情報保護學會論文誌
第17卷 第6號, 2007. 12

터널링 기반 IPv4/IPv6 전이 기법을 위한 패킷 필터링 기능 개선

이 완 직[1†], 허 석 렬[1], 이 원 열[2], 신 범 주[1‡]
[1]부산대학교, [2]영산대학교

An Improvement of Packet Filtering Functions for Tunneling Based IPv4/IPv6 Transition Mechanisms

Wan-Jik Lee[1†], Seok-Yeol Heo[1], Won-Yeoul Lee[2], Bum-Joo Shin[1‡]
[1]Pusan University, [2]Youngsan University

요 약

IPv6가 현재의 IPv4 프로토콜을 완전히 대치하기 위해서는 상당한 시일이 소요될 것으로 예상된다. 이 기간 동안 인터넷은 두 개의 IP 프로토콜이 함께 사용될 것이다. 이 두 프로토콜의 공존을 위해 IETF에서는 여러 가지 IPv4/IPv6 전이 기법을 표준화하였다. 하지만 전이 기법에 주로 사용되는 터널링 때문에, IPsec 적용과 IPv6 패킷 필터링에 관한 보안 문제가 발생할 수 있다. 본 논문에서는 이러한 보안 문제 해결을 위해, 내부 헤더 필터링과 전이 기법 전용 필터링의 두 가지 패킷 필터링 개선 기법을 제안하였다. 또한 제안한 기법을 리눅스 넷필터(Netfilter) 프레임워크에서 구현하였으며, IPv4/IPv6 전이 기법 테스트 환경에서 구현 기능을 테스트하고, 시험적인 성능 평가를 수행하였다. 이러한 기능 시험과 성능 평가를 통해, 본 논문의 패킷 필터링 개선 기능이 시스템의 큰 성능 저하 없이, IPv4/IPv6 전이 기법의 패킷 필터링 문제들을 해결할 수 있음을 보였다.

ABSTRACT

It will need a quite long time to replace IPv4 protocol, which currently used, with IPv6 protocol completely, thus we will use both IPv4 and IPv6 together in the Internet during the period. For coexisting protocols, IETF standardized various IPv4/IPv6 transition mechanisms. However, new security problems of IPsec adaptation and IPv6 packet filtering can be raised by tunneling mechanism which mainly used in transition mechanisms. To resolve these problems, we suggested two improved schemes for packet filtering functions, which consists of an inner header filtering scheme and a dedicated filtering scheme for IPv4/IPv6 transition mechanisms. Also we implemented our proposed schemes based on Linux Netfilter framework, and we tested their filtering functions and evaluated experimental performance of our implementation on IPv4/IPv6 transition testbed. These evaluation tests indicated that our improved packet filtering functions can solve packet filtering problems of IPv4/IPv6 transition mechanisms without severely affecting system performance.

Keywords : IPv6, Tunneling, Packet filtering, security

Ⅰ. 서 론

IPv6 프로토콜은 이미 20여 년 전에 표준화되었기

접수일: 2007년 8월 17일; 채택일: 2007년 10월 24일
† 주저자, wjlee@pusan.ac.kr
‡ 교신저자, bjshin@pusan.ac.kr

때문에, IPv6 도입에 필요한 표준화와 기술 개발도 대부분 완료된 상태이다. 현재의 IPv6 보급 현황은 기대보다 미흡한 형편이지만, IPv4 주소 고갈이 심각해지는 2010년경부터 신규 인터넷 망을 중심으로 IPv6 도입이 본격화될 것으로 전망된다. 그러나 IPv6 보급이 활발해지더라도 비용 및 기술적인 제약 등으로 단기간 내에 현재의 IPv4를 대체하는 것은 힘들다. 따라서 향후 상당한 기간 동안 인터넷은 IPv4와 IPv6, 두 가지의 IP 프로토콜이 함께 운용될 것으로 예상된다. 이를 위해 IETF에서는 연동의 용도와 망 환경 등에 따라 여러 가지 IPv4/IPv6 전이 기법(Transition Mechanism)을 표준화하였다.

하지만 최근의 연구에 의해, IPv4/IPv6 연동을 위한 전이 기법 적용이 새로운 보안 문제를 발생시킬 수 있다고 지적되고 있다[1,2]. 이는 IPv6 프로토콜 자체에서 발생할 수 있는 보안 위협 요소 이외, 전이 기법에서 사용하는 터널링 등에 의해 IPsec[3] 적용이 불가능해지고, 터널링을 위해 생성된 이중 헤더 때문에 방화벽에서 수행하는 패킷 필터링이 정상적으로 동작하지 않는 등의 문제들이 발생하기 때문이다.

본 논문에서는 이러한 보안 문제들을 개선하기 위해, 터널링 기반의 IPv4/IPv6 전이 기법을 위한 내부 헤더 필터링 기능과 전이 기법 전용 패킷 필터링 기능을 제안하였다. 제안된 필터링 기능들은 터널링을 위해 생성된 이중 IP 헤더를 감지하여 각 헤더의 IP 프로토콜 버전에 적합한 패킷 필터링 규칙을 적용할 수 있게 하고, 각종 전이 기법에 적합한 새로운 패킷 필터링 기능을 제공한다. 또한 본 논문에서는 제안한 패킷 필터링 기능을 리눅스 2.6.10 커널과 넷필터(Netfilter) 프레임워크 기반으로 구현하였으며, 리눅스 PC들로 구성된 테스트 환경에서 기능 검사와 시험적인 성능 측정을 수행하였다.

본 논문의 2장에서는 IPv4/IPv6 전이 기법에서 발생할 수 있는 보안 문제에 대해 분석하고, 3장에서는 터널링 기반 IPv4/IPv6 전이 기법의 패킷 필터링 기능 개선 방안과 구현에 대해 기술한다. 4장에서는 구현 기능의 동작 검사와 성능 측정 내용을 설명하고, 마지막 5장에서 결론을 맺는다.

Ⅱ. IPv4/IPv6 전이 기법의 보안 문제

2.1. 표준 IPv4/IPv6 전이 기법

IPv6가 현재 IPv4 기반의 인터넷에 도입되기 위해서는 IPv4에서 IPv6로의 자연스러운 이전을 지원하는 전이 기법에 관한 연구가 필수적이다. IETF에서는 NGTrans 워킹그룹을 별도로 구성하여, IPv6 망으로 완전한 전이 이전까지 IPv4/IPv6 연동을 위한 전이 기법들을 표준화해왔다.

