침입자 추적을 위한 적극적 대응체계

박기형*, 최진우*, 하성진*, 황선태*, 우종우*,
고재영**, 정주영**, 최대식**
*국민대학교 컴퓨터학부
**한국전자통신연구소 부설 국가보안기술연구소
e-mail : cwwoo@kookmin.ac.kr, dschoi@etri.re.kr

Active Response Mechanism for Tracing Intruders

Ki-Hyung park*, Jinwoo Choi*, Sungjin Ha*, Suntae Hwang*, Chongwoo Woo*
Jaeyoung Koh**, Juyoung Jung**, Daesik Choi**
*School of Computer Science, Kookmin University
**National Security Research Institute

요 약

인터넷의 발달과 더불어, 인터넷을 통한 전자상거래, 홈 뱅킹 등의 서비스가 급속히 성장하면서 이러한 사회기반 시설에 대한 침입피해 사례가 급증하고 있는 추세이다. 이러한 침입 피해를 방지하기 위하여 방화벽과 침입탐지 시스템이 개발되어 설치되고 있으나, 침입피해를 감소 시키기에는 문제점들이 있다. 본 논문에서는 기존 침입탐지 시스템들의 소극적인 탐지기능에서 벗어나, 보다 적극적으로 침입에 대응할 수 있는 방안을 연구 하였다. 즉, 침입탐지가 되면, 즉시 침입자들을 역으로 추적할 수 있는 시스템을 설계하였다. 또한 이러한 시스템은 자율적으로 행동하여야 하기 때문에 멀티 에이전트 기반 시스템으로 설계하였다.

1. 서론 [1]

인터넷의 발달과 더불어 최근 네트워크 상에서의 침입 유형은 급속도로 다양화되어 가고 있다. 또한, 인터넷을 통한 전자상거래, 홈 뱅킹 등의 서비스가 급속히 성장하면서 이러한 사회기반 시설에 대한 침입 피해 사례가 급증하고 있는 추세이다. 이러한 침입 피해를 방지 하기 위하여 방화벽(Firewall)과 침입탐지 시스템(Intrusion Detection System: IDS)이 개발되어 설치되고 있다. 방화벽에 비해 IDS 는 보다 능동적으로 침입사실을 탐지 할 수 있으나 침입피해를 감소하거나 완전히 방어 하기에는 문제점들이 있다. 예를 들면, 첫째, 기존의 IDS 는 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. 둘째, 침입피해가 점차 심각해가는 현실에 비해 현재의 침입 탐지이상으로 적극적인 대응방안이 없는 문제점이 있다. 침입자가 시스템에 피해를 주는 상황에서도 적극적으로 물리치거나 방어할 수 있는 방법이 부재하다고 할 수 있다. 이러한 문제점을 극복하기 위하여 해킹 피해 시스템에 대한 분석과 대응을 인간의 수동적인 분석 및 대응을 의존 하기보다는, 자동적인 분석 및 대응이 가능한 시스템에 관한 연구가 요구되고 있다.

본 논문에서는 이러한 기능들을 멀티 에이전트 개념을 도입 함으로서, 보다 자율적이며, 지능적이고, 적극적인 침입 탐지 시스템을 설계하고자 한다. 이러한 시스템의 구현을 위해서 우선, 침입 상황 시에 적극적으로 대처할 수 있는 방안으로 '역 추적(tracing intruder)'을 우선적으로 제안 하였고, 또한 역 추적을 체계적으로 구현하기 위해서 시스템의 전반적인 구현에 앞서, 가상의 역 추적 시나리오를 우선 설계하였다.

이 연구는 ETRI 부설 국가보안 기술연구소 2001 년 위탁 연구과제에서 지원 받았음

2. 관련연구

2.1 침입탐지 시스템

침입 탐지란 불법적인 행위를 실시간으로 감시하여 탐지 하는 것을 말하며, 이러한 행위를 실시간으로 탐

지하여 보고하는 시스템을 침입탐지 시스템이라 한다. 방화벽이 단순히 불법적인 접근을 막는데 중점을 두고 있다면 침입탐지 시스템은 방화벽이 효과적인 차단에의 실패 시, 이에 따른 피해를 최소화하고 관리자가 없을 때에도 불법적인 침입에 적절히 대응할 수 있는 보안 해결방안 이라고 할 수 있다.

