분산네트워크를 위한 KeyNote 기반의 보안 관리

배현철*, 송병욱*, 강철범*, 장희진*, 김상욱*, 이상훈**, 김도환**, 박응기**

*경북대학교 정보호호학과, 경북대학교 컴퓨터과학과

**국가 보안 기술 연구소

Network Security Policy Management based on KeyNote

Hyun-chul Bae*, Byung-wook Song*, Chul-bum Kang*, Hee-jin Jang*, Sang-wook Kim*
Sanghun Lee**, Dowhan Kim**, Eungki Paek**

Department of *Information Security and Computer Science of Kyungpook National Univ.
**National Security Research Institute

요 약

KeyNote 기반의 네트워크 보안 정책 관리는 분산 환경에서 보안 정책을 관리하고 적용하는 시스템이다. 기존의 네트워크에서 보안 정책 관리는 각각의 시스템에 보안 정책을 설정하여야 하는 구조로 구성되어 있다. 본 논문에서는 이러한 문제점을 개선하기 위하여 KeyNote 기반의 분산 네트워크 환경 하에서 보안 정책 관리를 이용한 보안정책의 정의 및 설정, 변환 등에 관련된 구조에 대한 설계와 이를 구현한다.

I. 서론

인터넷의 발전으로 네트워크의 규모가 확대되고 있다. 이에 따라 관리 시스템들은 보안 정책에 따라 자원에 대한 접근을 허용하거나 제한해야 할 필요가 있다. 그런데 관리 시스템 또는 응용들은 자원에 대해서 서로 다른 개념을 가지고 있으며, 또한 서로 다른 접근 허용 및 제한 정책을 가진다. 따라서 이때 사용되는 보안 메커니즘이 일반적인 보안 메커니즘이 되기 위해서는 임의 형태와 개수의 자원을 처리할 수 있어야 하고, 서로 다른 기준(정책)을 처리할 수 있어야 한다.

ACL(Access Control List)은 운영체제에서 사용되는 대표적인 보안 메커니즘이다. ACL은 주체(principal)가 자원(객체, object)에 대해서 어떤 접근 권한을 가지고 있는지를 기술하는 리스트이다. ACL은 이해하기 쉽고, 이를 다룬 문헌이 많기 때문에 네트워크 관리 시스템에서 많이 사용되어 왔지만 다음과 같은 이유 때문에 ACL은 네트워크 관리 시스템에서 사용하기에 부적절하다.

먼저, 별도 인증(authentication) 필요하다. 운영체제의 경우 주체의 신원을 쉽게 알 수 있지만 분산 환경에서는 자원에 대한 접근을 허용할 것인가를 결정하기 전에 인증이 먼저 이루어져야 한다. 이때 사용되는 인증 메커니즘에 문제점이 있다.

두 번째로 권한 위임(delegation)의 문제로 관리 작업의 고수준 권한을 가진 관리자는 전체를 포괄하는 보안 정책을 지정하는 대신 저수준에서 지정하여야한다. 이러한 인가 구조는 지역적으로 지정된 하위 정책들 사이에서의 불일치를 초래하기 쉽다. 세 번째로 표현력 및 확장성 부족으로 인하여 일반적인 형태의 보안 메커니즘은 새로운 다양한 상태와 제약을 다룰 수 있어야 한다. 기존 ACL 형태의 메커니즘은 표현력과 확장성에 문제가 있다. 마지막으로 지역 신뢰 정책 표현이 곤란한데 네트워크 관리 시스템에서는 많은 수의 관리자가 있을 수 있으며 그들은 각각 다른 사용자와 다른 개체에 대해서 서로 다른 신뢰 모델을 가지고 있을 수 있다. 따라서 네트워크 관리 시스템의 보안 메커니즘은 이를 표현할 수 있어야 하지만 기존 메커니즘은 이를 표현하는데 한계가 있다.

본 연구는 국가 보안 기술 연구소의 지원으로 수행되었음.

이러한 문제들 때문에 ACL과 결합된 사용자명/패스워드, One-Time 패스워드, 하드웨어 토큰 인증, 신원(identity) 기반 공개키 시스템 등은 네트워크 관리 시스템 보안 문제에 부적절한 해결책이다. 또한 접근과 관련된 결정이 "all-or-nothing" 형태인 바이너리 인가 모델에서는 자원에 대한 접근이, 접근을 요청한 주체가 특정 CA(Certificate Authority)가 발행한 인증서를 가지고 있는지의 여부에 따라 결정되게 되므로 경우의 수가 적은 경우(예: 웹 페이지에 대한 읽기 권한이 있는지 결정하면 되는 경우)와 같이 간단한 경우에는 적용 가능하지만 확장성에는 문제가 있다.

