제 19 회 한국정보처리학회 춘계학술발표대회 논문집 제 10 권 제 1 호 (2003. 5)

액티브 네트워크 환경에서의 에이전트 기반 침입탐지 시스템

최진우, 우종우
국민대학교 컴퓨터학부
e-mail : cwwoo@kookmin.ac.kr

Agent-based IDS in the Active Network Environment

Jinwoo Choi, Chongwoo Woo
School of Computer Science, Kookmin University

요 약

단일 호스트 환경에 특화되어 설계되어온 기존 침입탐지 시스템(Intrusion Detection System: IDS)은 침입 시 도메인의 보호만을 그 목적으로 하는 수동적인 성격으로써, 새로운 공격 기법에 대한 탐지 및 대응, 그리고 보다 그 규모가 큰 네트워크로의 확장 면에서 구조적인 결함을 가지고 있다. 이러한 IDS 의 구조적 문제점의 해결방안으로 액티브 네트워크 기반의 IDS 에 관한 연구가 진행되고 있다. 액티브 네트워크(Active network)란 패킷 스위칭 네트워크 상에 프로그램 가능한 라우터 등인 액티브 노드들을 배치하고, 사용자의 요구에 상응하는 적절한 연산을 위한 데이터와 프로그램으로 구성된 스마트 패킷(smart packet)에 대하여 수행 가능하게 하는 접근 방법이다.

본 논문에서는 이를 기반으로 자율적이며 지능적인 에이전트로 구성된 멀티 에이전트 기술을 액티브 노드에 적용함으로써 기존 IDS 보안메커니즘에서 보다 더 진보된 능동적이고, 적극적인 대응을 위한 보안 메커니즘을 제공하여 네트워크 공격에 의한 피해 최소화와 신속한 대응이 가능한 멀티 에이전트 기반 공격 대응 메커니즘을 제시하고, 이를 적용 가능한 액티브 네트워크 기반 프레임 설계를 제안한다.

1. 서론

최근 네트워크 환경을 침해하는 부정적인 사례들이 국내외에서 빈번히 발생하고 있으며, 이에 대한 대응책으로 침입 탐지 기술의 연구가 많이 수행되고 있다.이들의 연구 방향은 수동적이고 방어적이기 보다는 능동적이고 공격적으로 변모해 하고 있으며, 이러한 연구의 중심은 공격자로부터 공격이 감행된 근원지를 추적해 나아가는 방향으로 진행해 나아가고 있다. 이러한 능동적인 보안 메커니즘은 이론적으로는 합당한 반면, 실제 적용 시 다음과 같은 가장 큰 두 가지의 문제점을 안고 있다.

첫째, 공격자들에 의한 공격이 단일 시스템에서 이루어지는 것이 아닌 인터넷의 일부를 자신의 공격을 위한 분산 거점으로 삼아 표적이 되는 호스트로의 공격을 감행하기 때문이다. 이러한 대표적인 공격이 분산 서비스 거부(DDoS, Distributed Denial of Service) 공격이다. 또한 최근에는 인터넷 일부를 징검다리 호스트들로 사용함으로써 공격에 이용된 시스템이 자신이 공격에 가담 되어진 사실 또한 모르게 이용당하는 실정이어서 공격 근원지로의 공격 경로를 역추적한다는 것은 매우 어려운 문제이다.

둘째, 기존 IDS는 단일 호스트 환경에 특화되어 설계되어 공격 시 도메인의 보호에 치중된 수동적인 성격이므로, 대규모의 네트워크 환경에서의 침입 탐지 및 대응 면에서 구조적인 결함을 가지고 있다.

이에 본 논문에서는 액티브 네트워크 환경 하에 자율적이며 지능적인 에이전트로 구성된 멀티 에이전트 기술을 액티브 노드에 적용함으로써 기존 IDS 보안메커니즘에서 보다 더 진보된 능동적이고, 적극적인 대응을 위한 보안 메커니즘을 제안한다.

논문 구성은 공격 대응 메커니즘에 초점을 두는 대표적인 연구사례들과 이들이 명시하는 공통적인 문제점, 그리고 그 방안을 제 2장. 관련 연구에서 기술하고, 제 3장에서 본 논문에서 제안하는 시스템의 설계, 그리고 4장. 결론을 통해 맺는다.

2. 관련 연구

본 장에서는 대표적인 공격 대응 메커니즘들을 명시하는 연구 사례들과 이를 위한 대안으로 제시되는 기술들을 소개한다.

