논문 03-28-6C-11 한국통신학회논문지 '03-6 Vol.28 No.6C

IPSec VPN 테스트 베드의 구성 및 암호화 식별 도구 개발

준회원 김 윤 희*, 정회원 이 계 상**

Configuration of an IPSec VPN Testbed and Development of an Encryption Verification Tool

Yun-Hee Kim* *Associate Member*, Kye-Sang Lee** *Regular Member*

요 약

IPSec은 네트워크 계층의 IP 패킷 보호를 위한 인터넷 표준 보안 방식으로, IP 패킷의 무결성, 인증 및 기밀성 보안 서비스를 지원할 수 있는 보안 프로토콜과 암호 알고리즘의 집합을 가리킨다. 즉, IPSec은 보안 게이트웨이 간에 접근제어, 비연결형 패킷 무결성, 데이터 근원 인증, 재전송 공격 방지, 기밀성 서비스를 보장하며, IP 계층 뿐만 아니라 상위 계층에 대한 보호를 제공하는 프로토콜이다. 본 연구에서는 리눅스 게이트웨이에 FreeS/WAN IPSec 프로토콜을 설치하여 전형적인 VPN(Virtual Private Network, 가상사설망) 테스트베드를 구성하고, 두 게이트웨이간의 IKE SA 수립과정 중에 교환되는 ISAKMP 메시지를 분석하였다. 또한, VPN 테스트베드의 두 보안 게이트웨이 간에 수립된 IPSec SA의 올바른 동작을 관리자가 효율적으로 확인할 수 있는 편리한 암호화 식별 도구를 개발하였다.

Key Words: Internet Security, IPSec, VPN, FreeS/WAN, Testbed

ABSTRACT

IPsec refers to a standardized set of security protocols and algorithms which can provide the integrity, the authentication and the confidentiality services for IP packets in the Internet. Between two security gateways, IPsec provides the access control, the connectionless integrity, data origin authentication, the anti-replay, and the confidentiality services, not only to the IP layer but also to the upper layers. In this paper, we describe a VPN (Virtual Private Network) testbed configuration using the FreeS/WAN and analyze the ISAKMP messages exchanged between the linux security gateway during the IKE SA setup. Also, we describe our development of an IPSEC encryption verification tool which can be used conveniently by VPN administrators.

I. 서 론

오늘날 기업 활동에서 기업망이 차지하는 비중은 날로 증대되고 있다. 과거 기업망은 임대 전용회선을 사용하는 고비용의 사설망으로 구축되었다. 최근들어 기업망은 공중 ATM망 또는 공중 Frame Relay 망을 근간으로 한 가상 사설망 (VPN: Virtual Private Network)의 형태로 많이 구축되어 왔다. 더욱 최근에는 현재 전 지구적으로 연결이 보편화된 공중 인터넷 망을 기반으로 한 가상 사설망, 즉, IP 기반 VPN의 사용이 확산 되는 추세에 있다. 하지만, 공중 인터넷 망은 보안에 매우 취약하여 기

* 동의대학교 정보통신공학과 석사과정, ** 동의대학교 정보통신공학과 교수
※ 본 연구는 2002년도 동의대학교 교내 연구비 지원에 의해 수행되었습니다.

업망의 기본 요구사항인 보안성을 충족시키지 못한다. IPSec 프로토콜은 공중 인터넷 망을 통해 전달되는 패킷의 보호를 위해 고안되었다. 즉, IPSec 프로토콜은 IP VPN 구축의 필수 요소 기술로 설계되었다. IPSec 프로토콜은 네트워트 계층에서 IP 패킷의 데이터 근원 인증, 데이터 무결성 및 기밀성 서비스를 제공한다[1]. IPSec 프로토콜은 보안 프로토콜과 알고리즘의 집합으로 구성된다. 보안 프로토콜은 AH (Authentication Header)와 ESP (Encapsulating Security Payload) 프로토콜과 IKE (Internet Key Exchange) 프로토콜로 구성되고, 보안 프로토콜은 일련의 암호 알고리즘과 함께 동작한다.

