IPv6를 위한 XML 기반 안전한 터널 브로커
(XML based Secure Tunnel Broker for IPv6)

서 창 호 † 윤 보 현 ††
(Chang-Ho Seo) (Bo-Hyun Yun)

요 약 터널 브로커(Tunnel Broker)는 IPv6 주소환경에서 전용 서버를 구축하고, 사용자의 터널 요청을 자동으로 관리하는 방법이다. 이 개념은 IPv6의 호스트 증가를 시뮬레이션하거나 IPv6 네트워크에 쉽게 접근하는데 유용하다. 그러나 기존의 터널 브로커는 악의의 사용자의 네트워크 자원과 서비스에 대한 공격에 취약하다. 따라서 이러한 터널 브로커의 보호문제를 극복하기 위해 본 논문에서는 TSP(Tunnel Setup Protocol) 기반 안전한 IPv6 터널 브로커를 제안한다. 클라이언트와 TB구간에서 서로 통신하기 위해 HTTP에 기초하기 않고 SHTTP(Secure HTTP)에 기초하며, XML 메시지를 송수신시 암호화/복호화하여 통신한다. 마지막으로 클라이언트와 터널 서버사이에서는 IPSec을 적용하여 중요한 정보를 암호화 및 복호화한다.

키워드 : IPv6, 터널설정 프로토콜, IPSec, XML

Abstract The Tunnel Broker is to provide dedicated servers and to automatically manage tunnel requests coming from the users. This approach is useful to stimulate the growth of IPv6 interconnected hosts and to provide easy access to their IPv6 networks. However, the existing tunnel broker is vulnerable to attacks of malicious users about network resources and services. Therefore, to solve the secure problem of tunnel broker, this paper presents secure IPv6 tunnel broker based on TSP(Tunnel Setup Protocol). The clients and the tunnel broker are communicated based on SHTTP(Secure HTTP) and the XML message of plain text is converted to XML signature by encryption and decryption. Finally, Clients and tunnel server use the IPsec method to protect the important information.

Key words : IPv6, TSP, IPSec, XML

1. 서 론

IPv4의 주소 고갈 문제로 인하여 새로운 IP 기술에 대한 요구가 발생하고 있으며, IETF에서는 IPng(IP next generation) 작업 그룹을 구성하여, 새로운 프로토콜 개발을 목표로 1994년부터 표준화 작업을 진행하였다. 이 그룹의 활동 결과로 IPv6 프로토콜이 개발되었으며, 1996년부터 세계적으로 IPv6의 연동 및 시험을 목적으로 6Bone이라는 가상망이 구축되어 운영되고 있다[1].

IPv6 환경에서 터널 전용 서버를 구축하여 사용자의 터널 요청을 자동으로 관리하는 방법인 터널 브로커(Tunnel Broker: TB)의 개념이 1998년 Orlando의 IETF 문서에 처음으로 제안되었다. 그러나 기존의 터널 브로커는 클라이언트와 터널 브로커(TB), 터널 브로커(TB)와 터널 서버(TS) 그리고 터널 브로커(TB)와 DNS 구간에서 보안문제가 있다[2,3]

그 이유는 클라이언트와 터널 브로커(TB) 구간에서 서로 통신하기 위해 HTTP에 기초하고 있으며, 터널을 생성하기 위한 중요한 정보 전송시 암호화하지 않고 텍스트로 송신하기 때문이다. 또한 터널 서버는 IPv4 주소가 다이얼업 ISP의 다른 가입자에 동적으로 할당할 수 있는 동시에 사실에 관계없이 오래된 IPv4의 사용자는 IPv6 트래픽 주소의 터널링을 계속 유지하지 못하도록 하여야 한다. 아울러 확실한 많은 터널들의 요청에 대비하여 만들어진 터널 서버(TS)에서의 모든 자원들을 악의의 사용자가 다 써버리는 서비스 거부 공격의 부정에 대해 보호되도록 구현되어야 한다.

따라서 본 논문에서는 이와 같은 보안문제를 해결하

· 본 논문은 2005년도 한국과학재단 NO. R01-2005-000-10200-0 연구비에 의해 연구 되었음
† 정 회 원 : 공주대학교 응용수학과 교수
chseo@kongju.ac.kr
†† 정 회 원 : 목원대학교 컴퓨터교육과 교수
ybh@mokwon.ac.kr
논문접수 : 2005년 4월 12일
심사완료 : 2005년 9월 5일

기 위해 TSP(Tunnel Setup Protocol) 기능을 적용한 안전한 터널 브로커에 대하여 제안한다. 클라이언트와 터널 브로커(TB)구간에서 서로 통신하기 위해 HTTP에 기초하기 않고 SHTTP(Secure HTTP)에 기초하며, XML 메시지를 송수신시 암호화/복호화하여 XML Signature로 송신한다. 마지막으로 클라이언트와 터널 서버사이에서는 IPSec을 적용하여 IP를 암호화 및 복호화한다.

