는 침입탐지시스템은 에이전트 학습을 운영자가 개입한 피드백(feedback) 방식으로 수행하므로 자치적인 학습기능이 없어 오랜 시간동안 학습이 요구되며, 특히 에이전트 학습에 필요한 침입 시나리오를 운영자가 작성해야 하는 어려운 문제를 안고 있다.

4. 제안한 분산 침입 탐지 에이전트를 기반으로 한 지능형 침입탐지시스템(Distributed Agent Base Intelligent intrusion Detection System : DABIDS)

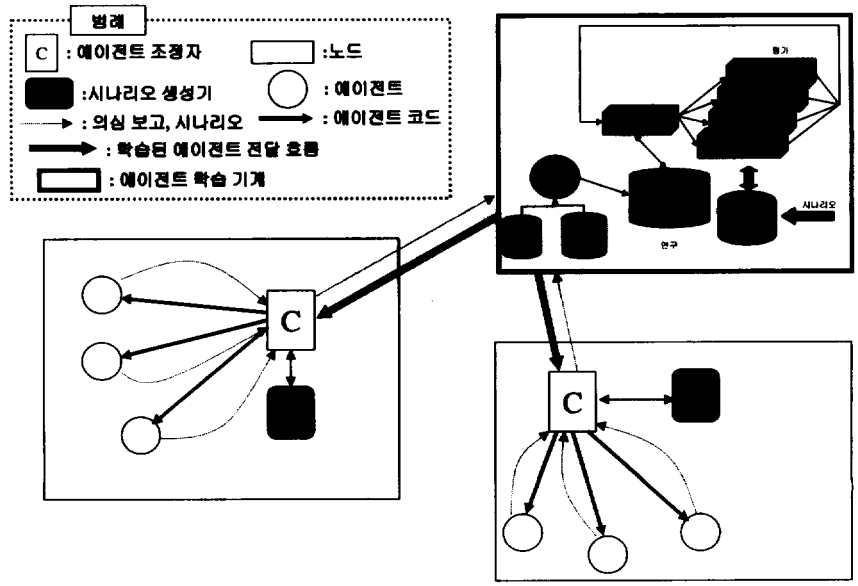
DABIDS는 (그림 2)와 같이 분산된 각각의 탐지 에이전트들로부터 해커의 침입 정보를 동적으로 수집하여 이를 통해 에이전트들에 침입 패턴을 학습시켜 학습된 탐지자 코드를 분배하여 침입 발생시 발생된 침입에 가장 적합한 탐지자를 통해 침입에 대응하게 한다.

DABIDS를 구성하는 각각의 구성요소를 살펴보면 다음과 같다.

4.1 탐지자(Detector)

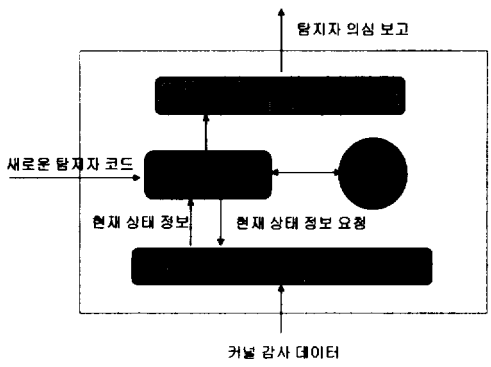
DABIDS의 탐지자는 (그림 3)과 같이 구성되어 IDS 엔진(IDS Engine)에 의해 탐지자 코드에 따라 시스템에서 제공하는 커널 감사 데이터를 얻어 이를 통해 의심 정도를 파악하여 탐지자 조정자에게 전달한다.

이때, 시스템 상태 정보 추출기는 탐지자와 침입탐지 대상 시스템간에 투명성을 제공하며, 평균적인



(그림 2) 분산 침입탐지 에이전트를 기반으로 한 침입탐지시스템(DABIDS)의 구조

CPU 사용시간, 어떤 사용자의 평균적인 로그인 시도 회수, IP주소, 특정포트에 대한 접근시도 간격 등에 대한 탐지자가 요청하는 시스템 현재 상태 정보를 커널 감사 데이터(kernel audit data)로부터 추출하는 역할을 수행한다.