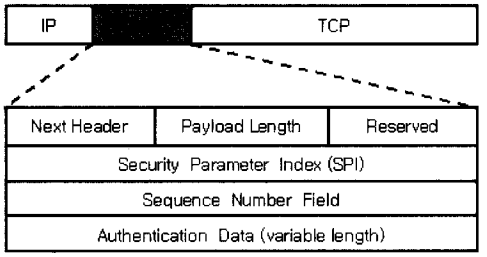
본 연구에서는 IPSec 프로토콜의 공개 구현 소프트웨어인 FreeS/WAN[2]을 이용하여 실험실내에 VPN 테스트베드를 구성하였고, VPN 테스트베드의 두 리눅스 게이트웨이간 보안 통신 수립시의 IKE SA 수립을 위한 ISAKMP 메시지 교환을 분석하였다. 또한, 보안 게이트웨이 간 VPN 터널의 설정 후 IPSec의 동작이 기대된 바와 같이 수행되는지의 여부를 쉽고 효율적으로 판별할 수 있는 IPSec 암호화 식별 도구를 개발하였다. 본 논문은 우선, II장에서, IPSec 보안 프로토콜인 AH와 ESP 및 그 동작 모드를 간략히 기술하고, 키관리 프로토콜인 IKE의 동작을 요약한다. 말미에IKE 버전2 표준화 동향을 약술한다. III장에서는 FreeS/WAN을 이용한 IPSec VPN 테스트베드의 구성을 기술하고, IKE 프로토콜의 동작을 IKE SA 수립시 보안 게이트웨이 간에 교환되는 ISAKMP 메시지를 통해 분석하였으며, 다음 IV장에서는 VPN 게이트웨이 간 설정된 IPSec SA의 동작이 관리자의 의도대로 수행되고 있는지를 판정할 수 있는 IPSec 암호화 식별 도구 프로그램의 개발을 요약한다.

II. IPSec 프로토콜

IPSec은 IP Security의 약자로 IP 패킷에 대한 보안을 제공하며 트래픽의 인증 및 무결성, 기밀성 서비스를 제공한다. 여기서, IP 패킷 보안이란 공중망을 지나가는 사설망 트래픽의 보호를 의미한다. IPSec은 AH(Authentication Header)와ESP(Encapsulating Security Protocol), IKE(Internet Key Exchange) 등의 세 가지의 프로토콜로 구성된다[1]. 이 중, AH나 ESP 프로토콜은 대칭키를 기반으로 한 암호 알고리즘을 사용하고 있는데, 이 프로토콜들을 위한 키의 생성, 유지, 갱신 및 분배에 대한 프로토콜을 필요로 하게 되었다. 이를 위해, IKE 프로토콜은 키 교환 메시지를 이용해 키를 생성 및 교환하게 된다. IPSec 프로토콜은 인터넷 표준 기구인 IETF (Internet Engineering Task Forces)의 IPSEC 워킹그룹에서 표준화되어 왔으며, 주요 기본 프로토콜은 1998년 말 RFC (Request for Comments)로 발간되었으나, IKE 프로토콜은 현재 후속 버전 프로토콜 표준화가 진행 중에 있다. 다음에 IETF에서 표준화된 IPSec 프로토콜과그 동작 모드 및 키관리 프로토콜에 대해 간략히 기술한다

2.1 AH 프로토콜

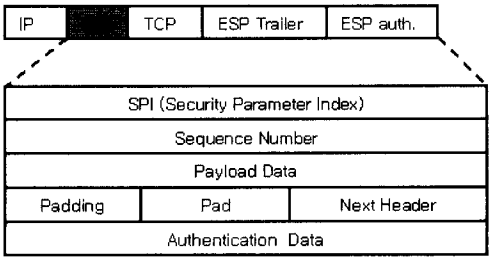
AH 프로토콜은 인증 알고리즘을 이용하여 패킷 무결성 서비스와 데이터 근원 인증 서비스를 제공한다. (그림 1)은 이와 같은 AH의 헤더 포맷을 보여준다[3].



