액티브 네트워크 상에서 액티브 노드의 보안 강화를 위한 보안 엔진 구현

김 옥 경[†]·임 지 영[††]·나 현 정[†]·나 가 진[†]
김 여 진[†]·채 기 준[†††]·김 동 영[††††]

요 약

액티브 네트워크는 패킷이 목적지에 도착하기 전에 거치게 되는 중간노드에서 새로운 정책이나 코드를 추가하여 프로그램을 실행시킬 수 있는 소프트웨어 기반 차세대 네트워크이다. 이러한 액티브 네트워크 환경에서는 다이나믹하게 패킷을 처리할 수 있지만 중간 노드에서 패킷을 처리하기 위해서는 그 노드의 자원을 사용해야 한다. 따라서 액티브 네트워크에서는 기존의 네트워크와는 달리 중간노드의 자원에 접근함으로써 발생하게 되는 네트워크 보안상의 문제점들을 고려하지 않으면 안 된다. 본 논문에서는 액티브 네트워크 환경에서 액티브 노드의 자원에 접근 시 발생되는 보안상의 문제점들을 해결하기 위한 보안 강화 엔진을 제안하고, 보안 강화 엔진 내에 보안, 인증, 권한부여 모듈을 두어 액티브 네트워크 환경에 노출되어있는 악의적인 위협 요소들로부터 보호하고자 한다.

Implementation of Security Enforcement Engine for Active Nodes in Active Networks

Okkyeung Kim[†]·Jiyoung Lim[††]·Hyunjung Na[†]·Gajin Na[†]
Yeojin Kim[†]·Kijoon Chae[†††]·Dongyoung Kim[††††]

ABSTRACT

An active network is a new generation network based on a software-intensive network architecture in which applications are able to inject new strategies or code into the infrastructure for their immediate needs. Therefore, the secure active node architecture is needed to give the capability defending an active node against threats that may be more dynamic and powerful than those in traditional networks. In this paper, a security enforcement engine is proposed to secure active networks. We implemented an operating engine with security, authentication and authorization modules. Using this engine, it is possible that active networks are protected from threats of the malicious active node.

키워드 : 액티브 네트워크(Active Network), 보안 강화 엔진(Security Enforcement Engine), 인증(Authentication), 권한부여(Authorization)

1. 서 론

인터넷의 급격한 확산과 그에 따른 사용자의 증가로 인해 네트워크에 대한 요구는 점점 다양화되고 있다. 그러나 현재의 네트워크 시스템은 새로운 기술이나 이와 관련된 표준을 망에 적용하기까지 많은 시간과 비용을 필요로 한다. 즉, 새로운 프로토콜이나 서비스를 네트워크에 통합시키는 것이 쉽지 않다. 게다가 여러 프로토콜 계층 간의 중복되는 기능으로 인하여 성능이 저하될 수 있다. 또한 기존의 네트워크는 흐름 제어나 경로 설정과 같은 패킷 처리를 종단의 단말에서만 처리할 수 있다.

종단의 단말에만 집중되어있는 네트워크의 기능을 분산시키고 사용자의 망에 대한 요구를 적절하고 빠르게 반영하여 네트워크에 유연성을 제공하기 위해 액티브 네트워크라는 새로운 패러다임이 등장하였다. 액티브 네트워크란 라우터나 스위치가 프로그램 실행 능력을 가지고 있어서 프로그램을 포함하거나 또는 중간 노드의 프로그램을 실행하도록 하는 패킷을 이용하여 다양하고 유동적인 처리를 패킷이 행할 수 있는 환경을 가진 망을 말한다[1-5]. 즉, 사용자가 원하는 프로그램을 패킷 내에 가지고 있거나 혹은 중간 노드에서 일부 특별한 관리자가 미리 제공하는 프로그램을 실행하여 단순한 처리를 넘어서는 매우 다양하고 유

※ 본 연구는 한국전자통신연구원 정보보호연구본부 네크워크보안연구부 위탁연구 과제에 의한 것임.
† 준 회 원 : 이화여자대학교 과학기술대학원 컴퓨터학과
†† 정 회 원 : 한국성서대학교 정보과학부 교수
††† 종신회원 : 이화여자대학교 컴퓨터학과 교수
†††† 정 회 원 : 한국전자통신연구원
논문접수 : 2003년 4월 28일, 심사완료 : 2003년 6월 11일

동적인 처리를 패킷이 실행할 수 있다. 이러한 패킷을 액티브 패킷이라고 하고, 프로그래밍 수행 능력을 가지고 액티브 패킷을 처리할 수 있는 라우터를 액티브 라우터, 또는 액티브 노드라고 한다.

이렇게 네트워크의 새로운 패러다임으로 나타난 액티브 네트워크 기술은 중간 노드에서 여러 가지 처리를 가능하게 함으로써 기존의 네트워크가 제공하지 못하는 유연성과 다양한 장점을 제공할 수 있다. 즉, 기존의 네트워크가 가지고 있던 비효율성이나 수동적인 측면을 개선할 수 있고 보안이나 성능 그리고 새로운 기술과 서비스의 도입, 네트워크 관리 문제 등을 개선할 수 있는 새로운 접근 방법이다.

하지만 동적이고 유연한 액티브 네트워크의 장점들이 보안의 측면에서는 매우 위험한 요소가 될 수 있다. 네트워크의 악의적인 사용자가 잘못된 코드 혹은 악의적인 코드를 네트워크 내에서 실행함으로써 전체 네트워크에 결정적인 영향을 끼칠 수 있기 때문이다. 게다가 악의 없는 사용자라 할지라도 실수로 잘못된 코드를 실행할 경우, 그 결과는 전자와 같기 때문에 액티브 네트워크에서의 보안은 기존의 네트워크에서보다 훨씬 더 중요한 이슈가 되고 있고, 이에 대한 활발한 연구가 진행중이다.

액티브 네트워크에서 보안을 제공하기 위해서는 기본적으로 인증과 권한부여 기능이 필요하다. 인증이란 네트워크 개체의 신분 증명으로, 인증된 개체에 대해서만 또 다른 보안 서비스가 실행될 수 있다. 따라서 권한부여 역시 반드시 인증된 개체들만을 대상으로 삼고 있다. 액티브 네트워크에서의 인증 시스템은 사용자에 기반을 둔 인증과 노드에 기반을 둔 인증으로 구성된다. 개체가 인증이 되면 사용자의 신분과 추가적인 정보가 담긴 인증서(Credential)를 받게 되는데, 이는 여러 가지 보안 서비스를 위해 사용된다. 권한부여란 네트워크의 개체가 무엇을 할 수 있는지를 결정하며, 누가(who) 어떤 것(what)에 접근할 수 있는지에 대한 접근 제어를 하는 것을 말한다. 접근 제어를 위한 많은 메커니즘이 존재하며 대부분의 액티브 네트워크 연구에서는 유닉스 기반의 시스템에서 사용하는 접근 제어 리스트(Access Control List, ACL)를 확장하여 자원 할당을 하고 있다. 이러한 인증과 권한부여에 대한 정보는 패킷 내에 첨가되어 전송된다. 또한 패킷 자체의 무결성을 보장하기 위해 전자 서명이 패킷 내에 첨가된다.