(그림 3) 탐지자 내부 구조

탐지자 코드는 IDS 엔진에 의해 수행되며, 탐지자 코드를 변경하면 탐지자는 시스템 상태 정보 추출기를 통해 시스템의 다른 상태 정보를 요청하여 이를 통해 다른 각도로 침입을 탐지하게 된다. 따라서, 탐지자 코드는 탐지자의 침입 탐지 성능을 좌우하는 중요한 요소이다.

한편, 탐지자 의심 보고기는 IDS 엔진에 의해 탐지자 코드를 수행한 결과 발생하는 탐지자의 침입 판단에 따라 의심 사항을 수집하여 탐지자가 최종적으로 침입이라고 판단되는 내용을 탐지자 조정자(detector coordinator)에 전달한다. 이때, 탐지자 의심 보고기는 침입탐지대상시스템에서 일련의 명령이 수행된 결과 침입이라고 판단되지 않지만 시스템이 비정상상태인 경우 수행된 일련의 명령을 추후 에이전트 학습을 위한 시나리오로 사용하기 위해 관리하고 시나리오 생성기(scenario generator)에 제공한다.

탐지자에 대한 클래스 및 멤버 함수는 다음과 같다.

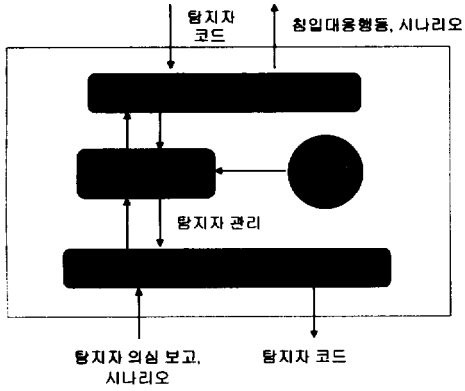
```
Class SIP_Extractor{
int RecvRequest(int PrimitiveID);
   //프리미티브에 대한 시스템정보 요청 수신
int GetSIP(int PrimitiveID);
      //프리미티브에 대한 시스템정보 추출
   void SendSIP(); //시스템정보 전송
}

Class IDS_Engine{
RecvDCodeFromCoordinator();
   //갱신된 탐지자코드 수신
CountCondition();
   //탐지자 코드 중 조건문 개수 카운트
int InferenceDetecterCode();
   //탐지자 코드를 읽으며 요청할 프리미티브 선택
int RequestSIP(int PrimitiveID);
   //선택된 프리미티브를 요청하여 시스템 상태정보
     획득
SendSIP();
   //획득된 시스템 상태정보를 탐지자 의심 보고기로
     전달
}

Class Reporter{
RecvResult(); //탐지자 수행결과 수신
Define_Semi_Intrusion();
   //총 조건에 대해 참인 개수가 일정 범위 내에 존
     재하면 이를 침입 시나리오로 정의
SendSuspension();
   //만약 침입이면 의심 카운터 값을 하나 증가하여
     탐지자 조정자에 전달
Send_Semi_Intrusion_Result();
   //탐지자 수행 결과 리스트를 시나리오 생성기로 전달
SendDC_state(); //탐지자 상태 정보 전송
}
```

탐지자 의심 보고기는 탐지자 코드에서 조건문에 존재하는 총 조건에 대해 참(true)인 개수가 일정 범위 내에 존재하면 이를 침입 시나리오로 정의하는데, 범위를 낮추면 탐지자는 자주 수행 결과를 시나리오 생성기로 전달하므로 전체 통신 오버헤드가 증가하고, 반면에 범위를 높이면 침입 시나리오가 종래에 알려진 침입과 같은 내용이 되는 문제가 발생된다.

