IPv4/IPv6 터널링 환경에 적합한 패킷 필터링 기능 설계 및 구현

(Design and Implementation of Packet Filtering System for IPv4/IPv6 Tunneling Environment)

허 석 렬 † 이 완 직 †† 김 경 준 ††† 정 상 진 ††††
(Seok-Yeol Heo) (Wan-Jik Lee) (Kyung-Jun Kim) (Sang-Jin Jeong)

신 명 기 †††† 김 형 준 †††† 한 기 준 †††††
(Myung-Ki Shin) (Hyoung-Jun Kim) (Ki-Jun Han)

요 약 현재의 모든 IPv4망을 향후 단기간 내에 IPv6망으로 대체하는 것은 높은 비용과 기술적인 제약이 예상되며 이런 점 때문에 상당한 기간 동안 IPv4와 IPv6가 공존하게 될 것이다. IPv4와 IPv6가 상호 공존하는 환경에서는 각각의 프로토콜을 기반으로 하는 보안에 문제가 없더라도 연동에 따른 보안 문제가 새롭게 발생한다. 따라서 IPv6로의 효과적인 이전과 정착을 위해서는 IPv4/IPv6 연동과정에서 발생하는 다양한 보안 위협에 대한 분석과 이에 대한 해결 방안이 반드시 필요하다. 본 논문에서는 IPv4/IPv6 연동 환경에서 발생할 수 있는 보안 위협요소를 막기 위해 터널링 연동환경에 적합한 패킷 필터링 규칙을 제시하였다. 또한, 제시된 패킷 필터링 규칙을 기반으로 리눅스 시스템의 넷필터(netfilter)와 ip6tables 형태로 터널링 환경에 적합한 패킷 필터링 기능을 설계·구현하였다. 그리고 시험용 테스트베드 터널링 연동 환경에서 정상적으로 동작하는 것을 확인하였다.

키워드 : 보안, IPv6, 패킷 필터링, 터널링, 변환

Abstract As substituting IPv6 network for all IPv4 network in a short time seems unattainable due to high cost and technical limitation, IPv4 and IPv6 are expected to coexist for a certain period of time. Under the coexisting environment of IPv4 and IPv6, interworking brings a number of extra security considerations even if it may have no security problem for each protocol respectively. Thus, the analysis and solutions for those various attacks toward IPv4/IPv6 interworking-related security are inevitably required for the sake of effective transition and settlement to IPv6.

In this paper, we carried out a proper rule of packet filtering for IPv6-in-IPv4 tunneling interworking environment to protect the IPv4/IPv6 interworking-related security attacks. Design and implementation of the packet filtering system suitable for IPv4/IPv6 tunneling environment in the form of Linux netfilter and ip6tables are also shown. Thru this study, the packet filtering system was found operating correctly in the tunneling mechanism.

Key words : Security, IPv6, Packet filtering, Tunneling, Transition

· 본 연구는 한국전자통신연구원 위탁연구(1010-2005-0099) 지원으로 수행되었음

† 종신회원 : 부산대학교 바이오시스템공학부 교수
syheo@pusan.ac.kr
†† 비 회 원 : 부산대학교 바이오시스템공학부 교수
wjlee@pusan.ac.kr
††† 비 회 원 : 호남대학교 전파이동통신공학과 교수
kjkim@honam.ac.kr
†††† 비 회 원 : 한국전자통신연구원 표준연구센터 연구원
sjjeong@etri.re.kr
mkshin@etri.re.kr
khj@etri.re.kr
††††† 종신회원 : 경북대학교 컴퓨터공학과 교수
kjhan@bh.knu.ac.kr

논문접수 : 2006년 5월 3일
심사완료 : 2006년 10월 31일

1. 서 론

차세대 인터넷 표준인 IPv6는 IPv4의 주소고갈 문제를 포함하여 라우팅, 보안, 이동성 지원 QoS보장 등과 같은 여러 가지 이슈에 대한 훌륭한 해법들을 제시하고 있다. IPv6는 빠른 속도로 진화하고 있는 인터넷 환경에서 궁극적으로는 IPv4를 대체할 것으로 예상되지만, 모든 IPv4 망을 짧은 시간에 IPv6 망으로 대체하는 것은 어렵기 때문에 IPv4와 IPv6 망이 상당한 기간 동안 공존할 수 있다. 이렇게 IPv4와 IPv6가 상호 공존하는 망 환경에서는 각각의 프로토콜을 기반으로 하는 보안에 문제가 없더라도 연동에 따른 보안 문제가 새롭게 발생할 수 있다[1-3].

IPv4와 IPv6는 각각 IPsec 보안 프로토콜을 사용하여 네트워크 계층에서 메시지들을 보호하지만, IPv4/IPv6 연동환경에서는 연동 지점에서 주소나 세션의 변환이 일어난다. 변환기법에서는 IPv4/IPv6의 헤더가 연동부분을 통과할 때 IPv4의 패킷헤더가 IPv6 헤더로 변환되거나 혹은 역으로 IPv6의 패킷 헤더가 IPv4의 헤더로 변환된다. 터널링 기법에서는 연동부분에서 "[IPv4][IPv6]" 패킷헤더 혹은 "[IPv6][IPv4]"패킷헤더의 형태로 기존의 헤더에 통과지점의 헤더가 각각의 헤더에 추가 된다. 이러한 변환 과정 때문에 IPsec만으로는 모든 보안 문제를 해결할 수 없으며, 변환 과정 상에서 DoS(Denial of Service)나 MITM(Man-in-the middle) 공격 등 다수의 다양한 보안 취약점이 발생할 수 있다.

지금까지 IPv4 프로토콜에 대한 보안은 많은 연구와 보완이 이루어져 왔으며 IPv6에 대해서도 많은 연구가 진행 중에 있다. 하지만 IPv6는 IPv4에 비해 프로토콜 자체의 여러 확장 헤더와 ICMPv6의 확장된 기능으로 인한 고려 요소가 상당히 많기 때문에 아직까지 연구 수준은 초보적인 단계에 머물러 있으며, 더욱이 IPv4/IPv6 연동환경에서 보안에 대한 연구는 아주 미미한 수준이다. 따라서 IPv6로의 효과적인 이전과 정착을 위해서는 IPv4/IPv6 연동과정에서 발생하는 다양한 보안 위협에 대한 분석과 이에 대한 해결 방안이 반드시 필요하다.

본 논문에서는 IPv4/IPv6 터널링 연동환경에서 발생할 수 있는 다양한 보안 문제점을 분석하고, 터널링 기법에서 발생할 수 있는 문제들을 차단하는 새로운 패킷 필터링 규칙을 설계하였다. 그리고 새로운 필터링 규칙을 기반으로 설계된 패킷 필터링 기능을 리눅스의 넷필터(netfilter)와 ip6tables 형태로 구현하였다.

본 논문의 2장에서는 대표적인 터널링 연동기법을 기술하고 Pv4/IPv6 연동환경의 보안 관련연구를 살펴본다. 3장과 4장에 걸쳐 터널링 환경에 적합한 패킷 필터링 기능 설계와 구현에 대한 내용을 기술한다. 마지막으로 5장에서 결론과 차후 연구과제에 대해 언급하였다.

2. IPv4/IPv6 터널링 연동에 따른 보안 문제점 및 관련연구

2.1 IPv4/IPv6 터널링 연동기술

앞으로 구축될 IPv6 망들은 IPv4/IPv6 듀얼(dual) 망 또는 IPv6 전용(native) 망 형태로 구성될 것이며, IPv4와 IPv6가 상호 공존하는 망 상에서 두 망간의 통신을 자연스럽게 하기 위해서는 IPv4/IPv6 연동 기술이 필요하다. 이러한 연동기술은 크게 변환기법과 터널링 기법으로 나눌 수 있다[4].

변환기법은 일반적으로 게이트웨이 상에서 계층에 따른 변환을 수행하는 기법으로서 헤더 변환, 주소 매핑, 프로토콜 변환과 같은 동작을 수행한다. 터널링 기법은 기존의 IPv4 인프라를 활용하여 IPv6 트래픽을 전송하는 방법을 제공하는 기법이다. 위의 두 가지 연동 기법 중에서 일반적으로 많이 사용되는 것은 터널링 기법이며, 변환 기법은 IPv6 전용노드와 IPv4 전용 노드간의 통신과 같이 터널링 기법을 적용 할 수 없는 상황에서 사용된다. 터널링 기법은 다시 설정 터널링(configured tunneling) 방식과 자동 터널링(automatic tunneling) 방식으로 구분된다. IPv6 도입 초기에는 설정 터널링 방식이 더 많이 사용되겠지만 IPv6 망이 점진적으로 확산되면 자동 터널링 방식이 대부분 사용될 것이다.

