침입자 추적 시스템의 에이전트 통신 보안을 위한 메커니즘

(A Secure Agent Communication Mechanism for Intruder Tracing System)

최 진 우 † 황 선 태 †† 우 종 우 ††† 정 주 영 †††† 최 대 식 ††††

(Jinwoo Choi) (Suntae Hwang) (Chongwoo Woo) (Jooyoung Jung) (Daesik Choi)

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. 따라서 최근 해킹에 대한 대응도 소극적 탐지 기능을 벗어나 적극적인 역 추적 등의 방법을 동원하는 추세로 가고 있다. 본 논문에서는 자율적인 기능을 가질 수 있는, 멀티 에이전트 기반의 침입자 역 추적 시스템에 대해서 기술한다. 특히 각 에이전트 간의 통신 메시지를 보호하기 위한 통신보안 메커니즘을 중점적으로 제안하는데, 이를 위해서 KQML 레벨에서 파라미터를 확장하고 공개키 암호화 방식을 도입하였다. 제안된 메커니즘에서 각 에이전트는 모든 통신에 앞서 중개자 에이전트를 통해서 서로를 인증하기 때문에 효율성은 떨어지지만 통신 메시지의 보안이 보다 확실하게 된다. 이는 역 추적 수행 시 에이전트 및 수행 서버의 안전을 위해서는 매우 중요한 요소 중에 하나이다.

키워드 : 침입자 추적, KQML, 멀티 에이전트 시스템

Abstract As the Internet technology becomes a major information infrastructure, an emerging problem is the tremendous increase of malicious computer intrusions. The present Intrusion Detection System (IDS) serves a useful purpose for detecting such intrusions, but the current situation requires more active response mechanism other than simple detection. This paper describes a multi-agent based tracing system against the intruders when the system is attacked. The focus of the study lies on the secure communication mechanism for the agent message communication. We have extended parameters on the KQML protocol, and applied the public key encryption approach. The limitation might be the requirements of two-way authentication for every communication through the broker agent. This model may not improve the efficiency, but it provides a concrete secure communication. Also this is one important factor to protect the agent and the tracing server during the tracing process.

Key words : Intruder Tracing, KQML, Multi-agent System

1. 서 론

인터넷의 발달과 더불어 최근 네트워크 상에서의 침입 유형은 급속도로 다양화되어 가고 있다. 또한, 인터넷을 통한 전자상거래, 홈뱅킹 등의 서비스가 급속히 성장하면서 이러한 사회기반 시설에 대한 침입피해 사례가 급증하고 있는 추세이다. 이러한 침입 피해를 방지하기 위하여 방화벽(Firewall)과 침입탐지 시스템(Intrusion Detection System: IDS)이 개발되어 설치되고 있다. 방화벽에 비해 IDS는 보다 능동적으로 침입사실을 탐지할 수 있으나 침입피해를 감소하거나 완전히 방어하기에는 문제점들이 있다. 예를 들면 기존의 IDS는 데이터를 수집하고 분석하는데 중앙 집중화 되어있고 통일된 구조를 가지고 있다. 따라서 실시간으로 탐지하고 이에 따른 대응을 하기에는 구조적인 문제점이 있고, 또한 시스템의 변경 및 확장에도 문제점이 노출된다. 그리고 침

† 학생회원 : 국민대학교 전산과학과
jwchoi@cs.kookmin.ac.kr
†† 종신회원 : 국민대학교 컴퓨터학부 교수
sthwang@kookmin.ac.kr
††† 정 회 원 : 국민대학교 컴퓨터학부 교수
cwwoo@kookmin.ac.kr
†††† 비 회 원 : 국가보안기술연구소 연구원
prgmaker@etri.re.kr
dschoi@etri.re.kr
논문접수 : 2002년 3월 11일
심사완료 : 2002년 10월 2일

입피해가 점차 심각해 가는 현실에 비해 기존의 IDS는 침입 탐지 이상의 적극적인 대응방안이 없다는 문제점이 있다. 즉, 침입자가 시스템에 피해를 주는 상황에서도 이를 적극적으로 물리치거나 방어할 수 있는 방법이 없다고 할 수 있다. 이러한 문제점을 극복하기 위해서 최근에는 해킹 피해 시스템에 대한 분석과 대응을 인간에 의한 수동적인 방법에 의존하기보다는, 자동 분석 및 대응이 가능한 시스템에 관한 연구가 요구되고 있다.

본 논문에서는 멀티 에이전트 개념을 도입함으로서, 보다 자율적이며, 지능적이고, 적극적인 침입 추적 시스템을 제시하고자 한다. 이러한 시스템의 구현을 위해서 우선, 침입 상황 시에 적극적으로 대처할 수 있는 한 가지 방안으로 '역 추적(tracing intruder)' 방안을 제시하였고, 또한 역 추적을 체계적으로 구현하기 위해서 시스템의 전반적인 구현에 앞서, 가상의 역 추적 시나리오를 설정하였다. 또한 이 시나리오에 따른 역 추적 에이전트 간의 메시지 전송에 대한 메시지 보호 및 에이전트 보호에 대한 모델을 제시하였다.

2. 관련연구

2.1 침입탐지 시스템

침입 탐지란 불법적인 행위를 실시간으로 감시하여 탐지 하는 것을 말하며, 이러한 행위를 실시간으로 탐지하여 보고하는 시스템을 침입탐지 시스템이라 한다. 방화벽이 단순히 불법적인 접근을 막는데 중점을 두고 있다면 침입탐지 시스템은 방화벽이 효과적이지 못할 때, 이에 따른 피해를 최소화하고 관리자가 없을 때에도 불법적인 침입에 적절히 대응할 수 있는 해결방안이라고 할 수 있다.