2. 관련연구

터널링 기술은 말이 의미하는 바와 같이 IPv6망에서 IPv4망을 거쳐서 IPv6 망으로 이동할시 IPv4 망에 터널을 만들어 IPv6 패킷이 지나갈 수 있도록 하는 개념을 의미한다. IPv4/ IPv6 듀얼 스택 호스트[17]와 라우터는 IPv6 데이타그램을 IPv4 패킷에 캡슐화하여 IPv4 라우팅 토폴로지 영역을 통해 터널링 할 수 있다[4,6,7].

그 중 대표적인 것을 살펴보면 'Configured tunnel', '6to4', '6over4', 'ISATAP'(Intra-Site Automatic Tunnel Addressing Protocol), 'Teredo', 'IPv6 over MPLS' 등이 있다[5,8,9].

IPv6 in IPv4 터널링 기술은 크게 설정 터널링(Configured Tunneling)과 자동 터널링(Automatic Tunneling)으로 구분할 수 있다. 설정 터널링은 6Bone에서 주로 사용하는 방식으로 실제 통신이 일어나기 전에 터널 종단간의 라우터를 미리 설정하는 방식으로 발신 호스트에서 생성된 IPv6 패킷의 목적지 주소는 최종 목적지의 IPv6 호스트 주소를 포함하고 있게 된다.

자동 터널링은 설정 터널링과 달리 실제 통신이 일어나면 자동으로 터널 종단을 설정하는 방식입니다. 이때 발신 호스트에서 생성된 IPv6 패킷은 IPv4 주소를 포함하는 IPv4 호환의 IPv6 주소 패킷을 사용한다[10,12,13].

기본적으로 터널링 기술을 이용하면서 추가적인 기술을 통하여 기능을 향상시킨 터널링 메커니즘은 다음과 같다[14-16]. 6to4는 명시적인 터널의 설정 없이 IPv6 네트워크 사이에 IPv4 네트워크를 통해 상호간에 통신하기 위한 메커니즘으로 하나 이상의 유일한 IPv4 주소를 가지고 있는 IPv6 전용 사이트에 "2002:IPv4 주소::/48" 단일 IPv6 프리픽스를 할당하여 외부 IPv6 네트워크와 자동 터널링을 가능하도록 하는 기술이다.

터널 브로커는 IPv6 네트워크에 안정적이고 지속적인 IPv6 주소와 DNS 이름을 중계하기 위해 도입된 개념으로 터널 브로커라는 전용 서버를 구축하여 사용자의 터널 요구를 자동으로 관리하는 방법이다. 현재 대부분의 6Bone 네트워크는 수동으로 설정된 터널을 사용하고 있다. 이는 관리자의 관리 작업이 지나치게 많아진다는 단점이 있는데, 관리 부하를 감소시키는 방법이다.

터널 브로커(TB)와 6to4 장치의 차이점은 그들이 IPv6 커뮤니티의 서로 다른 세그먼트를 제공한다. 터널 브로커(TB)는 소형의 고립된 IPv6 사이트에 잘 맞고, 기존의 IPv6 네트워크에 쉽게 연결을 원하는 IPv4 인터넷의 고립된 IPv6 호스트에 특히 잘 맞는다[3]. 6to4 방법은 고립된 IPv6 사이트들이 IPv4 ISPs에 처음 IPv6 서비스를 전달하기를 기다리지 않고 쉽게 연결될 수 있도록 설계되었다. 이것은 엑스트라넷과 가상 개인 네트워크 사이트에 매우 잘 맞는다. 6to4 중계 장치를 사용하여 6to4 사이트들은 IPv6 인터넷 사이트에 도달할 수 있다. 또한, 터널 브로커 방법은 네트워크 자원을 이용하여 그들의 고유한 방법을 강요하는 사용자들을 쉽게 접근 제어 수행하는 IPv6 ISP를 지원한다.

ISATAP(Intra Site Automatic Tunnel Addressing Protocol)은 듀얼스택 호스트와 듀얼스택 라우터들을 IPv4 네트워크 상에서 연결하기 위한 메커니즘으로 IPv6 게이트웨이와 공통 데이타 링크를 공유하지 않는 듀얼스택 노드가 사이트 내에서 IPv4라우팅 인프라를 통해 IPv6 메시지를 자동으로 터널링 함으로서 글로벌 IPv6 네트워크에 결합할 수 있도록 한다..