설정 터널링

듀얼스택 노드는 IPv6 데이타그램을 IPv4 패킷에 캡슐화하여 IPv4 라우팅 영역을 터널링 할 수 있다. IPv6-in-IPv4 터널링은 IPv4 라우팅 인프라를 활용하여 IPv6 트래픽을 전송하는 방법은 제공하는데, 설정 터널링 또는 자동 터널링 방법으로 동작된다. 설정 터널링 방법은 6Bone에서 주로 사용하는 것으로써 두 노드의 IPv4 주소를 이용해서 수동으로 터널을 설정하는 방법이다. 설정 터널링에서 터널의 종단 주소는 설정 정보에 의해서 결정된다. 설정 터널링 방법은 관리자에 의해 수동으로 운영되기 때문에 실험 망이나 소규모 망에서는 큰 문제가 없지만 대규모 망에서는 적용하기 어렵다.

자동터널링

자동 터널링 방법은 사용자나 관리자의 별도 설정 없이 IPv4 구간을 통과할 때 IPv6 주소에 포함되어 있는 IPv4 주소를 이용해서 자동으로 터널링 하는 방식이다. 지금까지는 IPv4 호환(IPv4-compatible) 주소를 이용하는 자동 터널링 방식을 많이 사용하였지만 앞으로는 6to4[5], ISATAP[6]과 같은 향상된 자동 터널링 방식을 많이 사용할 것이다.

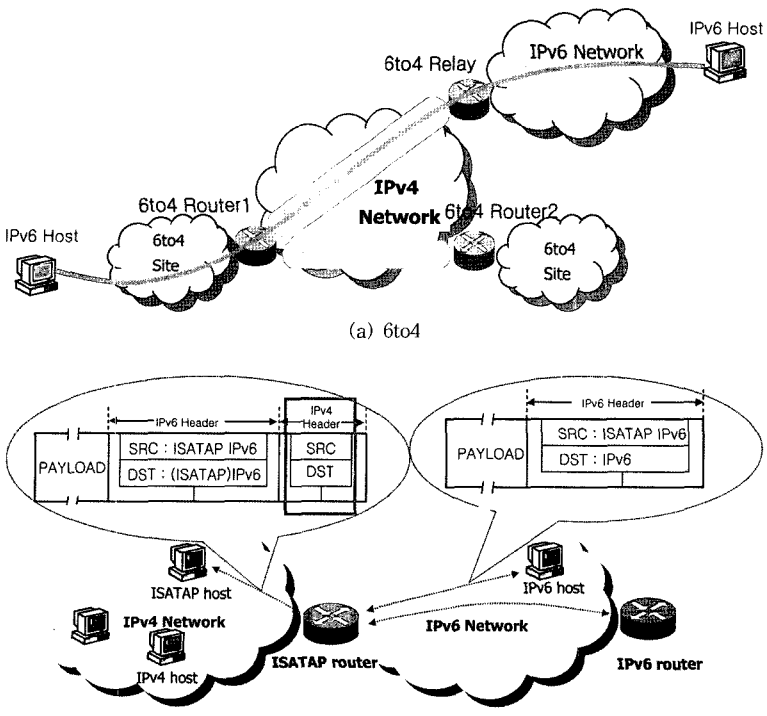
6to4는 IPv6를 지원하지 않는 광역 IPv4 네트워크에 연결되어 있는 고립된(isolated) IPv6 사이트나 호스트가 자동 터널링을 통하여 다른 IPv6 도메인의 호스트 또는 6to4 사이트의 호스트와 통신하도록 하는 연동 기법이다. 6to4 사이트에는 적어도 하나 이상의 광역 IPv4 주소가 필요하며, 이 주소를 이용하여 6to4 사이트에 사용할 프리픽스 "2002:IPv4::/48"을 생성한다. 6to4 라우터는 6to4 사이트와 광역 IPv4 망 사이의 경계 라우터로서 터널링을 통해 다른 라우터나 릴레이로 연동을 하고 6to4 릴레이 라우터는 6to4 사이트들과 순수 IPv6 사이트를 연동한다.

ISATAP은 IPv4 기반의 인트라넷에서 고립된 IPv6 노드가 외부 IPv6 망과 연결하는 방법을 제공한다. 듀얼스택을 지원하는 ISATAP 노드 IPv4망을 IPv6를 위한 링크계층으로 간주하고 IPv4망을 통과하는 IPv6 패킷을 자동 터널링 해서 IPv6망에 전달한다. ISATAP은 64비트 EUI-64 인터페이스 식별자와 표준 64비트 IPv6 주소 프리픽스를 포함하는 통합 가능 글로벌 유니캐스트 주소 형식을 기반으로 한다. IPv6 망과 직접 연결되지 않은 듀얼스택 노드는 사이트 내에서 IPv4 라우팅 인프라를 통해 IPv6 메시지를 자동 터널링 함으로써 광역 IPv6 네트워크에 연결할 수 있다. 그림 1은 대표적인 자동터널링 기법인 6to4와 ISATAP의 연동 망 환경이다.

2.2 IPv4/IPv6 터널링 연동에 따른 보안 문제점

IPv4 망과 IPv6 망이 상호 공존하는 형태가 복잡해질수록 IPv4/IPv6 연동으로 인한 새로운 보안 문제가 발생한다. IPv4/IPv6 터널링 연동 방법은 다양한 형태의 IP-in-IP 터널링 방법을 이용하는데 터널링 연동 환경에서 운영되는 방화벽이 IPv4와 IPv6 두 가지 프로토콜을 동시에 검사하는 기능을 갖지 못할 경우, 터널링된 패킷에 대해서 적절한 보안 정책을 수립할 수 없다. IPv4/IPv6 터널링 연동 환경에서는 다음과 같은 보안 문제가 발생할 수 있다[1,7,12].

1) 수동 설정된 터널링에서 IPsec을 이용하는 방법은 제안되었으나, 자동 터널링 기법은 터널 종단 간에 사전 설정된 연계가 없기 때문에 IPsec을 이용할 수 없다. 결과적으로 자동 터널링에서 패킷을 수신하는 터널 종단에서는 신뢰할 수 없는 임의의 송신지에서 온 모든 패킷을 수신하고 역 캡슐화를 수행한다.
2) 터널링을 통해 인그레스 필터링과 같이 네트워크 토폴로지를 기반으로 하는 보안 기법을 피할 수 있다.
3) 터널링 기반 연동 기법에서는 IPv4 주소가 IPv6 주소 내에 포함되는 경우(6to4, ISATAP, Teredo)가



(a) 6to4

(b) ISATAP

그림 1 IPv4/IPv6 자동터널링 연동 망 환경

존재할 수 있으며, 이런 경우 내포된 IP 주소가 실제로 유효한 주소인가에 대한 검사를 수행해야 한다. 예를 들어 6to4에서 목적지 주소가 2002:203.232.244.255::0a00:0002인 경우 내포된 IPv4 주소는 브로드캐스트 IP 주소인데, 이를 방화벽에서 처리하지 않으면 IPv4망에서 브로드캐스트 전송이 발생된다.

4) IPv6-in-IPv4 터널링에서 외부 패킷 헤더(IPv4 헤더)의 IPv4주소 또는 내부 패킷 헤더(IPv6 헤더)의 주소가 스푸핑이나 DoS 공격에 이용될 수 있다. 따라서 방화벽에서 터널링을 허용하는 경우에는 패킷의 외부 헤더뿐만 아니라 내부에 포함된 패킷에 대해서도 검사를 수행해야 한다.
5) 자동 터널링 연동 환경에서 정당하지 않은 터널링 주소를 이용해서 터널링을 생성한 뒤 이를 이용해서 악의적인 공격을 할 수 있기 때문에 터널링 주소가 올바른지에 대한 검사가 이루어져야 한다.