그러나, 모든 침입탐지 작업이 단일화 되어 있어서 과중한 부하를 유발할 수 있고 탐지 및 대응 모듈이 오 동작하거나 및 파괴되는 것에 따른 안정성 제고 문제 등이 야기 될 수 있다. 따라서 기존의 침입탐지 시스템이 가지고 있는 여러 가지 단점을 보완하기 위해, 최근에는 에이전트 개념을 도입한 IDS 시스템이 개발되고 있다. 대표적인 시스템으로는 IDA(An Intrusion Detection Agent System)[1]시스템과 AAFID(Autonomous Agent For Intrusion Detection)[2]시스템 등이 있다. IDA는 이동 에이전트(mobile agent)에 의해 침입자들을 추적할 수 있는 기능과 침입한 경로를 따라 침입과 관련된 정보를 수집하는 기능을 가진다. 그리고 IDA는 이런 수집된 여러 정보들을 통해서 침입의 발생여부를 결정하는 역할을 하기도 한다. AAFID에서 제안된 모델은 다중의 독립적인 요소들이 정보를 수집하는 분산 침입 탐지 시스템이다. 이 시스템에서는 최하위 에이전트가 데이터를 수집하고 가공하여 상위 계층으로 전송하는 계층적 구조를 제안하고 있다. 이 두 시스템은 분산 환경에서의 수행과 독립적인 에이전트의 기능 면에서는 IDS와 다르지만 역 추적 등 능동적 방안을 제시하고 있지 못하다.

2.2 멀티 에이전트

IDS가 가지고 있는 구조적 문제점의 해결방안으로 최근에는 멀티 에이전트 개념을 적용한 역 추적 시스템이 연구되고 있다. 멀티 에이전트 시스템은, 한 개의 중앙 집중화 된 시스템이 해결하기에는 너무 크거나 복잡한 문제를, 자율성을 가지는 다수의 서브 에이전트들이 동적으로 분산된 임무를 수행 가능하게 함으로써, 해결하도록 하였다. 에이전트 시스템의 장점은 인공지능에서 연구되어온 많은 연구 결과인 지식베이스, 추론 능력을 가지고 있어 지능적으로 행동할 수 있다는 것과, 네트워크를 통해 분산된 에이전트끼리 협동하여 작업을 수행하기도 한다는 점이다[3,4,5].

이러한 분산된 환경 하에서 에이전트들끼리 작업을 수행하기 위해서는 에이전트간 통신을 위한 프로토콜이 요구되는데 KQML(Knowledge Query and Manipulation Language)[6,7]이 대표적인 예이다. KQML은 분산되어 있는 지능적 소프트웨어 에이전트들 간의 상호 통신을 지원하기 위하여 설계된 언어이다. 이러한 에이전트 기반 통신 프로토콜을 사용함으로써 에이전트간 통신 전반에 대한 의미 구조를 설계하는 것이 가능하며 보다 효율적인 멀티 에이전트 시스템 구현이 가능하다. 본 논문에서는 이러한 장점이 있는 멀티 에이전트 개념을 적용하여 역 추적 에이전트 시스템을 설계한다.

2.3 에이전트 보호 및 메시지 보호

KQML 프로토콜을 이용하면 에이전트간의 지식 전달을 위한 통신이 매우 효과적으로 이루어진다. 그러나 KQML 프로토콜 자체에는 에이전트 간의 통신에 따른 메시지 보호에 관련된 메커니즘을 포함하지 않고 있다. 따라서 에이전트간의 통신 시 서로 다른 에이전트를 신뢰하고 메시지를 보호할 수 있는 새로운 메커니즘이 요구된다. 이와 관련해서 최근에는 많은 연구가 이루어지고 있으며 그 중에서도 마스터 키를 이용한 메시지 보호 및 단 방향 인증을 위해 퍼포머티브와 파라미터를 확장한 모델[12]과 이를 바탕으로 보안 레벨의 강화를 위한 공개키 와 개인키에 의한 공개키 암호화 방식의 사용과 효율성 향상을 위해 단지 파라미터만을 확장 시켜 양방향 인증을 수행하는 모델[14] 등 대표적인 연구 사례가 있다.

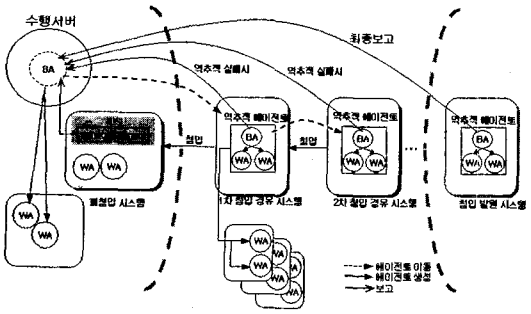


그림 1 역 추적 시스템 구성도

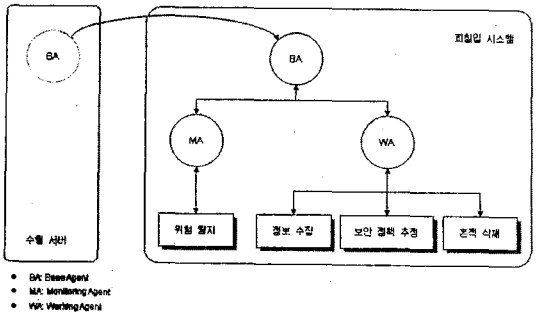


그림 2 역 추적 메시지 시나리오

3. 역 추적 시나리오

해킹에 대해서 능동적 대응을 위한 역 추적이 발생하는 환경은 다음 그림 1과 같이 가정하였고 이러한 환경에서 가상적으로 역 추적 시나리오를 설정하였다.