2.1 공격 대응 메커니즘

최근의 공격 대응 메커니즘은 스위칭 구조의 네트워크를 경유하여 패킷이 이동하여 온 정상적인 흐름을 역으로 그 패킷이 통과해온 통로를 추적해 나가는 패킷 근원지 식별(packet source identification)에 관한 문제에 그 핵심을 두고 있다. 이러한 문제에 있어서의 다양한 후보 해결 방안들이 소개되어 오고 있으며, 이들은 다음과 같이 분류되어 질 수 있다.

첫째, 패킷 사이즈의 증가 없이 기존 패킷 자체의 내부에서의 경로(route) 식별을 문제로 하는 접근 방안으로써 라우팅 장치들의 IP 주소에 근거를 두는 코드를 패킷 자체 내부에 삽입하여 이를 기반으로 문제를 해결하는 방안을 제안하고 있다[1].

둘째, 라우팅 장치들로부터 생성되는 가외 패킷(extra packet)들을 사용하여 경로 식별을 문제로 하는 접근 방안으로써 라우터들이 제어하는 각 패킷에 관하여 추적을 위한 2차적인 추적 패킷(ICMP traceback message, iTrace)을 발생 가능한 라우팅 장치들을 이용하는 방안을 제안하고 있다[2].

셋째, 보안 인증을 확립하기 위한 표준 프로토콜을 기반으로 하는 프레임워크를 이용하여 문제를 해결하려는 접근 방안으로써 보안 인증 확립을 위한 IPSec(IP Security protocol)과 인증 헤더(AH, Authentication Header)를 가지고 확립된 IPSec SA(Security Association)을 이용하는 방안을 제안하고 있다[3].

넷째, 라우팅 장치들에 의해 제어 가능한 트래픽을 직접 질의하여 해결하려는 접근 방안으로써 라우터는 응답 가능한 주변 라우터들에게 질의하고, 이를 상위(upstream) 라우터들에게 반복 적용하는 방안을 제안하고 있다[4].

다섯째, 액티브 네트워크 환경을 위하여 프로그래밍 가능한 노드(router, gateway)를 기존 네트워크 구성에 참여 시킴으로써 중간 단계의 네트워크 구성요소에서 보다 능동적인 연산을 수행하도록 하는 방안을 제안하고 있다[5][6][7].

이들의 연구에서 비롯하는 공통적이고, 기본적인 문제는 라우터로 진입하는 패킷들을 네트워크 레벨에서 제어 하는 것이다. 그러나 기존 라우터들과 같은 네트워크 엔터티들에서의 패킷 처리는 매우 제한적이다. 예를 들어 이들에 의해서 진입 패킷의 헤더를 제어할 수 있더라도, 이는 필터링 정도에 불과하고 특히 패킷의 데이터 영역에 관한 검사 없이 대부분 단순한 전송만을 담당하고 있다. 이를 위한 대안으로 라우팅 장치와 같은 네트워크 구성요소 상에서 NBAR(Network Based Application Recognition)과 같은 보다 적극적인 제어가 필수적이라는 것은 사실이며, 액티브 네트워크는 이러한 요구를 충분히 만족시킨다.

2.2 액티브 네트워크

액티브 네트워크(Active network)란 패킷 스위칭 네트워크에 프로그램 가능한 라우터 또는 스위치를 배치함으로써 전송되는 패킷들을 특정 서비스나 사용자의 요구에 적합하게 연산할 수 있는 차세대 네트워크 구조에 대한 새로운 접근 방안이다. 기존 네트워크에서의 네트워크 노드의 역할은 패킷을 목적지까지 전달하는 단조로우며 수동적인 기능인 반면, 액티브 네트워크 환경에서는 중간 노드인 액티브 노드들로 하여금 패킷에 대한 연산 및 수정 가능하도록 한다. 더욱이 이러한 절차들은 사용자마다 혹은 서비스 응용마다 각각의 특화된 연산을 제공해야만 한다[8][9][10].

이에 관하여 액티브 네트워크 설계 시 각 노드 상에서 이동 코드를 수행시켜줄 수 있는 수행환경과 이동 코드의 개발에 관한 기술언어 또한 고려해야만 한다. 액티브 네트워크에서의 이동 코드는 이종의 다양한 노드들에서 실행되어야 하고, 신뢰할 수 있고 안정성이 뛰어나야 하므로 이동코드 기술 언어는 이식성, 안정성, 보안성, 그리고 효율성과 같은 특징을 가져야만 한다.