[표 1]은 IETF에서 현재까지 표준화된 대표적인 IPv4/IPv6 전이 기법들을 요약한 것이다. [표 1]의 '분류' 열에서 보는 바와 같이 NAT-PT를 제외한 모든 전이 기법은 IPv6-over-IPv4 용도의 전이 기법이다. IPv6-over-IPv4는 광역 IPv4 전용망을 경유한 두 IPv6 망 혹은 단말 간의 통신을 지원하는 목적의 전이 기법을 의미한다. 또한 [표 1]의 설정 터널링, 6to4, ISATAP 전이 기법은 IPv4 영역 통과를 위해 IPv6 패킷이 IPv4 헤더에 캡슐화 되는 IPv6-in-IPv4 형태의 터널링을 사용한다. 다만 Teredo는 IPv4 NAT (Network Address Translator) 하의 IPv6(듀얼 스택) 단말을 위한 전이 기법으로써, IPv6 패킷이 IPv4 영역을 통과할 때

〔표 1〕 대표적인 IPv4/IPv6 전이 기법

기 법	분 류	표준화	설 명
설정 터널링	IPv6-over-IPv4 수동 터널링	RFC 4312	듀얼스택 IPv6 망 또는 단말에서 IPv4 망을 경유하여 외부 IPv6 망 또는 단말을 접속
6to4	IPv6-over-IPv4 자동 터널링	RFC 3056	듀얼스택 IPv6 망 또는 단말이 IPv4 망을 경유하여 외부 IPv6 망 또는 단말을 자동 접속
ISATAP	IPv6-over-IPv4 자동 터널링	RFC 4124	IPv4 전용망 내의 듀얼스택 IPv6 단말이 외부 IPv6 망 또는 단말을 자동 접속
Teredo	IPv6-over-UDP 자동 터널링	RFC 4380	IPv4 전용망 내의 NAT 하의 듀얼스택 IPv6 단말이 외부 IPv6 망 또는 단말과 접속
NAT-PT	IPv6-to-IPv4 프로토콜 변환	RFC 2766	전용 IPv6 단말이 외부 IPv4 단말에게 접속, IPv6 패킷을 IPv4 패킷으로 변환 또는 그 역을 수행함

IPv6-in-IPv4/UDP 형태로 전송된다. [표 1]의 '분류' 열에 표시된 수동 터널링과 자동 터널링의 구분은 IPv4 영역 통과를 위한 터널을 수동으로 생성하는가 아니면, IPv6 패킷의 주소 정보 등을 이용하여 자동으로 생성하는가에 따라 분류한 것이다.

2.2. 터널링 기반 IPv4/IPv6 전이 기법 보안 문제

2.2.1. IPsec 적용 문제

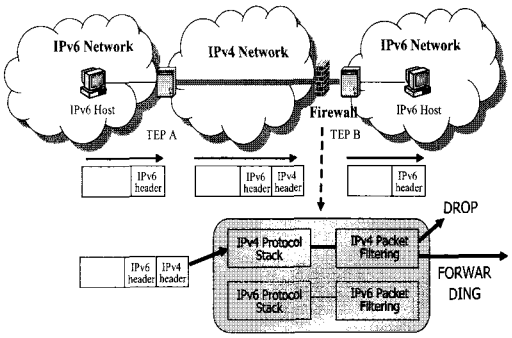
IPv4에서는 IPsec이 VPN(Virtual Private Network) 등의 높은 수준의 보안이 요구되는 통신 서비스에 주로 사용되었다. 그러나 IPv6는 IPsec을 모든 종단 간의 통신에 사용하도록 규정하고 있다. 따라서 IPv6에서는 IPsec 구현이 필수적이며, IPv4에 비해 IPsec 사용이 많이 활성화될 것으로 예상된다. 하지만 실제 활용 면에서 보면, 아직 인터넷 전역에서 사용 가능한 공개키 기반(Public Key Infrastructure: PKI)과 키 교환 기법 등의 실현에 많은 의구심을 품고 있다[2].

또한 현재의 IPsec 보안 방식은 6to4, ISATAP과 같은 자동 터널링 기반 전이 기법 상에서는 적용이 불가능하다는 문제가 발생한다. 설정 터널과 같은 IPv4/IPv6 전이 기법이나 VPN을 위한 터널링 상에서는 터널을 생성하는 두 라우터(또는 단말)의 종단 주소를 서로 알 수 있으며, 이 주소가 고정됨으로 IPsec의 보안 연결(Security Association) 생성이 가능하다. 하지만 6to4, ISATAP과 같은 자동 터널링 방식에서는 터널이 동적으로 생성되었다가 해지되며, 터널의 종단 주소도 동적으로 변경되기 때문에 IPsec의 보안 연결 생성이 불가능하다[4]. 따라서 자동 터널을 이용하는 전이 기법에서는 패킷 필터링 등의 기존 보안 기술을 적용할 수밖에 없다.

그리고 설정 터널링 방식에서도 IPsec 적용을 위한 메커니즘이 규정되지 않았기 때문에, 현재 IETF에서 설정 터널링 IPv4/IPv6 전이 기법 상에서 IPsec 적용 메커니즘을 표준화하고 있다[5].

2.2.2. 방화벽에서 터널링 패킷의 필터링 문제

터널 기반 전이 기법들의 IPv6 패킷은 호환되지 않는 IPv4 영역을 통과하기 위한 외부 IPv4 헤더와 실제 사용되는 내부 IPv6 헤더, 이렇게 두 가지의 IP 헤더를 가지게 된다. 하지만 현재 대부분의 방화벽에서는 외부 IPv4 헤더에 대한 필터링만 수행할 수 있고, 내부 IPv6 헤더에 대한 필터링 기능을 수행하지 못한다.