이에 효율적인 보안관리를 위하여 KeyNote 신뢰 관리을 이용하여 네트워크를 구성하는 이기종간의 정책 조정 및 협조가 필요함에 따라, 고수준 언어를 이용하여 대규모 네트워크를 위한 보안정책 기술 및 적용 메커니즘을 제시하고자 한다. [3]

본 논문의 제 2장에서는 관련 연구에 대하여 소개를 하고 제3장에서는 네트워크 보안 정책 관리에 대하여 정의한다. 제 4장에서는 이를 이용한 구현상황을 제시한다. 제 5 장에서는 네트워크 보안 정책 관리의 구현 및 적용 후 개선방안을 제시하고 결론을 맺는다.

II. KeyNote

기존 시스템 보안 접근법에서는 행위 수행에 대한 서명된 요청을 인증과 접근 통제의 결합으로 처리하였다. 요청을 수신한 시스템은 우선 요청이 누구에 의해서 서명된 것인지 결정하고(인증), 서명자가 요청된 행위를 수행하기 위해서 필요한 자원에 대한 접근을 허용할 것인지를 결정(접근 통제)하기 위해서 내부 DB를 검색해야 했다. 이에 비해서 신뢰 관리에서는 서명된 요청이 보안 정책에 부합되는지를 결정하기 위해서 증명서를 사용한다. 즉, 증명서는 공개키 시스템의 인증서와 달리 그 자체로서 권한을 위임하는 데 사용될 수 있다. 신뢰 관리는 보안 정책, 증명서(credential), 및 관계(relationship)를 지정하고 해석하는 통일된 접근방법으로, 신뢰 관리를 이용하면 보안과 관련된 중요한 행위를 직접적으로 인가할 수 있다. [4]

신뢰 관리는 응용에 보안 정책과 증명서를 지정하기 위한 표준화된, 일반적인 메커니즘을 제공한다. 신뢰 관리 증명서는 특정한 신뢰의 위임을 기술하며, 공개키 인증서의 역할도 함께 한다. 즉, 전통적인 인증서는 키를 이름에 결합시키는데 비해서 신뢰 관리 증명서는 키를 특정 작업 수행 여부를 인가하는 것에 직접 결합시킬 수 있다.[1]

대표적인 신뢰 관리에는 신뢰 관리의 개념을 처음 소개한 PolicyMaker와 KeyNote가 있다. PolicyMaker는 연구를 목적으로 만들어진것인데 비해서 KeyNote는 이를 발전시켜 실용화한 것이다. KeyNote는 현재 적용 가능한 최신 신뢰 관리로서 특징은 다음과 같다.

- 행위(action)는이름-값의 쌍으로 표현된다.
- 주체는 임의의 문자열이나 공개키로 표현된다.
- 정책과 증명서를 표현하는데 동일한 언어를 사용하며, 사용되는 언어는 간결하고, 표현력이 뛰어나며, 사람이 읽고 쓸 수 있는 형태이다. 또한 다양한 저장 장치에 저장 가능하며, 전자 메일 등을 통하여 전송 가능하다.
- 순응 검사기는 요청이 응용에 의해서 어떻게 처리되어야 하는 지를 기술하는 값을 결과값으로 제공하는 데, 이 값은 응용에서 설정한 '정책 순응 값'이며, 항상 지정된 정책과 증명서로부터 도출되기 분석이 용이하게 된다.
- 순응 검사는 고성능 실시간 응용에 적용할 수 있을 정도로 효율적이다.

KeyNote는 소규모에서 대규모에 이르는 다양한 인터넷 기반의 응용에 잘 적용될 수 있도록 설계된 시스템으로, 동일한 언어를 이용하여 지역 정책과 증명서를 기술할 수 있게 해 준다.[2]

III. 네트워크 보안 정책 관리

1. 개요

네트워크 보안 정책 관리는 그림 1와 같은 구조로 구성이 된다. 정책 에디터를 이용하여 고수준 언어로 정책을 작성하고 이를 KeyNote 정책으로 변환을 한 다음, 그림2와 같이 KeyNote 서버로 전송을 하게 된다.