Teredo는 IPv4 NAT 상에서 위치한 노드에 UDP 상의 터널링 패킷을 통하여 IPv6 연결성을 제공하는 기술이다.

DSTM(Dual Stack Transition Mechanism)은 임시의 글로벌 IPv4 주소를 IPv6 노드에 제공하는 방법과 IPv6 네트워크에서 동적 터널을 사용한 IPv4 트래픽 전송, 그리고 지원 인프라에 대한 정의를 하고 있는 기술로 IPv6 초기에 IPv6 네트워크 내에서 IPv4 주소를 철저하게 사용하는 장점이 있다. DSTM은 IPv6 패킷 내부에서 IPv4 패킷의 동적 터널링을 수행하고, IPv6 네트워크의 DSTM 도메인 내에서 IPv4 순수 패킷의 노출을 억제하는 능력과 관련이 있다[11].

그리고, IPv6 라우팅 테이블만 있으면 라우터가 IPv6 라우팅을 통해 IPv4 패킷을 이동할 수 있으므로, IPv6 배치의 네트워크관리가 간단하다.

3. 터널 브로커 및 IPsec

3.1 터널 브로커(TB) 개념

IPv6 네트워크의 성장은 현재의 인터넷에 의해 제공된 전송 시설을 사용하면서 시작했다. 이것은 IPv4 터널위에서 IPv6를 관리하기 위한 몇몇 기술들의 발전을 가져왔다. 현재, 대부분의 6Bone 네트워크는 수동으로 설정된 터널을 사용하여 구축된다. 이 방법의 단점은 네트워크 관리자의 관리 작업이 지나치게 많다는 점이다. 관리자는 각 터널마다 광범위한 수동설정을 수행해야 한다. 이 관리 오버헤드를 줄이려는 방법 중의 하나가 바로 터널 브로커 메커니즘이다[2].

터널 브로커(TB) 개념은 터널 브로커라는 전용 서버를 구축, 사용자의 터널 요청을 자동으로 관리하는 방법이다. 이 방법은 IPv6로 연결된 호스트의 성장을 촉진시키고, 초기 IPv6 네트워크 제공자들이 그들의 IPv6 네트워크에 쉽게 접근할 수 있도록 해주는 데 유용하다.

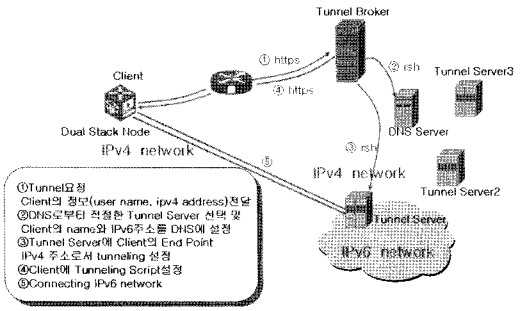

그림 1 터널 브로커 모델

터널 브로커(TB)는 IPv4 인터넷에 연결된 사용자에게 IPv6 연결을 제공하는 가상 IPv6이다. 새로운 IPv6 인터넷에서 터널 브로커(TB)를 이용하여 사용자는 IPv6 서비스를 받을 수 있다. 터널 브로커(TB)의 목록은 웹 페이지 (IPv4 HTTP)를 통해 접근하게 된다. 터널 브로커(TB) 모델은 그림 1에서 보는 바와 같으며, 클라이언트(Client), 터널브로커(TB), 터널 서버(Tunnel Server:TS) 등으로 구성하고 있다.

3.2 IPsec

IPsec은 IETF에서 1995년 8월 RFC로 채택된 이후 IPsec working group에서 현재까지 표준화가 진행 중이다. IPsec은 어플리케이션에 의존적인 기존의 보안 프로토콜들과 달리 IP 계층에서 보안을 제공하여 사용자에게 투명한 보안 서비스를 어플리켄이션 프로그램에 독립적으로 제공할 수 있다.

1) IPsec과 보안 연계

IPv6 인터넷에서의 정보보호는 IPsec으로 대표된다. IPsec은 이제 인터넷에서 필수적인 암호와 인증 서비스를 구조적으로 제공하면서 안전한 키교환과 재현 공격 등을 방어할 수 있다. IPsec은 IP 계층을 지나는 패킷에 AH와 ESP 헤더를 처리한다[4,5]. AH는 IP 패킷 전체에 무결성 및 인증을 위해 필요한 헤더이며, ESP는 페이로드내에 데이타를 암호화하는데 사용되는데, 두 헤더를 사용하여 패킷 단위로 제공되는 보안 서비스는 그림 2와 같다.