그러나, 모든 침입탐지 작업이 단일화 되어 있어 과중한 부하의 유발과 탐지와 대응 모듈의 오 동작 및 파괴에 따른 안정성 문제 등이 있다. 따라서 기존의 침입탐지 시스템이 가지고 있는 여러 가지 단점을 보완하기 위해, 최근에는 에이전트 개념을 도입한 IDS 시스템이 개발되어지고 있다. 대표적인 시스템으로는 일본 와세다 대학의 IDA(An Intrusion Detection Agent System)[1]시스템이 있고, 미국의 퍼듀 대학의 COAST Laboratory 에서 진행해 온 연구인 AAFID(Autonomous Agent For Intrusion Detection)[2]시스템이 있다. IDA 는 이동 에이전트(mobile agent)에 의해 침입자들을 추적할 수 있는 기능과 침입한 경로를 따라 침입과 관련된 정보를 수집하는 기능을 가진다. 그리고 IDA 는 이런 수집된 여러 정보들을 통해서 침입의 발생여부를 결정하는 역할 또한 수행한다. AAFID 의 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 보다 분산 환경에서의 수행과 독립적인 에이전트의 기능 면에서 IDS 와 다르지만 역 추적 등 능동적 방안을 제시하고 있지 못하다.

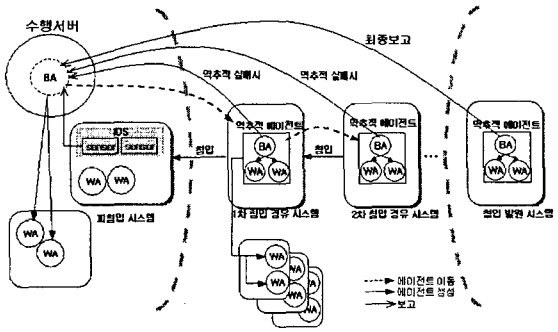
2.2 멀티 에이전트

IDS 가 가지고 있는 구조적 문제점의 해결방안으로 최근에는 멀티 에이전트 개념을 적용한 역 추적 시스템이 연구되고 있다. 멀티 에이전트 시스템은 자치성을 가지는 여러 개의 소프트웨어 에이전트를 동적으로 통합하여 동시에 수행 가능하게 함으로써 한 개의 중앙 집중화 된 에이전트 시스템이 해결하기에는 너무 큰 문제 또는 복잡한 문제를 해결가능 하도록 하였다. 에이전트 시스템의 장점은 인공지능에서 연구되어온 많은 연구 결과인 지식베이스, 추론 능력을 가지고 있어 지능적으로 행동할 수 있으며, 또한 네트워크를 통한 분산된 에이전트 끼리 협동하여 작업을 수행하기도 한다 [3][4][5].

따라서, 이러한 분산된 환경 하에서 에이전트들 끼리 작업을 수행하기 위해서는 에이전트간 통신을 위해 프로토콜이 요구 되는데 이를 KQML(Knowledge Query and Manipulation Language)[6][7]이라 한다. KQML 은 분산되어 있는 지능적 소프트웨어 에이전트들 간의 상호 통신을 지원하기 위하여 설계된 언어이다. 이러한 에이전트 기반 통신 프로토콜을 사용함으로써 에이전트간 통신 전반에 대한 의미 구조를 설계 가능하며 보다 효율적인 멀티 에이전트 시스템 구현이 가능하다. 본 논문에서는 이러한 장점이 있는 멀티 에이전트 개념을 적용하여 역 추적 에이전트 시스템을 설계 한다.

3. 역 추적 시나리오

역 추적 시스템은 다음 [그림 1]과 같이 구성하였고 가상적으로 추적 시나리오를 설계하였다.

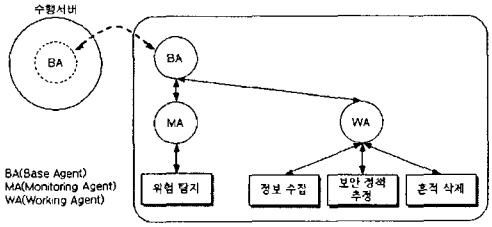


[그림 1] 역 추적 시스템 구성도

3.1 역 추적 시스템

역 추적 시스템의 구성은 수행 서버, 침입 경유 시스템, 침입 발원 시스템들과, 역 추적의 기능을 수행할 다수의 에이전트들로 구성된다[그림 1]. 각 시스템들의 기능은 다음과 같다.

