분산 침입 탐지를 위한 계약망 프로토콜의 적용

An Application of Contract Net Protocol for The Distributed Intrusion Detection

서희석
성균관대학교 정보통신공학부
김희완
삼육대학교 컴퓨터과학과

중심어 : 계약망 프로토콜, 네트워크 보안, 연동

요 약

분산 문제 해결 방법은 문제 해결 능력을 갖는 knowledge -sources(KS's)들이 분산되지만 느슨한 연결을 유지하며 서로 협력하여 문제를 해결하는 수단을 제공한다. 계약망 프로토콜(Contract Net Protocol)은 이러한 분산 문제 해결 분야에서 KS 간의 통신과 제어를 위해 제안된 방법이다. 역할의 분담은 협상 과정에 의해서 결정이 되며 협상의 결과 주어진 역할을 수행하게 된다. 본 논문에서는 분산 침입 탐지 시스템(Distributed Intrusion Detection System)의 침입 성능을 향상시키며, 침입 차단 시스템(firewall)과의 통신을 위해서 계약망 프로토콜을 사용하여 연동하는 방법을 소개한다. IDS와 firewall의 모델을 계층적으로 구성하기 위해서 DEVS (Discrete EVent system Specification) 방법론을 사용하였다. 각 침입 탐지 에이전트는 계약망 프로토콜을 사용하여 침입을 탐지하게 된다. 침입 탐지의 내용은 바로 방화벽에 알려지고 방화벽은 이러한 침입 사실을 바탕으로 유해 트래픽이 네트워크로 유입되는 것을 막는다. 즉 한 침입 탐지 시스템이 침입을 탐지하게 되면 이를 침입 차단 시스템에 알리게 되어 해당 침입 패킷을 차단하게 된다. 이러한 방법을 사용하여 네트워크의 피해를 막게 된다.

Hee-Suk Seo (hisstone@hanmail.net)
School of Info. & Com. Eng, Sungkyunkwan University
Hee-Wan Kim (hwkim@syu.ac.kr),
School of Information Communication, Sahmyook University

Keyword : Contract Net Protocol, Network Security, Coordination

Abstract

Distributed problem solving is the cooperative solution of problem by a decentralized and loosely coupled collection of knowledge-sources (KS's), located in a number of distinct processor nodes. The contract net protocol has been developed to specify problem-solving communication and control for nodes in a distributed problem solver. Task distribution is affected by a negotiation process, a discussion carried on between nodes with tasks to be executed and nodes that may be able to execute tasks. In this paper, we present the coordination method among distributed intrusion detection system and firewall by the contract net protocol. The method enhances the intrusion detection performance and provides the communication methods. To model IDS and firewall, security models have been hierarchically constructed based on the DEVS (Discrete EVent system Specification) formalism. Each ID agent cooperates through the contract net protocol for detecting intrusions. The IDS which detects the intrusion informs to firewall, so the harmful network traffic is blocked. If an agent detects intrusions, the agent transfers attacker's information to a firewall. Using this mechanism attacker's packets detected by IDS can be prevented from damaging the network.

I. 서론

20세기 중반 이후에 세계 각국은 정보 통신 기술의 급속한 발전을 바탕으로 초고속 통신망 구축에 박차를 가했다. 또한 초고속 통신망 구축으로 인해 인터넷은 급속도로 확산되고 발전하게 되었다. 인터넷의 발전은 사용자가 각종 최신의 정보를 수집하고 서로 교환하는 것을 가능하게 하여 업무의 효율을 향상시켰고 생활의 질을 높여 주며 국가 경쟁력을 강화시켜 주는 긍정적인 효과를 거두고 있다. 하지만 그 이면에는 인터넷 확장으로 인한 외부 사용자의 시스템 불법 침입, 중요 정보의 유출 및 변경·훼손·불법적인 사용, 컴퓨터 바이러스 및 서비스 거부 등 역기능이 날로 증대되어 피해 규모가 심

접수번호 ': #030729-002
접수일자 : 2003년 7월 29일, 심사완료일 : 2003년 11월 25일
*교신저자 : 김희완, e-mail : hwkim@syu.ac.kr

각한 수준에 이르고 있다[1]. 이러한 문제를 해결하기 위해 여러 보안 서스템들이 소개되고 있으며 대표적인 시스템으로는 침입 탐지 시스템[2],[3]과 침입 차단 시스템[1],[4] 가상 사설망 (Virtual Private Network)[5] 등이 존재한다.