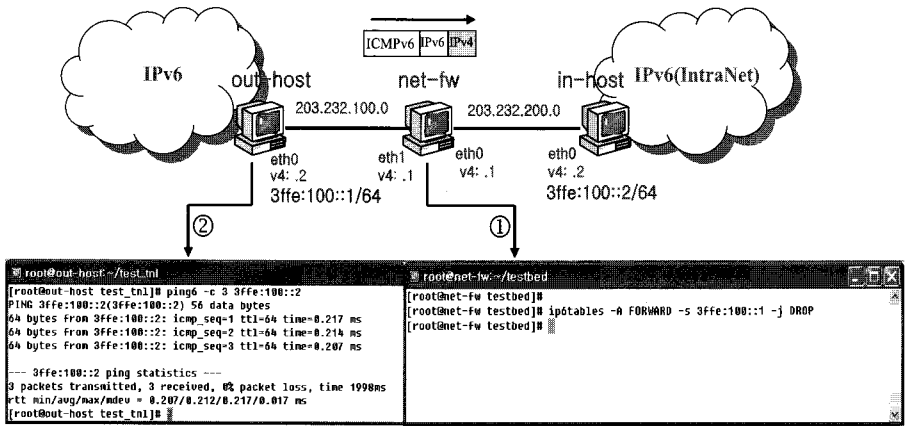


〔그림 1〕 IPv4/IPv6 전이 기법 상에서 발생하는 터널링 패킷에 대한 기존 방화벽의 필터링 과정

[그림 1]은 IPv4/IPv6 전이 기법에서 발생하는 터널링 패킷에 대한 기존 방화벽의 패킷 필터링 과정을 보여준다. [그림 1]과 같은 환경에서는 TEP (Tunnel End Point) A와 TEP B 사이에서 IPv4 터널이 생성되며, 이 터널 사이에는 이중 헤더를 가진 IPv6-in-IPv4 터널링 패킷이 전송된다. [그림 1]에서 방화벽(Firewall)은 외부 망에서 유입되는 IP 패킷들의 버전에 따라, 적합한 IP 프로토콜 스택이 선택되고, 이 프로토콜 스택의 포워딩 과정에서 패킷 필터링 규칙을 적용한다. [그림 1]의 방화벽에 이미 IPv4, IPv6 필터링 규칙이 모두 설정되어 있더라도, 기존 방화벽은 터널링 패킷을 일반적인 IPv4 패킷으로 오인하여 처리하기 때문에 IPv4 내부에 포함된 실제 IPv6 패킷에 대한 필터링은 전혀 수행할 수 없다.

따라서 터널링 기법이 방화벽 등의 보안 시스템을 우회하기 위한 용도로 사용된다면 아주 심각한 보안 위협 요소가 될 수 있다.

[그림 2]는 이와 같은 필터링 문제를 보여주기 위해, 간단히 구성된 IPv4/IPv6 전이 환경에서 기존 방화벽의 터널링 패킷 필터링 기능 시험을 보여준다. [그림 2]의 세 가지 호스트(out-host, net-fw, in-host)는 모두 리눅스 운영체제로 동작되며, netfw 시스템이 방화벽 기능을 수행하며, in-host가 방화벽이 보호하는 망 내부의 호스트, out-host가 외부 망의 호스트 역할을 수행한다. out-host와 in-host 사이에 수동 설정에 의한 IPv6-



〔그림 2〕 기존 방화벽의 터널링 패킷 필터링 기능 시험

over-IPv4 터널이 생성된 상태이다. out-host의 IPv6 TEP 주소는 3ffe:100::1이며 in-host의 IPv6 TEP 주소는 3ffe:100:2로 설정하였다. 따라서 out-host와 in-host 사이에는 IPv4 프로토콜만 동작하지만, out-host와 in-host 간의 IPv6 패킷은 IPv4 터널에 의해 전송된다.

[그림 2]에서 ①과 같이 방화벽 역할을 수행하는 net-fw에서 "송신지 주소가 3ffe:100::1인 모든 패킷을 폐기하라"는 리눅스 IPv6 패킷 필터링 명령어(ip6tables)를 수행시켰음에도 불구하고, ②에 out-host(3ffe:100::1)에서 in-host(3ffe:100::2)로 수행한 ping6 명령어가 정상적으로 동작함을 확인할 수 있다. 이러한 결과는 앞에서 설명한 바와 같이 터널링된 IPv6 패킷을 방화벽(net-fw)에서 전혀 필터링하지 못함을 보여준다.

2.2.3. 관련 연구

IPv6 보급이 활발하지 않은 현재 시점에서, IPv6 보안에 관한 연구는 아직 시작 단계로 볼 수 있으며, 일부 수행 중인 연구도 업체나 학계보다는 IETF의 IPv6 표준화 그룹 중심으로 진행되고 있다. 현재 IETF에서는 IPv6 도입에 의한 보안 문제를 크게 IPv6 프로토콜 자체에 기인한 문제와 IPv4/IPv6 전이 환경에서 발생할 수 있는 보안 문제로 구분하고 있다[1]. 이들 중, IPv6 프로토콜에 자체에 기인한 보안 문제에 비해 IPv4/IPv6 전이 환경에서 발생할 수 있는 보안 연구는 문제 해결보다도 보안 위협 요소와 발생 원인을 제시하고 있는 상황이다. 특히 본 논문과 같이 IPv4/IPv6 전이 기법 상의 방화벽에 관련된 연구는 매우 미흡한 실정이다. 참고문헌 [6]에서 각종 IPv4/IPv6 연동 기법에 대한 패킷 필터링 방식을 제안하였지만, 본 논문과는 달리, [6]에서 제안한 패킷 필터링 방식은 개인 방화벽이나 터널의 종단 라우터와 같이 터널링 과정이 종료되어 내부 IPv6 헤더에 대한 필터링이 정상적으로 동작하는 환경에서만 적용 가능하다는 제한점을 가진다.

Ⅲ. 터널링 기반의 IPv4/IPv6 전이 기법의 패킷 필터링 기능 개선

3.1. 패킷 필터링 기능 개선 방안

본 논문에서는 터널링 기반 IPv4/IPv6 전이 기법의 보안 문제를 해결하기 위해 터널링 패킷의 내부 헤더 필터링과 전이 기법 전용 패킷 필터링의 두 가지 개선 방안을 제안한다.

1) 터널링 패킷의 내부 헤더 필터링
- 방화벽에 유입되는 패킷들 중 터널링 패킷을 인식, 이중 헤더를 가진 터널링 패킷일 경우, 패킷의 외부 헤더(IPv4)에 대한 패킷 필터링 후, 내부 헤더(IPv6)에 대한 패킷 필터링 기능을 적용

2) 전이 기법 전용 패킷 필터링 기능 지원
- 전이 기법 종류에 따른 패킷 허용/차단 기능
- 터널 종단 주소에 따라 패킷 허용/차단 기능
- 전이 기법의 내장 IPv4 주소 검사에 의한 패킷 차단 기능

첫 번째 개선 방안은 기존 방화벽의 패킷 필터링 방식을 터널링 패킷 내부 IPv6 헤더 필터링이 가능하도록 수정하는 것이다. 리눅스 넷필터와 freeBSD의 ipfw2와 같은 범용 패킷 필터링 엔진들은 IPv4 필터링 루틴과 IPv6 필터링 루틴이 분리되어 있다. 따라서 터널링 패킷과 같은 이중 헤더를 가진 패킷에 대해서는 외부 IPv4 헤더의 필터링 루틴을 수행한 후, 내부 IPv6 헤더의 필터링 루틴을 수행하도록 패킷 처리 구조를 수정해야 한다. 이러한 내부 헤더 필터링 기능이 지원된다면, [그림 2]에서 수행한 IPv6 패킷 필터링 명령이 바르게 동작하게 된다. 즉 터널링된 IPv6 패킷도 일반적인 IPv6 패킷 필터링 규칙에 의한 필터링이 가능해진다.