본 논문에서는 액티브 패킷이 중간 노드에서 실행되는 경우 발생하게 될 여러 가지 보안상의 문제점을 해결하기 위한 보안 강화 엔진을 제안하고자 한다. 중간노드에서 실행되는 액티브 패킷은 송신 노드에서 효율적이고 안전한 인증 및 암호화 기법을 통해 생성된다. 또한 중간노드에 도착한 액티브 패킷은 인증과 무결성 체크, 권한부여 모듈을 통과해야만 노드의 자원에 접근 할 수 있다. 따라서 보다 안전하고 효율적인 액티브 네트워크 환경을 제공할 수 있다.

본 논문의 구성은 다음과 같다. 1장의 서론에 이어 2장에서의 기존 액티브 네트워크 보안 연구에 대한 간략한 설명과 3장에서는 제안한 보안 강화 엔진(Security Enforcement Engine) 구조에 대해 설명하고, 4장에서는 보안 강화 엔진의 각 모듈별 상세 동작 과정과 구현 결과에 대해 설명하며 마지막으로 5장에서는 본 논문의 결론을 제시한다.

2. 기존 액티브 네트워크 보안 연구

액티브 네트워크에 대한 연구는 현재 DARPA[6]의 지원으로 이루어지는 것만도 50여개에 이를 정도로 여러 곳에서 독립적으로 활발하게 이루어지고 있다. 현재 연구되고 있는 주제는 구조, 네트워크 관리, 네트워크 보안, 응용 분야 등으로 나누어 볼 수 있다. 이 장에서는 주요 연구 기관 및 대학에서 수행된 액티브 네트워크의 보안 구조에 관한 연구를 소개하고자 한다.

펜실베니아 대학에서 수행된 SANE(Secure Active Network Environment)[7]은 안전한 액티브 네트워크를 구현하기 위하여 SwitchWare 환경에서 구현되었으며, 계층화된 보안 구조로 인증, 무결성, 암호 등의 보안 서비스를 제공한다. 일리노이 대학에서 연구된 Seraphim[3]은 다양한 보안 정책과 메커니즘을 수용할 수 있게 확장 가능하고, 재구성 가능한 보안 구조이다. Seraphim에서는 사용자에게 동적으로 상황에 맞는 정책을 생성하고 수용할 수 있는 능력을 제공한다. 시드니 대학에서 연구된 CANSA(Certificate Active Network Security Architecture)[4]에서는 보안에 관한 두 가지 중요한 주제인 인증 프로토콜과 권한부여를 제안한다. 액티브 네트워크에서의 인증은 사용자나 패킷 요소, 또는 다른 어떤 객체, 실행 환경 그리고 노드가 될 수 있는 주된 개체에 관한 신원확인이고, 권한부여 정책은 네트워크 내의 개체들의 접근을 어떻게 통제할 것인가에 대해 설명하고 있으며, 오직 인증 프로토콜이 작동된 상태에서만 올바른 기능 구현이 가능하다. FAIN[8]은 2000년부터 UCL(University College London)이 주축이 되어 진행중인 프로젝트로 액티브 노드를 기반으로 개방적이고 프로그램 가능하며 신뢰성 있는 액티브 네트워크 구조를 개발하는데 목적을 두고 있다. Georgia Tech(Georgia Institute of Technology)에서 연구중인 CANEs(Composable Active Network Elements)[9] 프로젝트는 액티브 네트워크의 특정 어플리케이션이나 새로운 서비스를 제공하기 위하여 동적으로 네트워크를 변경하는 동안 높은 성능을 제공할 수 있는 방안에 초점을 두었다. CANEs 프로젝트의 목적은 새로운 서비스가 소프트웨어로 설치되는 동안 최적의 빠른 경로(fast-path)

를 제공할 수 있는 사용자 인터페이스를 설계 및 구현하고 액티브 네트워크를 위한 전반적인 구조를 구성하고자 하는 것이다.

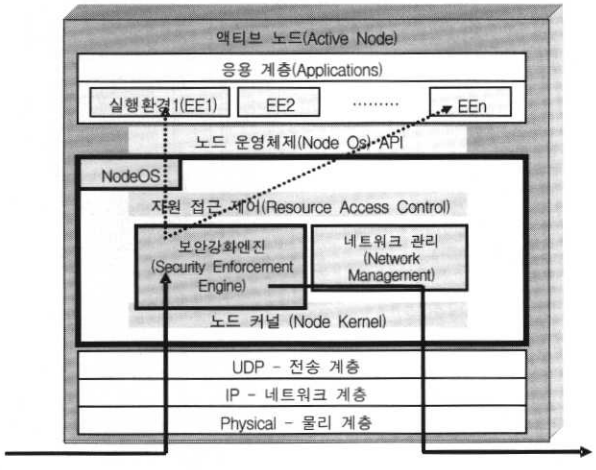
그 외에도 여러 연구들이 이루어지고 있으나 기존의 연구들은 액티브 네트워크 환경에서 체계적이고, 효율적인 보안 처리가 부족할 뿐만 아니라 실제적으로 네트워크 상에서 사용 가능한 구현은 이루어지지 않은 상태이다. 이러한 점을 고려하여 본 논문에서는 향상된 구조 제안과 네트워크 활용에 있어서 유용한 보안 엔진 구현에 초점을 맞추었다. 다시 말하면, 액티브 네트워크 환경에서 액티브 노드를 위한 보안 강화 엔진의 구조를 제안하고 이를 구현하였는데, 본 논문에서 제안한 보안 강화 엔진과 액티브 패킷 구조는 기존 연구를 바탕으로 하여 보다 향상된 기능을 첨가하였고, 기존 엔진의 OS에서도 동작이 가능하도록 독립적인 보안 모듈을 구현하였다. 즉, 기존 네트워크뿐만 아니라 액티브 네트워크 환경에서도 유연성 있게 사용될 수 있는 엔진을 제안하였다. 다음은 제안한 보안 강화 엔진 구조에 대해서 기술한다.