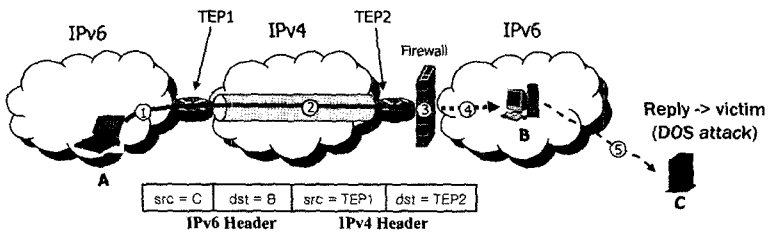
위에서 설명한 연동환경의 보안 문제점 중에서 1)번의 경우에는 자동터널링을 사용하는 환경에서 IPsec을 적용하기 힘든 문제이기 때문에 방화벽으로는 해결할 수 없지만 나머지는 터널링 종단 지점의 방화벽에서 IPv4와 IPv6, 두 가지 프로토콜을 동시에 검사하는 패킷 필터링 기능만으로 해결이 가능하다. 그림 2는 터널링 환경에서 발생할 수 있는 송신 주소 스푸핑 공격을 보여 주고 있다. 그림에서는 스푸핑 공격만을 보여 주고 있지만 터널 종단 지점의 방화벽이 터널링 된 패킷의 안쪽 헤더에 대한 검사 기능을 갖고 있지 못하다면, 고의 또는 실수에 의한 잘못된 패킷으로 인해 문제가 발생할 수 있다.

2.3 관련연구

IPv6가 도입되는 과정에서 기존의 IPv4 환경과 다른 새로운 보안 문제가 발생할 수 있다. 이러한 문제점들을 [1]에서는 IPv6 프로토콜 자체의 문제점에 기인한 것과 IPv4/IPv6 연동 환경에서 발생할 수 있는 보안 취약점, 그리고 IPv6를 도입하는 과정에서 고려해야 사항으로 구분하여 설명하고 있다. [8]에서는 라우팅 헤더와 홈 주소 옵션의 사용으로 인해 발생할 수 있는 필터링 회피 문제와 스푸핑 문제를 다루고 있으며 [9]에서는 icmpv6의 여러 가지 메시지들에 대한 보안 취약점을 분석하고 있다. 연동환경에 대해서는 [10]에서 6to4와 관련된 보안 문제들을 다루고 있으며 [11]은 NAT-PT를 도입할 경우 발생할 수 있는 보안 취약점을 분석하였다.

IPv6는 IPv4에서 부가적 기능(add-on)이었던 IPsec을 필수 기능으로 포함시킴으로써 IPv4보다 훨씬 높은 수준의 보안 환경을 제공하지만, 모든 IPv6 트래픽에 대해 IPsec 보안 기법을 적용할 수 없다. 특히, 변환 기법이 적용되는 환경에서는 종단간 IPsec 적용이 불가능하다. [12]에서는 터널링 환경에서 IPsec을 사용하는 방법을 제시하고 있으나 적용 환경을 자동터널링이 아닌 설정터널링 환경을 대상으로 하고 있다.

위에서 언급한 여러 가지 형태의 보안 위협 요소들 중에서 icmpv6와 IPv4/IPv6 터널링 연동과 관련된 위협 요소들은 방화벽을 이용한 패킷 필터링만으로 비교적 간단하게 해결할 수 있지만, IPv6 확장 헤더와 관련한 DoS나 스푸핑 문제와 같은 위협 요소들은 IDS나 IPS를 통해서만 해결이 가능하다. IPv6를 지원하는 방화벽 현황은 다음과 같다. 리눅스에서는 커널 2.6.12에서부터 IPv6를 지원하는 방화벽을 탑재하였으며 FreeBSD 6.1에서 IPv6를 지원하는 IPFW2를 제공하고 있다[13]. 외국에서는 Cisco, Nokia 등에서 2005년 하반기부터 IPv6를 지원하는 방화벽을 출시하고 있으며 NTT에서는 2006년부터 IPv6 방화벽 서비스를 제공하고 있다. 국내에서는 시큐어닷컴과 퓨쳐시스템에서 2006년 상반기에 IPv6를 지원하는 방화벽을 출시하였다. 그러나 아직까지 IPv6를 지원하는 방화벽들은 기본헤더 또는 확장헤더 정도만 지원하고 있으며 터널링 연동과 같은 환경을 고려한 방화벽은 아직까지 없다.



① 송신지를 C로 가장한 패킷을 B에게 전송
② IPv6-in-IPv4 터널링
③ 방화벽에서는 외부 IPv4헤더에 대해서만 필터링
④ IPv6 패킷을 B에게 전송
⑤ B는 C에게 응답 패킷을 보냄

그림 2 터널링 환경에서 스푸핑 공격 예

3. IPv4/IPv6 터널링 연동 환경의 패킷 필터링 기능 설계

IPv6 도입을 위해 개발된 다양한 IPv4/IPv6 연동 기술을 실제 네트워크에 도입하는 경우에는 IPv6 프로토콜 자체의 보안 문제 이외에 연동으로 인한 새로운 보안 문제가 발생한다. 특히 터널링 기법을 이용한 연동 기술에는 IPv4, IPv6 두 개의 헤더를 가진 패킷이 존재하므로 이에 대한 처리를 수행할 수 있도록 패킷 필터링 규칙을 설계해야 한다.

다음의 그림 3은 터널링 형태의 연동 기법을 사용하는 대표적인 방화벽 환경을 나타낸 것이다. 그림에서 보는 바와 같이 방화벽을 포함한 내부 망은 IPv6로 운영되고, 외부의 IPv4 망과 연동을 위해 IPv6-in-IPv4 터널링을 사용한다. 방화벽 시스템은 사이트의 에지(edge) 라우터에 적용하고, 내부로 유입되는 패킷의 필터링(ingress filtering)만 고려한다.

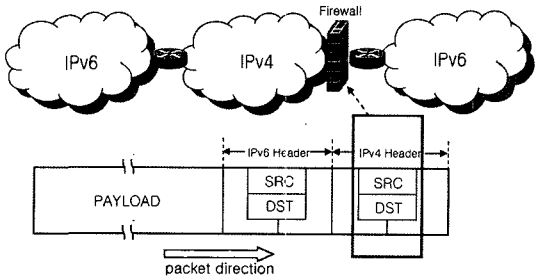


그림 3 방화벽 기능을 갖는 터널링 연동 망 구조

3.1 IPv4/IPv6 터널링 연동 기술의 패킷 필터링 규칙 설계[14]

이중 헤더 처리

IPv4 헤더는 IPv4 필터링 규칙에 의해 처리되고, IPv6 헤더는 IPv6 필터링 규칙에 의해 처리되도록 필터링 규칙을 설계해야 한다. 현재 사용되는 대부분의 방화벽은 외부 헤더(IPv4 헤더)만 필터링 하도록 구현되어 있다. 이 경우에는 내부의 악의적인 IPv6 패킷이 방화벽을 통과할 수 있으므로 두 가지 헤더에 대한 필터링이 모두 적용되도록 필터링 규칙을 설계 한다.

외부 IPv4 터널링 헤더의 필터링 규칙

IPv6 패킷이 포함된 패킷(Protocol 필드 41)이면서 목적지 IPv4 주소가 사이트의 에지 라우터 터널링 주소(또는 사이트에서 허가된 내부의 IPv4 터널링 주소)가 아닌 경우에는 패킷을 폐기한다. 그 외 IPv4 헤더의 각 필드에 대해서는 현재 적용하고 있는 방화벽의 IPv4 필터링 규칙을 그대로 적용한다.

내부 IPv6 패킷의 필터링 규칙

방화벽에서 설정된 IPv6 필터링 규칙을 적용한다. IPv6 필터링 기법에 대해서는 현재 많이 연구되고 있는 상황이며, 본 논문에서는 [9]를 참조하여 다음과 같은 IPv6 필터링 규칙을 설계하였다. 내부 IPv6 패킷에 대한 필터링 규칙은 표 1과 같다.

다음의 그림 4는 IPv6 헤더 관련 패킷 필터링 규칙을 순서도 형태로 표현한 것이다.

3.2 6to4 및 ISATAP 연동 기법의 패킷 필터링 기본 규칙 설계

6to4

6to4 역시 터널링을 이용한 연동 기술이므로 3.1절에 기술한 패킷 필터링 규칙이 동일하게 적용되어야 한다. 하지만 6to4 패킷의 IPv6 주소 내에는 IPv4 주소가 포함되며, 여기에 포함된 IPv4 주소는 터널링을 위해 사용되는 외부 IPv4 헤더의 IPv4 주소와 동일해야 한다. 공격자는 6to4 패킷을 가장하여 특정 IPv4 주소를 가진 호스트에 대해 DoS 공격을 할 수도 있다. 따라서 2002::/16으로 시작되는 6to4 주소를 가진 IPv6 패킷들에 대해서 적절한 필터링을 할 수 있도록 설계하였다.