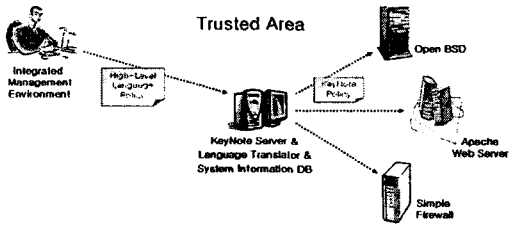

그림 1: 네트워크 보안 정책 관리 구조

그림 1에서 KeyNote 서버는 전송받은 정책을 KeyNote 인터프리터를 이용하여 해당 시스템에 적용을 하게 된다.

2. 보안 정책 설정

보안 정책 에디터는 다양한 고수준 언어를 지원하도록 플러그-인 기능을 제공한다. 또한, 다이어그램으로 표현의 범위를 확장함으로써 정책 설정을 표현한다. 다음은 고수준 언어로 보안정책을 작성한 것이다.

```
[on f.connect()][s.state="Restricted"] s.login(id,pass)
```

사용자가 Firewall을 통해 서버로 접속하고자 하는 connect() 동작이 발생했을 때 서버의 상태가 제한적이면 서버로 로그인시에 id와 pass를 필요로하는 것을 언어로 표현한 것이다.

3. KeyNote 정책으로 변환

설정된 보안 정책을 KeyNote 형식의 정책으로 변환하기 위하여 그림 2와 같이 인터프리터를 통하여 변환을 한다. 보안 정책 분석과정에서 참조정보를 이용하여 해당 노드의 실제 객체로 맵핑을 하게 된다. [13]

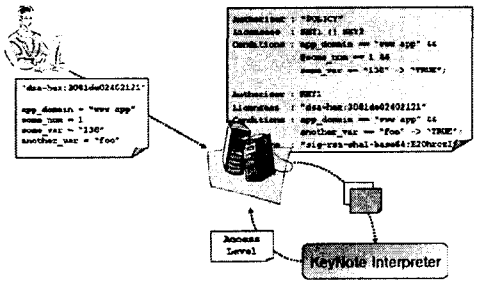

그림 2: 고수준 언어의 KeyNote 정책 변환과정

각 노드에 적용시에는 중간언어로 변환하여 적용이 되며, 마찬가지로 데이터베이스를 참조하여 최종 설정값으로 변환이 된다. [5][6]

4. 해당 노드의 보안정책 설정과 처리

고수준언어로 작성된 것을 해당 노드에 적용하기 위해서 데이터베이스를 이용한다. 이 데이터베이스는 노드 관련된 제반 정보를 가지고 있으며, 이는 고수준 언어에서 KeyNote 정책으로 번역시에 사용되어 진다.

```
Authorizer: "POLICY"
Licensees: "DSA:1f43dee5f001b177fe"
Conditions: application = "firewall"
            && protocol == "tcp"
            && (port == "134" || port == "21")
            && source == "10.0.0.2"
            && dest == "192.168.0.5"
```

위와 같은 정책이 있는 경우, 라이센스 정보와 port 와 source, dest 정보는 데이터베이스의 정보를 이용하여 검증되어지거나 추가되어지며 라우터나 호스트, 방화벽 등의 설정을 변경하는 부분은 시스템 개발자가 만들어주어야 한다.

IV. 구현

그림 3은 구현한 시스템의 구조이다.

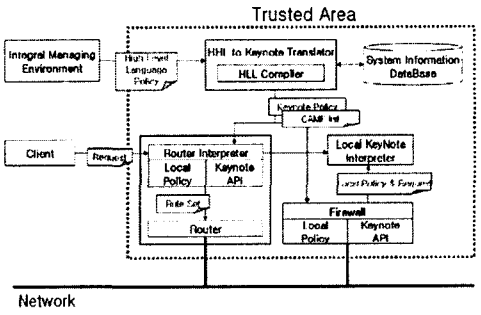

그림 3: 시스템 구조

그림 4와 같이 관리환경은 윈도우즈 환경에서 동작하며, 고수준언어와 다이어그램으로 작성된 정책은 데이터베이스의 정보를 이용하여 컴파일되어 KeyNote 정책이 생성된다. [7]