(그림 1) AH 프로토콜 헤더 포맷

2.2 ESP 프로토콜

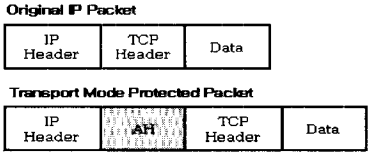
인증 헤더(AH)의 경우와 마찬가지로 ESP (캡슐화 보안 페이로드) 프로토콜은 IP 프로토콜의 보안성을 향상시키기 위해 설계되었다[4]. ESP는 AH가 제공하는 서비스에 데이터 기밀성과 제한된 트래픽 흐름 기밀성의 두 가지 서비스를 추가로 제공한다. (그림 2)는 ESP 프로토콜의 헤더 포맷을 보인다.



(그림 2) ESP 프로토콜 헤더 포맷

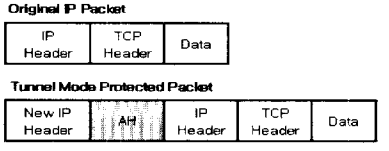
2.3 프로토콜 동작 모드

AH와 ESP 헤더의 위치는 프로토콜의 동작 모드에 따라 달라지는데, 각 프로토콜의 동작 모드에는 트랜스포트 모드와 터널모드가 있다. 먼저, AH 프로토콜의 트랜스포트 모드의 동작을 보면, (그림 3)과 같이 IP 헤더의 뒤, 전송 계층 프로토콜 헤더의 앞, 또는 다른 IPSec 프로토콜 헤더들의 앞에 삽입된다. 트랜스포트 모드로 동작되는 ESP 프로토콜의 경우도 이와 유사하게 ESP 헤더가 AH 헤더 위치에 삽입된다. 물론, ESP 프로토콜의 경우 상위 계층 데이터의 뒤에 ESP 트레일러가 뒤따른다.



(그림 3) 트랜스포트 모드의 AH 헤더의 위치

또한, AH와 ESP 프로토콜은 터널 모드로 동작할 수 있다. (그림 4)는 AH 프로토콜이 터널 모드로 동작될 때 헤더의 위치를 가리켜 준다. AH헤더는 원래의 IP 헤더 앞에 삽입되며, 새로운IP 헤더가 AH 헤더 앞에 삽입된다. 그림에 나타내지는

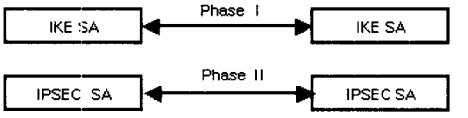


(그림 4) 터널 모드의 AH 헤더의 위치

않았지만, ESP 헤더도 동일한 위치에 삽입된다.

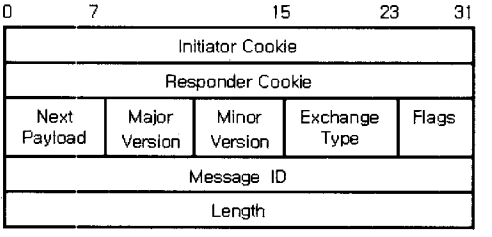
2.4 IKE 프로토콜

IKE는 보안연계(SA: Security Association)의 수립을 위하여 인증된 키 자료를 보호된 방식으로 협상하고 제공하는 프로토콜이다[5]. IKE는 3개의 서로 다른 프로토콜의 관련 부분을 결합한 하이브리드 프로토콜로서, ISAKMP(Internet Security Association and Key Management Protocol)[6], Oakley Key Determination Protocol[7]과 SKEME[8] 프로토콜로 이루어져 있다. ISAKMP 프로토콜에서는 프레임워크, 메시지 포맷 및 phase (단계) 개념을 채용해 왔고, Oakley 프로토콜에서는 두 가지 키 교환 모드, 그리고 SKEME 프로토콜에서는 공개키 암호방식을 가져왔다. IKE는 서비스 거부 공격 및 man-in-the-middle attack을 방지하며 Perfect Forward Secrecy(PFS)를 제공하도록 설계되었다. IKE는 ISAKMP 협상의 단계에서 동작하는 여러 교환 모드를 제공하며, 1단계 교환에서 개시자와 응답자간에 ISAKMP SA를 수립하고, 2단계에서는 AH, ESP와 같은 다른 보안 서비스를 위한 SA를 수립하는데 이용된다. (그림 5)에서 IKE의 단계를 나타내었다.