3.1 침입 및 역 추적 환경

역 추적 환경의 구성은 수행 서버, 침입 경유 시스템, 침입 발원 시스템과 역 추적의 기능을 수행할 다수의 에이전트들로 구성된다(그림 1). 여기서 각 시스템들의 정의는 다음과 같다.

- 수행 서버: 침입 탐지 시스템과 에이전트를 제어하는 역할 수행하는 호스트
- 피침입 시스템: 실제 침입을 당한 로컬 시스템 내의 호스트
- 침입 경유 시스템: 침입자가 피침입 시스템에 침입을 하기 위한 중간 경유지의 호스트
- 침입 발원 시스템: 피침입 시스템에 침입한 침입자가 침입을 시작한 호스트

3.2 가상 역 추적 시나리오

침입이 탐지된 후, 적극적으로 침입피해 상황에 대처하기 위해 침입자를 역 추적하는 과정을 단계별로 설명하면 다음과 같다(그림 2 참조). 특히 그림 2는 그림 1을 기능적 측면에서 단순화한 것이며 여기서 사용되는 각 서브 에이전트의 기능 및 구조는 다음 절에서 설명한다. 이러한 가상 시나리오는 역 추적 에이전트 시스템을 설계하기 위한 것이다.

- 제 1단계, IDS가 침입을 탐지
- 제 2단계, IDS가 수행서버에게 침입을 보고
- 제 3단계, 수행서버는 침입 발원지를 분석
- 제 4단계, 기지 에이전트(BA: Base Agent)가 침입 경유 시스템으로 이동
- 제 5단계, 이동한 에이전트가 감시 에이전트(MA: Monitoring Agent)와 작업 에이전트를 실행
- 제 6단계, 실행된 감시 에이전트와 작업 에이전트(WA: Working Agent)는 각각 에이전트 보호를 위해 위협을 탐지하고 정보를 수집하는 활동
- 제 7단계, 필요한 정보를 수집한 에이전트는 수행을 마치고 다음 침입 경유 시스템으로 이동하기 전에 히스토리 파일과 로그 파일 삭제한 후 이동
- 제 8단계, 에이전트가 이동한 시스템이 침입 발원 시스템일 에이전트는 목표를 마치고 수행서버에 그동안 의 정보를 수행서버에 전송

4. 역 추적 에이전트 시스템 설계

역 추적 에이전트 시스템은 크게 세 개의 서브 에이전트로써, 기지 에이전트(Base agent), 작업 에이전트(Working agent), 그리고 감시 에이전트(Monitoring agent)로 구성된다(그림 3).

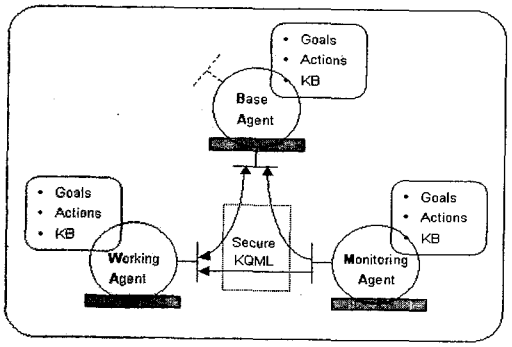


그림 3 역 추적 시스템 구조

4.1 서브 에이전트의 기능

역 추적 시스템 내 각각의 에이전트는 자신의 고유의 도메인 지식을 가지고 있으며, 이를 기반으로 자신에게 주어진 목표를 해결하게 된다. 그리고 목표 수행 과정

중 문제를 해결하기 위하여 다른 에이전트를 수행시킬 수 있으며, 또는 다른 에이전트와 상호 통신을 함으로써 협력하여 문제를 해결할 수 있다.

4.1.1 기지 에이전트(Base Agent)

침입 경유 시스템의 기지 에이전트 침투 후에는 자신의 목표를 수행하기 위한 작업 에이전트를 실행시킨다. 기지 에이전트는 자신의 도메인 지식을 기반으로 자신의 행동 지침에 반영되는 사실들을 수집한다. 예를 들어, 기지 에이전트의 최종 목표인 다음 침입 경유 시스템에 대한 사실들을 수집하기 위하여 요구되는 절차를 수행하는 작업 에이전트를 실행한다. 만일 주어진 목표를 성취한 경우에는 동작 중인 모든 에이전트들의 삭제에 관한 절차를 수행한다. 그렇지 못한 경우에는 실패한 절차를 즉시 수행 서버로 전송한다.

4.1.2 작업 에이전트(Working Agent)

작업 에이전트는 기지 에이전트에 의하여 초기에 실행되며, 수행 중인 작업 에이전트가 요구할 때 새로운 작업 에이전트로 생성될 수 있다. 작업 에이전트는 침입 경유지 시스템의 보안 정책에 대한 추정과 사실들에 대한 수집, 그리고 흔적 삭제를 위한 지식 베이스를 가지고 있다. 예를 들어, 작업 에이전트가 보안 정책의 추정 결과에 따른 위험도를 기지 에이전트에게 전송하게 되고, 기지 에이전트의 행동 지침에 따라서 재구성된 작업 목표를 기지 에이전트로부터 수신하여 자신의 작업에 반영할 수 있다. 그리고 자신의 수행 절차를 기록하여 작업 실패 시의 기록을 기지 에이전트에게 전송하게 된다.

4.1.3 감시 에이전트(Monitoring Agent)

감시 에이전트는 자기 자신과 다른 에이전트의 보호를 위하여 독립적으로 시스템을 감시하는 에이전트로 기지 에이전트에 의하여 초기에 생성된다. 다른 에이전트들과 기본 구조는 동일하지만, 에이전트간 통신 메커니즘에서 그 차이점이 명백히 구별된다. 차이점은 다른 에이전트들은 상호 통신, 즉 양방향 통신을 하는데 반하여, 감시 에이전트는 단 방향 통신을 수행한다. 다른 에이전트들에 의하여 생성된 메시지의 유입은 없으며, 침입 경유 시스템 내에서 발각이 추정되는 경우에 경보 메시지를 동작 중인 모든 에이전트에게 전파하는 기능을 담당한다.