두 번째 개선 방안은 앞으로 활용 가능성이 높아지는 터널링 기반 IPv4/IPv6 전이 기법에 필요한 새로운 필터링 기능을 방화벽에서 추가하는 것이다. 이 전이 기법 전용 필터링 기능은 방화벽에서 전이 기법의 종류와 터널 종단(TEP) 주소에 따른 패킷 필터링 기능을 제공한다. 만약 이러한 기능이 제공된다면, 네트워크 관리자가 설정 터널링 전이 기법의 패킷만 허용하고, 6to4나 ISATAP 전이 기법에 의한 패킷은 모두 차단할 수도 있다. 또한 터널 종단 주소에 의한 패킷 필터링이 제공됨으로 전이 기법 패킷 중, 공인된 외부 터널 서버에서 유입된 패킷만 허용할 수도 있다.

자동 터널링 방식의 전이 기법에서는 TEP에서 IPv4 터널을 자동으로 생성할 수 있도록 6to4, ISATAP IPv6 주소 내에 IPv4 주소가 내장되어 있다. 이러한 내장 IPv4 주소는 유니캐스트 형식의 공인된 IPv4 주소만 허용되는데, 이러한 내장 IPv4 주소는 DoS(Denial of Service)나 DDoS(Distrbuted DoS)를 위해 악용될 수 있으므로 이러한 내장 IPv4 주소에 대한 패킷 필터링이 필요하다[2]. 두 번째 개선 방안의 세 번째, 전이 기법의 내장 IPv4 주소 검사 기능은 이러한 IPv6 내장 IPv4 주소의 필터링 기능을 지원한다는 것이다.

3.2. 내부 헤더 필터링 기능 설계 및 구현

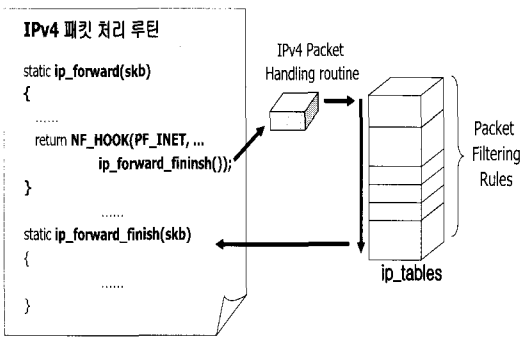
앞서 기술한 바와 같이, 터널링 패킷의 내부 헤더에 대한 필터링 규칙들이 제대로 적용되기 위해서는, 터널링 패킷 인식과 패킷들의 내부 IPv6 패킷에 대한 필터링 테이블 진입이 추가적으로 수행되어야 한다.

[그림 3]은 기존 방화벽 커널에서 IPv4 패킷의 필터링 처리 절차를 나타낸 것인데, 패킷을 포워딩하는 함수인 ip_forward()에서 패킷을 훅킹하여 넷필터의 패킷 필터링 처리 루틴으로 진입하는 과정(NF_HOOK 매크로 함수 호출)을 볼 수 있다. 따라서 2.2.2절에서 기술한 바와 같이, 터널링 패킷은 외부 IPv4 헤더에 의해 판단되어 처리되고 IPv4 패킷에 적용되는 패킷 필터링 규칙들만 적용된다.

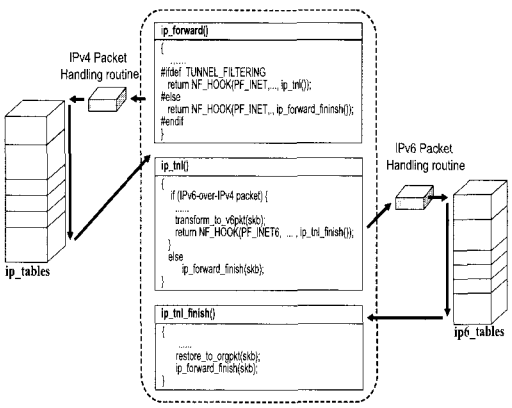
본 논문에서는 터널링 IPv6 패킷의 IPv6 필터링 규칙 적용을 위해, 터널링 패킷 인식과 내부 IPv6 패킷에 대한 필터링 루틴 진입 기능을 추가, 구현하였다.

[그림 4]는 이러한 기능 추가를 위해, [그림 3]의 기존 방화벽의 패킷 처리 과정을 수정·설계한 내용을 표현한 것이다.

[그림 4]의 ip_forward() 함수에서 컴파일 옵션이 TUNNEL_FILTERING이 설정된 경우, IPv4 패킷의 필터링 테이블 통과한 패킷은 본 논문에서 추가된



〔그림 3〕 기존 방화벽의 패킷 필터링 처리 과정



〔그림 4〕 수정된 패킷 필터링 처리 과정

〔표 2〕 전이 기법 전용 필터링(v6tnl) 매칭 문법

옵 션	인 수	설 명
mode	configured|6to4|isatap	전이 기법의 종류를 명시하며, 내부 망에서 허용(차단)할 수 있는 전이 기법의 종류를 명시할 수 있다.
tep-src| tep-dst	[!] v4addr[1][,v4addr]*[2]	터널의 송신자(수신자) 주소에 따른 패킷 매칭을 지원한다. !은 역매칭을 위해 사용된다.
malformed		인수가 없으며, 6to4, ISATAP의 내장 IPv4 주소가 적법성 등의 자동 터널링 기법에서 생성되지 않는 비정상적인 패킷 매칭에 사용된다.