한편, 탐지자 의심 보고기는 침입 시나리오를 정의할 때 탐지자 코드를 구성하는 총 조건문에 대해 참인 조건의 개수 비율을 침입확률로 간주하여 탐지자 조정자를 통해 시나리오 생성기에 전달한다. 즉, 탐지자에 구성된 탐지자 코드에 존재하는 전체 총 조건문이 10개이고, 조건이 참인 개수가 8개이면 침입확률은 80%가 되고, 8개가 참이 되는 과정이 새로운 침입 시나리오가 된다. 추후, 탐지자 학습기는 80% 침입확률을 갖는 시나리오를 통해 탐지 에이전트를 학습시킨다.

4.2 탐지자 조정자(Detectors Coordinator)

탐지자 조정자는 (그림 4)에 도시된 바와 같이 각각의 탐지자들로부터 전달되는 침입 의심 정도를 수신하여 현재 컴퓨터 노드의 침입 여부를 판단하여 보안 관리자에게 침입 경고를 알리거나 침입에 대한 적절한 조치를 취하는 침입대응행동을 자동으로 수행시킨다. 한편, 탐지자 조정자는 탐지자 조정자 엔진에 의해 행동 지식(behavior knowledge)에 따라 시나리오 생성기(scenario generator)에 의해 생성된 학습 시나리오를 상위 수준 통신부를 통해 탐지자 학습기로 전달하고, 탐지자 학습기로부터 전달되는 학습된 탐지자 코드를 수신하여 이를 하위 수준 통신부를 통해 탐지자에 전달하고 탐지자들을 관리하는 기능을 수행한다.



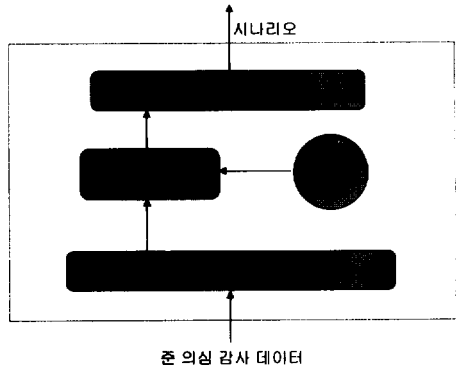
(그림 4) 탐지자 조정자 내부 구조

탐지자 조정자를 구성하는 클래스 및 멤버 함수 중 탐지자 조정자 엔진에 대한 클래스 및 멤버함수는 다음과 같다.

```
Class DT_Coord_Engine{
RecvSuspension(); //탐지자 의심 보고를 수신
RecvScenario();
   //시나리오 생성기로부터 생성된 시나리오 수신
SendDConde(int DT_ID);
   //탐지자 코드를 해당되는 탐지자에게 전달
RequestDC_state(); //탐지자 상태 정보 요청
Calculate_Node_state(); //노드 침입 여부 판단
RecvDC(); //탐지자 코드 수신
ReactionInstruc();
//침입을 판단한 경우 행동지식에 따라 침입에 대
  응 지시
}
```

4.3 시나리오 생성기(Scenario Generator)

시나리오는 그 시나리오가 침입일 확률을 나타내는 확률 값을 갖는데, 예를 들어 "예약된 포트에 접속하고, 접근간격이 1초이면 침입일 확률은 90%"라고 하고, 정상적인 시스템 동작에 관한 시나리오인 경우 침입 확률을 낮춘다. 이와 같은 시나리오를 이용한 학습은 COAST 프로젝트에서 제안[11,12,13]되었으나, COAST 프로젝트에서는 시나리오 제작을 보안 관리자에 의해 작성되어 학습을 시키는 문제점이 있었다.



(그림 5) 시나리오 생성기 내부 구조

따라서, 제시된 DABIDS에서는 탐지 에이전트들이 탐지자 조정자에게 현재 시스템의 상황에 대한 의심 정도를 보고하는데 일정 임계치(총 조건문 중에서 만족되는 개수) 이상이면 침입을 의심함을 보고하고, 어느 이하인 경우 정상 상태라고 판정하는데, 만일 그 중간 정도이면 이를 준 의심상태로 하여 이에 대한 감사 데이터를 획득하여 시나리오를 생성한다. 일반적으로 컴퓨터 시스템에 대한 침입은 컴퓨터 시스템이 정상동작상태에서 다소 벗어남을 의미하며 고수준 침입은 그 벗어나는 정도가 적기 때문에 전술한 준 의심 상태에 대한 시나리오를 작성하여 탐지에이전트를 학습시키면 그만큼 탐지 에이전트는 새로운 침입에 더욱 강해질 수 있다.