침입 탐지의 초기에는 네트워크에 단일 침입탐지 시스템을 설치하여 네트워크를 감시하도록 하였다. 단일 침입 탐지 시스템의 사용은 상대적으로 시스템의 부하가 커져서 자체적으로 우수한 컴퓨팅 능력이 요구되며, 단일 침입 탐지 시스템이 한정된 규칙만을 가지고 침입을 탐지하기 때문에 새로운 침입을 탐지하는데 여러 문제점이 발생되었다. 이와 같은 문제점들을 보완하고 탐지에 대한 성능을 향상시키기 위한 방법은 다중 침입 탐지 시스템을 사용하는 것이다[6],[7]. 무엇보다도 다중 침입 탐지 시스템에서는 침입을 탐지하는 에이전트들의 연동이 시스템의 성능을 높이기 위한 중요한 요소로 부각된다. 에이전트에 근거한 연동을 설계하는데 있어서 높은 수행 능력을 성취하기 위해서는 분산된 에이전트에게 효과적인 작업의 할당이 이루어져야 한다. 이러한 이유로 인공 지능과 자동 제어(Automation Control)에 근거한 IC(Intelligent Control)가 필요하게 되었고 인공 지능의 모든 분야의 아이디어와 연구 결과가 문제 영역을 조정하기 위해 적용되었다. IC의 각 부문에서 Expert System Control, Fuzzy Control, Neural Network Control 그리고 Simulating Human Intelligent Control 등과 같은 시스템들이 나타났다[8]. 본 연구에서 침입 탐지 에이전트들을 상호 연동하며 이를 통해 문제를 해결하기 위해 계약망 프로토콜[9]을 적용하였다.

2장에서는 본 연구의 배경이론에 대해서 설명할 것이고 3장에서는 모델링을 수행할 대상 네트워크의 구조에 대해서 설명할 것이다. 4장에서는 침입 탐지 모델을 설명하고 5장에서는 계약망 프로토콜을 사용한 모델들의 연동에 대해서 설명한다. 6장에서는 시뮬레이션 결과에 대해서 설명하고 7장에서 결론을 설명한다.

II. 배경 이론

1. 계약망 프로토콜

계약망 프로토콜은 분산된 문제를 해결하는데 있어 통신하고 조정하기 위한 도구로서 제안되었다[8]. 계약망 프로토콜의 사용은 분산 감지 시스템과 분산 전달 시스템을 위해서 시도되었다[9]. 계약망 프로토콜은 에이전트들이 계약 (contract)에 의하여 분산된 문제를 해결하기 위하여 협상하고 통신하는 메커니즘을 제공한다. 에이전트들은 수행될 필요가 있는 작업을 알리고 다른 에이전트들에 의해 공지된 작업들을 수행하기 위해 bid를 만들고, Command Console은 각 에이전트들이 제출한 bid를 평가하여 계약을 체결하게 된다. 계약망 프로토콜의 적용은 다중 침입 탐지 시스템에 있어서 서로 보완하고 협력하여 탐지의 성능을 향상시키고 정확도를 높일 수 있다.

계약망 프로토콜에서 모든 IDS 모델과 Firewall 모델의 에이전트들을 통제하게 되는 Command Console 모델의 모듈과 각 모듈의 기능은 그림 1과 같다.

Messenger 모델은 메시지의 송수신을 관리하는데 Receiver와 Sender로 구성된다. Receiver는 IDS에서 보낸 메시지를 받고 Sender는 Selector나 Commander에서 만들어진 메시지를 해당 IDS나 모든 IDS에게 unicast, multicast, broadcast의 방법으로 보낸다. Selector 모델은 모든 IDS가 보낸 bid로 내부 네트워크를 감시할 IDS를 선택한다. Commander 모델은 내부 네트워크의 상태에 따라 IDS와 Firewall를 통제하는 메시지를 결정한다.

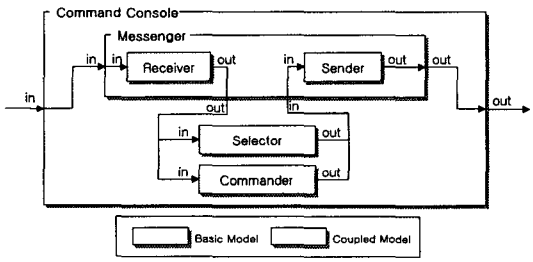

그림 1. Command Console 모델의 구조도

2. DEVS 방법론

$M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, t_a \rangle$
- X : 입력 사건의 집합
- S : 상태들의 집합
- Y : 출력 사건의 집합
- δ_{int} : 내부 상태 변이 함수
- δ_{ext} : 외부 상태 변이 함수
- λ : 출력 함수
- t_a : 시간 갱신 함수

$DN = \langle D, \{M_i\}, \{I_i\}, \{Z_{i,j}\}, select \rangle$
- D : 구성 요소 이름의 집합
- M_i : 구성 모델
- I_i : 모델 i와 연관된 모델의 집합
- $Z_{i,j}$: 모델 i와 j모델간의 연결 함수
- $select$: $tie-breaking\ selection$ 함수

Zeigler에 의해 정립된 DEVS 방법론은 연속적인 시간상에

서 발생되는 이산 사건을 처리하는 시스템을 시뮬레이션 하기 위해 이론적으로 정립된 모델링 방법론이다[10-12]. 이는 모델의 구조와 행동을 시뮬레이션 수행으로부터 추상화시키기 위해 모델을 집합 이론적 방법으로 이용한 것으로, 시스템을 계층적(hierarchical)이고 모듈화(modular)된 형식으로 기술한다. DEVS에서는 기본(Basic) 모델과 결합(Coupled) 모델을 정의한다. 기본 모델은 시스템의 동적인 특성을 표현하기 위한 모델이고, 결합 모델은 시스템의 구성 요소간의 상호작용을 표현하기 위한 모델이다. 이 모델들은 다음의 항들로 명세할 수 있다.