3. 제안한 보안 강화 엔진 구조

기존의 패킷 교환 네트워크는 해킹과 같은 보안 공격에 많은 취약점을 드러내고 있다. 침입차단 시스템(Firewall System)과 침입탐지 시스템(Intrusion Detection System)[10]같은 보안 시스템들이 개발되고 있지만 기존의 네트워크는 공격에 대해 적극적으로 대처하기 어렵다. DARPA에서는 전송중인 액티브 패킷이 라우터에서 코드를 실행할 수 있으며 코드의 실행결과에 따라 라우터의 상태를 변경할 수 있음을 제안하였다. 이러한 액티브 네트워크는 패킷을 단순히 전달하는 기능만을 지닌 기존의 패시브 네트워크에 비해 유연성이 있는 반면, 심각한 보안 문제를 가지고 있다. 악의적인 코드를 가진 패킷이 액티브 노드를 공격할 수도 있고, 악의적인 노드가 정상적인 액티브 패킷의 실행을 방해할 수도 있다. 이것은 기존 네트워크에서의 공격보다 더욱 심각한 영향을 미칠 수 있다. 따라서 외부의 의도적인 공격에 대한 방어와 노드 자체 내의 안전성을 위해서 다양한 위협에 대응할 수 있는 안전한 액티브 노드 구조가 필요하다.

본 논문에서는 액티브 네트워크에서의 안전한 통신을 위해 보안성이 추가된 액티브 노드 구조를 제안한다. (그림 1)은 제안한 액티브 노드 구조이며 노드 내에는 액티브 보안 엔진 기능을 담당하는 보안 강화 엔진을 포함한다. 액티브 노드 구조는 크게 물리 계층, 네트워크 계층, 전송 계층, 노드 운영체제(NodeOS), 실행 환경(Execution Environment, EE) 그리고 응용 계층으로 구성되어 있다.

액티브 노드의 노드 운영체제 내에 위치하는 보안 강화 엔진은 다음과 같은 기능을 수행한다. 노드로 들어온 패킷에 대한 무결성 검사와 복호화 그리고 검증을 수행하고 수행한 결과가 안전하다고 판정되면 패킷에게 적절한 자원을 할당하고 패킷이 실행 환경에서 실행할 수 있도록 처리된다. 모두 처리된 후에는 다시 보안 강화 엔진으로 이동하여 패킷을 암호화하고 전자 서명을 처리한 후 다음 액티브 노드로 전달된다.



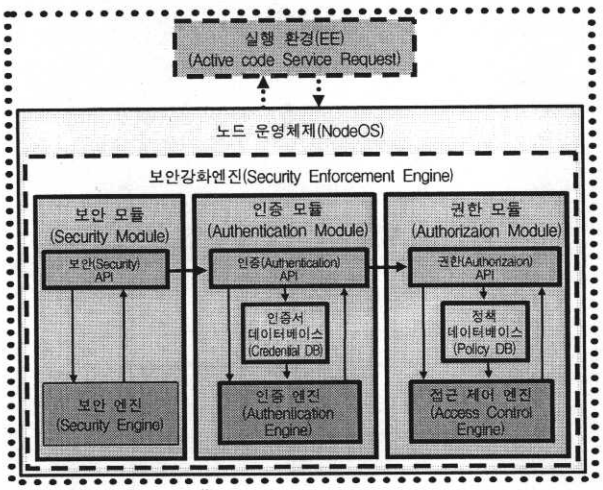
(그림 1) 액티브 노드 구조

(그림 2)는 네트워크에서 안전한 통신을 위한 요구 사항을 바탕으로 설계한 액티브 노드 구조로, 여러 계층들 중 노드 운영체제 내에 보안 강화 엔진이 위치하고, 본 논문에서 제안한 보안 강화 엔진은 보안(Security) 모듈과 인증(Authentication) 모듈, 권한부여(Authorization) 모듈로 구성된다. 이 세 모듈간의 긴밀한 상호작용으로 액티브 네트워크 상에서의 안전한 통신을 가능하게 한다. 보안 모듈과 인증 모듈을 통해 인증되고, 권한부여 모듈을 통해 권한부여 서비스가 제공된다.

보안 모듈은 액티브 패킷에 의해 전달되어지는 메시지의 무결성을 검증하기 위한 보안기능을 제공한다. 인증 모듈은 메시지 송신 노드와 송신자의 신분을 인증하기 위한 기능을 담당한다. 권한부여 모듈은 수신한 메시지나 프로그램이 요구한 자원에 대한 권한을 가졌는지를 검증하고, 그 결과에 따라 패킷이 요청한 자원을 제한적으로 제공하도록 제어한다.

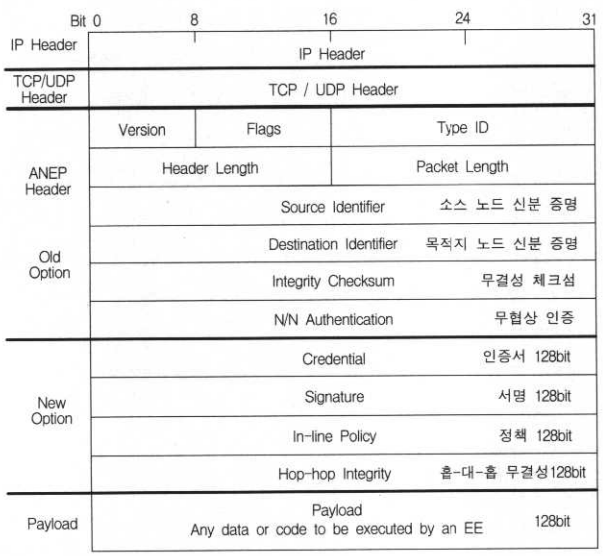
전체적인 흐름을 보면, 먼저 액티브 패킷을 생성하여 전달하는 기능을 가진 클라이언트 액티브 노드는 보넬 메시지와 코드 등을 보안 모듈에서 처리하여 전송하고, 액티브 패킷을 수신 받는 서버 액티브 노드는 보안 모듈에서 메시지의 무결성을 검증하고, 인증 모듈에서 송신 액티브 노드

및 사용자에 대해 인증하며, 권한부여 모듈에서 얼마만큼의 자원을 쓸 수 있는지를 고려하여 액티브 패킷의 실행 여부를 결정한다. 각 모듈별 상세 처리 기법은 다음 4장에서 더 깊이 다루도록 한다.



(그림 2) 보안 강화 엔진(Security Enforcement Engine)

본 논문에서는 액티브 패킷의 포맷으로 ANEP(Active Network Encapsulation Protocol)[11] 패킷 구조를 이용하였다. (그림 3)은 액티브 네트워크에서 안전한 통신을 위한 요구사항을 바탕으로 설계한 액티브 패킷의 포맷이다.