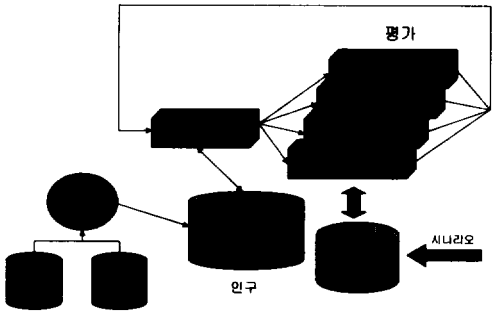
이처럼 각각의 노드의 시나리오 생성기는 (그림 5)에 도시된 바와 같이 시나리오 생성기 엔진에 의해 행동지식에 따라 정상 상태 및 비정상 상태의 침입 정보(감사 데이터)를 준 의심 감사데이터 수집기를 통해 수집하여 정상 상태 동작과 비정상 상태 동작 작동 순서를 가상적으로 생성하여 시나리오 전송기를 통해 탐지자 학습기에 전달한다.

시나리오 생성기를 구성하는 클래스 및 멤버 함수 중 시나리오 생성기 엔진에 대한 클래스 및 멤버함수는 다음과 같다.

```
Class ScenarioGeneratorEngine{
Recv_Semi_Intrusion_Result();
   //탐지자 수행 결과 리스트를 수신
Generate_Scenario();
   //수신된 탐지자 수행 결과를 통해 시나리오 생성
     즉, 참인 조건문의 집합을 새로운 시나리오로 간주
SendScenario();
   //생성된 시나리오를 탐지자 학습기로 전송
}
```

4.4 탐지자 학습기(Detector Trainer)

DABIDS의 탐지자 학습기를 (그림 6)을 참조하여 주요 구성요소를 중심으로 살펴보면 다음과 같다.



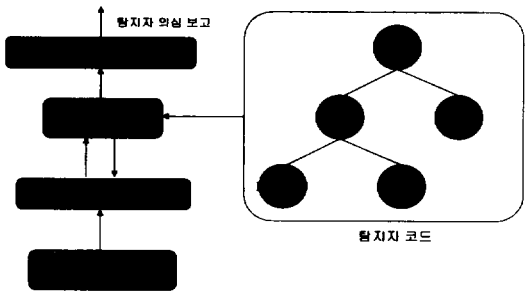
(그림 6) 탐지자 학습기

● 학습 시나리오 데이터베이스

각각의 노드에서 발생하는 정상 상태 및 비정상 상태의 침입 정보를 저장 관리하는 모듈로서, 예를 들어 "예약된 포트에 접속하고, 접근간격이 1초이면 침입일 확률은 90%"라는 시나리오를 시나리오 생성기로부터 수신하여 저장한다.

● 연산자 집합

유전자 프로그래밍(genetic programming)에서는 함수 집합(function set)이라고 칭하며, 탐지 에이전트가 수행하는 연산자들(이를테면, 산술, 조건, 논리 연산자)을 의미하며 (그림 7)의 경우 "IF", "IP-NEQ"가 이에 속한다.



(그림 7) 분석 트리로 구성된 탐지자 코드 예

● 프리미티브 집합(a set of primitive)

유전자 프로그래밍에서는 터미널 집합(terminal set)이라고 칭하며, 탐지자가 시스템 상태 정보 추출기를 통해 커널 감사 데이터로부터 획득한 상수들을 의미하며, (그림 7)의 경우 "IP-DEST", "MY-IP", "RAISE"가 이에 속하며, "IP-DEST"는 현재 시스템에 입력된 패킷의 목적지 주소를 요청하는 프리미티브이고, "MY-IP"는 현재 시스템 자신의 IP 주소를 요청하는 프리미티브이며, "RAISE"는 침입 의심을 알리라는 프리미티브이다. 따라서 (그림 7)을 통해 예시한 분석 트리는 시스템에 입력되는 모든 패킷들에 대한 IP주소와 시스템의 IP 주소를 비교하여 일치하지 않으면 의심을 알리는 분석 트리이다.