III. 대상 네트워크 구조

그림 2는 본 논문을 위해 구성한 네트워크 구조를 나타낸다. Network Security 모델은 크게 네트워크 장비 모델과 각 호스트 모델로 구성된다. 네트워크에는 3개의 서브 네트워크가 존재하는 경우를 가정하였다.

SES(System Entity Structure)[10],[11]는 시스템의 구조적인 지식을 효과적으로 표현할 수 있는 방법을 제공한다. SES는 분해(decomposition), 분류(taxonomy)와 연결 관계(coupling relationship)가 결합된 지식 표현 방법이다. 각 개체(entity)와 개체와의 관계는 분해와 세분화(specialization)의 관계로 표현된다.

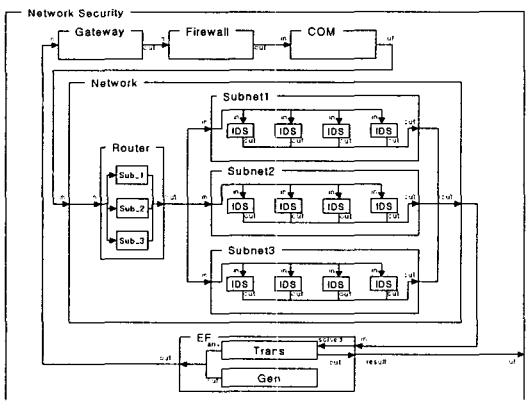

그림 2. 대상 네트워크의 구조도

그림 3, 4는 그림 2의 대상 네트워크를 SES 형식론을 사용하여 표현한 것이다. Network Security 모델은 Gateway, Firewall, Network, COM 그리고 EF 모델로 구성된다. Firewall 모델은 다시 Controller 모델, Inbound_Filter 모델, Outbound_Filter 모델로 구성된다. 이 중에서 Inbound_Filter 모델과 Outbound_Filter 모델은 Protocol, Address, Port 그리고 Data 모델과 specialization 관계를 가진다. 계약망 프로토콜을 사용하기 위해 구성된 COM 모델의 Commander 모델은 적당한 IDS를 선택하기 위한 Selector 모델과 계망망을 사용하기 위해 메시지를 송신하고 수신하기 위해 구성된 Messenger 모델로 구성된다. Messenger 모델은 다시 bid를 송신하는 Sender 모델과 bid를 수신하는 Receiver 모델로 구성된다.

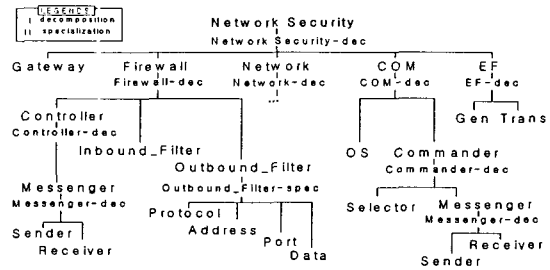

그림 3. Network Security 모델의 SES

Network 모델은 Router, Subnet1, Subnet2 모델과 Subnet3 모델로 구성된다. 각 서브넷 모델은 내부에 여러 시스템(UNIX, Windows NT, Windows 2K, LINUX 등)들을 갖고 있다. 이 시스템들은 침입을 탐지하는 IDS 모델과 운영 체제의 특징을 갖는 OS 모델로 구성된다. IDS 모델은 실제 침입을 탐지하는 모델인 Detector 모델과 입찰 참여에 관여하는 Agent 모델로 구성된다. Agent 모델은 다시 자신이 최적화된 처리를 할 수 있는지를 판단하기 위한 Local_Optimizer 모델과 Messenger 모델, 구성된 Bid를 가지고 입찰에 참여하기 위해 구성된 Bidder 모델로 구성된다. 위의 모델들은 DEVS 방법론[10],[12]에 의해서 구성된 모델이다.

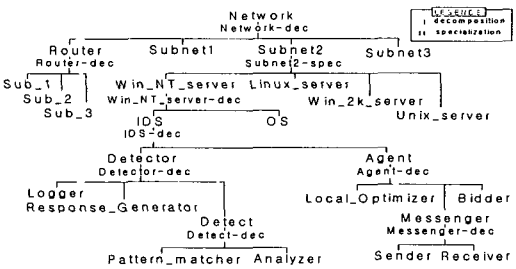

그림 4. Network 모델의 SES

IV. 침입탐지 모델

1. 침입 탐지 모델의 구조 및 기능

IDS 모델은 실질적으로 공격자의 침입을 탐지하는 모델로서 침입을 탐지를 수행하는 모델인 Detector 모델, 입찰에 참여하는 통신 기능을 수행하는 모델인 Agent 모델로 구성된다. 침입 탐지 후의 대응을 생성하기 위한 Response _Generator 그리고 감사 기록을 남기기 위한 Logger 모델이 Detector 모델 내에 존재한다. Detector 모델 내부의 Detect 모델은 Pattern_matcher와 Analyzer 모델로 구성되는데 기능은 다음과 같다.

Pattern_matcher는 전문가 시스템을 사용해서 구성된 규칙에 의해서 침입이 발생했는지 발생할 것인지를 결정한다[13]. 제안한 모델에서는 패킷 정보를 규칙과 비교하여 침입을 탐지하게 된다. 예를 들면 Smurf 공격은 다음과 같이 표현될 수 있다.