(그림 5) IKE 동작 단계

1단계에서는 안전하고 인증된 통신 채널을 생성하고, 인증된 키 교환을 수행하는데 사전공유키 (Preshared Keys), 디지털 서명 (Digital Signature), 두 가지의 공개키 암호방식 (Public Key Encryption, Revised Public Key Encryption)의 4가지 방식이 지원되며, 메인 (Main) 모드 또는 어그레시브 (Aggressive) 모드 중 한 모드로 동작함으로써 안전한 IKE SA를 수립한다. 메시지 교환은 모두 ISAKMP 메시지를 이용하며, ISAKMP 헤더 포맷은 (그림 6)과 같다. [6]



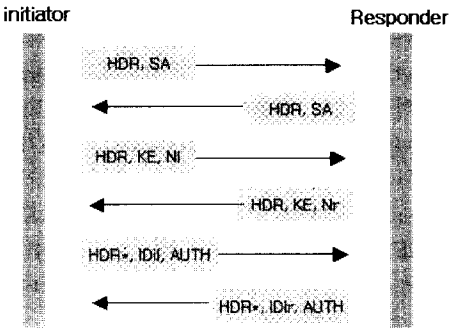
(그림 6) ISAKMP 헤더

Phase I 에서 메인 모드의 경우 메시지 교환 절차는 (그림 7)와 같다. 그림에서 보는 것과 같이 모두 여섯 개의 메시지 교환으로 이루어진다. 첫째와 둘째 메시지 교환에서 양측은IKE SA를 협상하고, 셋째와 넷째 메시지 교환에서는Diffie-Hellman 키공개값을 교환한다. 마지막, 다섯째와 여섯째 메시지 교환에서는 이상

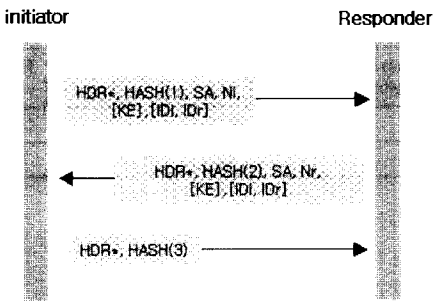
의 통신 내용과 상대방의 신원 (identity)을 인증한다. Phase Ⅰ에 의해 IKE SA가 수립되면, Phase Ⅱ에서는 실제로 보안 통신용 사용자 SA가 생성되는데, 그 중 하나가 IPSec SA이다. 즉, DOI(Domain of Interpretation)에 따라 다른 보안 프로토콜의 SA 생성이 가능하다[9]. 이를 위한 Phase Ⅱ의 퀵 (Quick) 모드 메시지 교환은 (그림 8)과 같다.

2.5 IKE 버전2 프로토콜 표준화 동향

RFC 2409로 표준화 되어 있는 현 IKE 프로토콜은 기능이 복잡하고 표준 규격의 기술(description)도 난해하여 이들을 구현한 이기종 제품간에 상호 연동성이 아직 까지도 확보되지 않고 있다. 이러한 배경에서 IETF IPSEC 워킹그룹은 2001년 8월 새로운 키 관리 프로토콜을 개발키로 하였고, 그동안, 두 프로토콜. 즉, IKE version 2 (IKEv2)와 Just Fast Keying (JFK) 프로토콜이 경합되어 왔으나, 최근 IKEv2로 표준화되어 가고 있는 중이다[10].