ISATAP

ISATAP을 사용하는 패킷은 IPv4 망의 ISATAP 라우터를 통과하면 IPv6 패킷이 전송되므로, IPv6 망의 방화벽에서는 IPv6 헤더에 대한 검사만 수행하면 된다. 다만, ISATAP 주소 내부에 IPv4 주소가 포함되어 있으므로, 포함된 IPv4 주소에 대한 검사를 수행할 필요가 있다. 따라서, ISATAP의 필터링 규칙은 IPv4 주소에 대한 필터링 적용 후, IPv6 헤더에 대한 필터링을 적용하도록 한다. ISATAP 주소 내에 포함된 IPv4 주소에 대한 인그레스 필터링 규칙은 [15]를 참조하여 주소의 적절성을 검사할 수 있도록 설계하였다.

다음의 그림 5는 6to4와 ISATAP을 위한 패킷 필터링 규칙을 순서도 형태로 표현한 것이다.

4. IPv4/IPv6 터널링 연동 환경에 적합한 패킷 필터링 기능 구현

4.1 구현 시스템 환경

구현 및 테스트에 사용된 시스템들은 리눅스 운영체제를 탑재한 PC들로 구성하였고 패킷 필터링 기능은 리눅스의 iptables[16] 명령어와 커널 내부의 넷필터(Netfilter)[17,18] 코드로 구현되었다. 그림 6에 구현에 사용한 IPv4/IPv6 연동 환경 및 방화벽 시스템 구성을 나타내었다. 그림에 나타낸 3개의 라우터들은 2개 이상의 네트워크 카드를 가진 PC로서 모두 리눅스 운영체제를 설치하였고 커널은 2.4.22로 구성하였다.

그림 6의 IPv4/IPv6 연동은 기본적인 설정 터널링(IPv6-in-IPv4) 환경이다. Router1과 Router3은 IPv4/

표 1 IPv6 내부 패킷에 대한 필터링 규칙

구분	필터링 조건		동작
IPv6 주소	IPv6 송신지주소	::	폐기
		멀티캐스트	폐기
		애니캐스트	폐기
		사이트로컬 유니캐스트	폐기
		링크로컬 유니캐스트	허용
		글로벌 IPv6주소	허용
		6to4	허용
		6bone	허용
		그 외 다른 송신지주소	폐기
	IPv6 목적지주소	::	폐기
		링크로컬	허용
		멀티캐스트	허용
		글로벌 IPv6	허용
		6to4	6to4 규칙 적용
		6bone	허용
		그 외 다른 목적지주소	폐기
IPv6 확장헤더	라우팅헤더 존재		헤더내 주소를 이용한 필터링
	단편 헤더	목적지 주소가 방화벽이나 라우터주소	폐기
		마지막패킷을 제외한 1280 바이트 미만 패킷	폐기
	그외 다른 확장 헤더 및 미확인 확장 헤더		폐기
ICMPv6	Packet Too Big 메시지 (Type 2)		허용
	Parameter Problem 메시지 (Type 4)		허용
	Echo Request/Reply 메시지 (Type 128, 129)		허용
	RS/RA 메시지 (Type 133, 134)		허용
	NS/NA 메시지 (Type 135, 136)		허용
	이외의 다른 메시지 및 미확인 메시지 패킷		폐기

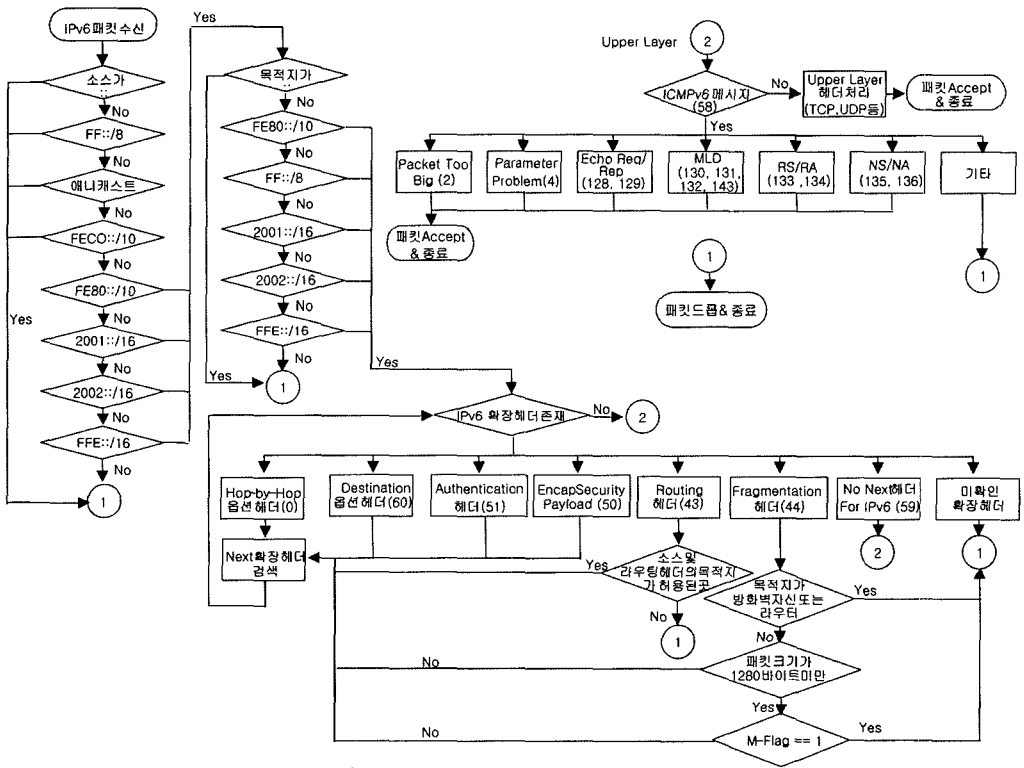


그림 4 IPv6 헤더 관련 패킷 필터링 규칙

표 2 6to4와 ISATAP 필터링 규칙

구분	필터링 조건		동작
6to4	IPv4 송신지 주소	0.0.0.0/8	폐기
		루프백 주소	폐기
		링크로컬 주소	폐기
		192.0.0.0/24, 192.18.0.0/15	폐기
		사설주소	폐기
		멀티캐스트주소	폐기
		240.0.0.0/4	폐기
	ICMP	Echo Request/Reply	허용
	6to4 주소 매칭	6to4 패킷의 소스 주소에 포함된 IPv4 패킷의 주소와 수신한 IPv4 패킷의 소스 주소가 다름	폐기
		6to4 패킷의 목적지 주소에 포함된 IPv4 패킷의 주소와 수신한 IPv4 패킷의 목적지 주소가 다름	폐기
ISATAP	IPv4 송신지 주소	0.0.0.0/8	폐기
		루프백 주소	폐기
		링크로컬 주소	폐기
		192.0.0.0/24, 192.18.0.0/15	폐기
		사설주소	폐기
		멀티캐스트주소	폐기
		240.0.0.0/4	폐기
	IPv6 주소	위의 규칙들을 통과한 ISATAP 패킷	IPv6패킷 필터링검사