(그림 3) ANEP 패킷 형식[17]

통신 프로토콜 스택은 기본적으로 IP 프로토콜을 사용하며 상위 계층으로 TCP 또는 UDP 프로토콜 모두 가능함을 제안한다. 본 논문에서는 IP 헤더와 UDP 헤더 다음에 ANEP 헤더를 붙이고, ANEP 헤더의 기존의 옵션(Old Option)으로는 소스 노드 신분 증명(Source Identification), 목적지 노드 신분 증명(Destination Identification), 무결성 체크섬(Integrity Checksum), 무협상 인증(N/N(Non-Negotication) Authentication) 네 가지 필드를 가지며, 새로 도입한 옵션(New Option)으로는 인증서(Credential), 서명(Signature), 정책(In-line Policy), 홉-대-홉 무결성(Hop-hop Integrity) 필드를 추가한다.

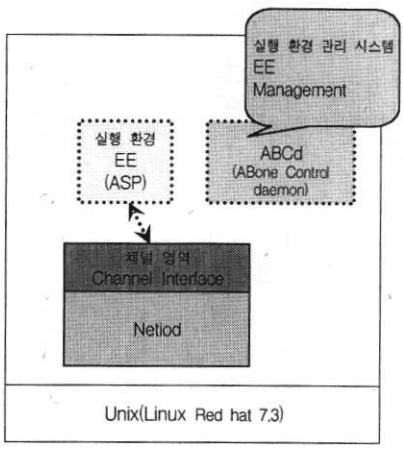
새로 도입한 옵션의 모든 필드와 페이로드(Payload) 필드의 메시지는 메시지 축약 알고리즘인 MD5 알고리즘을 수행한 결과가 입력되므로 128비트로 정해진다. 액티브 노드에 전달하고자 하는 데이터나 실행할 코드들은 페이로드 필드에 담아서 보내고, 액티브 패킷을 생성한 사용자의 신분을 검증하고 송신측 액티브 노드를 인증하기 위해 인증서 필드를 이용하며, 본 논문에서는 X.509[12] 포맷의 인증서를 이용하여 구현하였다. 그리고 수신한 액티브 패킷의 변경여부를 검증하기 위해 서명 필드를 생성하며, 수신한 액티브 패킷의 데이터나 실행할 프로그램을 위한 권한부여를 위해 정책 필드를 이용한다. 또한 액티브 패킷을 생성한 송신 액티브 노드뿐만이 아니라 중간 홉에서의 인증, 메시지 무결성 등을 검증하기 위해 홉-대-홉 무결성 필드를 이용할 수 있다. 만약 중간 홉에서의 인증을 추가할 경우, 인증서와 서명 필드는 각 액티브 노드별로 액티브 패킷에 추가되어야 한다. 이렇게 처리함으로써 단말 액티브 노드뿐만이 아니라 중간 홉의 액티브 노드에서도 인증이 가능하며, 보다 강력한 공격자의 공격에도 안심할 수 있게된다.

그러나 여기서의 문제점은 중간 홉 액티브 노드를 거침으로써 계속 쌓이게 되는 인증서와 서명 필드로 인한 부하가 막대하여 트래픽의 증가를 초래하는 것이다. 이에 중간 홉 액티브 노드에서의 인증문제를 어떻게 해결할 것인가 하는 것이 추후 고려해 볼 사항이다.

4. 보안 강화 엔진의 구현

네트워크에서 안전한 통신을 위한 요구 사항을 바탕으로 설계한 액티브 노드 내에 보안 강화 엔진은 보안 모듈과 인증 모듈, 권한부여 모듈로 구성되며, 이 세 모듈간의 긴밀한 상호작용으로 액티브 네트워크 상에서의 안전한 통신이 가능하다. 이 장에서는 보안 강화 엔진을 위한 구현 환경과 각 모듈의 동작과정 그리고 모듈에서 처리된 결과들을 살펴볼 것이다. 사전에 이루어져야 할 가정사항은 액티브 노드와 CA(Certificate Authority)간의 통신과 액티브 노드간의 키 교환이 이루어져야 한다는 것이다. (그림 4)는 구현된 엔진을 테스트하기 위한 시스템을 보여준다. 플랫폼으로는 리눅스 레드 햇 7.3(Linux Red hat 7.3) 버전으로 하고 노드 운영체제의 기능을 대체하기 위해 ABONE에서 제

공하는 Anetd(Active Networks Daemon)를 설치하였다. Anetd는 실행 환경을 관리하는 ABCd(Abone Control deamon)와 유닉스/리눅스(UNIX/Linux) 플랫폼에서 네트워크 I/O를 처리하고 실행 환경과 노드 운영체제간의 채널 인터페이스를 제공하는 Netiod(Network I/O daemon)로 구성되어 있다. 능동 패킷이 수행될 실행 환경은 Ants(Active Networks Transport System)나 ASP(Active Signaling Protocol) 혹은 PLAN(Packet Language for Active Networks) 실행 환경 모두 가능하다.



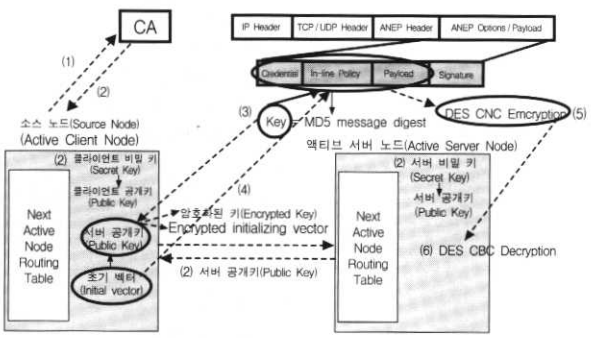
(그림 4) 시스템 환경[13]

4.1 보안(Security) 모듈

이 모듈에서는 메시지의 무결성을 검증하기 위해 처리되어 지는 부분으로 암호화, 복호화와 전자 서명을 수행한다. 이를 위해 사용한 알고리즘은 다음과 같다. 액티브 패킷에 담아 보낼 메시지나 코드의 길이를 128bit로 축약하는 메시지 압축 MD5[14] 알고리즘을 적용하고, 액티브 패킷이 이동 중에 어떠한 공격자에 의해서도 정보가 유지될 수 있도록 DES CBC(Chiper Block Chaining)[15] 암호화와 복호화 알고리즘을 적용하였으며, 수신한 메시지의 변경여부를 검증하기 위해 RSA 전자 서명(RSA Digital Signature)[16]과 검증 알고리즘을 적용하였다. 그리고 이 알고리즘들은 액티브 패킷내의 인증서, 서명, 정책, 페이로드 필드에 각각 적용하여 구현하였다.