4.2 에이전트 구조(Agent Architecture)

본 논문의 에이전트 시스템은 다음과 같이 여섯 개의 구성부인 KQML 프로세서, 인터페이스(interface), 인터프리터(interpreter), 워크플로우(workflow), 절차적 수행부(procedural action), 그리고 에이전트 지식베이스(knowledgebase)로 설계되었다(그림 4).

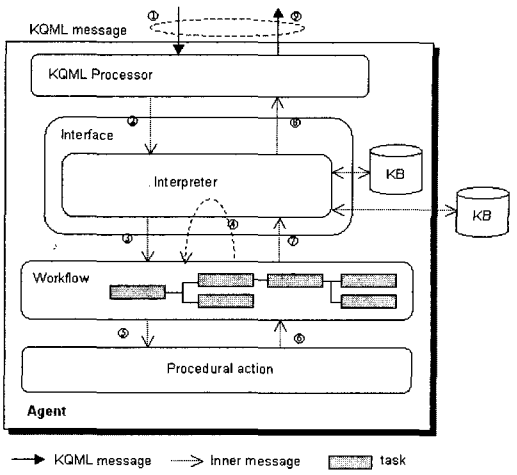


그림 4 에이전트 구조

4.2.1 KQML 프로세서

에이전트로 유입된 KQML 메시지(그림 4의 ①)는 KQML 프로세서에 의하여 분석이 된다. 메시지 분석은 명세서를 바탕으로 이루어지며, 명세서는 설계자에 의하여 확장될 수 있다. 명세서의 :language 파라미터에 명시된 내용을 독립적으로 설계하는 것이 가능하게 함으로써 KQML 프로세서가 사용된 언어의 해석기를 유연하게 선택할 수 있도록 한다. 본 시스템에서는 내부언어(internal language)로써 CISL (Common Intrusion Specification Language)[8,9]을 확장한 언어인 IL을 사용하였다. 또한 KIF와 같은 다른 언어를 내부 언어로 사용하는 것도 가능하도록 설계하였다. 그리고 메시지의 재구성은 송신 에이전트로 가는 응답 메시지를 응답 결과와 함께 재구성하는 것을 뜻한다.

4.2.2 인터페이스

그림 4의 인터페이스 부분은 KQML 프로세서로 유입된 KQML 메시지를 분석하여 :language 필드로부터 내부 언어를 해석하기위한 인터프리터를 선택하는데 이를 나타낸 것이 그림 5이다. 인터페이스로 유입되는 KQML 메시지(그림 4의 ②)는 퍼포머티브(performative)의 분석을 담당한다. 기본적으로 제공되어야 하는 KQML 퍼포머티브를 비롯하여 설계자에 의하여 확장, 추가된 퍼포머티브를 분석 가능하도록 설계되었다. 이러한 퍼포머티브의 확장을 통해서 본 시스템의 보안에 관련된 부분이 설계 되었으며, 또한 내부 언어를 위한 확장을 위해서도 이용되었다. 인터프리터는 KQML 프로세서를 통해 처리된 :content 파라미터에 명시된 내용을 해석하는 부분이며 파서(parser)와 바인더(binder), 추론부(reasoner)로

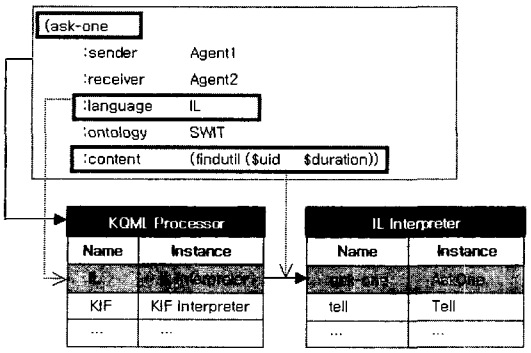


그림 5 KQML 프로세서

구성되어 있다. 이 중 바인더는 content의 내용을 수행하기 위한 오퍼레이션과의 연동을 위한 엔진 부분이다. 인터프리터에 연결된 에이전트 도메인 지식은 에이전트의 외부 지식으로도 명시가 가능하도록 설계하여 확장성을 제공하였다.

4.2.3 워크플로우

워크플로우 부는 인터프리터에 의해 해석되어진 : content 파라미터에서 명시되어 있는 절차(그림 4의 ③)를 해석하여 하나의 프로시저를 유지하는 부분이다. 유입되는 하나의 KQML 메시지에 하나의 워크플로우가 생성되어 유지된다. 절차적 수행부(procedural action)로부터 유입되는 각각의 결과들을 지식베이스를 참조하여 통합(그림 4의 ④)하고 분석한다. 절차적 수행부는 KQML 프로세서와 인터프리터를 경유하여 유입되는 입력으로부터 바인딩 된 오퍼레이션을 직접 실행하는 부분으로써 그 절차를 제어하는 부분이다. 실행부는 유입되는 워크플로우의 작업(task) 메시지(그림 4의 ⑤)에 하나씩 각각의 작업을 부여하는 방식으로 설계하였고, 그 실행 결과는 역 방향(그림 4의 ⑥-⑨)으로 전송된다. 마지막 단계에서 KQML 프로세서를 통해 재구성된 KQML 메시지를 에이전트로 전송한다.