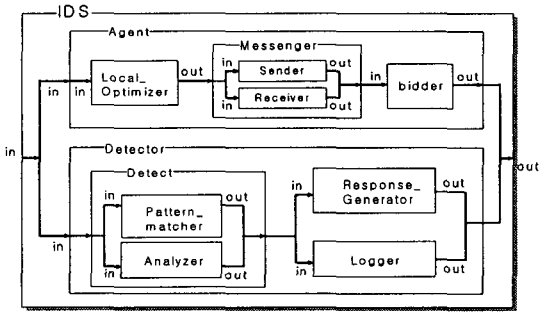

그림 5. IDS 모델의 구조

```
icmp number(receive()) >= m and addr()
                        = (host, broadcast)
```

Smurf 공격은 대상 호스트에서 목적지 주소가 브로드 캐스트 주소와 같은 패킷의 개수가 임계값을 초과하는 것을 의미한다[14].

Analyzer는 호스트에 근거한 상태 전이 분석을 통해 침입을 탐지하게 된다. 이 모델은 시작 상태에서 침입을 탐지할 때까지 하나 이상의 중간 상태를 사용한다. 시작 상태 이후에 key action에 의해 다음 상태로 인식되는데 여기서 인식된 action을 signature action이라고 한다. 침입은 일련의 signature action인 공격 시나리오에 따라 이루어진다. 따라서 signature action에 근거한 규칙을 통해서 침입을 탐지하게 된다.

Response_Generator 모델은 침입을 탐지한 후 침입에 대한 response를 생성한다. 예를 들면 경보를 울리거나 report를 생성하기도 하고 능동적으로 침입 차단 시스템을 통해 패킷을 차단하는 역할을 수행하게 된다.

Logger 모델은 Detector 모델에 의해 침입이 탐지되는 과정에 대한 정보를 로그파일에 기록한다.

2. 침입 탐지 방법

2.1. Pattern_matcher 모델에 사용된 클래스

아래에 소개되는 클래스는 Detector 모델의 Pattern_matcher 모델에서 사용된 클래스의 일부이다.

- IDALandAttack //land attack을 탐지하는 클래스
- IDAMailBomb //mailbomb 공격을 탐지하는 클래스
- IDAPortProb //port probing공격을 탐지하는 클래스
- IDAPortScan //port scan 공격을 탐지하는 클래스

```
IDAMailBomb::IDAMailBomb(): IDATwo(){
  m_PSR = new MailBombRule;
  //rule을 생성
}
IDAMailBomb::~IDAMailBomb(){
  delete m_PSR;//rule 파괴
}
void IDAMailBomb::SetView(CView* v){
  m_View = (CEditView *)v;
  //view에 대한 포인터를 얻어 옴.
  m_PSR->SetView(v);
  //rule에서도 view를 접근하도록 view 포인터를 넘겨줌.
}
void IDAMailBomb::InferStart(DataList* imsy){
...
  DataList *print = imsy;
  if(print==NULL||print->IsEmpty())
    ... //null 이라는 메시지 출력
  else{
   while( !(print->IsEmpty()) ){
   Slot_List fact;
```

```
MakeFact(print,fact);
  //전문가시스템의 사실 생성
  ...
  if( m_PSR->Inference(fact) ){
    if(alarm==0) //침입 탐지를 알림.
  }
  print = print->GetNext();
 }//출력하기 위한 while의 끝부분.
}//end else
```

그림 6. IDAMailBomb 클래스의 구현

- IDAProtocolProb //protocol probing을 탐지하는 클래스
- IDAWinNuk //winnuk 공격을 탐지하는 클래스
- IDASmurf //smurf 공격을 탐지하는 클래스
- IDAJoltAttack //jolt 공격을 탐지하는 클래스
- IDASYNFlood //SYN flood 공격을 탐지하는 클래스
- IDAPWCrack //패스워드 크랙 공격을 탐지하는 클래스

Pattern_matcher 모델은 이러한 클래스들을 통해서 공격을 탐지하게 된다. 그림 6는 IDAMailBomb 클래스 구현의 일부이다. IDAMailBomb 클래스의 생성자에서는 mailbomb 공격을 탐지하기 위한 규칙을 담고 있는 클래스인 MailBombRule의 인스턴스를 생성하게 되고, 소멸자에서 규칙을 파괴하게 된다. SetView 함수에서는 MailBombRule에 view에 대한 포인터를 넘겨주기 위한 작업을 한다. InferStart 함수에서는 실제적인 추론 작업을 수행하게 된다. Generator 모델에서 생성한 패킷을 계속적으로 전문가 시스템에서 사용할 수 있는 규칙의 형태로 만들어 주는 함수인 MakeFact 함수를 계속적으로 호출한다. 이 함수를 호출하여 만들어진 규칙을 Inference 함수에 넘겨주므로 추론을 하게 된다.