〔표 3〕 v6tnl 구현 프로그램

프로그램 명	용 도	수행 기능
ip6t_v6tnl.h	공유 자료구조 정의	사용자가 입력한 ip6tables의 v6tnl 매치의 옵션과 인수 값을 저장하는 공유 자료구조(v6tnl_info) 정의
lib6t_v6tnl.c	ip6tables 확장 라이브러리	ip6tables의 확장 라이브러리로 컴파일되며, v6tnl 확장 매치의 관련 옵션과 인수를 파싱하고, 파싱된 옵션과 인수를 v6tnl_info 자료구조에 저장
ip6t_v6tnl.c	넷필터 확장 매치 커널 모듈	리눅스 커널 모듈로 컴파일되며, 실제 시스템으로 유입된 패킷들을 전달받은 v6tnl_info에 설정된 정의된 옵션들과 비교 판단하는 기능을 수행

ip_tnl() 함수를 수행하도록 NF_HOOK() 매크로 함수 호출을 변경하였다. ip_tnl() 함수는 터널링 패킷 여부를 판단하고, 내부 헤더가 IPv6일 경우, ip6_tables 필터링 테이블로 다시 진입시키는 기능을 수행한다. ip6_tables에 저장된 IPv6 패킷 필터링 규칙까지 모두 통과한 패킷들은 ip_tnl_finish() 함수로 리턴되고 다시 원래의 패킷 처리 루틴인 ip_forward_finish() 함수로 진입된다. 그런데, IPv6 패킷 필터링 테이블의 필터링 진입이 올바르게 수행되기 위해서는 내부 IPv6 패킷 앞에 IPv4 헤더가 존재하지 않는, 일반적인 IPv6 패킷인 것처럼 처리되어야 한다. 이를 위해 ip_tnl() 함수에서 호출되는 transform_to_ipv6pkt() 함수가 리눅스 커널의 패킷 자료 구조인, skb 구조체의 길이, 헤더 위치 등의 필드를 수정하여 터널링 패킷을 일반 IPv6 패킷인 것처럼 변환하고, ip_tnl_finish()함수에서 호출되는 restore_to_orgpkt() 함수가 다시 패킷 구조로 환원하는 기능을 수행한다.

3.3. 전이 기법 전용 필터링 기능 설계 및 구현

IPv4/IPv6 전이 기법 전용 패킷 필터링 기능은 넷필터의 확장 모드[7]로 설계하고, 구현하였다. 넷필터는 리눅스 커널의 네트워크 프레임워크로서, 사용자 프로그램인 iptables/ip6tables와 함께 동작하여, 패킷 필터링, NAT(Network Address Translation) 등의 기능을 지원하며, 새로운 패킷 필터링 기능 추가가 용이하도록 확장 모드를 지원한다.

전이 기법 전용 필터링 기능은 이 넷필터의 확장 모드 형태로 구현되었으며, 전이 기법 관련 패킷을 매칭하기 ip6tables 명령어에 새로 추가되는 모듈은 "v6tnl"로 정의하였다. v6tnl 확장 모듈에 의해 패킷을 매칭하는 문법을 [표 2]와 같이 설계하였다. 이렇게 설계된 옵션과 인수는 넷필터의 사용자 도구인 iptables 명령어에서 지정됨으로써 전이 기법에 대한 전용 패킷 필터링 규칙들이 방화벽에 적용된다.

1 v4addr : 점십진표기(dotted decimal form)나 DNS 형태의 IPv4 주소
2 * : '['와 ']' 사이의 내용이 0번 이상 반복될 수 있음

[표 2]의 mode 옵션은 필터링 규칙을 적용할 전이 기법을 지정하는 용도로 사용한다. 만약 이 옵션이 지정되지 않으면 모든 전이 기법 대상으로 v6tnl 필터링 규칙이 적용된다. tep-src 및 tep-dst 옵션은 공인되지 않은 외부 터널 서버로부터 유입된 전이 기법 패킷들을 필터링하는 용도로 사용할 수 있다. malformed 옵션은 지정된 전이 기법에서 생성될 수 없는 패킷을 자동 매칭하는 옵션으로써 크게 두 가지 기능으로 구성된다. 첫 번째 기능은 지정된 전이 기법이 6to4나 ISATAP과 같이 자동 터널링을 사용하는 전이 기법일 경우, 6to4나 ISATAP IPv6 주소에 내장된 IPv4 주소를 검사하여, 이 주소가 전이 기법에 정의된 것과 같이 라우팅 가능한 유니캐스트 IPv4 주소가 아닐 경우에는 매칭된다. 즉 사설 주소, 브로드캐스트 주소, 네트워크 주소와 같이 정상적이지 않은 내장 IPv4 주소를 가진 패킷들이 매칭된다. 두 번째 기능은 6to4에서만 적용되는 기능으로써, 6to4에는 터널 시에 6to4 주소에 내장된 IPv4 주소와 실제 외부 헤더의 터널용 IPv4 주소가 동일해야 한다. 따라서 6to4 주소에 내장된 IPv4 주소와 터널링을 위한 외부 헤더의 IPv4가 동일하지 않은 6to4 패킷들은 매칭된다.

[표 2]의 문법을 활용하여, 내부 망에 유입되는 전이 기법의 IPv6 패킷들 중 "6to4 전이 기법 패킷만 허용하고, 6to4 패킷들도 공인된 100.1.1.1 주소의 터널 서버에서 생성된 패킷만 허용하며, 6to4 전이 기법 규칙에서 생성이 불가능한 비정상적인 패킷들은 모두 폐기" 하는 규칙을 적용하기 위해서는 다음과 같은 ip6tables 명령어를 사용할 수 있다.

```
$ ip6tables -m v6tnl --mode configured -j DROP
$ ip6tables -m v6tnl --mode isatap -j DROP
$ ip6tables -m v6tnl --mode 6to4 --tep-src
  !100.1.1.1 -j DROP
$ ip6tables -m v6tnl --mode 6to4 --malformed -j
  DROP
```

이렇게 설계된 v6tnl의 구현은 [표 3]과 같이 하나의 헤더 파일과 두 개의 C 프로그램 파일로 구성된다. 구현 프로그램 중, [표 3]의 lib6t_v6tnl.c는 ip6tables 명령어의 동적 라이브러리 형태로 컴파일되어, ip6tables 명령어 중, "-m v6tnl" 다음에 입력된 각종 옵션과 관련 인수를 파싱 시에 자동 로딩된다. [표 3]의 ip6t_v6tnl.c는 커널의 모듈 형태로 컴파일되며, 넷필터 동작에 의해 실제 패킷과 v6tnl 옵션에 의해 설정된 필터링 규칙을 비교하여 매칭 여부를 판단한다.

Ⅳ. 구현된 패킷 필터링의 기능 검사 및 시험적인 성능 평가

4.1. 패킷 필터링 기능 검사

본 논문의 기능 시험은 [그림 2]와 같은 리눅스 시스템들로 구성된 테스트 네트워크에서 주로 수행되었다.