4.2.4 IL 인터프리터의 설계

IL 인터프리터는 IL의 번역을 위한 어휘 분석기(lexical analyzer)와 구문 분석기(parser)로 구성된다[그림 6]. IL 인터프리터는 JDK 1.3.1을 사용하여 Java로 설계 및 구현하였으며, IL 파서 설계 시 사용된 파싱 기법은 recursive-descent 파싱 기법을 사용하였다

IL 구문은 S-expression[10,11] 형식으로써 지식 스킴(knowledge scheme)을 표현하며, 어휘 분석기가 IL 구문을 토큰(token)으로 분리하여 하나의 토큰 집합으로 구성한 뒤, 구문 분석기는 토큰을 사용하여 IL 문법 검

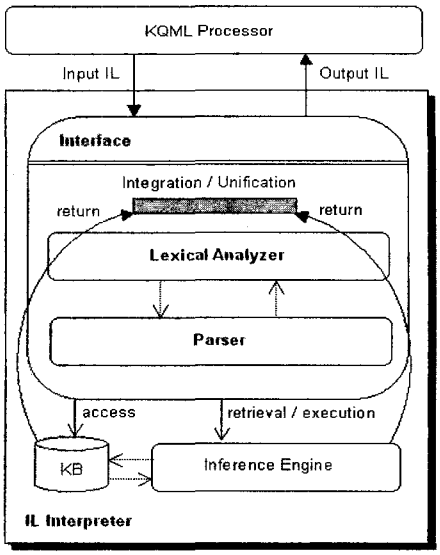


그림 6 IL 인터프리터

사를 한다. 모든 문법 검사가 성공적으로 끝나면, 인터페이스로부터 지식베이스를 로드, 검색, 참조해가며 결과를 생성해 내거나, 또는 추론 과정을 거쳐 결과를 생성한다. 이러한 결과는 지식베이스로부터의 검색된 단순한 정보일 수도 있고, 바인딩 된 오퍼레이션의 수행일 수도 있고, 그리고 규칙을 기반으로 하는 작업들의 나열일 수도 있다. 작업들의 나열인 경우는 워크플로우에 유지되며, 하나의 작업 단위에 하나의 절차적 수행이 생성된다. 모든 절차들을 수행한 뒤 워크플로우에 유지되어 있는 조건을 만족하면 결과를 다시 IL 인터프리터에 전달한다. 전달된 결과를 IL 인터프리터는 통합을 위한 인터페이스를 사용하여 새로운 IL로 통합하고, KQML 프로세서로 전달한다.

5. 에이전트의 보호

모든 에이전트간의 통신 시에는 보안 구조가 확립된 메시지를 상호 교환해야 하며 에이전트와 에이전트, 에이전트와 플랫폼, 플랫폼과 에이전트 사이에 발생하는 모든 통신에 대한 보안이 요구된다. 또한 역 추적 임무를 수행 중인 에이전트 및 에이전트 플랫폼의 보호를 위한 은닉도 요구된다. 만일 누군가에 의해 수행 중인 역 추적 에이전트로 위장한 악성 에이전트를 생성하여 수행 서버와의 통신을 시도하려 한다면 역 추적 에이전트를 파견한 위치의 노출과 역 공격에 의한 기밀 유출 등의 피해를 우려할 수 있다. 이러한 피해를 줄이기 위

해 역 추적 에이전트가 보호되어야 하는데 감시 에이전트가 에이전트의 노출을 감시하는 것 이외에도 에이전트끼리 주고 받는 메시지들이 보호될 필요가 있다. 이를 위해 KQML의 퍼포머티브와 파라미터를 확장하여 KQML 메시지의 비동기 상호 통신 및 세션키 교환을 통해서 보안 채널을 구성하는 모델이 있다[12,13]. 또한 보안 채널 구성에 필요한 세션키를 통신상에서 주고 받을 시 세션키 노출 위험을 줄이기 위해 서로 약속된 내부 함수에 의해 이를 생성함으로써 KQML의 퍼포머티브 대신 단지 파라미터만을 확장한 모델[14]을 볼 수 있다. 그러나 본 논문의 시스템에서는, 메시지 보호 이외에도 에이전트 은닉에 대해서도 큰 비중을 두고, 다음 사항을 고려하여 메시지 전송 메커니즘을 설계하였다.

- 보다 안전한 통신을 위해서 단순한 마스터 키 방식이 아닌 공개키 암호화 방식을 이용한다.
- 약속된 내부 함수에 의한 세션키의 생성은 세션키를 통신으로 주고 받을 필요가 없으므로 보다 안전하다고 할 수 있으나 에이전트의 소스 코드가 노출될 수 있는 상황에서는 내부 함수의 알고리즘 역시 노출될 수 있으므로 적합치 않다.
- 중개자는 생성한 에이전트들에 관한 키와 같은 보안 사항은 자신의 내부에 유지하지 않는다. 따라서 에이전트 사이의 상호 인증은 중개자를 거쳐야만 하며, 그 결과 중개자의 노출 시에도 이미 성립된 에이전트들은 보호될 수 있다.

이러한 구조를 가지는 메시지 전송 메커니즘은 다음과 같다.