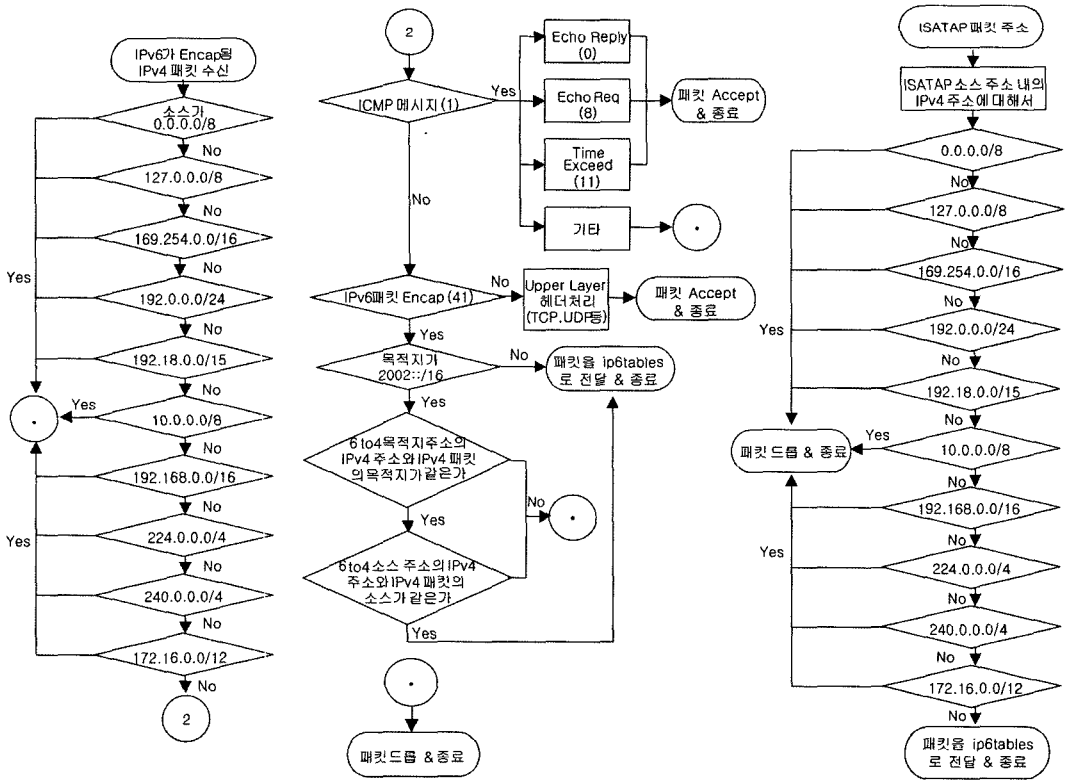


그림 5 IPv6 연동 기술 적용을 위한 필터링 규칙

IPv6 듀얼 스택을 가지며, 각 IPv6 사이트의 에지 라우터가 되고, Router2는 일반적인 IPv4 라우터로 설정하였다. 그림 6에서 네모 상자 안에 표시된 내용은 각 라우터의 설정 명령어를 나타낸다. Router1에는 sit30 이라는 TEP(Tunnel End Point) 인터페이스를 생성하였고, TEP 주소로 3ffe:106:9000::30/64를 주고, Router3

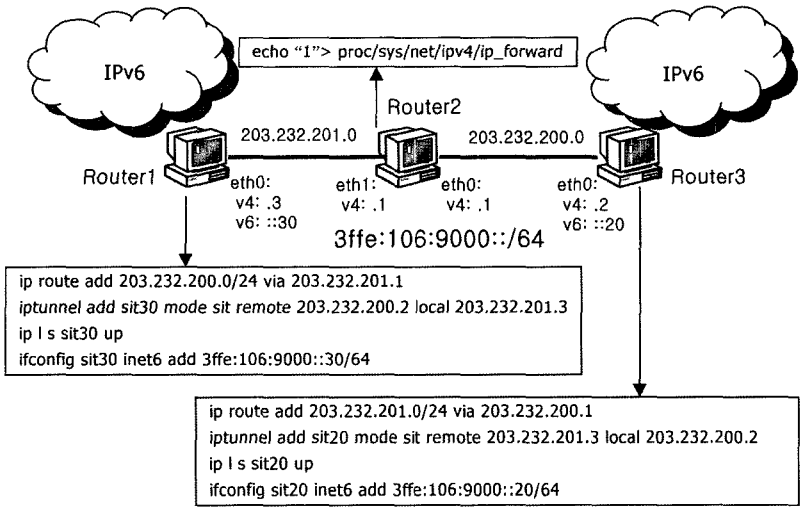


그림 6 IPv4/IPv6 연동 환경의 방화벽 시스템 구성

에는 sit20이라는 TEP 인터페이스를 생성하고, TEP 주소 3ffe:106:9000::20/64 주소를 설정하였다.

위와 같이 IPv6-in-IPv4 터널을 생성한 후, ping6로 Router1과 Router3을 연결을 테스트하고, Router3에 iptables 명령어와 ip6tables 명령어를 사용하여 여러 가지 패킷 필터링 기능을 수행하였다. 본 논문에서 설계한 패킷 필터링 규칙들은 기존의 리눅스 방화벽 기능(ip(6)tables, 넷필터)에 의해 지원되는 것도 있고, 추가로 구현이 필요한 규칙들도 있다. 표 3은 3장에서 설계한 필터링 규칙에 대한 현재 리눅스의 패킷 필터링 기능의 지원 여부를 나타낸 것이다.

표 3에서 보는 바와 같이 현재 리눅스에서 사용하고 있는 ip(6)tables 기능은 IPv4, IPv6의 2중 헤더를 가지는 일반적인 터널링 패킷에 대해서 올바르게 동작함을 알 수 있으며, IPv6의 기본 헤더 및 확장 헤더의 여러 필드에 대해서도 비교적 충실한 필터링 기능이 제공되고 있음을 알 수 있다. 하지만 6to4, ISATAP 등의 자동 터널링 기법에 필요한 터널링 규칙들은 제대로 지원하지 못하기 때문에 이러한 문제점을 이용한 악의적인 공격에는 보안 상의 취약점을 가질 수 있다.

4.2 6to4 패킷 필터링 기능 구현 및 테스트

3장에서 설명한 바와 같이 6to4 패킷의 경우에는 6to4 IPv6 주소의 v4 주소 영역과 외부 IPv4의 주소는 항상 일치해야 한다. 그러므로 이들 주소들이 서로 일치하는 가를 검사하고 일치하지 않는 6to4 패킷은 필터링하여 폐기할 수 있는 기능이 구현되어야 한다. 이러한 구현을 위해서는 IPv4 헤더와 IPv6 헤더를 동시에 보아야 하므로 IPv6 넷필터 부분에서는 처리할 수 없고(이미 IPv4 헤더를 처리되어 제거된 후이기 때문에), IPv4 넷필터 영역에서 패킷을 캡쳐 한 뒤, IPv4 관점에서 데이타 영역에 있는 IPv6 헤더를 검출하여 처리해야 한다. 아래의 표 4에 6to4 관련 필드를 매칭할 수 있도록 설계한 iptables 명령어를 나타내었다. 표 4에 나타난 것과 같이 기존의 iptables 명령어에 -m 옵션을 지정한 형태의 iptables 확장 모드 방식으로 6to4 패킷 매칭을 설계하였다.

표 3 적용한 IPv6 방화벽 필터링 규칙 요약

필터링규칙	세 부 항 목	구현여부
IPv6 주소에 대한 필터링 규칙	IPv6 송신지주소 적합성 검사	구현됨
	IPv6 목적지주소 적합성 검사	구현됨
IPv6 확장헤더 관련 필터링 규칙	라우팅 헤더 검사	구현됨
	단편 헤더 검사	구현됨
ICMPv6 관련 필터링 규칙	필요한 ICMPv6 메시지에 대한 검사	구현됨
설정 터널링 및 6to4 관련 필터링 규칙	IPv4 소스 주소 적합성 검사	구현됨
	ICMPv4 검사	구현됨
	6to4 주소 매칭 검사	구현필요
ISATAP 관련 필터링 규칙	IPv4 송신지 주소 검사	구현필요

표 4 6to4 관련 iptables 명령어

가능옵션	예제 및 설명
--srcmatch	**iptables -A INPUT -m m6to4 --srcmatch -j ACCEPT** 6to4 패킷이며, 캡슐화되는 IPv4 송신자 주소와 6to4 IPv6 송신자 주소의 IPv4 주소 값과 동일하면 매칭된다.
--dstmatch	**iptables -A INPUT -m m6to4 --dstmatch -j ACCEPT** 6to4 패킷이며, 캡슐화되는 IPv4 수신자 주소와 6to4 IPv6 수신자 주소의 IPv4 주소 값과 동일하면 매칭된다.
--srcnomatch	**iptables -A INPUT -m m6to4 --srcnomatch -j DROP** 6to4 패킷이며, 캡슐화되는 IPv4 송신자 주소와 6to4 IPv6 송신자 주소의 IPv4 주소 값과 동일하지 않으면 매칭된다.
--dstnotmatch	**iptables -A INPUT -m m6to4 --dstnomatch -j DROP** 6to4 패킷이며, 캡슐화되는 IPv4 수신자 주소와 6to4 IPv6 수신자 주소의 IPv4 주소 값과 동일하지 않으면 매칭된다.