(그림 7) Phase Ⅰ에서 메인 모드



(그림 8) Phase Ⅱ에서 퀵 모드

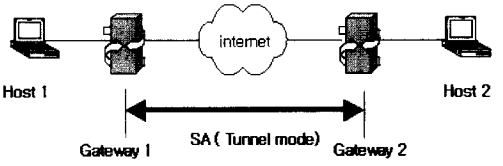
IKEv2는 기존의 IKE 프로토콜의 개념을 그대로 계승하면서 기능을 대폭 축약하였다. IKEv2는 새로운 키관리 프로토콜이 만족시켜야 하는 요구사항을 만족시키도록 설계되었다. IKEv2는 정상 상태에서 4개의 메시지 교환에 의해 통신 쌍방이 인증된 보안 채널을 수립한다. 기존 IKE가 6 개의 메시지 교환으로 이를 달성하는 것에 비해 경제적이다. 또한, IKEv2는 기존 IKE 프로토콜에서 취약했던 서비스 거부 공격 (DOS attack)에 대한 대응력을 갖고 있다. IKEv2는 DOS 공격시 2 개의 메시지 교환을 추가함으로써, 이를 달성한다. 또한, DOS 공격 대응력을 높이기 위해 IKE는 PFS를 타협할 수 있도록 허용하였다. IKEv2는 IKE에서 보다는 훨씬 선택 사항이 줄어들긴 했지만 아직도 암호 협상을 허용한다. 또한, IKEv2는 기존 IKE의 phase I/II 분리 개념을 계승하여 phase I에서 수립된 IKE SA를 후속 IPSEC SA들의 수립과 관리시 제어 채널로 활용할 수 있게 하였다. 인증 방식면에서는 IKEv2는 공개키 기반의 인증서에 의한 인증을 기본으로 하지만, 사전 공유키 사용도 지원한다. 현재 IPSEC WG은 IKEv2 문서를 최종 갱신 중에 있으며 2003년 6월 경 까지 워킹 그룹에서의 논의를 종결하고, 2003년 7월 비엔나 회의 때 까지는 표준 초안을 출간할 계획이다.

Ⅲ. IPSec VPN 테스트 베드의 구성

이 장에서는 당 연구실에서 IPSec 공개 소프트웨어를 사용하여 구축한 IPSec VPN 테스트 베드를 기술하고, 테스트 베드의 두 리눅스 IPSec 게이트웨이간 보안 통신 수립시 교환되는 IKE 메시지를 분석한다.

3.1 FreeS/WAN IPSec을 이용한 VPN 구성

당 연구실에서 구축한 IPSec VPN 구성은 (그림 9)와 같이, 두 대의 호스트와, 두 대의 보안 게이트웨이 그리고 중간의 라우터로 구성되었다. 각 Gateway에는 FreeS/WAN 버전 1.99 IPSec 소프트웨어를 설치하였다[2]. 두 게이트웨이는 본사와 지사의 보안 게이트웨이를 시뮬레이션 한다. 중간의 라우터는 공중 인터넷을 시뮬레이션한다. 각 게이트웨이에 리눅스 커널 버전 2.4.20을 사용하였고, 게이트웨이에는 FreeS/WAN 을 설치하기 위하여 커널