제안된 시스템의 프로토타입에 적용된 탐지자들에 해당되는 연산자 집합과 프리미티브 집합은 다음과 같다.

● I/O에 관련된 탐지자
연산자 집합 : IF, I/O-Interval-EQ, User-Permit
프리미티브 집합 : I/O-Interval, 상수, User, RAISE
● NFS에 관련된 탐지자
연산자 집합 : IF, IsREAD, IsWRITE, User-Permit
프리미티브 집합 : ThisOperation, User, RAISE
● TCP/IP에 관련된 탐지자
연산자 집합 : IF, IP-NEQ
프리미티브 집합 : IP-DEST, MY-IP, RAISE
● 파일에 관련된 탐지자
연산집합 : IF, IsFileDuplicate, IsFileMove, IsFile Remove, IsFileCreate,
프리미티브 집합 : owner, object, source, permit, RAISE

● 분석 트리 생성기(parse tree generator)

유전자 프로그래밍에서 함수 집합(function set)과 터미널 집합(terminal set)을 결합하여 해답 프로그램(solution program)을 생성하는 것과 동일하게 연산자 집합과 프리미티브 집합의 모든 요소들에 대해 (그림 7)과 같이 트리 구조의 결합 관계를 형성한다.

분석 트리 생성기를 구성하는 클래스 및 멤버 함수는 다음과 같다.

```
Class ParseTreeGenerator{
GetFunction();
  //특정 탐지 영역에 대한 연산자집합으로부터 연
    산자 획득
GetPrimitive();
  //특정 탐지 영역에 대한 프리미티브집합으로부터
    프리미티브 획득
Concatenate(); //분석 트리를 생성
}
```

● 유전자 풀(gene pool)

분석 트리 생성기에 의해 생성된 해당 문제영역에서 가질 수 있는 모든 경우에 대한 분석 트리를 저장한다.

● 학습 엔진(training engine)

탐지 에이전트들에 학습 시나리오 데이터베이스에 저장된 정상 상태 및 비정상 상태의 침입 정보를 제공하고, 각각의 에이전트는 유전자 풀에 저장된 특정 탐지 영역별 모든 분석 트리(즉, 탐지코드)에 따라 제공된 시나리오를 수행하여, 가장 적합한 탐지코드를 갖는 탐지 에이전트를 획득한다.

학습엔진에 대한 클래스 및 멤버 함수는 다음과 같다.

```
Class TrainerEngine{
GetParseTree();
//특정 탐지 영역(이를테면, I/O)에 대한 모든 분
  석 트리를 하나씩 획득
GetScenario(); //정리된 시나리오를 입력
RunAgent();  //시나리오 대한 에이전트 수행
CalculateFittness(); //정확도 체크
SelectGoodAgent();
  //특정 탐지자 영역(이를테면, I/O)에서 가장 정확
    한 탐지자를 선택
    모든 탐지자(I/O, NFS, TCP, 파일 등)들에 대해
    학습을 수행한 후, 학습된 모든 탐지자 코드를
    다시 다운로딩
}
```

가장 적합한 탐지코드를 찾기 위한 정확도는 아래 공식에 따라 계산되는데,

$$\text{정확도} = \frac{100}{|\text{침입확률} - \text{의심출력확률}|}$$

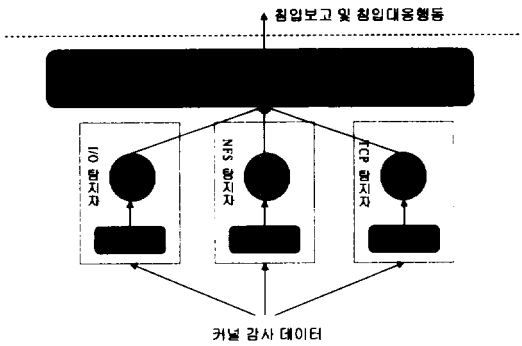
단, 침입확률과 의심출력확률이 같은 경우 정확도를 100으로 함

그 의미는 여러 개의 분석 트리를 갖는 탐지 에이전트에 의해 일정한 침입 확률을 갖는 시나리오를 수행한 결과, 시나리오의 침입확률과 의심을 출력할 확률의 차이를 의미한다.