2.2. 전문가 시스템의 규칙

그림 7은 mailbomb 공격에 사용된 규칙의 일부이다. Rule1은 프로토콜이 TCP 인지를 검사하는 것이고, Rule2는 패킷이 TCP이고 포트 번호가 25번인지를 검사한다. Rule3은 현재 공격이 진행 중인지를 검사한다. 현재 공격이 진행 중이고 localThreshold의 값을 넘으면 공격자의 출발지 주소(Source IP)를 저장하고, 변수 rule3를 참(True)으로 바꾼다. Rule4는 buffer clearing time을 검사하는 것이고, Rule5 부터 Rule9는 패킷의 개수가 정해진 임계값을 넘는지 검사하는 부분이다.

```
void MBRule::Rule1(Slot_List& fact){
... if(protocol==6) rule1 = true;
... }
void MBRule::Rule2(Slot_List& fact){
... if(rule1 && port==25) rule2 = true;
...  }
void MBRule::Rule3(Slot_List& fact){
... if(rule2 && Time==nowtime)
    if(timecount >= localThreshold){
       S_add.insert( Source_IP );
       rule3 = true;}
...  }
void MBRule::Rule4(Slot_List& fact){
... //check the buffer clearing time.
... }
void MBRule::Rule5(Slot_List& fact){
... if(rule4 && IsDanger() >= Minimal)
            rule5 = true;
...  }
void MBRule::Rule9(Slot_List& fact){
...
if(rule8 && IsDanger()>=Catastrophic)
            rule9 = true;
...  }
```

그림 7. 전문가 시스템에서 사용된 mailbomb 공격 규칙

V. 에이전트간의 연동

계약망 프로토콜은 분산된 문제를 해결하는 것에 있어 통신하고 조정하기 위한 도구로서 제안되었다. 분산 문제 해결 방법은 문제 해결 능력을 갖는 knowledge-sources(KS's)들이 분산되지만 느슨한 연결을 유지하며 서로 협력하여 문제를 해결하는 수단을 제공한다[15]. 분산되었다는 의미는 제어와 데이터가 논리적으로 분리되었음을 의미하고 느슨하게 연결되었다는 것은 각 에이전트 간의 통신보다는 자신이 수행해야 할 작업 처리에 많은 시간을 소비한다는 것을 의미한다. 계약망 프로토콜의 사용은 분산 감지 시스템과 분산 전달 시스템을 위해서 시도되었다[16]. 계약망 프로토콜은 에이전트들이 계약(contract)에 의하여 분산된 문제를 해결하기 위해서 협상하고 통신하는 메커니즘을 제공한다. 에이전트들은 수행

될 필요가 있는 작업을 알리고 다른 에이전트들에 의해 공지된 작업들을 수행하기 위해 비드(Bid)를 만들고, 중앙 콘솔(Command Console)은 각 에이전트들이 제출한 비드를 평가해서 계약을 체결한다.

1. 계약망 프로토콜의 구조

계약망 프로토콜은 모듈화와 객체 지향 설계에 근거한다. 침입 탐지 에이전트 모듈은 도메인에 독립적인 모듈과 도메인에 의존적인 모듈로 구성되는데 도메인에 독립적인 모듈에는 중앙 콘솔, Messenger, 그리고 Bidder가 있고 도메인에 의존적인 모듈은 응용 프로그램에 의존적인 함수들을 호출하는 Local Optimizer 모듈과 친밀하게 작동한다. 그림 8, 9는 계약망 프로토콜의 구조와 에이전트의 구조를 나타내며 각 모듈들은 다음과 같이 동작한다.

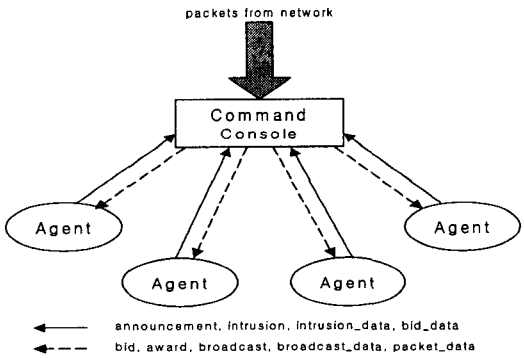

그림 8. 계약망 프로토콜의 구조 및 자료의 흐름

중앙 콘솔은 에이전트의 위치에 대한 정보를 가지고 있으며 모든 에이전트를 중앙에서 통제한다. Messenger는 에이전트들 사이에 메시지를 보내고 받는 것을 관리한다. 중앙 콘솔에서 선택된 에이전트로서 award 메시지를 받아들이며 announcement 메시지를 중앙 콘솔에 보낸다. 에이전트의 위치를 알기 위해 중앙 콘솔에 질의(query)한다. Bidder는 수신한 announcement에 대한 응답으로 Local Optimizer로부터 에이전트의 상태정보를 받아서 중앙 콘솔에 제출할 비드를 만든다. 비드를 만들 때는 시스템 부하를 고려한다. Local Optimizer는 비드의 정보가 되는 시스템 정보를 계산하고 상태에 따라 갱신된 최신의 정보를 유지한다.

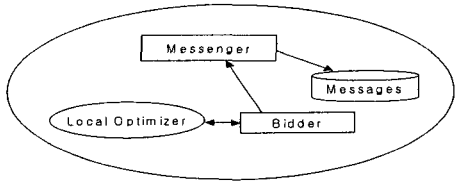

그림 9. 에이전트의 구조

2. 메시지의 종류 및 구조

중앙 콘솔과 침입 탐지 및 침입 차단 에이전트들은 모두 Messenger 모델을 가지고 있는데 이 모델은 메시지의 송신과 수신을 담당한다. 메시지의 종류는 크게 컨트롤 메시지(control message)와 데이터 메시지(data message)가 있다. 컨트롤 메시지에는 다음의 다섯 가지가 있다.