(그림 9) VPN 테스트 베드 구성도

컴파일을 한다. FreeS/WAN의 설치는 커널에 통합 설치되는 옵션과, 모듈로 설치되는 옵션을 갖는다. 당 연구실에서는 커널에 통합 설치 옵션을 선택하였다. IPSec 프로토콜이 성공적으로 커널에 설치되고 나면, IPSec 보안 통신을 위해 양 게이트웨이 간에 IPSec 보안 터널을 설정한다. 이 설정은 ipsec.conf 구성 파일에 새로운 connection을 정의함으로써 이루어진다. 보안 게이트웨이 1의 IP 주소는 10.0.0.1이며, Router는 두 개의 네트워크 카드로 보안 게이트웨이 1과 2를 연결하는데, 주소는 각각 10.0.0.254와 192.0.0.254이고, 보안 게이트웨이 2의 IP 주소는 192.0.0.1로 설정하였다. 구성 파일로 터널이 정의된 후, 이 IPSec SA 터널을 기동 시키기 위하여 (그림 10)와 같이 up 명령을 실행한다. 이 명령은 ipsec.conf에 정의된 내용대로 connection을 수립한다. 터널의 기동은 IKE phase I의 메시지 교환을 유발시키는데, (그림 10)에 IKE phase I의 메인 모드의 상태 천이가 표시된다. 이 그림은 또한, phase II의 quick 모드로 IPSec SA가 수립되는 것도 보여 준다.

```
[root@localhost root]# ipsec auto --up connection
104 "connection" #1: STATE_MAIN_I1: initiate
106 "connection" #1: STATE_MAIN_I2: sent MI2,
expecting MR2
108 "connection" #1: STATE_MAIN_I3: sent MI3,
expecting MR3
004 "connection" #1: STATE_MAIN_I4: ISAKMP SA
established
112 "connection" #2: STATE_QUICK_I1: initiate
004 "connection" #2: STATE_QUICK_I2: sent QI2,
IPsec SA established
```

(그림 10)보안 게이트웨이간 IPSec 터널의 형성

3.2 ISAKMP 메시지 해석

2.4절에서 기술한 바와 같이, IKE phase I의 메인 모드는 6 개의 ISAKMP 메시지를 교환하는 과정을 통해 SA의 협상, 키 교환, 인증을 수행하게 된다. 이 과정은 (그림 10)에서 보인 connection 수립시 일어나며, 이 과정에서 교환되는 ISAKMP 메시지는 중간 라우터에서 tcpdump 프로그램을 실행하여 확인해 볼 수 있다. (그림 11)은 tcpdump 프로그램[11]으로 확인한 메인모드 교환의 ISAKMP 메시지를 보인다. 16진수로 표시된 (그림 11)은 SA 프로포절을 포함하는 ISAKMP 메시지를 표시한다. IKE phase I에서 교환되는 첫 메시지이다. 메시지의 첫 20 옥텟은 IP 헤더이며, 다음 8 옥텟은 UDP 헤더이다, 다음 구간부터 ISAKMP 메시지가 시작된다. 첫 부분은 ISAKMP 헤더로서, 첫 16옥텟은 각각 8 옥텟씩의 initiator cookie와 responder cookie를 구성한다((그림 6) 참조). 그 다음의 코드는 next payload, version, Exchange Type, flag 등이 오고, message ID, message length가 코드화 되어 있다. ISAKMP 헤더 다음에는 프로포절 페이로드와 트랜스폼 페이로드가 따라온다.

```
[root@localhost root]# tcpdump -x -s 200
tcpdump: listening on eth0
19:09:18.057192 10.0.0.1.isakmp > 192.0.0.1.
isakmp: isakmp: phase 1 I ident: [|sa] (DF)
  4500 00cc 0000 4000 4011 701f 0a00 0001
  c000 0001 01f4 01f4 00b8 db2f b845 3901
  1871 fe3c 0000 0000 0000 0000 0110 0200
  0000 0000 0000 00b0 0000 0094 0000 0001
  0000 0001 0000 0088 0001 0004 0300 0020
  ... 중략 ...
```

(그림 11) ISAKMP 메시지의 예(16진수 표시)

IKE phase I 과 II에서의 ISAKMP 메시지 교환을 통해 IPSec SA가 생성되고 나면, 메시지는 암호화 되어 이미 생성된 터널을 통해 전송된다.
(그림 12)은 IKE Phase I 메인 모드에서 소요되는 ISAKMP 메시지 처리 시간을 보인다. 이 결과는 tcpdump의 메시지 수신 시간으로 부터 계산되었다.