AH와 ESP는 어떤 패킷을 암호화하여 전송한다면 두 호스트 및 게이트웨이가 같은 알고리즘과 키를 공유하여 IPsec 엔진이 패킷을 처리할 수 있게 한다. 각 보안

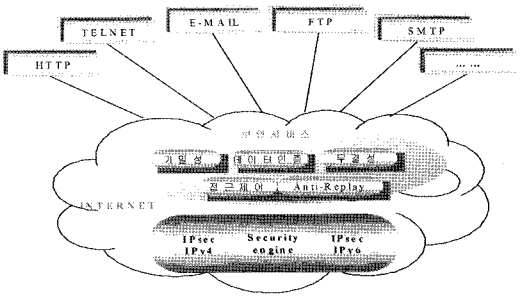

그림 2 IP 레벨의 보안 서비스

프로토콜과 관련된 보안 정보의 집합을 SA(Secu-rity Association, 보안연계)라 한다. SA는 단방향으로 적용되고, 한 연결이라 할지라도 outbound와inbound 프로세싱에 따라 다른 보안 연계를 적용해야 한다. 이러한 SA는 SPI(Security Parameter Index), IPv4나 IPv6 목적지 주소, 보안 프로토콜(AH,ESP)로서 식별하고 각 호스트 및 게이트웨이는 IPsec엔진과 같은 커널 계층에 SADB를 구축하여 패킷을 처리한다.

2) IPv4와 IPv6 환경에서의 IPsec

IPv4 기반 환경에서 IP계층의 트래픽 보안 서비스를 제공하는 목적으로IPsec이 다양한 플랫폼에서 구현되어 선택사항으로 적용되어지고 있다. 최근들어 공중망을 사설망과 같이 사용하고 비용측면에서 전용선보다 저렴하여 각광을 받고 있는 VPN(Virtual Private Network)은 이전의 터널링 프로토콜보다 현재 IPsec을 가장 많이 적용하여 서비스를 제공하고 있다. 이러한 IPsec은 IPv6 규격에서는 확장 헤더 중의 하나로서 AH와 ESP 헤더를 규정하여 필수사항으로 IPsec을 이용한 보안 서비스를 제공한다. IPv6에서는 확장 헤더를 이용하기 때문에, IPv6 기반 보안 기법은 보안 기능의 필요성과 망 효율성에 따라 쉽게 첨가 또는 제거할 수 있는 특징을 가진다. IPsec은 IP계층에서 제공되기 때문에 TCP, UDP, ICMP, BGP 등의 어떠한 상위 프로토콜에 의해서도 사용되는 어떠한 응용에도 보안 서비스를 제공할 수 있다.

IPv6가 IPv4 기반의 현재 인터넷상에 도입되기 시작함에 따라 논의되는 중요한 이슈들 중 하나는 바로 IPv4에서 IPv6로의 자연스러운 이전을 지원해주는 IPv6 전환 메커니즘에 관한 연구이다. 새로 구현될 IPv6 시스템은 IPv4/IPv6 듀얼스택(dual-stack)이거나 혹은 IPv6전용(native)형태로 구현될 것이다. 호스트나 라우터 같은 장비에서 가장 기본적인 전환 방법인 IPv4/IPv6 듀얼 스택은 호스트와 라우터에서 두 인터넷 프로토콜, 즉 IPv4와 IPv6를 모두 지원하는 방식이다. 최근들어 시스코, 주니퍼 등의 대표적인 라우터 장비들이 이를 지원하고 있으며, 리눅스, FreeBSD, Solaris, 윈도우

등 대부분의 운영체제들에도 구현되어져 있다. 특히 리눅스는 커널 버전 2.4.x부터 IPv6를 지원하고 있다.

4. 안전한 터널 브로커

4.1 터널 브로커 모델

터널링 매커니즘은 기존의 VPN에서와 같이 IPv6 데이타그램을 IPv4패킷에 캡슐화하여 영역을 통과하는 방법을 말한다. 제안된 터널 브로커(TB)는 터널설정을 서버를 통해 자동 관리 하는 것으로 생각하면 된다. 터널 브로커(TB)의 목록은 웹 페이지 (IPv4 S-HTTP)를 통해 접근하게 된다. TSP 및 IPsec 적용한 제안된 터널 브로커(TB) 모델은 그림 3에서 보는 바와 같으며, 클라이언트, 터널브로커(TB), 터널 서버(TS) 등으로 구성하고 있다.