리눅스의 넷필터 프레임워크는 필터링 기능 확장을 위해 넷필터 확장 모드를 지원한다. 이 확장 모드는 개발자가 새로운 패킷 매칭이나 타겟(Target 매칭된 패킷들의 인가, 폐기 등의 처리)을 기존의 iptables 명령어에 쉽게 추가하거나 변경할 수 있다. 이러한 확장 모드의 구현은 크게 사용자 도구인 iptables의 코드 부분과 실제 패킷을 매칭하는 넷필터의 커널 코드로 구성된다. iptables의 코드 부분은 iptables의 -m 옵션 뒤의 이름을 가진 공유 라이브러리 형식으로 시스템에 추가되며, 이 코드는 iptables에서 정의한 형식의 함수들로 구성되어 각 옵션들의 지정 여부와 사용된 문법의 적합성을 검사하게 된다.

확장 모드의 넷필터의 커널 코드는 독립적인 커널 모듈로 구현되며 상위 iptables의 확장 코드 부분에서 지정한 옵션에 따라 실제 패킷을 매칭하는 코드로 구성된다. 본 논문의 6to4 패킷 매칭 기능도 iptables의 확장 모드 형태로 구현되었다. 구현된 6to4 패킷 필터링 기능을 확인하기 위해 그림 7과 같이 테스트 환경을 구성하였다.

그림 7의 테스트 환경은 4.1절의 그림 6과 동일하지만 TEP 주소를 "2002"로 시작하는 6to4 주소를 수동으로 설정하였다. 그림 7의 테스트 환경에서 수동 IPv4 터널을 설정한 후 IPv6 TEP 주소만 6to4 주소로 설정하였지만, 실제 6to4 연동환경도 자동 터널링 설정한다는 것 이외에 그림과 동일하다. 먼저 테스트 시스템인 HA에서 본 논문에서 설정한 6to4 옵션이 동작하는 지를 확인하기 위해 그림 8과 같이 명령어를 입력하였다.

그림 8의 ①과 같이 6to4 IPv6 패킷의 송신자 IPv4주소 부분과 터널링 하는 IPv4 주소가 일치하면 ACCPET 하라는 명령어를 입력한 후, lsmod 명령어를 입력하여

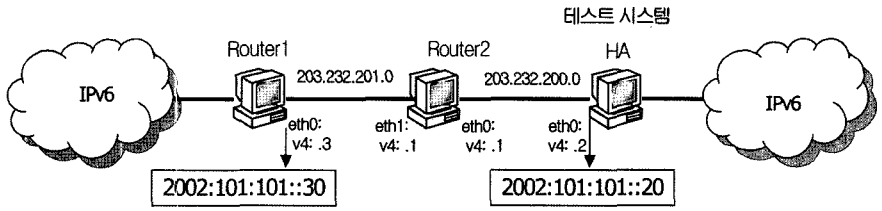


그림 7 6to4 패킷 필터링 동작 확인을 위한 테스트 환경

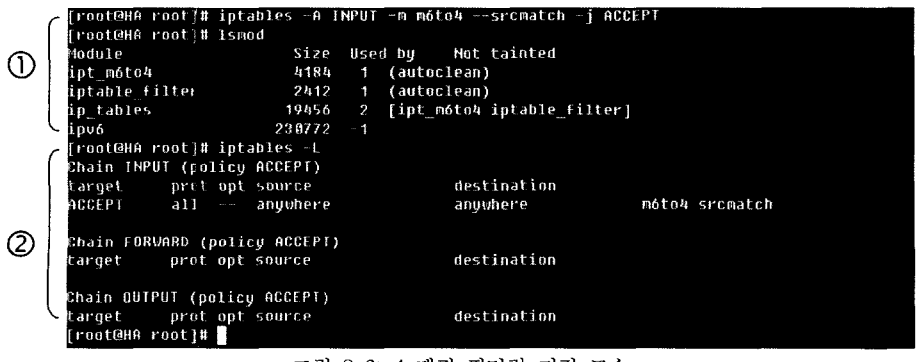


그림 8 6to4 패킷 필터링 지정 모습

6to4 커널 모듈(ipt_m6to4)이 제대로 등록이 되어있는지를 확인하였다. 그림 8의 ②는 iptables list 명령어를 사용하여 6to4 패킷 필터링 규칙이 제대로 등록된 것을 보여준다.

그림 9는 6to4 패킷 필터링에 의해 ping6(ICMPv6) 패킷이 폐기되는 예를 보여준다. ①에서는 그림 7의 HA 시스템이 Router1과 ping6로 연결이 되었음을 보여준다. 하지만 ②에서 6to4 IPv6 패킷의 목적지 IPv4주소 부분과 목적지 IPv4 주소가 일치하지 않으면 DROP 하라는 명령어를 입력한 후, 다시 ping6을 수행하면 ping6의 reply 패킷이 폐기되어 연결이 되지 않는 것을 보여준다. 이는 목적지인 HA의 eth0의 6to4 주소가 2002:101:101::20으로 지정되어 2002 다음의 IPv4 목적지 주소 부분 1.1.1.1 (v4 표기법)이 실제 IPv4 주소 203.232.200.2와 일치하지 않기 때문에 6to4 패킷 필터링에 의해 폐기되기 때문이다. ③에서 6to4 dstnotmatch 옵션을 제거한 후에는 Router1과 다시 정상적으로 연결되는 것을 볼 수 있다.

4.3 ISATAP 패킷 필터링 기능 구현 및 테스트

3장에서 설명한 바와 같이, 방화벽이 ISATAP 주소 내에 포함된 IPv4의 주소 영역에 대해 일반적인 IPv4 주소에 대한 필터링을 동일하게 적용할 수 있도록 해야 한다. 이를 위해 ISATAP에서 적용할 수 있는 IPv4 주소 영역에 대한 필터링 문법은 다음 표 5와 같이 설계하였다.

구현 내용 및 방법은 앞의 3.2절의 6to4와 동일하게 넷필터 확장 모드 형식을 따른다. 6to4의 경우에는 IPv4와 IPv6 주소를 함께 보아야 하기 때문에 IPv4 영역에서 동작하였지만 ISATAP 패킷 필터링에서는 IPv6 주소만 매칭하면 되므로 실제 구현 코드들은 IPv6 영역 즉, ip6tables의 확장 라이브러리와 ipv6 넷필터 커널 모듈, 헤더 파일로 구성된다. 구현한 ISATAP 패킷 필터링 동작을 확인하기 위해 그림 10과 같이 테스트 환경을 구축하였다.

그림 10에서 6to4 테스트 환경에서 TEP 주소 영역에 ISATAP 주소를 표시하기 위해 "5efe"주소를 지정하고 테스트 시스템인 HA에서 ISATAP 패킷 필터링 옵션을 지정하였다.

그림 11의 ①에서 ISATAP 송신자 IPv6 주소의 IPv4 주소 영역이 192.168.0.0/16 값으로 매칭이 되면

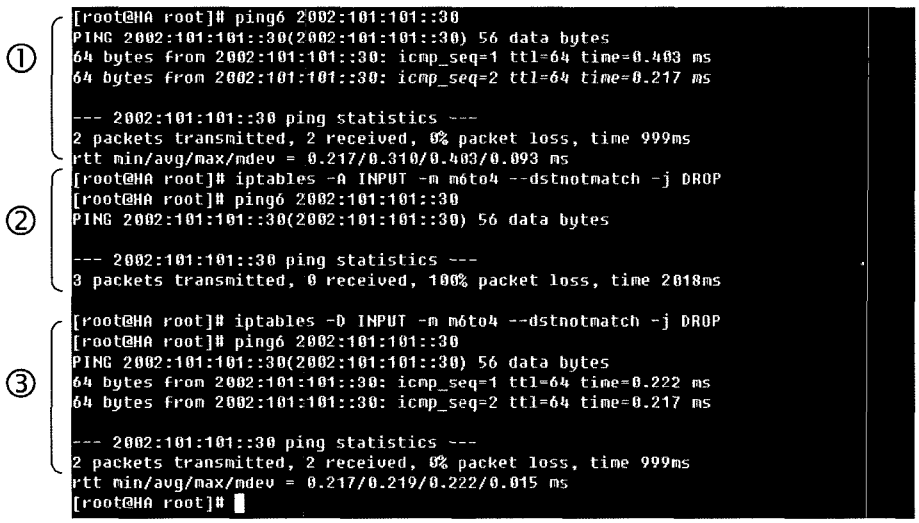


그림 9 6to4 패킷 필터링에 의한 패킷 폐기 확인

표 5 ISATAP 관련 iptables 명령어

가능옵션	예제 및 설명
--v4-source	**iptables -A INPUT -m isatap --v4-source 192.168.0.0/16 -j DROP** 이 옵션은 6to4 IPv6 송신자 주소 내에 IPv4 주소에 의해 패킷을 매칭할 수 있다. 옵션 지정 방법은 iptables -s 옵션과 동일하다.(즉 마스크 지정, ! 매치, DNS 주소 매치까지 지원함)
--v4-destination	**iptables -A INPUT -m isatap --v4-destination 192.168.0.0/16 -j DROP** 이 옵션은 6to4 IPv6 수신자 주소 내에 IPv4 주소에 의해 패킷을 매칭할 수 있다. 옵션 지정 방법은 iptables -s 옵션과 동일하다.(즉 마스크 지정, ! 매치, DNS 주소 매치까지 지원함)

