네트워크 접근 제어 목록 통합 관리를 위한 순응 메커니즘

이강희° 김장하 배현철 김상욱
경북대학교 컴퓨터과학과
{khlee°, jhkim, hcbae, swkim}@cs.knu.ac.kr

Adaptation Mechanism for Managing Integration of Network Access Control List

Kanghee Lee°, Jangha Kim, Hyunchul Bae, Sangwook Kim
Department of Computer Science, Kyungpook National University

요 약

본 논문에서는 네트워크의 구성 정보를 바탕으로 상위 수준에서 하위 수준으로 정책을 변환할 때 나타나는 기존 정책과의 충돌을 탐지하고 순응시키는 메커니즘을 소개한다. 대규모 네트워크는 라우터, 스위치, 방화벽, 침임 탐지 시스템, 일반 호스트 등과 같은 다양한 종류의 장비로 구성되어 있으며, 이러한 것들은 각기 다른 접근 및 제어 형식을 가지고 있다. 따라서 트래픽에 대한 일괄적인 통제가 어렵고, 외부의 공격에 대한 신속하고 효과적인 대응이 불가능하다.

또한 대규모 네트워크를 구성하고 있는 장비들을 제어하기 위해서는 그러한 장비들이 포함되어 있는 서브 네트워크의 세부 정보와 각 장비의 고유한 설정 규칙을 필요로 한다. 이러한 점은 대규모 네트워크를 상위 수준의 계층에서 관리를 어렵게 한다. 때문에 하부 계층의 구조나 정보와는 독립적으로 추상화된 고수준의 보안 정책 설정을 위한 도구가 요구된다. 이것은 상위 수준의 보안 정책 표현 기법, 하위 수준의 보안 정책 기법, 상위 수준의 보안 정책과 네트워크 구성 정보를 바탕으로 하위 수준의 보안 정책을 도출하는 기법, 하위 수준의 보안 정책을 실제 네트워크 구성 요소에 적용하는 기법 등의 네 가지 연구로 구분된다. 본 논문에서는 이 네 가지의 연구와 기법을 바탕으로 관리 네트워크에 새로운 정책이 전달될 때 기존의 단순한 정책 선택을 벗어난 서로의 정책을 변환한 ACL을 최대한 순응시키는 메커니즘을 제안한다.

1. 서 론

본 논문에서는 네트워크 보안 관리 시스템의 정책 충돌 상황에서의 접근 제어 목록을 순응 메커니즘을 소개하려 한다. 대규모 네트워크에서는 라우터, 스위치, 방화벽, 침임 탐지 시스템, 일반 호스트 등과 같은 다양한 종류의 장비로 구성되어 있으며, 이러한 것들은 각기 다른 접근 및 제어 형식을 가지고 있다. 따라서 트래픽에 대한 일괄적인 통제가 어렵고, 외부의 공격에 대한 신속하고 효과적인 대응이 불가능하다.

대규모 네트워크 관리 시스템에서는 일반적으로 하부의 구성 요소가 많기 때문에 정책을 작성하는데 다음과 같은 제약을 가지게 된다. 첫째 네트워크를 구성하는 요소, 장비의 공통적인 특징과 관계를 정확하게 알기 어렵고 전체 네트워크 구성 자체를 관리 대상으로 한다. 둘째 세부적인 네트워크 구조 및 속성보다는 일반적인 추상적 네트워크 모델을 기반으로 정책 설정을 한다. 마지막으로 네트워크 관리에 대한 방법의 기술보다는 관리자가 결과적인 상황으로 요구하는 목적의 기술에 초점을 맞춘다.