- 수행 서버: 침입 탐지 시스템과 에이전트를 제어하는 역할을 한다.
- 피침입 시스템: 실제 침입을 당한 로컬 시스템 내의 서브 시스템을 말한다.
- 침입 경유 시스템: 침입자가 피침입 시스템에 침입을 하기 위한 중간 경유 시스템이다.
- 침입 발원 시스템: 피침입 시스템에 침입한 실제 발원 시스템이다.



[그림 2] 역 추적 메시지 시나리오

3.2 메시지 시나리오

메시지 시나리오란 침입탐지가 된 후, 적극적으로 침입피해상황에 대처하기 위해, 침입자를 역 추적하는 과정을 단계별로 정의한 것이다. 메시지 시나리오를 쉽게 이해하기 위해 위 [그림 1]의 역 추적 시나리오 구성도를 기능적 측면에서 [그림 2]와 같이 단순화 시켰으며, 단계별 시나리오는 다음과 같다.

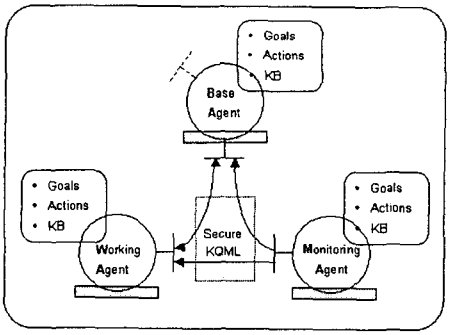
- 제 1 단계, IDS 가 침입을 탐지
- 제 2 단계, IDS 가 수행서버에게 침입을 보고
- 제 3 단계, 수행서버는 침입 발원지를 분석
- 제 4 단계, 기지 에이전트가 침입 경유 시스템으로 이동

- 제 5 단계, 이동한 에이전트가 감시 에이전트와 작업 에이전트를 실행
- 제 6 단계, 실행된 감시 에이전트와 작업 에이전트는 각각 에이전트 보호를 위해 위험을 탐지하고 정보를 수집하는 활동
- 제 7 단계, 필요한 정보를 수집한 에이전트는 수행을 마치고 다음 침입 경유 시스템으로 이동하기 전에 히스토리 파일과 로그 파일 삭제한 후 이동
- 제 8 단계, 에이전트가 이동한 시스템이 침입 발원 시스템일 에이전트는 목표를 마치고 수행서버에 그동안 의 정보를 수행서버에 전송

이러한 역 추적 메시지 시나리오에 기반 하여 다음과 같이 멀티 에이전트 시스템을 설계하였다

4. 역 추적 에이전트 시스템 설계

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



[그림 3] 역추적 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 실행되고, 수행 중인 작업 에이전트의 요구 시 새로운 작업 에이전트로 생성될 수 있다. 작업 에이전트는 침입 경유지 시스템의 보안 정책에 대한 추정, 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 따른 재구성된 작업 목표를 기지 에이전트로부터 수신하여 자신의 작업에 반영할 수 있다. 그리고 자신의 수행 절차를 기록하여 작업 실패 시 기록을 기지 에이전트에게 전송하게 된다.

4.1.3 감시 에이전트(Monitoring Agent)

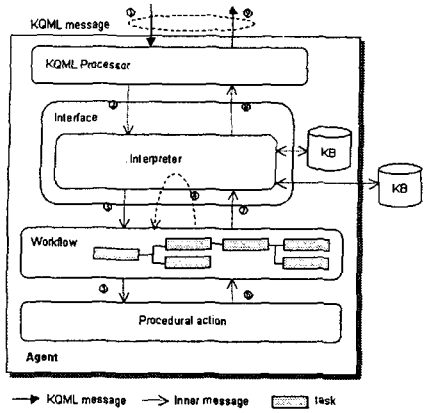
감시 에이전트는 자기 자신과 다른 에이전트의 보호를 위하여 독립적으로 시스템을 감시하는 에이전트로 기지 에이전트에 의하여 초기에 생성되어 진다. 다른 에이전트들과 기본 구조는 동일하지만, 에이전트간 통신 메커니즘에서 그 차이점이 명백히 구별된다. 차이점은 다른 에이전트들은 상호 통신, 즉 양방향 통신을 하는데 반하여, 감시 에이전트는 단 방향 통신을 수행한다. 다른 에이전트들에 의하여 생성된 메시지의 유입은 없으며, 침입 경유 시스템 내에서 발각이 추정되는 경우에 경보 메시지를 동작 중인 모든 에이전트에게 전파하는 기능을 담당한다.