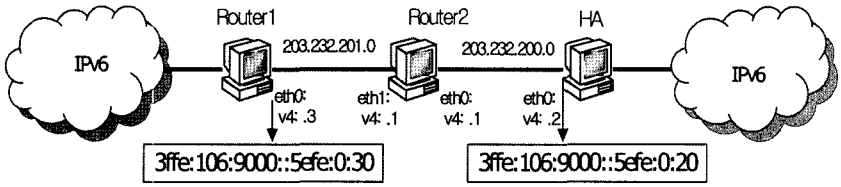


그림 10 ISATAP 패킷 필터링 기능 확인을 위한 테스트 환경

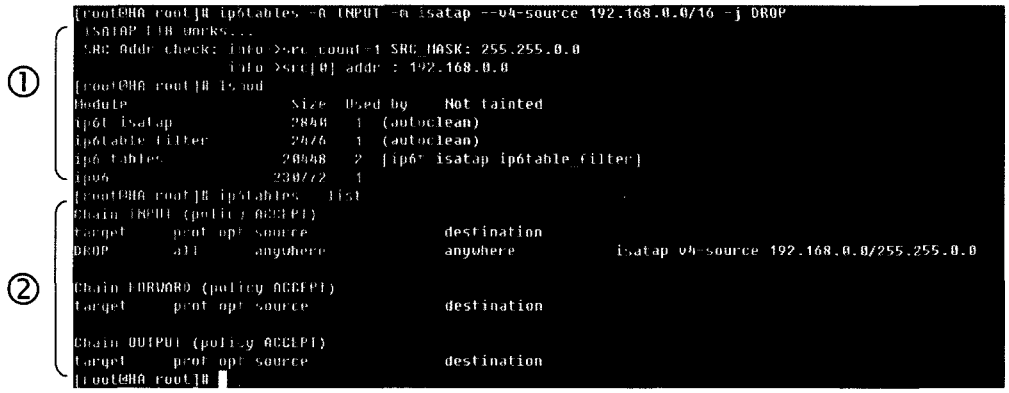


그림 11 ISATAP 패킷 필터링 지정 모습

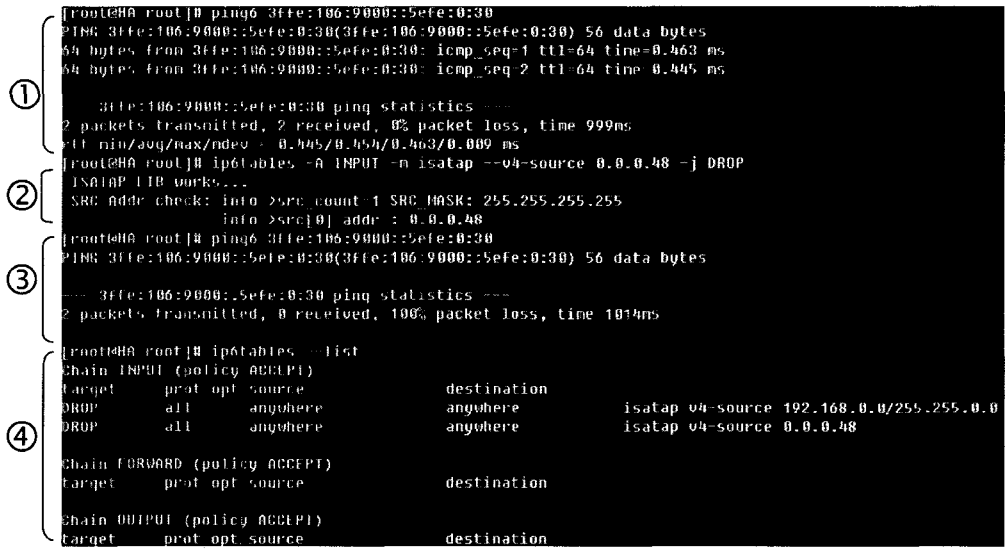


그림 12 ISATAP 패킷 필터링에 의한 패킷 드랍 확인

모두 폐기하라고 설정하였다. 그 후 lsmod 명령어를 입력하여 isatap 커널 모듈(ip6t_isatap)이 제대로 등록이 되어있는 지를 확인하였다. 또한 ② 부분에서는 iptables list 명령어를 사용하여 isatap 패킷 필터링 규칙이 제대로 등록된 것을 확인하였다. 그림 12는 ISATAP 송신자 IPv6 주소에 포함된 IPv4 주소에 대한 필터링 기능을 확인한 내용이다.

그림 12의 ①에서 ping6 명령어로 그림 10의 Router1과 연결성을 확인하였다. 현재 Router1의 주소가 "3ffe:106:9000::5efe::30"이므로 ISATAP 주소임을 표현하는 "5efe" 뒤의 네 바이트(0:30)가 v4 주소에 해당된다. 이를 v4 형식의 주소 값으로 표현하면 "0.0.0.48"이 되므로 이러한 송신자 v4 주소 값을 가진 ISATAP 패킷을 매칭하기 위해 ②부분에서는 옵션을 --v4-source 0.0.0.48"로

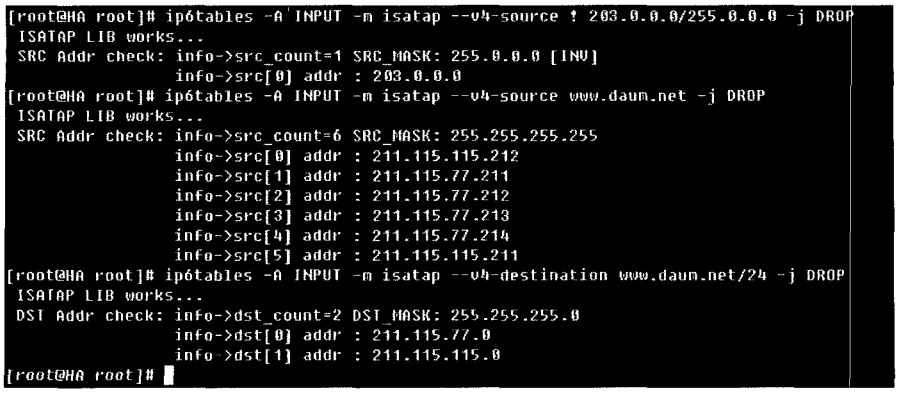


그림 13 ISATAP 필터링에서 여러 형식의 주소 지정 예

지정하였으며, 이런 ISATAP 패킷은 폐기하라고 설정하였다. 그 결과 ③과 같이 ping6 reply 패킷이 도착하지 않는 것을 확인할 수 있다. ④에서는 이러한 필터링 규칙이 ip6tables에 설정된 것을 보여준다. 그림 13은 여러 형태(inverse, net mask, DNS 지정 등)의 v4-source (destination) 옵션 지정이 가능한 것을 보여 준다.

2.2절에서 제시한 IPv4/IPv6 터널링 연동환경에서의 문제점은 방화벽이 터널링 된 패킷을 처리할 때 외부 헤더에 대해서만 필터링을 적용하고 내부에 캡슐화 된 헤더에 대해서 필터링을 수행하지 못하기 때문에 발생한다. 따라서 이를 해결하기 위해서는 방화벽에서 터널링 된 패킷을 인식하고 외부 헤더뿐만 아니라 내부 헤더에 대해서도 패킷 필터링을 수행하는 기능이 필요하다.