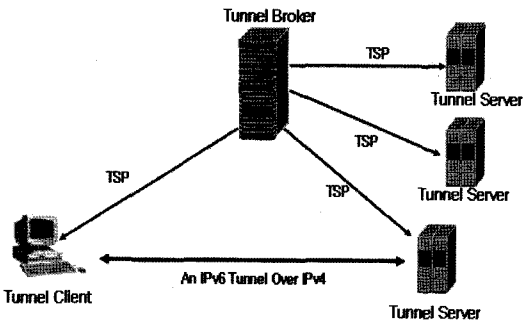

그림 3 안전한 터널 브로커 모델

4.2 TSP를 적용한 동작 과정

클라이언트는 IPv4 인터넷에 연결되어 있고, 터널 서버(TS)는 듀얼 스택 IPv6노드(호스트 또는 라우터)이다. 클라이언트가 터널 브로커(TB)에 접근하면 우선 적절한 사용자 인증, 허가 및 계산을 수행할 수 있도록 그 ID와 증명서를 제공하고 있으며, 다음과 같은 단계로 동작한다.

- 단계 1 : 터널 브로커(TB)관리자가 정의한 로드 공유 기준에 따라 네트워크 측에서 실제 터널 종단점으로 사용될 터널 서버(TS)를 지정한다.
- 단계 2 : 클라이언트에 할당할 IPv6프리픽스를 선택한다. 프리픽스 길이는 0-128범위이고 가장 일반적인 값은 48(사이트 프리픽스), 64(서브넷 프리픽스) 또는 128(호스트 프리픽스)이다.
- 단계 3 : 터널 수명을 고정한다.
- 단계 4 : 터널 종단점을 지정된 글로벌 IPv6 주소를 자동으로 DNS에 등록한다.
- 단계 5 : 터널의 서버 측을 설정한다.
- 단계 6 : 터널의 매개변수 및 DNS 이름을 비롯한 관련 설정 정보를 클라이언트에 통보(클라이언트의 설정을 포함하여) 이상의 설정 단계가 수행되고 나면, 클라이언트 호스트/라우터와 선택된 터널 서버(TS) 간에 IPv6 in IPv4 터널이 설정되어 작동되므로 터널 브로커(TB)사용자는 6Bone 또는 터널서버(TS)가 연결된 기타 임의의 IPv6 네트워크에 접근 할 수 있다.

TB로부터 설정 명령을 받으면 TS는 각 터널의 서버 측을 수정 또는 삭제한다. 그리고 모든 활성터널에 대해 사용 통계를 관리할 수 있다.

4.3 안전한 TSP 동작 과정

클라이언트와 터널 브로커 서버 사이의 셋업 터널에 대한 제어 프로토콜이다. 기존의 TSP(Tunnel Setup Protocol) 프로토콜은 메시지를 보내고 있는 단순한 XML에 기반에 두 개체간의 터널 파라미터를 협정을 위한 기반 구조인데[], 본 제안 시스템에서는 정보보호 서비스인 기밀성과 무결성 서비스를 제공하기 위해 XML_Signature 기능을 제공하여 보다 더 안전한 터널에서 협정이 가능하고 중요한 파라미터에 대한 보안 기능을 제공한다. 아래의 그림 4는 클라이언트와 터널 서버 사이의 안전하게 메시지를 XML 보안으로 동작하는 과정이다.

보안 기능을 추가한 TSP은 3가지인 인증단계, 명령단계 및 응답단계에 의해서 동작한다. 인증단계는 터널 브로커(TB)와 서버가 터널 클라이언트(TC)에 그 능력을 공시할 때 터널 클라이언트(TC)가 서버를 믿을 수 있는지 증명할 때, 명령단계는 클라이언트의 요청이나 터널 업데이트하고 자 하는 경우, 그리고 마지막인 응답단계는 클라이언트의 응답이다. 일단, 터널 클라이언트에 의해 보내진 각 명령은 서버에 의해 응답을 기다린다.