4.2 에이전트의 보호

모든 에이전트간 상호 통신 시에는 보안 구조가 확립된 메시지를 상호 교환되어야 함으로써 에이전트와 에이전트, 에이전트와 플랫폼, 플랫폼과 에이전트 사이의 통신 중에서의 보안이 요구된다. 물론 공격 근원지에서 수행 중인 에이전트, 에이전트 플랫폼의 보안을 위한 은닉성도 요구된다. 그러나 만일 누군가에 의해 수행 중인 에이전트가 발각된 후, 이로 위장한 악성 에이전트를 생성하여 수행 서버와의 통신을 시도하려 한다면 에이전트를 파견한 위치의 노출과 또는 역 공격에 의한 기밀 유출 등과 같은 더 큰 피해를 우려할 수 있다. 이러한 이유에서 메시지 계층에서의 통신 보안이 보다 더 큰 비중을 차지할 수 있다.

4.3 에이전트 구조 (Agent Architecture)

본 논문의 에이전트 설계는 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

4.3.1 KQML 프로세서

에이전트로 유입된 KQML 메시지(①)는 KQML 프로세서에 의하여 메시지 분석이 된다. 메시지 분석은 명세서를 바탕으로 분석되며, 명세서는 설계자에 의하여 확장될 수 있다. 명세서에는 KQML에 의하여 분석되어지는 :language 필드에 명시된 내용을 독립적으로 설계 가능하게 함으로써 사용 언어의 해석기의 선택시 그 유연성을 제공한다.

4.3.2 인터페이스

인터페이스로 유입되는 KQML 메시지(②)는 퍼포머티브(performative)의 분석을 담당한다. 기본적으로 제공되어야 KQML 퍼포머티브를 비롯하여 설계자에 의하여 확장, 추가된 퍼포머티브를 분석 가능하도록 설계되었다. 이러한 퍼포머티브의 확장은 본 시스템의 보안에 관련된 확장과 그 추가에서 그 예가 반영되며, 또한 내부 언어를 위한 확장을 위해서 설계되었다. 인터프리터는 KQML 프로세서를 통해 처리된 :content 필드에 명시된 내용을 해석하는 부분이며 파서(parser)와 수행을 위한 오퍼레이션과의 연동을 위한 엔진 부분인 바인더(binder), 그리고 추론부(reasoner)로 구성되어 있다. 에이전트 도메인 지식은 에이전트의 외부 지식으로 명시 가능하도록 설계하여 그 확장성을 제공하였다.

4.3.3 워크 플로우

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

5. 결론

본 논문에서는 최근 급증하고 있는 컴퓨터 침입 피해를 최소화 할 수 있는 방안을 연구하였다. 기존 IDS 들은 기본적인 침입탐지를 수행하고 있으나, 급증하는 침입피해 사례를 감소 시킬 수는 없다. 따라서 기존 IDS 들의 소극적인 탐지기능에서 벗어나, 보다 적극적으로 침입에 대응할 수 있는 방안으로, 침입탐지가 되면, 즉시 침입자들을 역으로 추적할 수 있는 시나리오를 중심으로 시스템을 설계하였다. 시나리오는 상위 레벨에서 이루어졌으며, 하부에서 필요로 하는 핵심 요소 기술은 해결된 것으로 가정하였다. 또한 이러한 시스템은 자율적으로 행동하여야 하기 때문에 멀티 에이전트의 개념을 도입하여 설계 하였다. 향후 과제는 역 추적을 수행하는 에이전트들 사이의 이동 및 보안에 관한 연구가 수행 되어야 할 것이다.

참고문헌

[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.gz

[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/kqmlovervi ew.ps.

[7] T. Finin, R. Fritzson, D. McKay, and R. McEntire, "KQML: An information and knowledge exchage protocol," In K. Funchi and T. Yokoi, editors, Knowledge Building and Knowledge Sharing. Ohmsha and IOS Press, 1994.