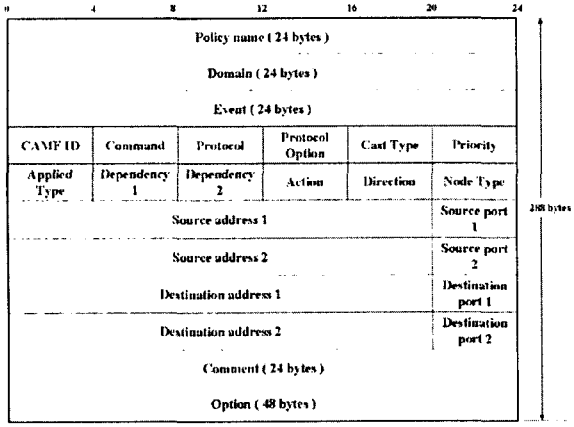
따라서 네트워크 접근 제어 목록 통합 관리를 위한 순응 메커니즘은 상위에서 고려하지 못한 정책 충돌을 탐지하고 네트워크의 각 구성 요소들에 대한 접근 제어 목록을 통합 관리 및 유지 하면서 상위에서 지시한 정책에 대해 최대한 일관성을 유지하는 것이 그 목적이다. 이 목적을 달성하기 위해서 Common Access Management Form을 정의하여 사용하며 정책을 ACL로 변환하게 된다. 그리고 이 변환 과정에 접근 제어 목록 통합 관리를 하기 위한 순응 메커니즘을 적용하여 최종적인 ACL을 생성하여 네트워크 노드에 적용하게 된다.

2. Common Access Management Form

다양한 하부 노드를 최 상위 관리 레벨에서 모두 표현하는 것은 불가능하다. 일반적으로 네트워크를 관리하기 위한 정책 작성 과정에서는 네트워크 구성과 결과적인 상황을 표현하는 것에 그친다. 그래서 이 정책을 최종 단계의 하부 노드의 적용되는 ACL로 변환하는 것은 많은 어려움이 따르게 된다. 이 어려움을 해결하기 위해 정책에서 ACL로 변환되기 전에 중간 단계의 형식 언어인 CAMF를 사용한다.

CAMF의 형식은 <그림 1>과 같고 전체가 288 Byte로 이루어져 있다. CAMF는 정책을 순응 메커니즘에 좀 더 쉽게 적용하기 위해서 ACL과 직접적으로 연관되는 필드를 가지고 있다. 관리를 위한 정책 이름, 적용 도메인 그리고 CAMF ID를 가진다. Event 필드는 특정 이벤트가 발생했을 때 정책을 적용시키기 위한 것이며 Command 필드는 정책을 삽입 혹은 삭제를 기술한다. Protocol 필드는 IP, TCP, UDP, ICMP 등의 프로토콜을 기술한다.

Protocol Option 과 Cast Type 필드는 기존 IP 프로토콜의 속성 값을 지정한다. Priority 필드는 정책의 우선순위를 지정하며 Applied Type 필드는 필수, 선택 정책임을 의미한다. Dependency 필드는 정책 간의 의존성을 Action 필드는 정책이 의도하는 허용 혹은 거부를 나타낸다. Direction 필드는 패킷의 incoming, outgoing을 나타내며 Node Type은 하부 노드의 종류를 기술한다. 그리고 Source Address, port 및 Destination Address, port의 지정 혹은 범위를 기술한다.



<그림 1> CAMF의 구조

3. 접근 제어 목록 관리를 위한 순응 메커니즘

네트워크의 최 상위에서 정책을 작성하게 되면 네트워크에 속한 노드들에게 정책이 전달되기 전에 이 노드들을 관리하는 도메인 서버에게 전달이 된다. 도메인 서버는 자신의 네트워크에 속한 노드들에 대한 접근 제어 목록도 관리 하고 뿐만 아니라 하위의 네트워크에 속한 도메인 서버도 관리하게 된다. 따라서 정책이 전달되면 관리 노드들에게 적용하고 하위의 네트워크에도 적용하는 계층적인 구조를 가진다.

이 계층적인 구조에서 정책이 전달되는 과정에 기존에 적용되고 있던 정책과 의미적인 충돌이 발생할 수 있다. 이것은 특정 범위가 같은 구간에 대해 CAMF의 Action의 상반된 지시에 의해 충돌을 감지할 수가 있게 된다. 예를 들어 호스트 A에 대해서 적용되고 있는 기존 ACL은 80번 포트를 deny 하는 것이고 새롭게 전달된 정책은 CAMF에서 ACL로 변환한 결과 80번 포트를 accept 한다면 충돌이 발생했다고 할 수 있다. 또한 범위에 대해서도 호스트 B의 80번에서 8080번 포트까지 deny 하라는 ACL이 적용되어 있는데 새롭게 내려온 정책의 ACL 변환 형태는 1024번 이상의 포트를 accept하게 나타났다면 이 또한 정책의 의미적인 충돌이 발생했음을 탐지할 수 있다.