4.1.1. 내부 헤더 필터링의 기능 시험

본 논문에서 구현한 IPv6-in-IPv4 터널링 패킷의 내부 헤더 필터링의 기능을 확인하기 위해, net-fw에 IPv6 주소에 대한 필터링을 설정한 다음, [그림 2]와 동일한 형태의 기능 시험을 [그림 5]와 같이 수행하였다.

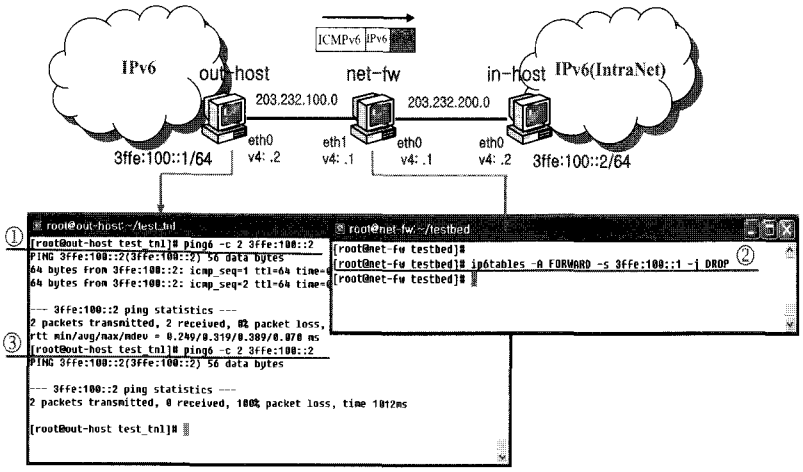
[그림 5] ①의 ping6가 제대로 동작하는 환경에서, net-fw의 ②로 표시된 것과 같이 "송신지 주소가 3ffe:100::1인 모든 패킷을 폐기하라"는 ip6tables 명령어를 수행하였다. 그 이후 ③과 같이, ping6 명령어가 전혀 동작하지 않음을 알 수 있다. 이와 같은 결과는 본 논문에서 구현한 내부 헤더 패킷 필터링 기능에 의해, IPv6 주소에 대한 패킷 필터링 기능이 정상적으로 동작함을 보여준다.

4.1.2. 전이 기법 전용 패킷 필터링(v6tnl) 기능 시험

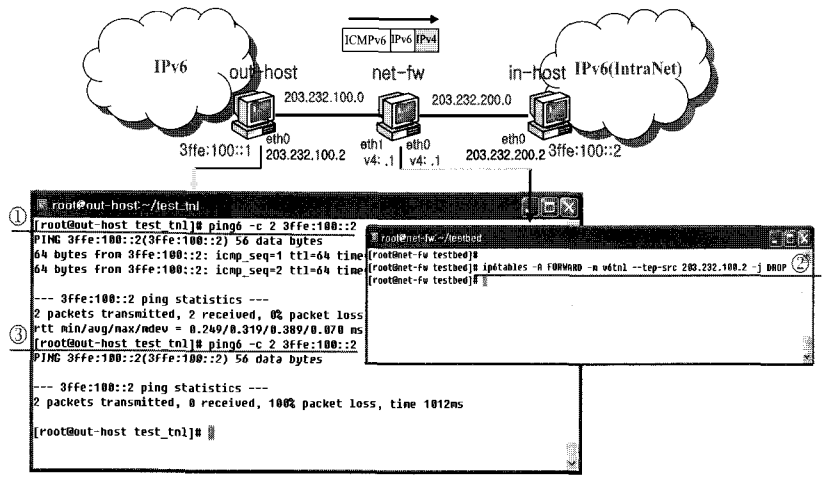
본 논문에서 설계하고 구현한 전이 기법 전용 필터링 기능 중, [그림 6]은 v6tnl의 'tep-src' 옵션을 사용하여 터널의 송신자 주소에 대한 전이 기법 패킷의 필터링 기능을 보여준다.

[그림 6]의 ①과 같이 ping6가 제대로 동작하는 환경에서, net-fw 터미널의 ②와 같이 v6tnl의 'tep-src' 옵션을 이용하여 "송신지 터널 종단 IPv4 주소가 203.232.100.2인 모든 패킷을 폐기하라"는 ip6tables 명령어를 수행하였다. 그 이후 ③과 같이 ping6 명령어가 전혀 동작하지 않음을 알 수 있다. 이는 v6tnl 패킷 필터링 기능에 의해 out-host의 터널 송신지가 "203.232.100.2"인 ping6 패킷이 모두 폐기되었기 때문이다.

[그림 6]에서 나타낸 v6tnl의 'tep-src' 옵션 이외,'mode', 'tep-dst', 'malformed' 옵션을 이용한 여러



〔그림 5〕 내부 헤더 필터링 기능에 의한 터널링 패킷 필터링 기능 시험



〔그림 6〕 v6tnl의 tep-src 옵션에 의한 터널링 패킷 필터링 기능 시험

가지 패킷 필터링 시험을 설정 터널, 6to4, ISATAP 등의 전이 기법 환경에서 수행하였으며, 모두 정상적으로 동작하는 것을 확인하였다.

4.2. 시험적인 성능 평가

본 논문에서 구현한 전이 기법을 위한 패킷 필터링 개선 기능이 기존 패킷 필터링에 추가된다면, 이 추가 기능 수행 때문에 기존 패킷 필터링 수행 지연 이외에 부가적인 지연이 발생할 것이다. 이 부가적인 지연을 측정하기 위해 기존 패킷 필터링 동작 환경에서 측정한 지연 및 처리량과 전이 기법을 위한 패킷 필터링 기능이 추가된 동작 환경에서 ping과 Netperf[8]를 이용하여 측정한 지연 및 처리량을 비교하였다.

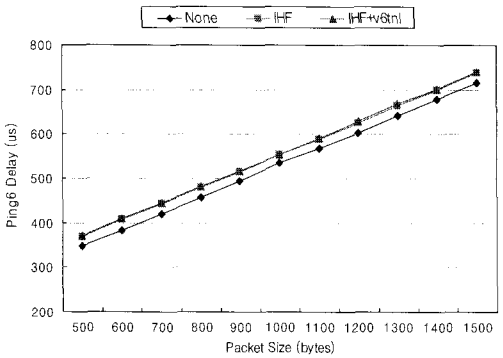
4.2.1. ping6에 의한 패킷 지연 측정

ping6에 의한 패킷 지연에 측정 시험에 대한 설정은 [표 4]에 나타내었다.