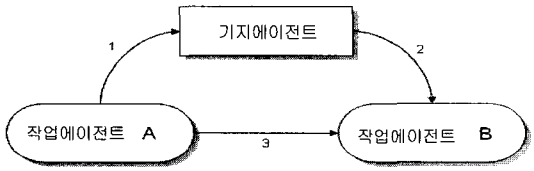


그림 7 메시지 전송을 위한 인증 구조

그림 7에서의 에이전트 보안 모델은 공개키 암호화 방식을 사용하며 KQML 프로토콜에서 보안 모델의 확장을 위해 파라미터를 확장하였다. 또한 이 모델에 있어서 기존에 연구된 모델과의 두드러진 차이점은 작업 에이전트 A에 의한 특정 목적을 수행하기 위해 알려지지 않은 에이전트로의 메시지 전송에 대해 기지 에이전트는 자신의 정보를 분석하여 해당 목적에 맞은 에이전트를 찾아 내고 이들 에이전트간에 메시지 전송을 위한 신뢰할 수 있는 인증 서버로의 역할을 수행하는 것이다. 기지 에이전트를 통해 통신을 하려는 두 작업 에이전트 사이의 상호 인증을 수행하며 기지 에이전트를 통해 상호 인증이 끝나면 상호 인증 과정에서 주고 받은 세션키를 이용하여 통신을 하려는 두 작업 에이전트 사이에 직접적인 보안 채널을 구성하여 세션키로 암호화된 KQML 메시지를 주고 받는다. 두 에이전트 사이에 메시지 전송을 위한 과정을 기술하면 다음과 같다.

- 작업 에이전트와 기지 에이전트를 위한 키의 생성 및 분배
- 작업 에이전트 A와 기지 에이전트간에 상호 인증을 통한 보안 채널 확보(step1)
- 작업 에이전트 A에 의해 보내진 중첩된 KQML 메시지 분석
- 분석된 KQML 메시지를 통한 작업 에이전트 B와의 보안 채널 확보(step2)
- 기지 에이전트를 경유한 두 작업 에이전트간의 세션키 교환
- 두 작업 에이전트간에 직접적인 보안 채널 구성(step3)
- 교환된 세션키를 통한 암호화된 메시지 교환

5.1 작업 에이전트와 기지에이전트를 위한 키의 생성 및 분배

공개키 암호화 방식을 위해서는 개인키와 공개키가 요구된다. 이들 키의 생성은 일반적으로 널리 사용되는 RSA알고리즘을 사용하여 기지에이전트에서 작업에이전트를 생성할 때마다 생성된다. 생성된 각각의 키들 중 공개키는 기지에이전트에 의해 관리되며 각각의 개인키는 해당 작업에이전트가 생성될 때 분배된다. 기지에이전트는 각각의 작업에이전트에 대한 개인키를 삭제하고 공개키만을 유지함으로써 각각의 작업에이전트의 개인키의 유출을 막을 수 있다.

5.2 작업 에이전트 A와 기지 에이전트사이의 보안 채널 구성(step1)

작업 에이전트 A가 임의의 목적을 수행하는 알려지지 않은 작업 에이전트에게 메시지를 전달하기 위해서는 신뢰할 수 있는 인증 에이전트인 기지 에이전트에 의뢰를 한 후 기지 에이전트는 해당 목적에 부합한 작업 에이전트를 찾아 그 에이전트와 상호 인증을 한 후 메시지를 보낼 수 있다. 그러기 위해서는 우선 작업 에이전트 A와 기지 에이전트가 상호 인증 절차를 수행한 후 기지 에이전트에 의해 선택된 작업 에이전트 B에게 두 작업 에이전트간의 독립적인 보안 채널에서 사용하게

될 세션키를 전달하기 위한 보안 채널을 구성해야 한다.

우선 작업 에이전트 A는 기지 에이전트에게 상호 인증 절차 수행을 요구하는 ask_one 퍼포머티브를 보냄으로써 인증 절차가 시작된다.

ask_one
: sender 작업 에이전트 A
: receiver 기지 에이전트
: reply-with <expression>
: auth-digest (<MDT><Eka.s(MDMDT(α))>)
: content <EkBA.p(α)>

기지 에이전트를 인증하기 위해 작업 에이전트 A는 임의의 문자열 α를 생성하여 기지 에이전트의 공개키를 통해 암호화 한다. 또한 메시지 digest를 이용하여 메시지 무결성을 유지한다.

tell
: sender 기지 에이전트
: receiver 작업 에이전트 A
: in-reply-to <expression>
: reply-with <expression>
: auth-digest (<MDT><EkBA.s(MDMDT(β))>)
: content (<Eka.p(α)> <Eka.p(β)>)

기지 에이전트는 작업 에이전트 A에게서 받은 암호화된 임의의 문자열 α를 자신의 개인키를 통해 복호화 한 후 작업 에이전트 A를 인증하기 위해 생성한 임의의 문자열 β와 함께 작업 에이전트 A의 공개키를 이용해서 tell 퍼포머티브를 보낸다.

reply
: sender 작업 에이전트 A
: receiver 기지 에이전트
: in-reply-to <expression>
: reply-with <expression>
: auth-digest (<MDT><Eka.s(MDMDT(β))>)
: auht-key(<bool><key-type>
<EkBA.p(session-key)>)
: content <EkBA.p(β)>

tell을 받은 작업 에이전트 A는 자신의 개인키로 기지 에이전트로부터 받은 KQML 메시지를 복호화 한 후 자신이 보낸 임의의 문자열 α와 같으면 기지 에이전트를 인증하고 자신이 인증을 받기 위해 기지 에이전트에서 보낸 임의의 문자열 β를 복호화 한 후 기지 에이전트의 공개키를 통해 암호화해서 보낸다. 이때 작업 에이전트 A는 기지 에이전트 사이에서 구성될 보안 채널에서 사용할 세션키를 보낸다.

reply
: sender 기지 에이전트
: receiver 작업 에이전트 A
: in-reply-to <expression>
: reply-with <expression>
: auth-digest (<MDT><EkBA.s(MDMDT(β))>)
: auth-key(<bool><key-type>
<Eka.p(session-key)>)
: content <Eka.p(β)>

기지 에이전트로는 작업 에이전트 A부터 받은 임의의 문자열 β가 자신이 보낸 β와 비교해서 같으면 작업 에이전트를 인증하고 작업 에이전트 A와 기지 에이전트 사이에 구성된 보안 채널에서 사용할 세션키를 보낸다.