예를 들어, 탐지자 에이전트가 여러 개의 분석 트리에 따라 침입일 확률이 90%인 "예약된 포트에 접속하고, 접근간격이 1초"라는 시나리오를 수행한 결과, 의심 출력이 100%, 50%, 90%이면, 정확도는 각각 10, 25, 100이 되고, 정확도가 높은 분석 트리를 그 시나리오에 가장 적합한 분석 트리 즉 탐지자 코드로 하여 추후에 탐지자들에 전달함으로 탐지자를 학습시킨다.

DABIDS의 주요 구성요소들 간의 작용을 살펴보면 다음과 같다.

먼저, 각각의 탐지자들은 각각의 노드에서 발생하는 정상 상태 및 비정상 상태의 침입 정보를 시나리오발생기에 의해 발생하여 (그림 6)에 도시된 탐지자 학습기에 전달하여 학습 시나리오 데이터베이스에 저장한 후 해당되는 문제 영역에 포함된 연산자 집합과 프리미티브 집합으로부터 유전자 프로그래밍 기법[18]에 의해 (그림 7)과 같은 분석 트리를 생성한 후, 에이전트들이 생성된 분석 트리를 이용하여 오퍼레이션 수행 후 정확도를 체크하고 이와 같은 과정을 유전자 연산(crossover, mutation)을 적용하며 여러 세대 수행하여 가장 적합한 분석 트리를 구한 후 각 노드들에 구한 분석 트리 코드를 담은 탐지자 코드를 노드들에 전송하여 탐지자를 갱신시켜 전체 침입탐지시스템의 성능을 향상시킨다.



(그림 8) 하나의 노드에 대한 DABIDS 예

한편, (그림 8)에 도시된 바와 같이 각 노드에는 여러 개의 학습된 탐지자들이 존재하며, (그림 8)과 같이 시스템 상태 정보 추출기를 통해 시스템 커널로부터 입력되는 감사 데이터(audit data)로부터 I/O 탐지자는 시스템의 I/O에 관련된 상태를 체크하고, NFS 탐지자는 NFS에 대한 요청 및 동작에 대해 체크하며, TCP 탐지자는 TCP 프로토콜을 통해 시스템에 접속하는 것을 체크한다. 이를 통해 탐지자 조정자는 현재 시스템의 침입 여부를 판단하여 시스템 관리자에게 알려준다.

5. 결론 및 향후 연구과제

제안한 DABIDS는 종래의 에이전트 기반 침입탐지 시스템에서 에이전트 학습을 사람이 개입한 피드백 방식으로 수행하므로 오랜 시간동안의 학습이 요구되는 문제를 극복할 수 있으며, 시나리오 작성의 문제를 각 노드에서 현재 비정상적으로 작동하는 패턴정보를 바탕으로 시나리오를 작성하여 탐지 에이전트 학습기에 전달하여 에이전트를 학습시키므로 인위적인 시나리오 개발에 드리는 노력을 줄일 수 있다.

향후 연구 과제는 DABIDS을 특정 시스템에 적용하고, 타당한 IDS 성능 평가 기준을 설정하여 그 기준에 따라 성능을 평가하는 것이 필요하다.

참 고 문 헌

[1] James Cannady, Jay Harrell, "A Comparative Analysis of Current Intrusion Detection Technologies," http://iw.gtri.gatech.edu/Papers/ids_rev.html, 1998.2.