본 논문에서는 4.2와 4.3절을 통해서 대표적인 터널링 기반 연동 기술인 6to4와 ISATAP 환경에서 터널링 된 패킷의 외부 헤더와 내부 헤더에 대해서 구현된 패킷 필터링 기능이 정상적으로 동작하는 것을 확인하였다.

5. 결론 및 향후 과제

현재의 모든 IPv4망을 단기간 내에 IPv6망으로 대체하는 것은 높은 비용과 기술적인 제약으로 인해 어렵기 때문에 상당한 기간 동안 IPv4와 IPv6가 공존하게 될 것이다. IPv4와 IPv6가 상호 공존하는 환경에서는 각각의 프로토콜을 기반으로 하는 보안에 문제가 없더라도 연동에 따른 보안 문제가 새롭게 발생한다. 따라서 IPv6로의 효과적인 이전과 정착을 위해서는 IPv4/IPv6 연동 과정에서 발생하는 다양한 보안 위협에 대한 분석과 이에 대한 해결 방안이 반드시 필요하다.

본 논문에서는 IPv4/IPv6 연동 환경에서 발생할 수 있는 보안 위험요소를 막기 위해 터널링 연동환경에 적합한 패킷 필터링 규칙을 제시하였다. 제시된 패킷 필터링 규칙은 기존 리눅스 방화벽에서 테스트를 통하여 필터링 규칙 지원 여부를 확인하였는데, 기존 리눅스 시스템은 IPv6의 기본 헤더 및 확장 헤더의 여러 필드에 대해서 비교적 충실한 필터링 기능이 제공하지만 6to4나 ISATAP 등의 자동 터널링 기법에 필요한 터널링 규칙에 대해서는 필터링 기능을 제공하지 못하는 것을 확인하였다. 이것을 바탕으로 하여, 리눅스 시스템의 넷필터(netfilter)와 ip6tables를 이용하여 터널링 환경에 적합한 패킷 필터링 기능을 설계·구현하였으며, 시험용 테스트베드 터널링 연동환경에서 정상적으로 동작하는 것을 확인하였다

향후 연구과제로는 현재 표준화가 진행 중이거나 표준화가 완성된 Teredo, NAT-PT, DSTM 등의 연동 메커니즘과 Mobile-IPv6를 지원하는 방화벽 기능의 설계·구현이 필요하다.

참 고 문 헌

[1] E. Davies et al., "IPv6 Transition/Co-existence Security Considerations," draft-ietf-v6ops-security-overview-04.txt(work in progress), March 6, 2006.

[2] P. Savola, "Firewalling Considerations for IPv6," draft-savola-v6ops-firewalling-01.txt(work in progress), March 2003.

[3] M-K Shin, "Security Implication of IPv6 and IPv6 Transition," ETRI, May 2005.

[4] R. Gilligan and E. Nordmark, "Transition Mechanisms for IPv6 Hosts and Routers," RFC 2893, August 2000.

[5] B. Carpenter and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds," RFC 3056, February 2001.

[6] F. Templin et al., "Intra-Site Automatic Tunnel Addressing Protocol(ISATAP)," draft-ietf-ngtrans-isatap-24.txt(work in progress), January 2005.

[7] 정상진, "IPv6 방화벽 요구사항 분석서," ETRI, Aug. 2005.

[8] P. Savola, "Security of IPv6 Routing Header and Home Address Options," draft-savola-ipv6-rh-ha-

security-03.txt(work in progress), December 2002.
[9] E. Davies et al., "Best Current Practice for Filtering ICMPv6 Messages in Firewalls," draft-davies-v6ops-icmpv6-filtering-bcp-00.txt(work in progress), July 2005.
[10] P. Savola, "Security Considerations for 6to4," RFC 3964, December 2004.
[11] Satomi Okazaki and Anand Desai, "NAT-PT Security Considerations," draft-okazaki-v6ops-natpt-security-00.txt(work in progress), June 2003.
[13] R. Graveman et al., "Using IPsec to Secure IPv6-in-IPv4 Tunnels," draft-ietf-v6ops-ipsec-tunnels-02.txt (work in progress), March 2006.
[14] FreeBSD Hypertext Man Pages, "IPFW-IP firewall and traffic shaper control program," www.freebsd. org.
[15] 정상진, "IPv6 방화벽 구현 설계서," ETRI, Aug. 2005.
[16] IANA, "Special-Use IPv4 Addresses," RFC 3330, September 2002.
[17] Rusty Russell, "Linux 2.4 Packet Filtering HOWTO," www.netfilter.org.
[18] Rusty Russell, "Linux Netfilter Hacking HOWTO," www.netfilter.org.
[19] Fabrice MARIE, "Netfilter Extensions HOWTO," www.netfilter.org.

허 석 렬

1986년 경북대학교 전자공학과 학사. 1991년 경북대학교 컴퓨터공학과 석사. 1993년 경북대학교 컴퓨터공학과 박사과정수료. 1992년 3월~2006년 2월 밀양대학교 컴퓨터공학부 교수. 2006년 3월~현재 부산대학교 바이오정보전자공학과 교수 관심분야는 RFID/USN, 컴퓨터 네트워크, 네트워크 보안, u-Health

이 완 직

1992년 경북대학교 통계학과 학사. 1994년 경북대학교 컴퓨터공학과 석사. 1997년 경북대학교 컴퓨터공학과 박사과정수료. 1997년 3월~2006년 2월 밀양대학교 정보통신공학부 교수. 2006년 3월~현재 부산대학교 바이오정보전자공학과 교수 관심분야는 통신 프로토콜, 프로토콜 구현, 네트워크 보안

김 경 준

1996년 경일대학교 컴퓨터공학과 학사 1999년 경북대학교 컴퓨터공학과 석사 2005년 경북대학교 정보통신학과 박사 2005년 9월~2006년 8월 대구대학교 정보통신공학부 초빙교수. 2006년 9월~현재 호남대학교 전파이동통신공학과 전임강사. 관심분야는 무선센서네트워크, 디지털홈, 유/무선네트워킹

정 상 진

1999년 KAIST 전산학과 이학사. 2001년 한국정보통신대학교(ICU) 통신공학부 공학석사. 2001년~현재 한국정보통신대학교(ICU) 공학부 박사과정. 2003년~현재 한국전자통신연구원(ETRI) 차세대인터넷표준연구팀 연구원. 관심분야는 IPv6, Mobility, WiBro

신 명 기

1992년 홍익대학교 전자계산학과 이학사 1994년 홍익대학교 대학원 전자계산학과 이학석사. 2003년 충남대학교 대학원 컴퓨터공학과 공학박사. 2004년~2005년 미국 국립표준기술연구원(NIST) 초빙연구원. 1994년~현재 한국전자통신연구원(ETRI) 차세대인터넷표준연구팀 선임연구원. 2001년~현재 IETF 국제표준화 기구 에디터(RFC 3338, RFC 4038 등) 2005년~현재 IPv6 Forum CTO Excom 집행위원. 관심분야는 차세대인터넷 기술, IPv6, 멀티캐스트, 이동성 관리, IP 보안 기술

김 형 준

1986년 광운대학교 컴퓨터공학과 학사 1988년 광운대학교 컴퓨터공학과 석사 2003년 충남대학교 컴퓨터과학과 박사과정 수료. 1988년~현재 한국전자통신연구원(ETRI) 차세대인터넷표준연구팀장 책임연구원. 2004년~현재 ITU-T SG13 Q9의장. 2006년~현재 ITU-T SG13 Q2 표준문서 에디터 2005년~현재 모바일 RFID 포럼 표준기획분과위 위원장 2005년~현재 OSIA RFID/USN TG 의장. 2003년~현재 IPv6 포럼 코리아 사무국장. 관심분야는 차세대인터넷 기술, IPv6, RFID/USN

한 기 준

1979년 서울대학교 전기공학과 학사 1981년 KAIST 전기 및 전자공학과 석사. 1985년 University of Arizona 전기 및 전산공학과 석사. 1987년 University of Arizona 전기 및 전산공학과 박사 1981년~1984년 국방과학연구소 연구원 1988년~현재 경북대학교 컴퓨터공학과 교수. 관심분야는 Ad-hoc Network, Wireless Personal Area Network, Home Network, Ubiquitous Sensor Network