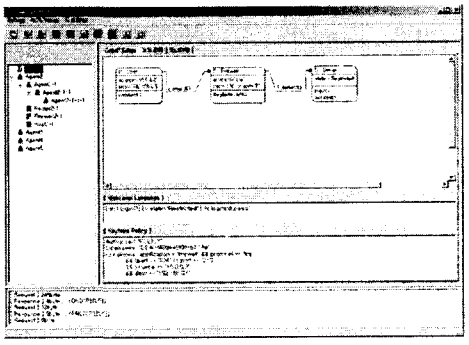

그림 4: 관리 통합 환경

생성된 KeyNote 정책은 라우터의 경우 직접적

인 정책적용이 어려움으로 인터프리터를 통하여 해당 라우터에 맞는 룰셋으로 변환되어 설정된다. 인터프리터는 리눅스 환경에서 동작하며, KeyNote 정책을 처리할 수 있는 KeyNote API와 지역정책 정보를 가지고 있게 된다.

방화벽이나 호스트의 경우에는 직접적으로 설정이 가능하므로 KeyNote API를 이용한 모듈을 이용하여 처리하며, 마찬가지로 지역정책을 가지고 있다. 생성된 KeyNote 정책은 지역정책과 인증과 검증을 통해 적용되며, 이때 KeyNote 정책에 따른 각각의 동작을 처리하는 부분은 직접적으로 제작을 해야 한다.

그림 5는 이러한 정책을 호스트와 라우터에 적용하고 이에 대한 결과를 조회할 수 있다.[8]

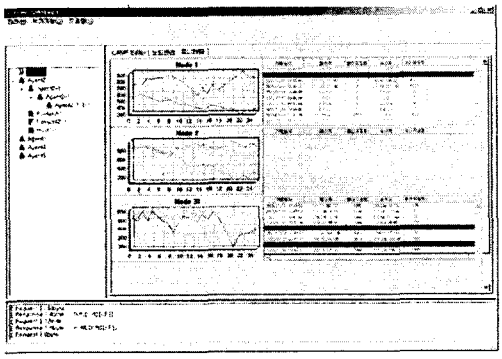

그림 5: 정책적용 유무확인

V. 결론

본 논문에서는 네트워크 보안 정책 관리를 구현하였다. 네트워크 상에서 동적으로 보안 정책을 적용하기 위해 KeyNote 기반 네트워크 보안 정책 관리를 설계 및 개발하였다. 현재 나와 있는 다양한 보안 정책 관리 시스템들은 현재 활발히 연구되고 있는 분산 환경에 적용하기에는 문제점이 존재하는 시스템이다. 이를 해결하기 위하여 DARPA에서 표준화한 KeyNote를 이용하여 기존의 네트워크 환경과 이질적인 분산 네트워크 환경간에 서로 호환성을 제공하였고, 앞서 개발한 KeyNote 기반의 네트워크 보안 정책 관리를 이용하여 보안정책을 적용하는 시스템을 개발하였다.

향후 연구방향은 아직은 구현모델이 적은 부분이 있지만 이를 이용하여 대규모 네트워크 환경에서 물리적인 디바이스(라우터) 환경까지도 보안 정책을 적용할 수 있는 시스템을 연구함으로써 실제 필드에서도 사용가능한 시스템을 개발하는 것이다.

[참고문헌]

[1] M. Blaze, J. Feigenbaum, J. Ioannidis, AKeromytis. "The KeyNote Trust Management System, Version 2." RFC-2704. IETF, September 1999.

[2] M. Blaze, J. Feigenbaum, M. Strauss."Compliance-Checking in the PolicyMaker Trust-Management System." Proc. 2nd Conference on Financial Cryptography. Anguilla 1998. LNCS 1465, pp 251-265, Springer-Verlag, 1998.

[3] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis. "The Role of Trust Management in Distributed Systems Security." Chapter in Secure Internet Programming: Security Issues for Mobile and Distributed Objects, (Vitek and Jensen, eds.) Springer-Verlag, 1999.

[4] Scalable Trust of Next Generation Management, "http://www.cis.upenn.edu/~dsl/STRONGMAN/"

[5] A. V. Aho, R. Sethi, J. D. Ullman , "Compilers : Principles, Techniques, & Tools," Addison Wesley, 1986

[6] K. C. Louden, "Compiler Comstruction Principles and Practice," PWS Publishing Company, 1997

[7] J. Levine, "Lex & Yacc," O'Reilly, 2001

[8] W. R. Stevens, "Unix Network Programming,", Prentice Hall, 1998