Bid 메시지는 비드를 제출하도록 모든 침입탐지 에이전트에게 알린다. Award 메시지는 중앙 콘솔에서 선택된 에이전트에게 침입탐지 에이전트로 선택된 것을 알린다. Intrusion 메시지는 침입이 발생하거나 상태전이가 이루어지면 침입 탐지 에이전트가 중앙 콘솔에 알린다. Intrusion 메시지를 받은 중앙 콘솔은 침입 차단 시스템에 알려서 침입에 대처하게 된다. Announcement 메시지는 에이전트에서 침입을 탐지할 수 없는 상황이 되거나 상태전이로 에이전트 선택을 다시 해야 할 경우 중앙 콘솔에 알린다. Broadcast 메시지는 침입 탐지 에이전트가 침입을 탐지하면 탐지에 대한 정보를 받은 중앙 콘솔이 모든 침입 탐지 에이전트에 탐지정보를 보낸다는 것을 알린다.

메시지의 종류는 msg_type 필드의 값에 의해 판단한다. 메시지의 구조는 그림 10과 같다.

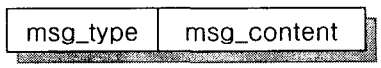

그림 10. 메시지의 구조

표 1. 메시지의 종류

msg_type	메 시 지 종 류
0	Broadcast
1	Announcement
2	Bid
3	Award
4	Intrusion

데이터 메시지는 네 가지가 있는데 첫 번째로 bid_data는 그림 11과 같이 네 가지 필드로 구성된다. address는 비드를 보내는 에이전트의 주소를 나타내고 expertise는 각 에이전트가 탐지할 수 있는 규칙과 전문성을 수치화한 값이며 experience는 전에 탐지한 경험이 있는 침입에 대한 것을 수치화한 값이다. 그리고 loading은 침입탐지 시스템을 장착한 시스템의 CPU에 대한 부하를 수치화한 값이다.

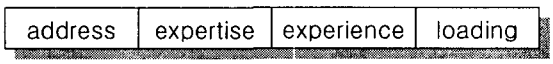

그림 11. bid_data 메시지의 구조

다른 데이터 메시지로 intrusion_data는 침입을 탐지하게 된 패킷의 정보를 가지고 있고 packet_data는 네트워크로 유입되는 패킷의 정보를 가지고 있다. 마지막으로 broadcast_data는 탐지된 침입에 관련된 정보를 포함한다.

3. 침입 탐지 에이전트의 선택과 연동

3.1. 선택 알고리즘

각 에이전트는 loading 필드의 값이 임계값을 넘지 않은 경우에 bid_data를 중앙 콘솔에 보내게 되는데 첫 번째로 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. 그림 12는 에이전트 선택 알고리즘이다.

3.2. 선택 및 연동과정

시뮬레이션이 시작되면 중앙 콘솔은 모든 침입 탐지 에이전트에게 bid 메시지를 보내고 이 메시지를 받은 에이전트들은 bid_data 메시지를 보낸다. 중앙 콘솔은 bid_data를 가지고 선택 알고리즘에 의해 침입을 탐지할 에이전트를 선택하게 되고 선택된 에이전트에게 award 메시지를 보낸다. award 메시지를 받은 에이전트는 packet_data를 기다리고 중앙 콘솔은 패킷 정보를 packet_data에 복사하여 선택된 에이전트에게 보낸다. 선택된 에이전트는 이 데이터를 가지고 침입을 탐지하게 된다.

침입 탐지 과정 중 상태전이가 발생한 경우에 에이전트는 중앙 콘솔에 announcement 메시지를 보내고 이 메시지를 받은 중앙 콘솔은 위의 과정과 마찬가지고 bid 메시지를 보내고 에이전트 선택과정을 반복하게 된다.

침입을 탐지한 경우에는 선택된 에이전트가 intrusion 메시지를 중앙 콘솔에 보내고 intrusion_data 메시지를 보낸다. 이 메시지를 받고 중앙 콘솔은 침입차단 시스템에 ntrusion, intrusion_data 메시지를 보낸다. 또한 중앙 콘솔은 broadcast 메시지와 침입에 대한 정보를 broadcast_ data 메시지로 보낸다. 그런 다음 다시 bid 메시지를 보내고 위의 에이전트 선택 과정을 반복한다.

```
Let  bid_i be bids
Let bid_list = ( bid_1, bid_2,·····, bid_n) be a list of bids
Set 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)
```

그림 12. 에이전트 선택 알고리즘

VI. 시뮬레이션 결과

본 논문에서는 하나의 침입 탐지 시스템이 침입을 탐지하는 경우와 여러 개의 침입 탐지 시스템이 서로 협력하여 침

입을 탐지하는 경우에 대하여 시뮬레이션을 수행하였다. 공격은 mailbomb 공격 및 jolt 공격을 시도하였다. mailbomb 공격은 메일 서버에 폭탄 메일을 보내는 DoS(Denial of Service) 공격의 한 종류이다. jolt 공격도 DoS 공격의 일종으로 패킷을 잘게 쪼개서 많은 수의 패킷을 만들고 이렇게 만들어진 많은 양의 패킷을 대상 시스템에 보냄으로 공격 대상 시스템에 과부하가 걸리도록 하는 공격이다. 시뮬레이션의 성능 지표로는 침입 탐지 시간(Intrusion Detecting Time), FNER (False Negative Error Ratio) 및 FRER(False Positive Error Ratio)를 택하였다.