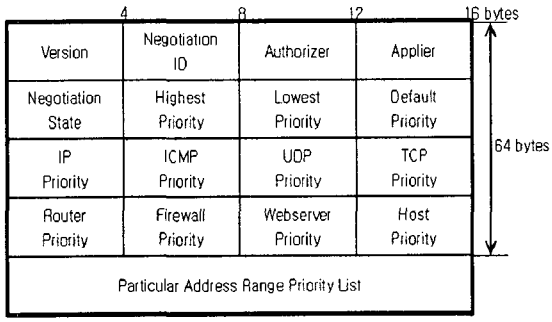
접근 제어 목록 관리를 위한 순응 메커니즘에서는 충돌을 패킷의 출발지와 목적지, 그리고 각각의 포트를 가지고 지정 대 지정, 지정 대 범위, 범위 대 범위의 형태로 나누어 서로의 구간을 검사하고 상반된 작동을 지시하는 ACL에 대해 충돌이라고 탐지하게 된다. 덧붙여 각 프로토콜마다 특징이 있으므로 [표 1]과 같은 검사 기준을 가진다.

[표 1] 프로토콜에 따른 충돌 검사 기준

기존 정책의 프로토콜	새로운 정책의 프로토콜	항목
IP	ALL	주소 범위
TCP	IP	주소 범위
	TCP	주소 범위 + 포트 범위
UDP	UDP	주소 범위 + 포트 범위
ICMP	IP	주소 범위
	ICMP	주소 범위 + ICMP 메시지

순응 메커니즘에서는 충돌된 정책을 조율시키기 위해 정책이 전달되면 '기본 순응 과정'이라고 정의한 다음과 같은 4가지 과정을 거치면서 네트워크의 접근 제어 목록의 일관성을 유지하며 관리하게 된다.

첫째 각 도메인 서버는 서로 사전 협상을 하게 된다. 이 사전 협상은 정책의 세부적인 필드에 대한 우선순위를 지정하는 것이다. 전달된 새 정책의 우선순위를 사전 협상 정보와 비교한다. 이때 정책의 내용이 사전 협상 정보의 필드에 해당되는 것이 있다면 그 필드의 우선순위와 전달된 정책의 우선순위를 단순히 비교하는 방법이다. 이 사전 협상은 상위의 도메인 서버가 하위의 도메인 서버의 자치적인 관리를 보장해주기 위해 사전 협상된 우선순위 이하는 상위 도메인 서버의 정책을 수용하고, 그 이상의 정책은 하위 도메인 서버 자체가 관리하게 된다. <그림 2>은 도메인 서버 간의 주고받는 사전 협상 정보를 나타낸 것이다.



<그림 2> 사전 협상 정보

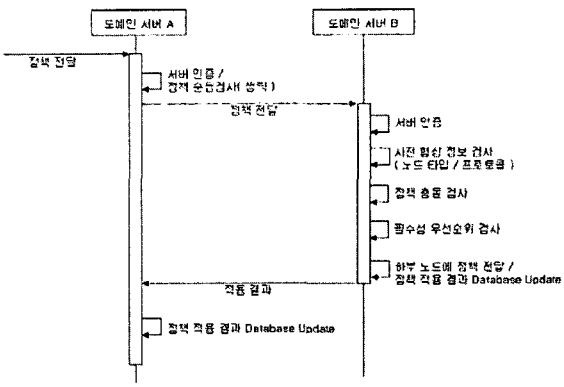
둘째 사전 협상 우선순위 보다 낮게 설정되어 전달된 정책은 첫 번째 단계를 통과하고 현재 노드에 적용되고 있는 정책과 우선순위를 비교하게 된다. 이 때에는 CAMF의 Priority 필드에 기술되어 있는 값을 서로 비교하게 된다. 이 값이 같다면 세 번째 과정으로 진입한다.