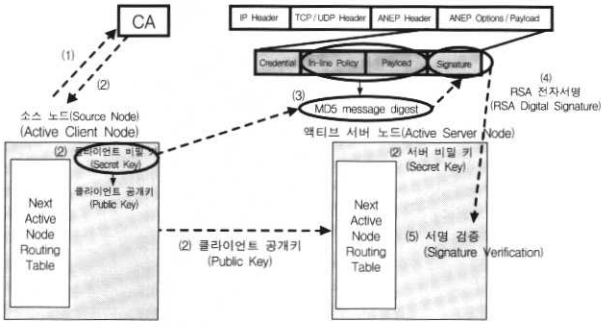
먼저 인증서와 정책 그리고 패이로드 필드에 적용한 알고리즘의 흐름에 대해 간략히 설명하고자 한다. (그림 5)는 보안 모듈에서 인증서와 정책 그리고 패이로드에 적용한 알고리즘의 처리과정을 보여준다.

인증서 필드에는 CA로부터 받은 인증서가 들어가며 (1)과 (2)는 인증서를 부여받는 과정이고 CA로부터 인증서를 받는 동시에 비밀키를 받는다. 패이로드 필드에는 실행하고자 하는 코드나 메시지들이 포함되고 정책 필드에는 권한부여를 위한 매개변수들이 포함된다. 예를 들어 호스트(Host), 사용자(User), 시간(Time), 권한(Privilege(생성(Create)/수정(Modify)/추가(Append)/읽기(Read)/삭제(Delete)/실행(Execute)))등이다. 인증서, 정책, 패이로드 필드에 들어가는 정보들은 네트워크를 통해 실행하고자 하는 다른 액티브 노드로 전달되어지는데 이때 공격자에 의해 정보들이 노출될 수 있고 조작될 수 있다. 이를 방지하기 위해 DES CBC 암호화/복호화 알고리즘을 적용한다. DES CBC 암호화를 하기 위해서는 서버의 공개키와 초기 벡터(Initial Vector)값이 필요하고, 복호화를 위해서는 암호화된 공개키와 암호화된 초기 벡터 값이 필요하다. 그리고 이 알고리즘을 적용하기 전에 메시지의 길이를 적정히 축약하기 위해 MD5 메시지 압축 알고리즘을 적용하였다. (3)은 MD5 메시지 압축 알고리즘을 적용하는 과정이고 (4)는 복호화를 위해 암호화된 공개키와 암호화된 초기 벡터 값을 다음 액티브 노드에 보내는 과정이다. (5)는 MD5 메시지 압축의 결과인 키 값과 초기 벡터 값으로 DES CBC 암호화를 수행하는 과정이고, (6)은 암호화된 공개키와 암호화된 초기 벡터 값으로 DES CBC 복호화 하는 과정이다.



(그림 5) 보안 모듈에서의 암호화/복호화 동작

다음으로 수신한 메시지의 변경여부를 검증하기 위해 RSA 전자 서명 알고리즘을 추가로 적용하였고, (그림 6)은 보안 모듈에서 수행하는 RSA 전자 서명 알고리즘의 처리과정을 보여준다.



(그림 6) 보안 모듈에서 RSA 전자 서명(Digital Signature) 동작

이 부분은 액티브 패킷을 수신한 액티브 노드에서 이 패킷이 전송되는 도중 악의적인 공격자에 의해서 내용이 바뀌어 졌는지를 검증해주는 부분이 된다. 암호/복호화 과정과 마찬가지로 (1)과 (2)과정에서 CA로부터 인증서와 비밀키를 받은 후, (3)과정에서 정책과 패이로드를 MD5 메시지 압축한 결과와 (4)과정에서 송신측 액티브 노드의 비밀키를 이용하여 RSA 전자 서명 알고리즘을 처리하고, (5)과정에서 받은 액티브 패킷의 서명 필드를 검증하는 과정이다. 이 과정들이 수신한 메시지의 무결성을 검증하기 위해 처리되어 지는 부분이다.

(그림 7)은 송신 액티브 노드에서 암호화 알고리즘을 적용한 액티브 패킷의 생성 결과를 나타낸 것이다. 만약 수신된 액티브 패킷이 액티브 네트워크 상에 이동하는 중 어떠한 악의적인 공격자나 네트워크 환경적인 잡음으로 인하여 변질되었다면, 인증 모듈로 넘어가지 않고 폐기된다. (그림 8)은 수신 노드에서 액티브 패킷이 제대로 검증되지 않아 다음 단계로 넘어가지 못하고 액티브 패킷이 바로 폐기됨을 보여주는 결과 화면이다. 이 결과에서 알 수 있듯이, 보안 모듈에서의 안전한 패킷 평가와, 단계적인 모듈 처리로 인하여, 기존의 액티브 보안 연구보다 효율적임을 알 수 있다.

```
[root@youuem linux-build]# ./clientauthengine
==================================
Message Digest & Signature Module Start!
Message Digest & Signature are OK!
==================================
Message Digest & Encryption Module Start!
Message Digest & Encryption are OK!
==================================
[root@youuem linux-build]#
```

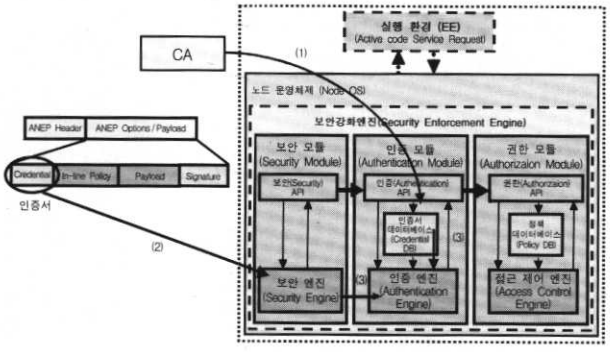
(그림 7) 송신 노드에서 액티브 패킷 생성 결과

```
[root@youuem linux-build]# ./serverauthengine
==================================
Security Module Start!
ERROR: Signature is incorrect while verifying file
[root@youuem linux-build]#
```

(그림 8) 수신 노드에서 액티브 패킷이 검증되지 않은 결과

4.2 인증(Authentication) 모듈

이 모듈에서는 송신측 액티브 노드와 액티브 패킷을 보낸 송신자의 신원을 인증하기 위해 처리되어 지는 부분을 구현하였다. 보안 모듈에서의 처리과정이 성공적으로 수행되고 나면, 인증 모듈에서는 액티브 패킷에서 인증서 필드를 복호화 한 뒤, 복호된 인증서 내용과 CA로부터 받은 데이터베이스에 저장중인 송신측 액티브 노드의 인증서 내용을 비교하여, 액티브 패킷이 유효한 액티브 노드와 사용자로부터 전송되어 온 액티브 패킷임을 보증해 주게 된다. 인증 모듈에서의 처리 후, 적합한 액티브 패킷이라 판단되면 권한부여 모듈로 넘겨주고, 그렇지 않다면 액티브 패킷을 폐기시켜 원하는 서비스를 제공해주지 않는다. (그림 9)는 인증 모듈에서 적용한 알고리즘의 처리과정을 보여준다.