지금까지 작업 에이전트 A와 기지 에이전트 사이에 상호 인증 과정을 통해 보안 채널이 확립되고 확립 과정에서 주고받은 세션키를 통해 암호화된 KQML 메시지로 통신한다. 이때 KQML 메시지는 작업 에이전트 A가 기지 에이전트에 의해 선택되어질 알려지지 않은 작업 에이전트 B에 보낼 KQML메시지를 포함한 중첩된 KQML 메시지이며 이 메시지의 :content에는 작업 에이전트 A와 B사이에 보안 채널 구성 후 메시지 암호화를 위해 사용될 세션키가 포함되어 있다. 또한 메시지 재전송(replay) 공격을 막기 위해 다음에 보낼 메시지의 ID를 기지에이전트의 공개키를 통해 암호화하여 전송한다. 이 때 전송되는 메시지 ID는 랜덤 값이다.

tell
: sender 작업 에이전트 A
: receiver 기지 에이전트
: auth-digest (<MDT><Eka.s(MDMDT(T))>)
: auth-msg-ID (<msg-id>
<EkBA.p(next-msg-ID)>)
: content <EkBA.p (tell
: sender 작업 에이전트 A
: receiver unknown에이전트
: reply-with <expression>:
: auth-digest (<MDT><Eka.s(MDMDT
(session-key))>)
: content <Ekb. p(session-key))>

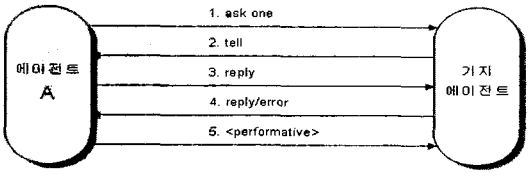


그림 8 보안 채널 구성(step1)

5.3 기지 에이전트와 작업 에이전트 B사이의 보안 채널 구성(step2)

기지 에이전트는 작업 에이전트 A가 보낸 tell 퍼포머티브를 통해 작업 에이전트 A가 어떤 목적을 수행하는 작업 에이전트에게 메시지를 보내려고 한다는 사실을 안다. 따라서 기지 에이전트는 자신이 가지고 있는 정보를 통해 작업 에이전트 A가 요구하는 목적의 임무를 수행하는 작업 에이전트 B를 찾아내고 작업 에이전트 A의 tell에 중첩된 KQML 메시지를 작업 에이전트 B에게 보내기 위해 기지 에이전트와 작업 에이전트 B사이의 상호 인증 과정을 통한 보안 채널을 구성한다. 따라서 작업 에이전트 A와 기지 에이전트간에 상호 인증을 위해 수행된 메커니즘을 통해 기지 에이전트와 작업 에이전트 B사이의 상호 인증이 이루어지고 기지 에이전트는 작업 에이전트 A의 tell에 중첩되어 있는 KQML 메시지를 이 보안 채널을 통해 tell 퍼포머티브를 통해 작업 에이전트 B에게 보낸다.

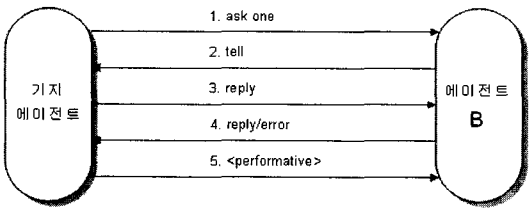


그림 9 보안채널 구성(step2)

5.4 작업 에이전트 A와 B 사이의 보안 채널구성(step3)

작업 에이전트 A가 작업 에이전트 B에게 메시지를 보내기 위해서는 신뢰할 수 있는 에이전트인 기지 에이전트를 통해 작업 에이전트 A와 B는 상호 인증된다. 이렇게 신뢰할 수 있는 기지 에이전트를 통해 작업 에이전트간의 상호 인증을 수행함으로써 절차상에 효율은 떨어지지만 에이전트 보안 및 보호의 정도를 높일 수 있다. 기지 에이전트에 의해서 상호 인증이 끝난 작업 에이전트 A와 B는 메시지 전송의 효율성을 높이기 위해 별도의 인증 절차 없이 두 에이전트간에 보안 채널을 구성한다. 또한 메시지 재전송 공격을 막기 위해 다음에 보낼 메시지의 ID를 미리 약속한 세션키를 통해 암호화한 후 전송한다.

tell
: sender 작업 에이전트 A
: receiver 작업 에이전트 B
: auth-digest (<MDT>
<Eka.s(MDMDT(MSG))>)
: auth-msg-ID (<msg-id>
<Ek.session-key(next-msg-ID)>)
: content <Ek.session-key(MSG)>

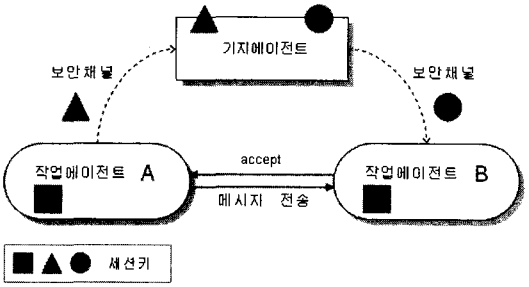


그림 10 두 에이전트간의 직접 보안 채널 구성(step3)