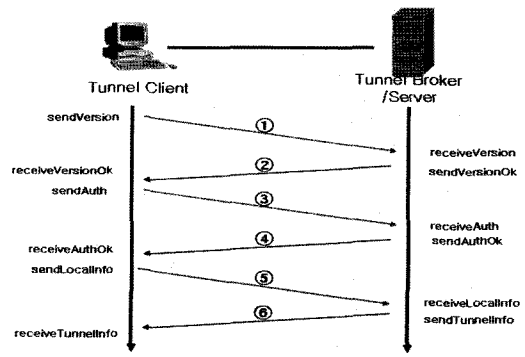

그림 4 안전한 TSP 동작과정

```
● 인증 단계
① C:Version=0.1 CR LF (string)
② S: CAPABILITY TUNNEL=V6V4
     AUTH=DIGEST-MD5
     AUTH=ANONYMOUS CR LF
③ C: AUTHENTICATE ANONYMOUS CR LF
④ S: 200 Authentication successful CR LF

● 명령 단계/응답 단계
⑤ C: Content-length: 123 CR LF
  <tunnel action="create" type="v6v4">
      <client>
         <address type="ipv4">1.1.1.1</address>
       </client>
      </tunnel> CR LF
⑥ S: Content-length: 234 CR LF
     200 OK CR LF
   <tunnel action="info" type="v6v4"
     lifetime ="1440">
   <server>
     <address type= "ipv4">
          206.123.31.114
     </address>
     <address type="ipv6">
     3ffe:b00:c18:ffff:0000:0000:0000:0000
     </address>
     </server>
     <client>
     <address type="ipv4">1.1.1.1</address>
     <address type="ipv6">
     3ffe:b00:c18:ffff::0000:0000:0000:0001</address>
     <address type="dn">
     userid.domain
     </address>
     </client>
     </tunnel> CR LF
```

4.4 제안된 터널 브로커의 안정성

제안된 안전한 터널 브로커(TB)의 각 구성간의 상호작용에 안전성에 대하여 설명한다. 모든 각 구성간의 상호 작용에 적용한 프로토콜은 TSP(Tunnel Setup Protocol)을 적용하고, TSP를 적용할 때 정보보호 서비스인 기밀성 및 무결성 서비스를 XML_Signataure 형식으로 제공한다.

1) 클라이언트와 터널브로커(TB)간의 상호작용

터널 생성 서비스를 받기 위해서 먼저 클라이언트와 터널 브로커(TB) 사이에 클라이언트 사용자의 ID와 패스워드를 터널 브로커(TB)로부터 받아야 한다. 인가를 받기위해서는 기존의 터널 브로커(TB)에서는 HTTP를 사용하여 웹 서버로 보내어지고 다운받은 자료를 암호화하는 SSL(Secure Socket Layer)과 같은 소켓를 이용하였다. 그러나 본 제안된 모델에서는 보다 더 보안기능이 포함된 S-HTTP 및 보안 소켓인 기밀성과 무결성을 추가된 TSP 프로토콜을 이용하여 인증과정과 접근제어를 설계 구현하였다. 또한, 클라이언트에 터널 매개변수를 제공하기 위해 터널 브로커(TB)가 사용할 새 MIME 컨텐츠 유형 (예, application/tunnel)을 정의하고, 이 정보를 처리하고 실제로 사용자를 위해 터널 종단점을 셋업하는 전용 에이전트가 클라이언트에서 실행한다.

2) 클라이언트와 터널 서버(TS)간의 상호작용

클라이언트와 터널 서버(TS) 구간의 사이의 보안을 해결하기 위해 안전한 SNMP 및 Secure-TSP 프로토콜을 적용하였다. 만약 동적인 DNS 업데이트 과정이 터널 브로커(TB)-DNS 상호 작용을 위해 사용되면, 안전성 문제에 대하여 논의하여야 한다[11]. 반대로, 만약 RSH 명령들에 의거하는 단순한 방법이 사용되면, IPsec 메커니즘이 적용될 수 있다.

클라이언트의 구축이 터널 브로커(TB)에 의해 준비된 실행 스크립트들이 얻어지면, 이런 스크립트들은 관리자나 루트의 역할이 같은 네트워크 인터페이스들을 관리 할 수 있는 권한과 함께 실행하도록 설계하였다. S-http의 MIME 타입에서 Secure-TSP 프로토콜을 이용한 파라미터의 전송은 보다 안전하다.

터널 브로커(TB)를 통해 전에 만들어진 터널의 손실없이 인터넷으로부터 다이얼업 사용자가 접속 중단 될 때 기밀의 손실은 일어날 수 있다. 실제, 터널 서버(TS)는 IPv4 주소가 다이얼업 ISP의 다른 가입자에 동적으로 할당할 수 있는 동시에 사실에 관계없이 오래된 IPv4의 사용자는 IPv6 트래픽 주소의 터널링을 유지한다. 따라서 터널 브로커(TB)가 연결 중지된 사용자들에 대한 IPv6 트래픽을 즉시 중지되도록 설계하였다. 이러한 방법은 전용 에이전트가 클라이언트에서 종료시 중지하는 방법이다.

결국, 터널 브로커(TB)는 아주 확실한 많은 터널들의 요청에 대비하여 만들어진 터널 서버(TS)에서의 모든 자원들을 악의의 사용자가 다 써버리는 서비스 공격의 부정에 대해 보호되도록 구현 하였다. 이 공격에 대한 가능한 보호는 싱글 유저가 같은 시간에 셋업시 허용되는 터널들 수의 관리자적인 제한에 의해 이루어진다.