(그림 9) 인증 모듈의 동작과정

(1)은 CA로부터 송신측 액티브 노드의 인증서를 사전에 수신 받아, 수신측 액티브 노드의 인증서 데이터베이스에 저장하는 과정이고, (2)는 수신된 액티브 패킷의 인증서 필드에서 인증서를 추출하여 앞서 언급한대로 보안 모듈에서 처리되는 과정이다. (3)은 두 개의 인증서를 받은 뒤 비교하여, 송신측이 보낸 인증서가 변하지 않았음을 인증한다. 이로써 액티브 패킷이 네트워크에서 전송되는 도중 악의적인 공격이나 오류로 인한 변질을 겪지 않았음을 보장할 수 있다. 또한 인증서의 유효기간을 검사하여 만기된 다른 인증서를 훔친 것이 아니라 송신측 액티브 노드가 직접 받은 유효한 인증서임을 확인한다. 즉 수신 받은 액티브 패킷은 유효함이 검증되고, 다음 단계인 권한부여 모듈에서 제공하는 적합한 권한부여 절차를 거칠 수 있게 된다.

(그림 10)은 수신된 액티브 패킷의 인증서 내용이 조작되어 인증 모듈 과정을 제대로 수행하지 못하고 액티브 패킷이 바로 폐기됨을 보여주는 결과 화면이다.

```
[root@youuem linux-build]# ./serverauthengine
==================================
Security Module Start!
Active Packet Signature verified.
==================================
Authentication Module Start!
It's Valid DATE!!!
no!! issuer name is not match!
[root@youuem linux-build]#
```

(그림 10) 수신된 액티브 패킷의 인증서가 조작된 결과

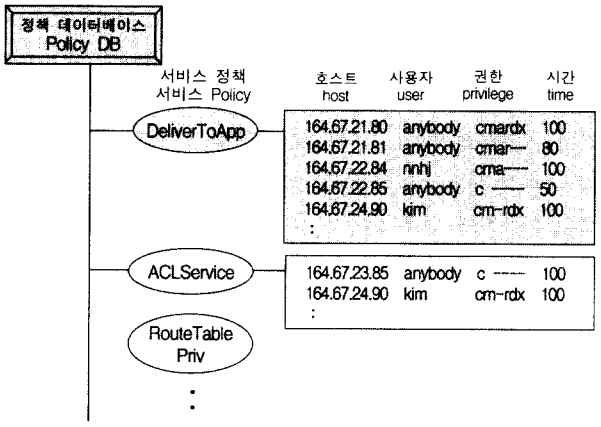
4.3 권한부여(Authorization) 모듈

권한부여 모듈은 액티브 패킷에 대한 인증이 완료된 후에 실제로 이 노드에서 액티브 패킷의 실행을 위한 권한을 줄 것인지 체크하는 기능을 담당한다. 먼저 패킷이 노드에서의 실행을 위해 요구하는 자원에 대한 접근이 정당한 것

인지 검증하고, 자원 사용에 대해 제한된 권한을 부여하게 된다.

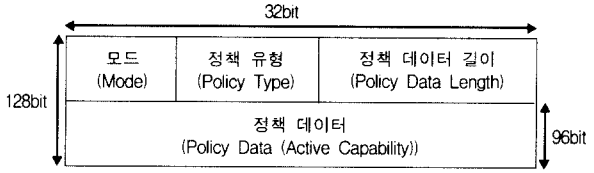
이 모듈에서는 패킷의 정책 옵션 필드의 내용을 기반으로 권한부여 기능을 수행한다. 인증된 액티브 패킷의 정책 옵션 필드에는 노드에서의 실행을 위한 자원 할당 요청이 들어 있다. 각 액티브 노드는 정책 데이터베이스(Policy DB)를 갖고 있으며, 정책 데이터베이스는 그 노드에서 실행 가능한 각 서비스 별 정책을 포함하고 있다. 해당 서비스의 정책에는 어떤 호스트와 사용자가 얼마만큼의 권한을 갖고 실행할 수 있는가가 명시되어 있다.

(그림 11)은 전반적인 정책 데이터베이스 구조와 간단한 예이다.

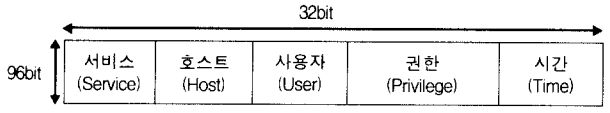


(그림 11) 정책 데이터베이스 구조

수신되는 패킷의 자원 할당을 위한 요청 관련 정보에는 호스트, 사용자, 권한 그리고 시간 정보가 포함된다. (그림 12)는 본 논문에서 제안한 패킷에 포함되는 정책 옵션 필드의 포맷이다. 정책 데이터 포맷은 (그림 13)과 같이 구성하였다.



(그림 12) 정책 필드 포맷



(그림 13) 정책 데이터 포맷

호스트와 사용자 필드에는 해당 서비스에 접근 하고자 하는 호스트와 사용자가 명시되며, 권한 필드에는 이 서비스를 실행하기 위한 권한 모드가 명시된다. 본 논문에서는 권한 모드를 C(생성.Create), M(수정.Modify), A(추가.Append), R(읽기.Read), D(삭제.Delete), E(실행.Execute)으로 명시하고 있다. 시간 필드에는 노드에서의 서비스 수행시 타임 아웃을 판단하기 위한 시간이 정해진다. 서비스 필드는 패킷이 실행을 요청하는 서비스를 의미한다. 노드 내에는 각 서비스 별로 정책 데이터베이스가 저장되어 있어서, 권한부여 모듈은 패킷이 요청한 서비스에 해당하는 정책 데이터베이스를 참고하여 적절한 권한을 부여한다.

수신된 패킷의 정책 데이터 필드에 명시된 정보들은 노드 내의 정책 데이터베이스에 명시된 정책들과 비교된다. 요청한 정보들이 모두 적절한 것이면 패킷을 받아서 실행 환경으로 넘겨주고 적절하지 못하면 폐기한다.