셋째 각 정책들이 필수 정책인지 선택 정책인지를 판별한다. 정책을 작성할 때에 반드시 적용되어야 할 정책이라면 CAMF 필드의 Applied Type을 essential로 설정하고, 그렇지 않다면 Optional로 설정한다. 예를 들어 호스트 C에서는 반드시 80번 포트를 열어두고 웹 서비스를 해야만 한다면 해당 네트워크 관리자는 정책을 작성할 때에 Applied Type 필드를 essential 로 설정하여 서비스를 지속하도록 한다.

넷째 정책이 사전 협상 우선순위 검사를 통과하고 기존 정책과 우선순위가 같고 필수 우선순위마저 같다면 기존의 정책을 그대로 고수하고 정책을 전달한 상위 도메인에게 어떤 정책과 충돌이 발생하며 모든 우선순위가 같아 적용되지 못하고 거부되었다는 상황을 전달한다.

'기본 순응 과정'의 각 단계에서 충돌된 새로운 정책이 우선순위가 높다면 정책 조율에 들어가게 된다. 정책 조율은 충돌 발생 범위에 따라 다르게 된다. 지점 대 지점이라면 두 정책 중 하나를 버려야하기 때문에 상관이 없지만, 충돌 범위가 지점 대 범위 혹은 범위 대 범위라면 '기본 순응 과정'을 거치고 우선순위가 높은 구간으로 정책을 순응하게 된다.

전체적인 순응 과정은 <그림 3>과 같다.



<그림 3> 전체적인 순응 과정

4. 구현 및 실행 결과

네트워크 접근 제어 목록 통합 관리를 위한 순응 메커니즘은 Fedora Core 1 배포판 리눅스 운영체제에서 g++ 3.2.2, MySQL 4.0.18, ACE 라이브러리를 이용하여 구현되고 테스트되었다. 테스트 노드는 시스코 라우터 1721과 3COM 라우터 5009 모델을 사용하였다. <그림 4>는 전달된 정책이 기존의 ACL과의 충돌을 탐지하고 서로를 순응시켜 새로운 정책으로 생성하는 테스트 화면이다.

<그림 4> 순응 메커니즘 테스트 화면

5. 결 론

대규모 네트워크에서 다양한 하부 노드를 제어하기 위한 다양한 연구가 진행되고 있다[1]. 도메인 간의 동적 협동에 관한 연구, 정책 표현 언어에 관한 연구, 신뢰 관리 시스템에 관한 연구 등이 활발히 진행되고 있다[2-4]. 그러나 접근 제어 목록을 통합 관리하여 네트워크 트래픽을 제어할 수 있는 연구는 미비하다.

본 논문에서는 다양한 노드가 존재하는 네트워크 관리 구조에서 일괄적인 정책 유지를 위한 순응 메커니즘을 소개하였다. 이를 위해 정책 언어의 특성에 따른 ACL 적용을 위하여 중간 형태의 CAMF를 정의하였다. CAMF는 기존의 노드에 의존적인 정책 작성을 벗어나 새롭게 나타는 노드에 대한 확장성 및 유연성을 지원한다. 이를 바탕으로 정책의 충돌 과정, 정책의 순응 과정, 정책의 선택에 대하여 연구하였다.

앞으로도 네트워크 관리 구조에서는 일괄적인 관리를 위한 정책 작성을 위하여 다양한 특징을 지원해 줄 수 있는 순응 메커니즘 개선을 위한 연구와 다양한 하부 노드를 지원하기 위한 인터프리터 엔진 기술가 필요하다.

참고 문헌

[1] DC-PREMISYS, http://govt.argreenhouse.com/DC-PREMISYS/, 2000

[2] Dynamic Coalitions, http://www.iaands.org/iaands2002/dc/index.html

[3] Ribeiro,C., A Zuquete, "SPL: An access control language for security policies with complex constraints," NDSS, 2001.

[4] Lobo,J., R.Bhatia, "A Policy Description Language," AAAI, 1999.8] Nicodemos Damianou, "The Ponder Policy Specification Language", Proc. of Workshop on Policies for Distributed Systems and Networks, Jan. 2001.