시뮬레이션 수행에 사용된 임계값은 40, 50, 60, 70, 80이다. 그림 13, 14에서와 같이 다수의 침입 탐지 에이전트가 침입을 탐지하는 경우가 하나의 에이전트가 침입을 탐지하는 것보다 빠름을 알 수 있다.

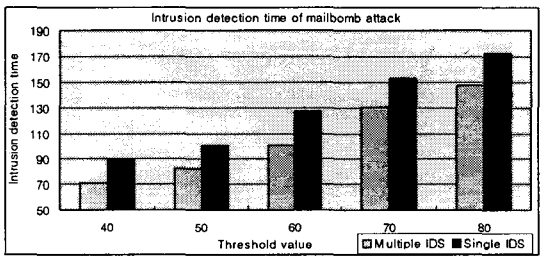

그림 13. mailbomb 공격의 침입 탐지 시간

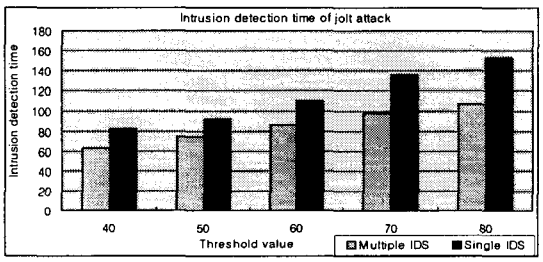

그림 14. jolt 공격의 침입 탐지 시간

서로 정보를 공유하면 침입을 빠르게 탐지하게 되면 관리자에 의한 신속한 대응으로 네트워크의 피해를 줄일 수 있다. 현재 인터넷과 같이 다른 네트워크와의 관계가 중요해지는 상황에서는 피해를 받고 있는 네트워크에만 공격의 영향이 미치는 것이 아니라 다른 네트워크에도 공격의 영향이 미치므로 빠른 침입 탐지는 인터넷을 보호하는 중요한 요인이 된다.

그림 15, 16는 보안의 수준을 강화하면(본 시스템에서는 임계값을 낮춤) False Negative 에러 비율이 감소함을 나타낸다. 그림 15, 16에서 여러 개의 침입 탐지 시스템을 사용했을 때의 에러율이 하나의 침입 탐지 시스템을 사용했을 경우의 에러율보다 낮음을 알 수 있다. 이것은 다수의 침입 탐지 에이전트가 계약망 프로토콜에 의해서 서로 충분한 정보를 공유하여 침입을 탐지했기 때문이다.

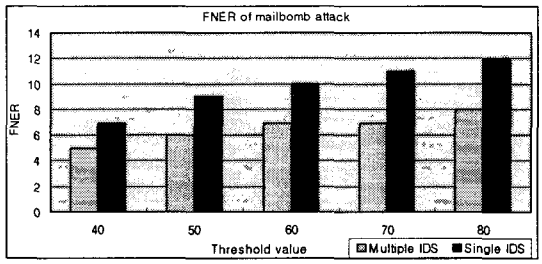

그림 15. mailbomb 공격의 FNER

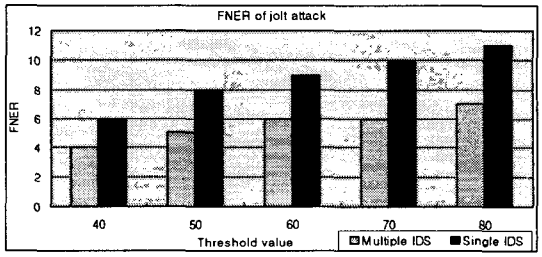

그림 16. jolt 공격의 FNER

그림 17, 18은 FPER로서 다양한 침입 에이전트가 계약망 프로토콜을 사용하여 침입을 탐지하는 경우의 에러율이 낮음을 볼 수 있다. 계약망 프로토콜을 사용하여 침입을 탐지하는 경우에는 각 침입 탐지 시스템의 상황이 고려되어 입찰이 진행되고 지역적인 최적화 과정을 거치게 되므로 보다 효과적으로 침입을 탐지하는 방법을 제공할 수 있다.