권한부여 모듈에서는 패킷의 권한부여 외에 노드내의 정책 데이터베이스를 관리하는 기능도 제공한다. 정책 데이터베이스 내의 정책 정보들은 수정 혹은 삭제될 수도 있으며, 새로운 서비스나 항목에 대한 정책이 정책 데이터베이스에 추가될 수도 있다. 이를 위해 제공되는 필드가 (그림 12)의 모드 필드이다. 모드 필드에 명시된 값은 권한부여 모듈에서 패킷이 어떤 수행을 할 것인가를 결정한다. 모드 값이 1이면 이 패킷이 요청하는 자원 실행에 대한 적절한 권한을 부여한다. 모드 값 2와 3은 정책 데이터베이스를 관리하기 위한 모드로, 2는 새로운 항목을 추가하거나 현재 있는 항목을 변경하고자 할 때 사용되고, 3은 현재 있는 정책 내용을 삭제하고자 할 때 사용된다.

(그림 14)는 권한부여 모듈의 수행 결과로 수신된 액티브 패킷의 정책 항목 중 적절하지 않은 권한을 요구하여 성공적으로 수행을 마치지 못하고 빠져나오는 결과이다. (그림 15)는 보안 모듈과 인증 모듈을 모두 성공적으로 통과하고 권한부여 모듈을 성공적으로 통과하여 액티브 보안 엔진에서 제공하는 모든 보안 관련 알고리즘의 수행 결과가 성공적으로 이루어 졌음을 보여주는 결과 화면이다.

```
[root@youyem linux-build]# ./serverauthengine
==================================
Security Module Start!
Active Packet Signature verified.
==================================
Authentication Module Start!
It's Valid DATE!!!
complete match!
Source Node is Authenticated!
==================================
Authorization Module Start!
Read request information..
Read request information more..
Call authorization policy..
Start authorization function..
Open DeliverToApp DB successfully.
host ok..
user ok..
Checking privilege..
CAN NOT ACCESS!
[root@youyem linux-build]#
```

(그림 14) 권한이 적절하지 않은 결과

```
[root@yoyuem linux-build]# ./serverauthengine
====================================
Security Module Start!
Active Packet Signature verified.
====================================
Authentication Module Start!
It's Valid DATE!!!
complete match!
Source Node is Authenticated!
====================================
Authorization Module Start!
Read request information..
Read request information more..
Call authorization policy..
Start authorization function..
Open DeliverToApp DB successfully.
host ok..
user ok..
Checking privilege..
Privilege ok..
Checking time..
Date is ok..
Authorization is OK!
====================================
[root@yoyuem linux-build]#
```

(그림 15) 권한부여 모듈이 성공적으로 수행된 결과

(그림 16)은 권한부여 모듈에서의 부가적인 기능으로 필요에 따라 즉, 액티브 네트워크의 상황에 따라 정책을 추가할 필요성이 있는데 이를 위하여 구현한 부분의 결과화면이다. (그림 17)은 권한부여 모듈에서 정책 추가 외에 또 다른 부가적인 기능으로 액티브 네트워크의 상황에 따라 정책을 삭제할 필요성이 있는데 이를 위하여 구현한 부분의 결과화면이다.

```
[root@yoyuem linux-build]# ./serverauthengine
====================================
Security Module Start!
Active Packet Signature verified.
====================================
Authentication Module Start!
It's Valid DATE!!!
complete match!
Source Node is Authenticated!
====================================
Authorization Module Start!
Read request information..
Read request information more..
Open  DeliverToApp DB file successfully.
Add successfully!
[root@yoyuem linux-build]#
```

(그림 16) 권한부여 모듈에서 정책 추가를 수행한 결과

```
[root@yoyuem linux-build]# ./serverauthengir
====================================
Security Module Start!
Active Packet Signature verified.
====================================
Authentication Module Start!
It's Valid DATE!!!
complete match!
Source Node is Authenticated!
====================================
Authorization Module Start!
Read request information..
Read request information more..
Open  DeliverToApp DB file successfully.
 Searching to remove..
Open  DeliverToApp DB file successfully.
164.67.21.80 anybody cmardx 100
164.67.21.81 anybody cmar-- 100
164.67.21.82 anybody cmard- 100
164.67.21.84 anybody cma--- 100
164.67.21.85 anybody c----- 100
164.67.21.90 anybody cmardx 100
Remove successfully!
Remove policy successfully!
[root@yoyuem linux-build]#
```

(그림 17) 권한부여 모듈에서 정책 삭제를 수행한 결과

5. 결 론

액티브 노드가 산재해 있는 액티브 네트워크는 네트워크의 상태에 따라 필요하다면 즉시 실행 가능한 코드나 데이터를 소프트웨어적으로 실행하여 적용함으로써 기존의 네트워크가 가지고 있던 많은 문제점들을 해결할 수 있는 차세대 네트워크 구조이다. 즉 액티브 네트워크는 기존의 네트워크와는 달리 네트워크를 거치는 도중에 어떤 처리를 수행함으로써 기존의 네트워크가 할 수 없었던 능동적이고 유동적인 패킷 처리 기능을 사용자에게 제공할 수 있다. 하지만 이렇게 패킷에 담긴 액티브 코드가 실행되기 위해 액티브 노드의 자원에 접근함으로써, 정당하지 못한 패킷이 액티브 노드의 자원이나 시스템 자체에 악영향을 끼칠 수 있는 가능성이 발생하게 된다.

액티브 네트워크의 장점에도 불구하고 액티브 네트워크는 아직 완전히 네트워크 망에서 정착되고 정립된 단계가 아니기 때문에 액티브 네트워크 상에는 아직 많은 보안상의 문제가 남아 있다. 또한 라우터 등의 중간 노드의 시스템에 사용자의 실행 가능한 코드가 직접 접근해서 그 노드의 기능을 무력화시킬 수도 있다는 점에서 액티브 네트워크의 보안은 기존의 네트워크 망보다 더 치밀하게 고려되어야 한다.

따라서 본 논문에서는 보안상의 문제점들을 해결하여 액티브 네트워크의 전체적인 보안성을 강화하기 위한 보안 강화 엔진을 제안하였고, 제안된 보안 강화 엔진은 보안 모듈, 인증 모듈 그리고 권한부여 모듈로 구성되어 액티브 네트워크 상에서 안전한 통신을 도와준다. 이때 소스 노드에서 목적지 노드로 액티브 패킷을 전송할 경우를 가정하여 인증과 권한부여 기능을 제공하는 액티브 엔진에 대해 구현하였다. 이 경우 "홉-바이-홉(Hop-by-Hop)"으로 확장했을 때 추가적으로 발생할 고려 사항에 대한 논의가 부족한 실정이다. 따라서, IP 망에서의 홉-바이-홉 경로로 액티브 패킷이 안전하게 전송되기 위하여, 여러 개의 액티브 노드들에 대해 각각을 구별해 인증과 권한부여를 제공해줄 수 있는 방안에 대한 연구가 앞으로 필요하다.