최근에는 액티브 네트워크 환경으로의 에이전트 기술을 적용한 연구와 개발이 활발히 이루어지고 있다.

2.3 멀티 에이전트 시스템

단일 호스트에서의 침입탐지시스템은 개인 시스템 환경에 특화되어 설계되어 대규모의 네트워크로의 확장 시 문제점을 가지고 있다. 이러한 IDS가 가지고 있는 구조적 문제점의 해결방안으로 최근에는 멀티 에이전트 개념을 적용한 침입 탐지 시스템들이 연구되고 있다.

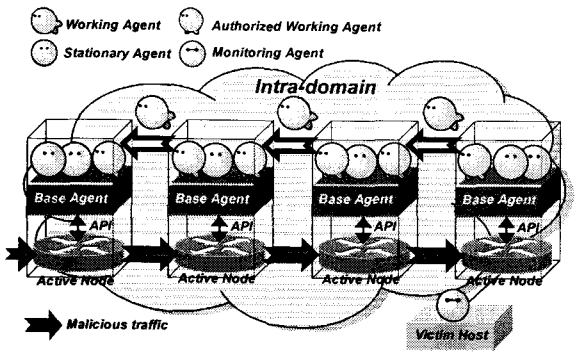
멀티 에이전트 시스템이란 한 개의 중앙 집중화 된 시스템이 해결하기에는 그 규모가 너무 크거나 복잡한 문제를 자율성을 가지는 다수의 서브 에이전트들이 동적으로 분산된 임무를 수행 가능하게 함으로써 해결 가능하도록 하는 시스템이다.

에이전트 시스템의 장점은 인공지능에서 연구되어 온 많은 연구 결과인 지식베이스, 추론 능력, 결정 지원 등의 기능을 가지고 있어 지능적으로 행동할 수 있다는 점과 네트워크를 통해 분산된 에이전트 끼리 협동하여 작업을 수행할 수 있다는 점을 들 수 있다[11][12][13]. 분산 환경 하에서 에이전트 상호 작업을 수행하기 위해서는 에이전트간 통신을 위한 프로토콜이 요구 되는데 KQML(Knowledge Query and Manipulation Language)[14][15]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 설계

하는 것이 가능하며 보다 효율적인 멀티 에이전트 시스템 구현이 가능하다. 그러나 다양한 응용 분야에서의 멀티 에이전트 시스템 적용 시 에이전트간 상호 협력을 위한 통신에서 반드시 고려해야 하는 문제점이 바로 메시지의 보호이며, 특히 네트워크 보안 도메인에서 더욱 그러하다[17].

3. 액티브 네트워크 기반 멀티 에이전트 시스템 설계

네트워크 공격에 관한 대응 메커니즘을 위한 액티브 네트워크 기반의 멀티 에이전트 시스템의 구성은 은 크게 에이전트 기지에 해당하는 기지 에이전트(Base Agent), 이동 에이전트에 해당하는 작업 에이전트(Working Agent), 그리고 네트워크 자원을 제어하는 고정 에이전트(Stationary Agent)로 구성된다[그림 1].



[그림 1] 액티브 네트워크 기반의 멀티 에이전트 시스템

3.1 대응 시나리오

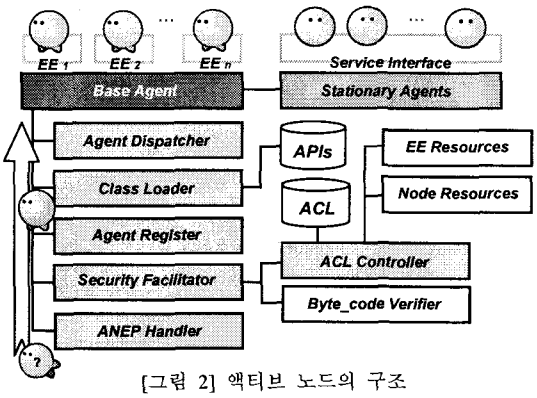
네트워크 공격이 탐지된 후, 공격에 대처하기 위한 즉각적인 단계별 대응 과정은 다음과 같다.

