FreeS/WAN IPSEC을 이용한 LINUX 라우터 VPN 구성

김한철* · 이계상*

*동의대학교 정보통신공학과

A Configuration of LINUX router VPN using FreeS/WAN IPSEC

Han-Chul Kim* · Kye-Sang Lee*

*Dong-eui University

E-mail : windy1129@hananet.net

요 약

FreeS/WAN[1] 은 LINUX 상에서 네트워크 보안 프로토콜표준인 IPSEC을 구현한 공개 S/W이다. 현재 LINUX Project로 수행되고 있으며 1.91 version 까지 나와 있다. 라우터와 라우터간에 IPSEC을 사용하여 통신함으로써 access control, connectionless integrity, data origin authentication, protection against replays, confidentiality의 서비스를 보장받을 수 있고, 또한 이러한 서비스들은 IP 계층에서 제공되기 때문에 IP 계층뿐만 아니라 그 이상의 계층에 대한 보호를 제공한다. [2]

본 논문에서는 LINUX router에 FreeS/WAN IPSEC을 설치하여 Security Gateway를 구성하고, 이 Security Gateway를 통해 전형적인 가상사설망을 구성할 수 있음을 보였다. 양단의 Security Gateway에 설치되어진 FreeS/WAN으로 VPN connection을 설정하고, 인증방법으로 RSA authentication key를 setup 하였다. IPSEC을 통하여 암호화되어진 데이터로 양단의 Gateway 구간에서 보안통신이 이루어짐을 알아본다.

Ⅰ. 서 론

항시 변화하는 글로벌 데이터 커뮤니케이션의 세계에서, 그리고 값싼 인터넷 연결이 가능한 현재에서, 또한 빠르게 움직이는 소프트웨어 개발에 있어서, 보안은 갈수록 중요한 문제로 떠오르고 있다. 예를 들어, 인터넷상에서 한 데이터가 A 지점에서 B 지점으로 흐르는 중간에 여러 지점에서, 다른 사용자들이 데이터를 가로채거나 변형해 버릴 수 있다.

우리가 풀어야 할 문제는 이 두 네트워크 사이의 안전한 통신을 하는 것이다. 인터넷상에서 두 개의 방화벽 사이에 보안 통신 채널이나 완벽한 개인정보보호, 인증과 데이터 무결성 등을 제공하는 것은 필수적인 것이다. 이런 필수적인 요구들로 IPSec이 생겨났다.

IPSec(Internet Protocol Security)은 인증과 암호화 서비스 모두를 제공하기 위한 강력한 암호화 기법을 사용하고 있다. 인증은 패킷이 확실한 사용자가 보냈음을 확인하고, 전송 중에 변형되지 않았음을 확인한다. 그리고 암호화는 인증되지 않은 패킷을 사전에 방지한다. IPSec은 IP상에서 돌아가는 어떠한 프로토콜이나 IP 밑에서 사용되는 장비들도 보호가 가능하다. 더욱이 암호화는 IP레벨에서 일어나기 때문에 복잡한 응용프로그램이나 상위 프로토콜의 구현에 영향을 받지 않고 보안을 제공한다. IPSec 서비스는 신뢰성 없는 네트워크를 통하여 보안 터널을 생성할 수 있도록 한다. 신뢰할 수 없는 네트워크를 지나는 모든 것들은 IPSec 게이트웨이 시스템에 의해 암호화되고, 다른 끝의 게이트웨이를 통하여 해독하게 된다. 이것을 VPN(Virtual Private Network)이라 한다. 이 논문에서는 FreeS/WAN을 이용하여 Linux 상에서 IPSec을 이용한 VPN을 구성해 보였다.[3][4]

II. VPN의 구성

VPN을 구성하기 위해 본 논문에서는 2개의 PC 보안 게이트웨이와 2개의 호스트를 사용하였다. 각각의 호스트는 여러대의 서브넷과 같은 의미로 사용된다. FreeS/WAN은 1.3 버전을 사용하였고, 리눅스는 레드햇 6.1사용하였다. 그리고, 커널은 2.2.16을 사용하여 VPN을 구성하였다.