참 고 문 헌

[1] K. Psounis, "Active Network : Applications, Security, Safety and Architecture," IEEE Communications Serveys, 1999.

[2] Security Architecture for Active Nets by AN Security Working Group : 1998, Modified by Seraphim Group, 2000.

[3] R. H. Campbell, Z. Liu, M. D. Mickunas, P. Naldurg and S. Yi, "Seraphim : Dynamic Interoperable Security Archi-

tecture for Active Networks," IEEE OPENARCH 2000, Tel-Aviv, Israel, Mar., 2000.

[4] L. Dang, "CANSA (Certificate Active Network Security Architecture)," Basser Department of Computer Science, University of Sydney, 1998.

[5] M. Blaze, J. Feigenbaum, J. Ioannidis and A. D. Keromytis, "The Role of Trust Management in Distributed System Security, Secure Internet Programming : Issues in Distributed and Mobile Object Systems," Lecture Notes on Computer Science, Springer-Verlag, 1999.

[6] Defense Advanced research Projects Agency, http://www.darpa.mil/ato/programs/activenetworks/actnet.htm.

[7] D. S. Alexander, W. A. Arbaugh, A. D. Keromytis and J. M. Smith, "A Secure Active Network Environment Architecture : Realization in SwitchWare," IEEE Network Magazine, special issue on Active and Programmable Networks, 12(3), 1998.

[8] A. Galis, B. Plattner, J. M.Smith, S. Denazis, E. Moeller, H. Guo, C. Klein, J. Serrat, J. Laarhuis, G. T. Karetsos and C. Todd "A Flexible IP Active Networks Architecture," IWAN 2000 Conference, Nov., 2000.

[9] S. Merugu, S. Bhattacharjee, Y. Chae, M. Sanders, K. Calvert and E. Zegura, "Bowman and CANEs : Implementation of an Active Network," Invited paper at 37th Annual Allerton Conference, Monticello, IL, Sep., 1999.

[10] M. Wood and M. Erlinger, "Intrusion Detection Message Exchange Requirements : draft-ietf-idwg-requirements-10.txt," Oct., 2002.

[11] D. S. Alexander, B. Braden, C. A. Gunter, A. W. Jackson, A. D. Keromytis, G. J. Minden and D. Wetherall, "Active Network Encapsulation Protocol (ANEP)," http://www.cis.upenn.edu/switchware/ANEP/docs/ANEP.txt, 1997.

[12] R. Houseley, W. Ford, W. Polk and D. Solo, "Internet X.509 Public Key Infrastructure : X.509 Certificate and CRL Profile," RFC2459, Jan., 1999.

[13] S. Berson, B. Braden and S. Dawson, "Evolution of an Active Networks Testbed," Presentation at DARPA Active Networks Conference and Exposition 2002, San Francisco, CA, May, 2002.

[14] M. Collins, "Improving Open Web Architectures," University of Dublin, Sep., 2000.

[15] M. Bellare, J. Killan and P. Rogaway, "The Security of the Cipher Block Chaining Message Authentication Code," Computer and System Science, Vol.61, No.3, pp.362-399, Dec., 2000.

[16] B. Kaliski, "RSA Digital Signature Standards," RSA Laboratories, RSA Conference, 2000.

[17] J. Hu, "Smart Packets : Applying Active Networks to Network Management," BBN Technologies, Sep., 2002.

김 옥 경

e-mail : kimok@ewha.ac.kr

2001년 신라대학교 컴퓨터학과(이학사)

2002년~현재 이화여자대학교 과학기술대학원 컴퓨터학과 석사과정

관심분야 : 액티브 네트워크, 홈 네트워크 보안, 네트워크 보안

임 지 영

e-mail : jylim@biblc.ac.kr

1994년 이화여자대학교 전자계산학과(이학사)

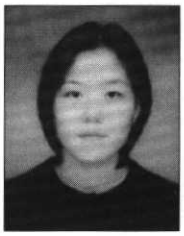
1996년 이화여자대학교 대학원 전자계산학과(이학석사)

2001년 이화여자대학교 과학기술대학원 컴퓨터학과(공학박사)

2001년~2003년 이화여자대학교 공과대학 컴퓨터학과 대우전임강사

2003년~현재 한국성서대학교 정보과학부 전임강사

관심분야 : 네트워크 보안, 액티브 네트워크, 네트워크 성능분석



나 현 정

e-mail : hjna@ewha.ac.kr

2002년 이화여자대학교 컴퓨터학과(공학사)

2002년~현재 이화여자대학교 과학기술대학원 컴퓨터학과 석사과정

관심분야 : 액티브 네트워크, DDOS 탐지, 무선 랜

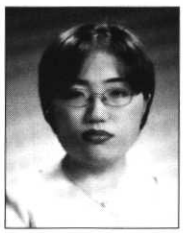
나 가 진

e-mail : nagajin@ewha.ac.kr

2002년 이화여자대학교 컴퓨터학과(공학사)

2002년~현재 이화여자대학교 과학기술대학원 컴퓨터학과 석사과정

관심분야 : 액티브 네트워크, 애드혹 네트워크 보안, 네트워크 보안



김 여 진

e-mail : zzin97@ewha.ac.kr

2001년 이화여자대학교 컴퓨터학과(공학사)

2002년~현재 이화여자대학교 과학기술대학원 컴퓨터학과 석사과정

관심분야 : 액티브 네트워크, 네트워크 보안

채 기 준

e-mail : kjchae@ewha.ac.kr

1982년 연세대학교 수학과(이학사)

1984년 미국 Syracuse University 컴퓨터학과(이학석사)

1990년 미국 North Carolina State University 컴퓨터공학과(공학박사)

1990년~1992년 미국 해군사관학교 컴퓨터학과 조교수

1992년~현재 이화여자대학교 컴퓨터학과 교수

관심분야 : 네트워크 보안, 액티브 네트워크 보안 및 관리, 인터넷/무선통신망/고속 통신망 프로토콜 설계 및 성능 분석

김 동 영

e-mail : kdy63281@etri.re.kr

1993년 건국대학교 전자계산학과(이학사)

1998년 건국대학교 전자계산학과 대학원(이학석사)

1998년~2001년 윈베이스 소프트웨어

2001년~현재 한국전자통신연구원

관심분야 : 네트워크 보안, 액티브 네트워크, 프로그래밍 언어