- 제 1단계, 피해지 호스트의 감시 에이전트 침입 탐지
- 제 2단계, 감시 에이전트 상위의 액티브 노드에게 침입 보고
- 제 3단계, 기지 에이전트에 의해 작업 에이전트 생성
- 제 4단계, 작업 에이전트가 자신의 임무를 가지고 이웃한 액티브 노드들로 이동
- 제 5단계, 이동한 액티브 노드에서의 고정 에이전트와의 상호 협력을 통한 문제 해결
- 제 6단계, 이러한 과정을 경계 라우터의 액티브 노드까지 반복 적용

3.2 액티브 노드 구조

라우터 내부로의 진입 패킷이 액티브 노드로 유입되기 이전 먼저 ANEP(Active Network Encapsulation Protocol) 헤더를 검사한다. 만일 ANEP 헤더가 존재하면 ANEP 제어기(ANEP handler)에 의해 헤더의 처리 과정이 이루어지며, 그렇지 않은 경우에는 모든 처리 과정을 라우터에게 위임한다. 이를 위한 스위칭 과정은 IPSec 프로토콜을 다루는 FreeS/WAN[16]에서 IPSec을 위한 가상 네트워크 디바이스를 사용하는 것과 같은 메커니즘으로 설계하였다. 그리고 액티브 노드의 설계는 지금까지 발표되어온 많은 연구 결과로부터 공통적이며 기본적인 기반구조를 모델로 설계되었다

액티브 노드 내의 기지 에이전트는 알려진 모든 고정 에이전트들과의 통신을 수행한다. 이들 고정 에이전트 들로는 위에서 언급한 ANEP handler, API의 클래스의 로드를 담당하는 Class Loader, Java에서 제공하는 보안 메커니즘인 byte_code verifier를 제공하는 Security Facilitator, 시스템 자원으로의 접근 제어를 담당하는 ACL Controller, 작업 에이전트 전송을 위한 패킷 형태로 가공하는 Agent Dispatcher, 그리고 에이전트의 등록을 담당하는 Agent Register가 존재한다[그림 2].



[그림 2] 액티브 노드의 구조

3.3 기지 에이전트(Base Agent)

기지 에이전트(BA)는 액티브 네트워크 측면과 멀티 에이전트 시스템 측면에 관하여 두 가지 관점을 고려하여 설계하였다.

첫째, 액티브 네트워크 관점에서의 핵심 구성은 크게 두 가지로 에이전트 플랫폼(agent platform)과 에이전트 전송자(agent dispatcher)로 구분된다. 에이전트 플랫폼은 에이전트의 실행을 제어하기 위한 에이전트 실행 환경(agent execution environment)을 제공하며, 에이전트 전송자는 이웃한 액티브 노드들로의 에이전트 파견의 기능을 담당한다.

둘째, 멀티 에이전트 시스템 관점에서의 핵심은 에이전트 조정자(agent coordinator)의 역할로써 자신의 고유 기능을 명시하기 위한 작업 에이전트의 등록과 에이전트 자체의 보안과 상호간 통신을 위한 통신 보안 메커니즘을 제공한다.

BA의 수행절차는 다음과 같이 진행된다. 파견된 작업 에이전트(WA)는 BA의 "byte_code verifier"에 의해 코드 레벨에서의 보안이 입증된 뒤, 자신의 ID와 작업 목표를 함께 등록을 한다. 등록 이유는 BA 상의 유일한 WA임을 증명하기 위함이다.

3.4 작업 에이전트(Working Agent)

작업 에이전트(WA)는 이동 에이전트(mobile agent)의 기능에 충실한 에이전트로써 액티브 노드들 사이를 이동하여 다니며 자신의 임무를 수행하는 로빙 프로그램(roving program)이다.

이동 코드에 해당하는 모든 WA는 최상위 클래스로부터 계승한 객체로써, 이것이 의미하는 바는 실제 사용되는 API는 현재 생성된 시점의 노드에서 호출되는 것이 아니라, 파견된 노드 상에서 서비스 요구 시 호출되는 것이므로, 이들 WA들에 의해 사용되는 모든 API들은 액티브 노드 상에서 클래스의 형태로 제공되고, 이를 위한 인터페이스만이 WA내에 포함된다. 이를 위하여 이동 코드의 작성시 기술 가능한 API들에 관한 명세서를 설계하였다. 허가된 WA가 안착된 뒤에는 자신의 임무와 상응하는 서비스를 SA에게 요구하고, SA에 의해서 네트워크 자원들이 사용된다. WA는 수행 시 진행되는 작업 목표를 BA에 기록하고, 네트워크 환경 설정에 관하여 시스템 제어를 담당하는 준비된 고정 에이전트(SA)들과 협력하게 된다. 예를 들어 WA에 의해 기록된 작업 목표가 하위 네트워크의 블로킹인 경우 SA는 다운 스트림의 블로킹을 위하여 라우터의 "access-list"를 조정하게 된다.