[표 4]에 나타낸 비교 대상과 같이 지연 시간 측정은, 본 논문의 패킷 필터링 개선 기능을 전혀 추가하지 않은 상태(None), 패킷 필터링 개선 기능 중 내부 헤더 필

〔표 4〕 ping6에 의한 지연 측정시험 설정

구 분	내 용
망 구성	설정 터널링 전이 기법(그림 5) 100Mbps Ethernet
패킷 크기	500~1500 byte (100Byte단위로 증가)
지연 산출	패킷 크기 당, 1000회 측정 후, 평균값
비교 대상	1)None 2)IHF 3)IHF+v6tnl



〔그림 7〕 ping6에 의한 지연 시간 측정 결과

터링 기능만 수행한 상태(IHF), 패킷 필터링 개선 기능의 내부 헤더 필터링과 v6tnl 패킷 필터링 기능을 함께 수행한 상태(IHF+v6tnl), 이렇게 세 가지 환경에서 수행되었다. 세 번째의 v6tnl 패킷 필터링은 'tep-src', 'tep-dst', 'malformed' 옵션 등을 이용하여 모두 6개의 v6tnl을 이용한 패킷 필터링 규칙을 net-fw 시스템에 설정하였다.

[표 4]의 비교 대상 항과 같이 지연 시간 측정은, 본 논문의 패킷 필터링 개선 기능을 전혀 추가하지 않은 상태(None), 패킷 필터링 개선 기능 중 내부 헤더 필터링 기능만 수행한 상태(IHF), 패킷 필터링 개선 기능의 내부 헤더 필터링과 v6tnl 패킷 필터링 기능을 함께 수행한 상태(IHF+v6tnl), 이렇게 세 가지 환경에서 수행되었다. 세 번째의 v6tnl 패킷 필터링은 'tep-src', 'tep-dst', 'malformed' 옵션 등을 이용하여 모두 6개의 v6tnl을 이용한 패킷 필터링 규칙을 net-fw 시스템에 설정하였다.

ping6에 의한 지연 시간 실험 결과를 [그림 7]에 나타내었다. [그림 7]의 결과를 보면, 1) None 환경과 2) IHF 환경의 지연 시간 차이는 전송 패킷 크기와 크게 상관없이 평균 22.2 us 정도이며, 1) None 환경과 3)IHF+v6tnl의 지연 시간 차이는 평균 24.4 us 정도인 것으로 측정되었다. 그런데 이 수치는 ping6의 왕복 지연 시간 값이며, 비교 환경의 지연 시간 차이도 두 개의 패킷(ping6 request와 ping6 reply)에 대한 필터링 지연 시간이 합산된 값이다. 따라서 하나의 패킷에 대한 비교 환경 간의 지연 차이는 1) None 환경과 2) IHF 환경 차이가 11.1 us, 1) None 환경과 3) IHF+v6tnl 환경과의 차이가 12.2 us 정도로 볼 수 있다. 이에 비해 2) IHF 환경과 3) IHF+v6tnl 환경과의 차이는 불과 1.1 us 정도에 불과한 것으로 나타났다.

이러한 결과를 종합해 볼 때, 내부 IPv6 헤더 필터링을 위해 부가적인 IPv6 필터링 테이블에 진입과 리턴 과정이 하나의 필터링 테이블 내에서 여러 개의 v6tnl 매치 검사 수행보다 오버헤드가 큰 것을 알 수 있다. 하지만 추가적인 IPv6 헤더의 필터링 검사를 위한 11 us 정도의 지연은 전체 방화벽 시스템의 성능을 크게 저하시킬 만한 오버헤드는 아니라고 생각된다.

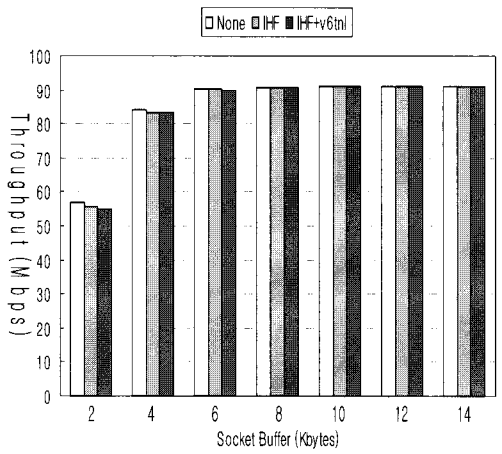
4.2.2 Netperf에 의한 데이터 처리량 측정

Netperf에 의한 데이터 전송 측정에 관한 상세한 설정은 [표 5]에 나타내었다. 측정 방법을 제외한 대부분의 측정 환경은 ping6에 의한 시험과 동일하다. [그림 8]은 Netperf에 의해 측정된 데이터 전송 처리량을 보여준다.

[그림 8]의 결과를 볼 때, 전송에 사용되는 소켓 버퍼의 크기가 작을 경우에는 부가적인 패킷 필터링 기능에

〔표 5〕 Netperf에 의한 데이터 처리량 측정시험 설정

구 분	내 용
망 구성	설정 터널링 전이 기법 환경(그림 5) 100Mbps Ethernet
전송 방식	설정된 소켓 버퍼 크기의 데이터를 TCP로 10초간 전송
처리량 산출	총 1,000회 측정 후, 평균값
비교 대상	1)None 2)IHF 3)IHF+v6tnl



〔그림 8〕 Netperf에 의한 데이터 처리량 측정 결과

의해 약간의 처리량 차이가 존재하지만, 소켓 버퍼의 크기가 일정 크기(8K) 이상이면 처리량의 차이가 거의 존재하지 않는다. 이는 소켓 버퍼가 큰 경우의 처리량 병목은 방화벽에서 수행되는 패킷 필터링 기능이 아니라, 송수신 응용 프로그램이나 송수신 시스템 버스 등의 다른 부분에서 발생하는 것으로 판단된다.

V. 결 론

앞으로 상당한 시간 동안 지속될 IPv4와 IPv6 공존 환경을 위해, 다양한 IPv4/IPv6 전이 기법이 활용되리라 예상된다. 이러한 형태의 IPv6 프로토콜 도입은 IPv6 프로토콜 자체의 보안 문제 이외, 전이 기법에서 사용하는 터널링 방식에 의한 IPsec 적용 문제와 이중 헤더 구성에 의한 패킷 필터링 동작 문제 등이 발생할 수 있다.