5. 설계 및 구현

5.1 운용체계 및 개발 환경

본 논문에서 개발을 위한 환경은 소프트웨어로 구현하였다. 소프트웨어로는 사용자 등록을 처리하는 프로그램, 터널 브로커(TB)와 사용자의 클라이언트간의 응용 프로그램, 그리고 클라이언트와 터널 서버(TB)사이의 터널을 생성하기 위한 프로그램으로 나눌 수 있다.

• 개발 환경

①TB :

Window2000 Server + Jakarta-tomcat 3.1 +MS SQL2000 +jdk 1.3 +Mail Server

②TS:DEBIAN LINUX 3.0r1 stable(woody)

-kernel patch:KERNEL:LINUX-2.4.17, IPsec patch 적용

-IPsec Utility:FreeS/Wan 1.96-01 with x509

③TC : (a) Windowxp

(b) DEBIAN LINUX 3.0r1 stable

-kernel patch:KERNEL:LINUX-2.4.17, IPsec patch 적용

-IPsec Utility:FreeS/Wan 1.96-01 with x509

5.2 TSP 기능을 적용한 터널 브로커 실행

TB_server 내부 실행과 동시에 클라이언트 내부에서 실행하는 화면이다(그림 5). 터널 서버(TS)의 자바 서버 프로그램은 터널 브로커(TB)의 데이타베이스를 참조하여 접속한 터널 클라이언트(TC)의 암호화된 계정과 패스워드를 인증한다. 그리고 터널 브로커(TB)와 터널 클라이언트(TC)의 정보를 터널 서버(TS) 내의 터널생성 스크립트에 대입하여 접속한 터널 클라이언트(TC)에 최적화된 스크립트를 생성하고 실행하여 터널 서버(TS)의 서버를 생성한다.

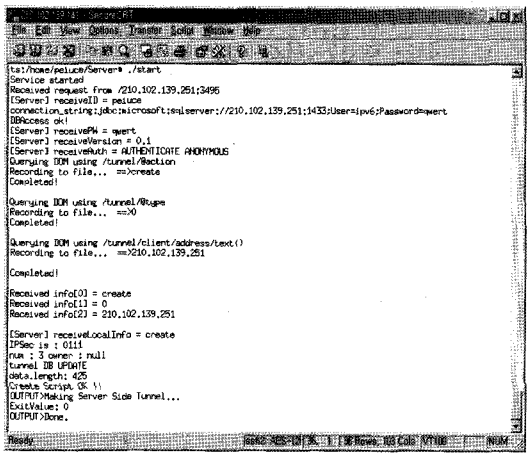

그림 5 TB_server 내부 실행 화면

그림 6은 Client 내부에서 실행하는 화면을 보여주고 있다.터널 서버(TS)에 터널 접속 준비를 완료한 터벌 클라이언트(TC)의 자바 클라이언트 프로그램은 또한 터널 브로커(TB)에서 받아온 TS 접속 정보를 이용하여 내부 스크립트에 대입, 생성하여 터널을 생성하여 터널 서버(TS)의 터널과 연결한다.

그림 6 Client 내부 실행 화면

5.3 구현시스템의 비교

표 1에서는 구현시스템과 외국시스템인 CSEIT와 Freenet을 비교한 결과를 보여준다. 구현시스템만이 TSP를 지원하며, 동적 IP까지 지원한다. 아울러 터널클라이언트와 터널서버간에 IPSec을 지원하며, 터널브로커와 터널클라이언트간에 XML 보안기능을 지원한다. 표에서 구현시스템이 기존의 시스템보다 기능면에서 우수함을 보이고 있다.

표 1 구현시스템의 비교표

	CSEIT	Freenet	구현 시스템
서비스 형태	호스트	호스트 & 네트워크	호스트
개발언어	cgi/perl	cgi/perl	java
TSP지원 여부	X	O	O
동적 IP	X	X	O
보안지원 (통신) (TC-TS:IPSec 적용)	X	X	O
보안지원 (세션) (TB-TC:XML 보안)	X	X	O

6. 결 론

IPv4에서 IPv6로의 전환과 IPv6 사용 활성화를 도모하기 위한 방법 등에 대한 기술적인 논의가 점차 커지고 있다. IPv6로의 전환만이 유일한 대안이 아니라고 할지라도, 대부분의 전문가들이 인정하는 것처럼 IPv4에서 IPv6로 전환되어지고 있다. 다만, 네트워크의 특성상 순식간에 변경되지는 않을 것이며, 상당히 오랜 기간 동안 IPv4와 IPv6 가 공존할 것으로 생각된다.