단위 : (ms)

	시작시각	완료시각	소요시간
Phase 1 메시지 1	114,050	116,754	2,704
Phase 1 메시지 2	116,754	133,740	16,986
Phase 1 메시지 3	133,740	171,967	38,227
Phase 1 메시지 4	171,967	214,481	42,514
Phase 1 메시지 5	214,481	237,728	23,247
Phase 1 메시지 6	237,728	261,013	계 : 123,678

(그림 12) ISAKMP 메시지 처리 시간

(그림 13)은 패턴 aaaaaaa를 갖는 ping 명령이 실행될 때 중간 라우터에서 tcpdump로 확인해 본 ping 패킷을 나타낸다. 그림에서 볼 수 있듯이, 20 옥텟의 IP 헤더 다음, ESP 헤더가 따라오고, 이 후는 암호화된 메시지로 ping 패킷에 실린 패턴 aaaaaaa를 식별해 낼 수 없다.

Ⅳ. 암호화 식별 도구

IPSEC으로 구현된 VPN에서 패킷이 기대된 바와 같이 실제로 암호화되어 네트워크로 전송되는지에 대한 효율적인 확인은 VPN 관리자 입장에서 매우 중요한 기능이다. 응용 계층은 네트워 크 계층에서

```
[root@localhost root]# tcpdump -x
tcpdump: listening on eth0
20:57:36.195499 192.0.0.1 > 10.0.0.1:
ESP(spi=0x0334fc69,seq=0x6)
        4500 0088 3a0c 0000 3f32 7736 c000 0001
        0a00 0001 0334 fc69 0000 0006 e804 3593
        0c3a 7e4d 08d6 3dca 6b34 5755 75d2 02b1
        8dd3 36ed 4379 2340 4c94 a978 504b aac3
        87c8 2629 fdb6 a90a 9221 5f0f 42e5 491b
        d27a        ... 중략 ...
```

(그림 13) 암호화된 ping 패킷

이루어지는 IPSEC 서비스가 완료된 상위 계층에 존재하므로 자신의 패킷이 암호화되어 송수신되었는지 알 수가 없다. 또한, 앞장에서 기술된 바와 같이, 패킷을 암호화하여 전송하고 수신하는 두 통신 주체 외에, 제3의 노드에서 tcpdump와 같은 네트워크 모니터링 프로그램을 실행하여 패킷의 암호화를 확인하는 방법은 또 다른 노드에서의 모니터링 프로그램 실행이 필요한 점에서 효과적이지 못하다. 특히, 하나의 IPSEC VPN 노드가 많은 수의 노드와 IPSEC 통신을 하고자 하는 경우, 이 방법은 최악의 경우 그 만큼 많은 개수의 제3의 노드에서의 모니터링을 필요하게 되므로 효율적이지 못하다.

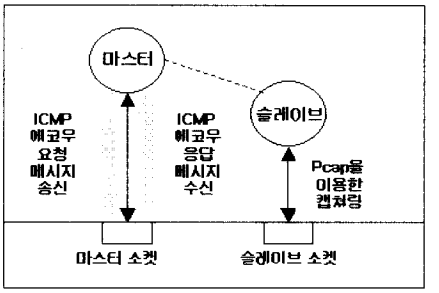
IPSEC VPN 노드에서 패킷 암호화가 과연 기대된 바와 같이 수행되는지를 같은 노드에서 실행되는 프로그램으로 확인할 수 있는 기능은 VPN 관리자에게 편리한 기능이 될 것이다. 즉, 충분한 개수의 테스트 패킷을 발생시키고, 이들 패킷이 기대된 바와 같이 적절히 암호화되는지를 자동으로 확인할 수 있는 프로그램은 VPN 관리자의 관리를 좀 더 편리하게 할 것이다. 이러한 기능은 사용자 입장에서도 유익한 기능이 될 것이다. 보안 통신을 원하는 사용자가 자신의 중요한 트래픽을 전송하기 전에 패킷 암호화 여부를 쉽게 확인할 수 있다면 바람직할 것이기 때문이다.