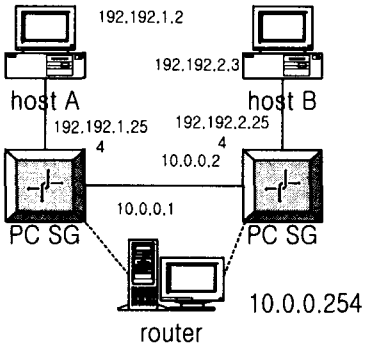

그림 1. VPN 구성도

2.1 FreeS/WAN 설치하기

FreeS/WAN을 설치할 때의 순서는 우선 커널을 컴파일 한다. 다음으로 FreeS/WAN을 풀고 다시 커널을 컴파일해서 FreeS/WAN을 커널에 삽입시킨다.

2.1.1 커널컴파일 하기

```
cd /usr/src/linux
make menuconfig
make dep;make clean;make bzImage;make
modules;make install_modules
```

리눅스 설치후 커널 컴파일을 하지 않으면 FreeS/WAN 설치시 네트워크 옵션에 에러가 나 설치가 되지 않을 수 있음으로 반드시 커널을 컴파일 하여야 한다. [5]

2.1.2 FreeS/WAN 설치

FreeS/WAN을 압축을 풀고 설치하기 위해서는 make insert; make programs; make install 명령을 입력해야한다. make insert 명령은 KLIPS 소스 디렉토리를가르키는 /usr/src/linux/net/ipsec 의 심볼릭 링크를 생성한다. make programs 명령은 Pluto 라이브러리와 다양한 사용자 레벨의 유틸리티를 생성한다. 마지막으로 make install 명령은 Pluto 데몬과 사용자 레벨의 유틸리티들을 설치하고 부팅시 활성화하도록 설정한다. 그런 후 다시 커널을 재 설정하여야 한다.

2.2 FreeS/WAN Configuration

2.2.1 ipsec.conf 설정

FreeS/WAN 설치를 마쳤으면 두 보안 게이트웨이 간에 라우팅 테이블을 구성하여 네트워크가 가능하도록 설정한 후 ipsec.conf와 ipsec.secrets 파일을 편집하여 FreeS/WAN이 실행되도록 한다. /etc/ipsec.conf 파일은 SG 사이의 IP주소와 host 간의 주소 등을 설정하는 conn 부분과 기본적인 설정을 담당하는 config 부분으로 나누어진다.

다음은 ipsec.conf의 config 부분을 나타낸다. interfaces는 IPSec을 사용하기에 적절한 가상 또는 물리적인 인터페이스를 나타낸다. "ipsec0=eth0"의 의미는 eth0을 ipsec0인 것처럼 사용하도록 지정해준다. klipsdebug=none 은 커널 IPSec 코드인 KLIPS 의 디비깅 결과를 지정한다. 디폴트 값은 none이고, 어떠한 디버깅 결과도 없다는 뜻이다. plutodebug=none은 Pluto 키의 디버깅 결과를 지정한다. 디폴트값은 none이고, 어떠한 디버깅 결과도 없다는 뜻이다.

```
# basic configuration
config setup
# THIS SETTING MUST BE CORRECT or almost nothing will work;
# %defaultroute is okay for most simple cases.
        interfaces="ipsec0=eth0"
# Debug-logging controls:  "none" for (almost) none, "all" for lots.
        klipsdebug=none
        plutodebug=none
# Use auto= parameters in conn descriptions to control startup actions.
        plutoload=%search
        plutostart=%search
```

아래의 ipse.conf 파일은 connectoin 부분을 나타낸다. conn은 IPSec을 이용하여 생성되는 접속내용을 확인하도록 이름을 지정한다. 예를 들어 두 SG 간의 접속명을 aaa-bbb라고 지정을 하면 두 SG간의 conn 라인에 aaa-bbb라고 적어넣어두면 된다. 즉 두 SG의 ipsec.conf 파일에는 같은 접속이름이 명시되어 있어야 한다. left, leftsubnet, leftnexthop은 letf SG의 IP주소, 숨어있는 네트워크, 그리고 SG에서 다음으로 넘어갈 라우터나 ISP의 라우터를 나타낸다. 각각의 SG간