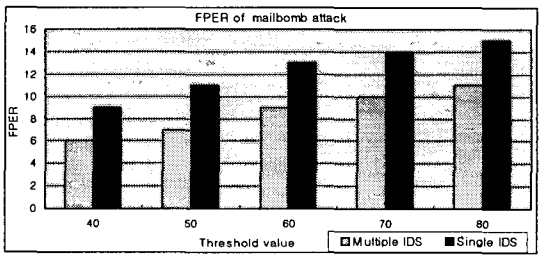

그림 17. mailbomb 공격의 FPER

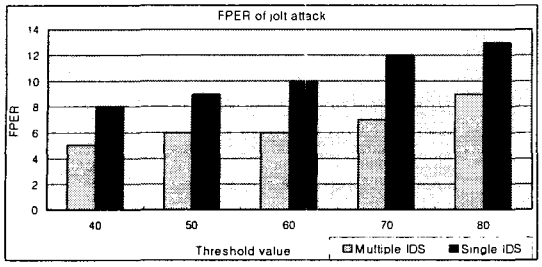

그림 18. jolt 공격의 FPER

VII. 결론 및 향후 과제

계약망 프로토콜을 적용한 네트워크 보안 모델의 설계를 바탕으로 DEVS 방법론을 사용하여 보안 시스템들의 모델링을 수행하였다. 계약망 프로토콜을 사용한 시스템은 각 에이전트의 상황에 맞게 작업량을 분산하여 처리할 수 있는 장점이 존재하기 때문에 어느 한 노드에 큰 부하가 걸리는 경우를 방지할 수 있다. 또한 입찰에 참여할 때 에이전트를 선택하는 과정에서 다양한 알고리즘을 정의할 수 있으므로 정책적으로 각 노드에 걸리는 부하를 조절할 수 있을 뿐만 아니라 선택 알고리즘의 개선만으로도 처리 속도를 크게 개선할 수 있는 장점이 존재한다.

향후 과제로는 다양한 에이전트 선택 알고리즘을 갖추고, 네트워크의 상황에 맞게 능동적으로 알고리즘을 바꿀 수 있는 연구가 진행될 것이다.

참 고 문 헌

[1] E. D. Zwicky, S. Cooper and D. B. Chapman, Building Internet Firewalls second edition, O'reilly & Associates, 2000.

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

[3] R. Bace, Intrusion Detection, Macmillan Technical Publishing, 2000.

[4] Duan Haixin, Wu Jianping and Li Xing, "Policy based access control framework for large networks," Proceedings of IEEE International Conference on ICON 2000, Sept. 2000.

[5] Matt Bishop, Computer Security, Addison Wesley, 2003.

[6] Seo, Hee Suk, Cho, Tae Ho and Chi, Sung Do, "Modeling and Simulation of Distributed Security Models," Lecture Notes on Computer Science, Springer Verlag, LNCS 2660, pp. 179-190, Jun. 2003.

[7] Seo, Hee Suk and Cho, Tae Ho, "Simulation of Network Security with Collaboration among IDS Models," Lecture Notes on Artificial Intelligence, Springer Verlag, LNAI 2256, pp. 438-448, Dec. 2001.

[8] Wang Junpu, Chen Hao, Xu Yang and Liu Shuhui, "An Architecture of Agent-Based Intelligent Control Systems," Proceedings of the 3rd World Congress on Intelligent Control and Automation, IEEE, June 28-July 2, pp. 404-407, 2000.

[9] Shungeng Hu, Li Zhang and Yixin Zhong, "Theories, Technology and Application of Multi-Agent Systems," Computer Science, Vol. 26, No. 9, pp. 20-24, 1999.

[10] B. P. Zeigler, Object-Oriented Simulation with Hierarchical, Modular Models, USA:Academic Press, San Diego CA, 1990

[11] Seo, Hee Suk and Cho, Tae Ho, "Modeling and Simulation for Detecting a Distributed Denial of Service Attack," Lecture Notes on Artificial Intelligence, Springer Verlag, LNAI 2557, pp. 179-190, Dec. 2002.

[12] B. P. Zeigler, Theory of Modeling and Simulation, John Wiley, NY, USA, 1976, reissued by Krieger, Malabar, FL, USA, 1985.

[13] Shan Zheng, Chen Peng, Xu Ying and Xu Ke, "A Network State Based Intrusion Detection Model," Computer Networks and Mobile Computing, 2001. Proceedings. International Conference on 2001, pp. 481-486, 2001.

[14] S Mclure, J. Scambray, G. Kurtz, Hacking Exposed: Network Security Secrets and Solutions, McGraw-Hill, 1999.

[15] Alan H. Bond and Les Gasser, Distributed Artificial Intelligence, Morgan Kaufmann Publishers, 1988.

[16] T. Sandholm, "An Implementation of the Contract Net Protocol based on Marginal Cost Calculations," in 11th National Conference on Artificial Intelligence (AAAI-93), Washington. DC, 1993.

서 희 석(Hee-Suk Seo) 정회원

2000년 2월 : 성균관대학교 산업공학과 (공학사)
2002년 2월 : 성균관대학교 전기전자 및 컴퓨터공학부(공학석사)
2002년 3월 ~ 현재 : 성균관대학교 정보통신공학부 박사과정
<관심분야 > : 네트워크 보안 시뮬레이션, 지식기반 시뮬레이션, 취약성 분석.

김 희 완(Hee-Wan Kim) 정회원

1987년 : 광운대학교 전자계산학과 (이학사)
1995년 : 성균관대학교 정보공학과 (공학석사)
2002년 : 성균관대학교 전기전자 및 컴퓨터공학부(공학박사)
1991년 : 한국전력공사 정보처리처
1996년 : 정보처리 기술사(정보관리) 취득
1999년 : 공인 정보시스템감리인 자격취득(한국전산원)
1996년 : 삼육의명대학 전산정보과 조교수
2001년 ~ 현재 : 삼육대학교 컴퓨터과학과 조교수
<관심분야> : 컴퓨터보안, 동시성제어, 분산DB, 보안 시뮬레이션