본 논문은 이러한 보안 문제를 패킷 필터링 측면에서 해결하기 위해, 내부 헤더 필터링 기능과 전이 기법 전용 필터링 기능의 두 가지 패킷 필터링 개선 방안을 제안하였다. 제안된 패킷 필터링 기능은 터널 상의 내부 IPv6 패킷 필터링을 가능하게 하고, 전이 기법 종류에 대한 수용 여부 설정, 터널 종단 주소에 따른 패킷 필터링, 전이 기법 표준에 위배된 비정상적인 패킷 차단 기능을 제공한다. 또한 본 논문에서는 제안한 패킷 필터링 개선 기능을 리눅스 커널과 넷필터 프레임워크에서 구현하였으며, 전이 기법 시험 환경에서 구현 기능을 확인하고, Netperf, ping6를 이용한 성능 측정을 수행하였다. 기능 시험과 성능 평가를 통해, 본 논문의 패킷 필터링 개선 기능이 방화벽의 큰 성능 저하 없이, 설정 터널, 6to4, ISATAP 전이 기법 환경의 패킷 필터링 문제를 해결할 수 있음을 보였다.

본 논문에서 제안한 개선 기능을 이용하면 앞으로 활용 가능성이 높은 터널링 기반의 IPv4/IPv6 전이 기법에 대한 방화벽의 보안 기능을 향상시킬 수 있으리라 기대되고, 특히 본 논문의 넷필터 방식의 구현은 임베디드 리눅스 기반의 방화벽, 라우터, 공유기 등에 쉽게 적용이 가능하다는 장점을 가진다.

현재 본 논문에서는 터널링 기반 전이 기법 중에서, 설정 터널링, 6to4, ISATAP 전이 기법의 보안 문제 해결만 연구 대상으로 삼았다. 2장에서 소개한 Teredo 전이 기법도 터널링 방식을 사용하지만, IPv6 패킷이 IPv4 패킷이 아닌 IPv4/UDP 패킷에 캡슐화 되기 때문에 본 논문의 연구 대상에서는 제외하였다. 따라서 향후 과제는 Teredo에 관한 패킷 필터링 기능 개선을 수행하는 것이다. Teredo는 IPv4 NAT (Network Address Translator) 통과를 위해 IPv6-in-UDP 터널링 방식을 사용하며, Teredo IPv6 주소도 IPv4 주소를 내장하고 있기 때문에, IPv4/UDP 내부에 포함된 IPv6 패킷 필터링 기능과 내장된 IPv4 주소에 대한 필터링 기능 개선이 필요하다.

참고문헌

[1] E. Davies, et al., *IPv6 Transition Co-existence Security Considerations*, draft-ietf-v6ops-security-overview-06.txt, April 2007.

[2] 신명기, 김형준, "IPv6 전환 환경에서의 보안 기술 분석," *전자통신 동향분석* 제21권 제5호, pp. 163~170, 2006.

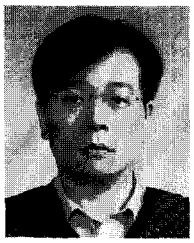
[3] U. Black, *Internet Security Protocols*, Prentice Hall PTR, 2000.

[4] W. Lee, et al., "A Secure Packet Filtering Mechanism for Tunneling over Internet," *ICESS 2007 LNCS 4523*, pp. 641~652, May 2007.

[5] R. Graveman, et al., *Using IPsec to Secure IPv6-in-IPv4 Tunnels*, draft-ietf-v6ops-ipsec-tunnels-05.txt, January 2007.

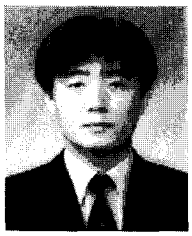
[6] 허석렬 외, "IPv4/IPv6 터널링 환경에 적합한 패킷 필터링 기능 설계 및 구현," *정보과학회 논문지:* 정보통신 제33권 6호, pp. 407~419, 2006.
[7] R.Russell, *Linux Netfilter Extension Howto*, http://www.netfileter.org/ documentation.
[8] R. Jones, *Care and Feeding of Netperf*, http://www.netperf.org/svn/netperf2.html
[9] The 6NET Consortium, *6net: An IPv6 Deployment Guide*, September 2005.
[10] C. Benevenuti, *Understanding LINUX Network Internals*, O'REILLY, 2005.

〈著者紹介〉

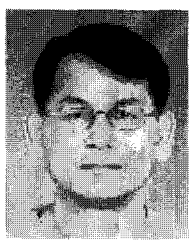


이 완 직 (Wan-JIk Lee) 정회원
1992년 2월 : 경북대학교 통계학과 학사
1994년 2월 : 경북대학교 컴퓨터공학과 석사
2007년 8월 : 경북대학교 컴퓨터공학과 박사
1997년 3월~2006년 2월 : 밀양대학교 컴퓨터공학과 교수
2006년 3월~현재 : 부산대학교 바이오정보전자공학과 교수
<관심분야> 통신 프로토콜, 네트워크 보안, 센서 네트워크

허 석 렬 (Seok-Yeol Heo) 정회원
1986년 2월 : 경북대학교 컴퓨터공학과 학사
1991년 2월 : 경북대학교 컴퓨터공학과 석사
1993년 2월 : 경북대학교 컴퓨터공학과 박사 수료
1992년 3월~2006년 2월 : 밀양대학교 컴퓨터공학과 교수
2006년 3월~현재 : 부산대학교 바이오정보전자공학과 교수
<관심분야> RFID/USN, u-Health, 네트워크 보안, 컴퓨터 네트워크



이 원 열 (Won-Yeoul Lee) 정회원
1987년 2월 : 경북대학교 전자공학과 학사
1993년 2월 : 경북대학교 컴퓨터공학과 석사
2002년 2월 : 경북대학교 컴퓨터공학과 박사
1997년 9월~2002년 2월 : 성심외국어대학 정보통신학과 조교수
2002년 3월~현재 : 영산대학교 IT건축대학 사이버경찰학과 조교수
<관심분야> USN, Mobility Management, Location Tracking, Resource Management



신 범 주 (Bum-Joo Shin) 정회원
1998년 8월 : 경북대학교 컴퓨터공학과 박사
1987년~2002년 : 한국전자통신연구원 연구원
2002년 3월~2006년 2월 : 밀양대학교 컴퓨터공학과 교수
2006년 3월~현재 : 부산대학교 바이오정보전자공학과 교수
<관심분야> 분산처리, 시스템소프트웨어, 컴퓨터 보안