에 직접 연결할 경우 leftnexthop은 생략할 수 있다. right, rightsubnet, rightnexthop도 left와 마찬가지이다. auto=start 는 IPSec이 부팅시 시나 활성화시 자동실행이 되도록 지정한다. 파라미터가 add가 되면 ipsec auto(manual) --up name 명령을 실행하게 되면 IPSec이 작동이 된다.

```
conn sample
# left security gateway (public-network address)
        left=10.0.0.1
# next hop to reach right
        #leftnexthop=
# subnet behind left (omit if there is no subnet)
        leftsubnet=192.192.1.0/24
# right s.g., subnet behind it, and next hop to reach
left
        right=10.0.0.2
        #rightnexthop=
        rightsubnet=192.192.2.0/24
        auto=start
```

2.2.2 ipsec.secrets 설정

각각의 SG를 위한 개별적인 RSA 키를 생성해야 한다. SG의 ipsec.secrets 파일에는 고유의 사설키를 가지고 있고, ipsec.conf 파일에 각각의 SG에서 생성된 공개키를 가지고 있어야 한다.

```
ipsec rsasigkey --verbose 1024 > left_Pkey
```

와 같은 입력을 하게되면 left_Pkey라는 파일에 1024비트의 RSA의 공개키와 사설키를 아래와 같이 생성하게 된다.

```
# RSA 1024 bits     Sat Sep 29 13:53:22 2001
# for signatures only, UNSAFE FOR
  ENCRYPTION
#pubkey=0sAQOF8tZ2NZt...1347
 Modulus: 0xcc2a86fcf440...cf1011abb82d1
 PublicExponent: 0x03
# everything after this point is secret
 PrivateExponent: 0x881c59fdf8...ab05c8c77d23
      Prime1: 0xf49fd1f779...46504c7bf3
      Prime2: 0xd5a9108453...321d43cb2b
      Exponent1: 0xa31536a4fb...536d98adda7f7
      Exponent2: 0x8e70b5ad8d...9142168d7dcc7
      Coefficient: 0xafb761d001...0c13e98d98
```

각각의 SG에 생성된 공개키를 ipsec.conf 파일에 추가를 하고 ipsec.conf에 자신만이 가지고 있는 사설키를 복사하여 둔다. ipsec.conf파일에서 아래와 같이 추가를 한다.

```
 # right s.g., subnet behind it, and next hop to
reach left
        right=10.0.0.2
        #rightnexthop=
        rightsubnet=192.192.2.0/24
        keyingtries=0
  authby=rsasig
  leftrsasigkey=<left SG에서 생성된 공개키>
  rightrsasigkey=<right SG에서 생성된 공개키>
  auto=add
```

ipsec.secret 파일도 아래처럼 편집을 한다.

```
10.0.0.1 10.0.0.2: RSA {

Modulus: 0xc · · ·
                · · · 3e98d98
                    }
```

ipsec.secrtes를 설정할 때 주의해야 할 점은 RSA는 각각의 SG에서 생성이 되어야 하고 공개키는 ipsec.conf에 서로가 다 같이 가지고 있어야 하고, ipsec.secrtes에 각자의 사설키를 가지고 있어야 한다. 한번생성한 키파일은 바로 삭제하여 키의 보안을 유지하여야 한다.

III. 테스트

FreeS/WAN 설정을 끝내고 나서 IPSec 제대로 작동을 하는지를 알아보는 방법에 대해 이야기 한다. 우선 FreeS/WAN 시작을 위해 재부팅을 한다. 부팅시 제대로 FreeS/WAN이 작동했는지 아니면 어떤 문제가 있었는지 /var/log/message 파일이나 secure 파일을 살펴보면 된다. 그리고 /proc/net/ 디렉토리 안에

```
ipsec_eroute
ipsec_spi
ipsec_spigrp
ipsec_spinew
ipsec_tncfg
ipsec_version
```