[2] Mansour Esmaili, Rei Safavi-Naini, "Case-Based Reasoning for Intrusion Detection," Computer Security Applications Conference, pp.214-222, 1996.

[3] Jai Sundar B. Spafford E, "Software Agents for Intrusion Detection," Technical Report, Purdue University, Department of Computer Science, 1997.

[4] 은유진, 박정호, "침입탐지 기술 분류 및 기술적 구성요소", 정보보호센터 정보보호 뉴스 1998.7 통권 13호.

[5] Dorothy E. Denning, "An Intrusion Detection Model," IEEE Trans. S.E., 1987. 2.

[6] Teresa F. Lunt, R. Jagannathan, "A Prototype Real-Time Intrusion Detection Expert Systems," 1988, IEEE S&P, 1988. 7.

[7] T.D Garvey, T. F. Lunt, "Model-Based Intrusion Detection," 14th NCSC, 1991.10.

[8] Steven R. Snapp, et al., "A System for Distributed Intrusion Detection," COMPCON Spring'91 New York, IEEE, pp.170-176. 1991.

[9] K.Ilgun, "USTAT : A Real-Time Intrusion Detection System for UNIX," IEEE Computer Society Symposium Research in Security and Privacy, pp.16-28, 1993.

[10] H.Debar. et al., "A Neural Network Component for an Intrusion Detection System," IEEE Computer Society Symposium Research in Security and Privacy, pp.240-250, 1992.

[11] Crosbie M, Spafford E, "Applying Genetic Programming to Intrusion Detection," Technical Report, Purdue University, Department of Computer Science, 1996.

[12] Crosbie M, Spafford E, "Defending a Computer System using Autonomous Agents," Technical Report, Purdue University, Department of Computer Science, 1994.

[13] Crosbie M, Spafford E, "Active Defense of a Computer System using Autonomous Agents," Technical Report, Purdue University, Department of Computer Science, 1995.

[14] 이종성, "미국 대학에서의 컴퓨터 보안 교육·연구

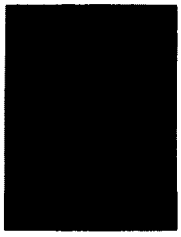
조사", 정보보호센터 보고서, 1997.12.

[15] S. Forrest, S. Hofmeyr, and A. Somayaji, "Computer Immunology", Communications of the ACM, Vol.40, No.10, 1997. pp.88-96.

[16] A. Somayaji, S. Hofmeyr, and S. Forrest, "Principle of a Computer Immune System," New Security Paradigms Workshops, 1997.

[17] 김민수, 노봉남, "퍼지 이론을 이용한 효율적인 침입탐지 방법", 한국통신정보보호학회 종합학술발표회 논문집. Vol.8, No.1, 1998.12.

[18] Thomas Weinbrenner, Adam Fraser, The Genetic Programming Kernel - Version 0.5.2, 1997.



이 종 성

e-mail : jslee@hanul.hangkong.ac.kr

1994년 2월 한국 항공대학교 전자계산학과 졸업(이학사)

1996년 2월 한국 항공대학교 전자계산학과 대학원 졸업(이학석사)

1996년~현재 항공대학교 컴퓨터공학과 대학원 박사과정

관심분야 : Computer Security, Intrusion Detection System, 인공생명, 병렬/분산처리, High Performance Computing 등

채 수 환

e-mail : chae@mail.hangkong.ac.kr

1973년 한국 항공대학교 항공전자공학과 졸업(공학사)

1985년 미국 Univ. of Alabama 전산공학과 졸업(공학석사)

1988년 미국 Univ. of Alabama 전기공학과 졸업(공학박사)

1973년~1977년 공군교육사령부 통신학교 교관

1977년~1983년 금성통신 근무(연구원)

1996년 9월~1997년 8월 영국 Newcastle upon tyne 교환 교수

1998년~1999년 3월 한국항공대학교 컴퓨터신기술연구소장

1989년~현재 한국항공대학교 컴퓨터공학과 정교수

관심분야 : 컴퓨터 구조, 병렬처리시스템, 컴퓨터 보안 등