3.5 감시 에이전트(Monitoring Agent)

감시 에이전트(MA)는 지능형 에이전트 기능에 충실한 에이전트로써 감사 데이터로부터 획득한 지식을 가지고 있으며, 피해지 호스트 상에 존재한다. 실질적인 액티브 노드의 구성요소로 참여하지는 않지만, 멀티 에이전트 시스템의 구성 요소이다. 일반적으로 네트워크 인터페이스를 경유하는 모든 패킷들은 저수준 데이터(raw data)이므로 MA는 이를 정형화된 표현식으로 표현하는 기능을 한다. 예를 들어 저수준 네트워크 모니터링인 "libcap"을 사용하여 획득한 데이터는 네트워크 패킷을 기술하고 있는 바이너리 포맷이므로, 우선적으로 이를 사람과 기계 가독의 지식 형태로 변환하는 처리 단계를 수행하게 된다. 이러한 MA는 다양한 공격 모델을 위해 감사 데이터로부터 학습한 지식을 이용하여 결정 지원 능력을 갖춤으로써 새로운 유형의 공격 시 예측 가능하도록 설계되었다.

4. 결론

현재 액티브 네트워크 기술은 여러 분야에 걸쳐 적용되고 있다. 특히 네트워크 보안 영역에 액티브 네트워크 기술을 적용하는 것은 기존의 네트워크 기반 IDS 의 문제점-즉 네트워크 공격에서 중간 단계인 라우터에서의 보다 적극적인 대처 방안의 요구와 이를 위한 프로그램 가능한 라우터의 필요성을 해결함으로써 한 차원 높은 수준의 네트워크 보안을 이룰 수 있게 하였다. 본 논문에서는 이러한 문제점을 해결할 수 있는 액티브 네트워크 기반의 멀티 에이전트 시스템을 제안하였으며, 멀티 에이전트 기술에 입각하여 각기 구별되는 성격의 에이전트들이 액티브 노드에서 담당하는 역할들을 기술하였다.

이 연구는 향후 검증을 통해 제안된 설계 구조의 타당성을 입증하고, 실제 적용 시 발생 할 수 있는 문제점에 대하여 해결할 수 있는 방안을 모색하고자 한다.

참고문헌

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

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

[3] H. Y. Chang, S. F. Wu, and et al, "DECIDUOUS: decentralized source identification for network-based intrusions," Proceedings of the Sixth IFIP/IEEE International Symposium on Integrated Network Management, May 1999.

[4] J. Rowe, "Intrusion Detection and Isolation Protocol: Automated Response to Attacks," Presentation at RAID'99, September 1999.

[5] X. Wang, D. Reeves, S. F. Wu, and J. Yuill, "Sleepy Watermark Tracing: An Active Network-Based Intrusion Response Framework", Proceedings of IFIP Conference on Security, Mar. 2001.

[6] S. Karnouskos, "Dealing with Denial-of-Service Attacks in Agent-enabled Active and Programmable Infrastructures," 25th IEEE International Computer Software and Applications Conference, October 2001.

[7] D. Schnackenberg, K. Djahandari, and D. Strene, Harley Holiday, Randall Smith, "Cooperative Intrusion Traceback and Response Architecture(CITRA)", Proceedings of the 2nd DARPA Information Survivability Conference and Exposition (DISCEXII), June 2001.

[8] Tennenhouse et. Al., "A Survey of Active Network Research", IEEE Communications Magazine, Vol. 35, No. 1, pp80-86. January 1997.

[9] D.L. Tennenhouse, and D. Wetherall. "Towards an Active Network Architecture", Computer Communication Review, Vol.26, No. 2, April 1996.

[10] S. Bhattacharjee, K. Calvert, and E. Zegura. "An Architecture for Active Networking", Proceedings of High Performance networking 97, White plains, April 1997.

[11] 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.

[12] 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.

[13] 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.

[14] 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, http://www.csee.umbc.edu/kqml/papers/kqmloverview.ps, 1992.

[15] 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.

[16] Linux FreeS/WAN Project. Available at http://www.freeswan.org.

[17] C. Woo, S. Hwang, and J. Choi, "Security Mechanism for Agent Communication of Intruder Tracing Multi-Agent System," Proceedings of the 4th ICACT, pp 893-897, 2002.