과 같은 파일들이 존재하여야 한다. 이중에 ipsec_tncfg 파일을 살펴보면 Ipsec0 -> eth0 mut =16260 -> 1500 과 같이 IPSec 인터페이스는 물리적인 인터페이스가 최상위에 위치하고 있어야 한다.

```
ipsec auto(manual) --up name
ipsec look
```

위 명령어로 IPSec을 활성화 시키고, ipsec이 어떻게 동작하는지 볼 수 수있다.

```
--------------------------
10.0.1.0/24 -> 11.0.1.0/24 => tun0x20 · · ·
--------------------------
tun0x200@11.0.0.1 IPv4_Encapsulati · · ·
sp0x203@10.0.0.1 3DES-MD · · ·
esp0x202@11.0.0.1 3DES-MD5-96_Encryp · · ·
Destination  Gateway  Genmask · · ·
11.0.0.0     0.0.0.0    255.255.255.0 · · ·
11.0.1.0     11.0.0.1   255.255.255.0 · · ·
```

또다른 방법으로는 제 3자의 PC를 이용해서 TCPdump[6] 명령어로 SG 사이를 패킷들을 들여다보고 무엇이 어떻게 지나가는지 해석할 수 있다.

```
tcpdump -s 512 -xnp -i eth0 > test
```

와 같이 실행을 하면 test 파일에 보안 게이트들 사이에서 교환되는 패킷들을 살펴볼 수 있다. SG 사이의 보안게이트들 사이에 접속을 하고 키를 교환하고, 인증을 하고 나서 모든 패킷이 IPSec에 의해 암호화되어 나아가는 것을 볼 수 있다.
이밖에 ping test를 이용하여 ping이 전달되는지를 가지고도 IPSec을 테스트 할 수도 있다.

설치시 유의할 사항은 FreeS/WAN 소프트웨어는 패키지를 다운받을 때 커널 버전별로 필요한 패키지 버전이 나눠져 있는데 이것을 알수 없다는 것이다. 따라서 README 파일을 꼼꼼이 읽고 각자의 상황에 맞게 패치해야 한다.

IV. 결론

최근들어 널리 이용되고 있는 인터넷의 전자상거래나 회사 네트워크 등의 개인정보나 회사의 정보에 대한 보안의 문제가 크게 대두되고 있다. 그러나 막상 보안을 위해서 너무 무리하게 보안을 하다보면 속도문제나 사용상의 불편함 등이 더 커 보이기도 한다. 그리고 예전에 이용되던 암호나 DES 등은 암호를 풀고 들어올 수 있어 보안성에 문제가 되고 있다. 여기에 제시되어진 IPSec은 그러한 문제에 진일보한 보안 프로토콜이다. LINUX FreeS/WAN을 이용한 IPSEC VPN 구성의 장점으로는 암호를 위해 드는 비용이 전혀 없다는 점과 강력한 암호 프로토콜인 IPSec을 사용할 수 있다는 점이다. SG와 SG의 간단한 구성뿐만 아니라 라우터 설정 ipsec.conf의 설정에 따라 VPN 네트워크의 확장이 용이하고 암호 알고리즘으로 RSA 뿐만 아니라 3DES, MD5등을 사용할 수 있고 AH와 ESP가 사용 가능하다.[7]

앞으로 연구해야할 과제는 VPN을 실험실 테스트 베드에서 벗어나 집과 학교, 모바일 사용자가 VPN에 접속할 수 있도록 구성하여 적용시킬 것이다.

참고문헌

[1] "Linux FreeS/WAN 1.3 HTML document" http://www.freeswan.org

[2] 이계상 "VPN 보안 기술 표준 해설서 작성에 관한연구", 한국정보보호진흥원, 2001

[3] Naganand Doraswamy, Dan Harkins "IPSec", Prentice Hall, 1999.

[4] Douglas E. Comer " Internetworking with TCP/IP Volume 1" Prentice Hall, 2000

[5] 김 장우, 추 인호 "Using Linux Fifth Edition", (주)교학사, 2000

[6] W. Richard Stevens "TCP/IP Illustrated, Volume 1", ADDISON-WESLEY PUBLISHING COMPANY, 1994

[7] 니츠 편저 "인터넷 보안기술. I " 동서, 2001