본 논문에서는 터널 브로커 시스템의 각 구성 요소인 클라이언트와 터널 브로커(TB), 터널 브로커(TB)와 터널 서버(TS) 구간에 TSP(Tunnel Setup Protocol) 프로토콜을 적용한 안전한 터널 브로커 시스템을 제안하였다. 제안된 시스템은 클라이언트와 터널 브로커(TB) 구간에서 서로 통신하기 위해 HTTP에 기초하기 않고 SHTTP(Secure HTTP)에 기초하며, XML 메시지를 송수신시 암호화/복호화하여 XML Signature로 송신한다. Java기술을 이용해서 다중플랫폼과 호환이 가능했으며, 기존 연구와 달리 리눅스기반으로 구축하였다. 구현 시스템이 기존의 시스템보다 여러 가지 기능면에서 우수함을 보였다.

참 고 문 헌

[1] S. Deering and R. Hinden, "Internet Protocol, Version 6(IPv6) Specification," RFC2460, 1998.

[2] R. Hinden and S. Deering, "IP Version 6 Addressing Architecture," RFC2373, 1998.

[3] 박정수외 4명, "차세대 인터넷 프로토콜(Internet Protocol Version 6) 기술 소개", 한국전자통신연구원, 주간기술동향 통권 965호, 2000.

[4] K. Yamamoto and M. Sumikawa, "Categorizing Translators between IPv4 and IPv6," draft-ietf-ngtrans-translator-01.txt, 1999.

[5] T. Larder, "Transition Scenarios and Solutions," draft-ietf-ngtrans-trans-scenes-00.txt, 1999.

[6] W. Biemlt, M. Kaat and etc., "A Guide to the Introduction of IPv6 in the IPv4 world," 1999.

[7] R. Gilligan and E. Nordmark, "Transition Mechanisms for IPv6 Hosts and Routers," draft-ietf-ngtrans-mech-04.txt, 1999.

[8] B. Carpenter and C. Jung, "Transmissioin of IPv6 over IPv4 Domains without Explicit Tunnels," RFC2529, 1999.

[9] B. Carpenter and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds without Explicit Tunnels," draft-ietf-ngtrans-6to4-03.txt, 1999.

[10] J. Bound, "Assignment of IPv4 Global Addresses to IPv6 Hosts (AIIH)," draft-ietf-ngtrans-assgn-IPv4-addrs-01.txt, 1999.

[11] J. Bound and L. Toutain, "Dual Stack Transition Mechanism(DSTM)," draft-ietf-ngtrans-dstm-00.txt, 2000.

[12] A. Durand, P. Fasano and etc, "IPv6 Tunnels Broker," draft-ietf-ngtrans-broker-02.txt, 1999.

[13] E. Nordmark, "Stateless IP/ICMP Translation Algorithm(SIIT)," RFC2765, 2000.

[14] A. Conta and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6(IPv6) Specification," draft-ietf-ipngwg-icmp-v3-00.txt, 1999.

[15] G. Tsirtsis and P. Srisuresh, "Network Address Translation-Protocol Translation(NAT-PT)," RFC2766, 2000.

[16] H. Kitamur, A. Jinzaki and S. Kobayashi, "A SOCKS-based IPv6/IPv4 Gateway Mechanism," draft-ietf-ngtrans-socks-gateway-02.txt, 1999.

[17] K. Tsuchiya, H. Higuchi and Y. Atarashi, "Dual Stack Hosts using the Bump-In-the-Stack Technique(BIS)," RFC2767, 2000.

서 창 호

1990년 고려대학교 수학과 졸업(학사) 1992년 고려대학교 일반대학원 수학과(이학석사). 1996년 고려대학교 일반대학원 수학과(이학박사). 1996년~1996년 국방과학연구소 선임연구원. 1996년~2000년 한국전자통신연구원 선임연구원, 팀장 2000년~현재 공주대학교 응용수학과(정보보호전공) 부교수. 관심분야는 암호 알고리즘, PKI, 무선 인터넷 보안,시스템 보안 등

윤 보 현

1999년 고려대학교 컴퓨터학과(이학박사). 1999년~2003년 한국전자통신연구원(ETRI) 선임연구원 및 팀장. 2001년~2003년 한국소프트웨어산업협회 언어정보산업협의회 운영위원. 2002년~2002년 광인터넷 기술정책 자문위원. 2003년~2004년 한국전자통신연구원(ETRI) 초빙연구원. 2004년 KRnet 컨퍼런스 운영위원. 2003년~현재 목원대학교 컴퓨터교육과 교수. 관심분야는 웹서비스, 정보추출 및 정보 검색 등