6. 결 론

최근 들어 컴퓨터 해킹에 대한 대응은 침입 탐지와 같은 소극적 방어보다는 역 추적과 같은 능동적인 방향으로 가고 있다. 본 논문에서는 보다 능동적인 침입 대응 방안에 활용할 수 있는 침입자 역 추적 시스템을 설계하였다. 이러한 시스템은 자율적으로 작업을 수행해야 하기 때문에 멀티 에이전트의 개념을 도입하였고 각 에이전트 사이의 통신은 KQML을 이용하였다. 또한 피침입 시스템으로부터 침입 경유 시스템과 최종 침입 발원 시스템으로의 추적을 위한 역 추적 가상 시나리오를 기술하였고 시나리오에 따른 에이전트간의 메시지 전송 및 침입 경유 시스템상의 에이전트를 보호하기 위한 모델을 제시하였다. 이를 위해서 에이전트 사이의 KQML 메시지를 보호하기 위한 통신 보안 메커니즘을 제안하였는데 공개키 암호화 방식을 도입하였으며 중개 에이전트가 통신을 원하는 에이전트 사이의 상호 인증을 대신 함으로서 성능보다는 보안성에 치중하였다. 또한, 본 시스템의 설계는 역 추적 과정의 가상 시나리오를 바탕으로 하여 상위 레벨에서 이루어졌으며, 하부에서 필요로 하는 핵심 요소 기술은 해결된 것으로 가정하였다.

참 고 문 헌

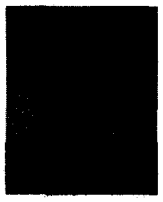
[1] Asaka, M., Taguchi, A., Goto, A., "Implementation of IDA: An Intrusion Detection Agent System," http://www.ipa.go.jp/STC/IDA/paper/first.ps.

[2] J.S.Balasubramaniyan, J.O.Garcia-Fernandez, D.Isacoff, E.Spafford,and D. Zamboni, "Architecture for

Intrusion Detection using Autonomous Agents," COAST Technical Report, COAST Laboratory, Purdue University, 1998.
[3] N. Bhandaru and W. Croft, "An architecture for supporting goal-based cooperative work," In Gibbs S. and Verrijn-Stuart A., eds., Multi-User Interfaces and Applications, pp 337-354, Elsevier Science Publishers B.V., North-Holand, 1990.
[4] P. Stone and M. Veloso, "Multiagent Systems: A Survey from a Machine Learning Perspective," Technical Report CMU-CS-97-193, School of Computer Science, Carnegie Mellon University, Pittsburg, PA 15213, 1997.
[5] M. N. Huhns and L. M. Stephens, "Multiagent Systems and Societies of Agents," In Multiagents Systems. A Modern Approach to Distributed Artificial Intelligence. Weiss, Gehrard, ed. Cambridge, Mass., MIT Press, pp 79-120, 1999.
[6] H. Chalupsky, T. Finin, R. Fritzson, D. McKay, S. Shapiro, and G. Weiderhold, "An overview of KQML: A knowledge query and manipulation language," Technical report, KQML Advisory Group, April 1992http://www.csee.umbc.edu/kqml/papers/kqmloverview.ps.
[7] T. Finin, R. Fritzson, D. McKay, and R. McEntire, "KQML: An information and knowledge exchange protocol," In K. Funchi and T. Yokoi, editors, Knowledge Building and Knowledge Sharing. Ohmsha and IOS Press, 1994.
[8] R. Feiertag, C. Kahn, P. Porras, D. Schnackenberg, S. Staniford-Chen, B. Tung, "A Common Intrusion Specification Language(CISL)," 10 March 2000, http://www.gidos.org/drafts/language.txt
[9] S. Staniford-Chen, B. Tung, and D. Schnackenberg, "The Common Intrusion Detection Framework(CIDF)," Position paper accepted to the Informatio Surviv-ability Workshop, Orlando FL, October 1998.
[10] J. C. Corbett, "The S-expression design language (SEDL)," ICS-TR-93-02, Information and Computer Science Department, University of Hawaii at Manoa, 1993.
[11] R. Rivest, "S-expressions," Internet Draft draft-rivest-sexp-00.txt, 1997.
[12] C. Thirunavukkarasu, T. Finin, J. Mayfield, "Secret Agents - A Security Architecture for the KQML Agent Communication Language," Proc. of CIKM '95 Intelligent Information Agents Workshop, 1995.
[13] Q. He, K. P. Sycara, "Personal Security Agent: KQML-Based PKI," to appear in Autonoumous Agents'98, Mineneapolis/St. Paul, May 10-13, 1998.
[14] Hando Kim, Min Soo Kim, Yeongho Kim, Suk Ho Kang, "Design of SKAP(Secure KQML Agent Protocol)," Journal of the Korean Institute of Industrial Engineers(JKIIE), '98 Fall Academic Conference, 1998.

최 진 우
1998년 한성대학교 전산학과 졸업(학사) 2000년 국민대학교 대학원 전산과학과 졸업(석사). 2000년 ~ 현재 국민대학교 대학원 전산과학과 박사과정. 관심분야는 인공지능, ITS, 에이전트, 정보 보호

황 선 태
1985년 서울대학교 전자계산기공학과 졸업(학사). 1987년 서울대학교 대학원 전자계산기공학과 졸업(석사). 1996년 맨체스터대학교 대학원 전산학과 졸업(박사) 1997년 ~ 현재 국민대학 컴퓨터학부 교수. 관심분야는 병렬처리, 시스템 소프트웨어, 그리드 시스템



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 국민대학교 컴퓨터학부 교수. 관심분야는 인공지능 지능형 교육시스템, 에이전트, 정보보호

정 주 영
2001년 2월 한국외국어대학교 컴퓨터공학과(공학사). 2001년 3월 ~ 현재 숭실대학교 컴퓨터학과 석사과정. 2001년 3월 ~ 현재 국가보안기술연구소 연구원 관심분야는 IPV6, 정보보호 대응

최 대 식
1997년 강원대학교 전자계산학과(이학사) 1999년 강원대학교 전자계산학과(석사) 2000년 ~ 현재 국가보안기술 연구소 연구원. 관심분야는 알고리즘, FTN, 정보보호 대응