이 장에서는 이러한 목적으로 개발한 암호화 식별 프로그램의 구조 및 동작을 기술한다. 암호화 식별 프로그램은 자체적으로 테스트 패킷을 발생시키고, 이들이 적절히 암호화되는지를 모니터링하여 그 결과를 사용자에 알려준다. 다음절에서 이 프로그램의 구조, 테스트 패킷의 발생 및 패킷 모니터링 방식에 대해 기술한다.

4.1 프로그램의 구조

프로그램의 구조는 (그림 14)과 같이, 마스터(master) 쓰레드 (thread)와 슬레이브 (slave) 쓰레드로 구성된다. 마스터 쓰레드는 명령 줄로부터 상대 주소를 받아들여, 이 노드로 테스트 패킷을 전송하고 이에 대한 응답을 받아들이며, 슬레이브 쓰레드는 패킷 모니터링을 실행한다. 마스터는 자신의 테스트 패킷 송수신 결과와 패킷 모니터링 결과를 비교하여 패킷 암호화 여부를 판정하고 이를 사용자에게 보고한다. 마스터에 의해 발생되는 테스트 패킷은 ICMP 에코우 요청 메시지이며 수신되는 테스트 패킷은 ICMP 에코우 응답 메시지로 구현하였다. 마스터와 슬레이브는 각각 자신의 소켓을 갖는다.

마스터 프로세스는 로 소켓 (raw socket)을 생성하여 에코우 요청/응답 메시지를 송수신하며, 슬레이브 프로세스는 pcap 라이브러리[13]를 이용하여 데이터 링크 계층의 프레임을 캡처링 한다. 다음 절에서 로 소켓을 이용한 ICMP 패킷의 생성과 pcap 라이브러리를 이용한 패킷 캡처링을 기술한다.



(그림 14) 프로그램 구조

4.2 로 소켓의 생성

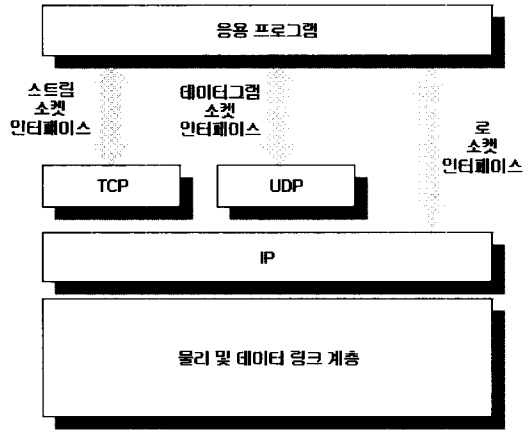
본 논문의 소켓 프로그램 작성에 사용한 소켓의 유형은 로 소켓으로서 통상적인 응용 프로그램에서는 잘 사용되지 않는 소켓 유형이므로 이 절에서 로 소켓의 생성을 간략히 소개한다[12]. 로 소켓은 ICMPv4, IGMPv4 패킷들을 읽고 쓸 수 있게 하며, OSPF 라우팅 프로토콜로 하여금 IP 패킷의 프로토콜 필드를 89로 설정함으로써 TCP나 UDP를 사용하지 않고 IP 프로토콜을 직

접 사용하게 된다. 로 소켓의 동작 위치를 TCP/IP 프로토콜 계층 구조에서 보면, (그림 15)와 같다.

(그림 15)에서 보듯이 로 소켓은 소켓 인터페이스에서 전송 계층 프로토콜을 거치지 않고 바로 IP 계층의 서비스를 이용한다. ICMP 프로토콜을 이용하는 로 소켓의 생성은 다음과 같은 입력인수를 갖는 시스템 콜에 의해 이루어진다.

socket(PF_INET,SOCK_RAW, IPPROTO_ICMP)

위의 소켓 시스템 콜의 인수 중, PF_INET은 protocol family로 인터넷 프로토콜의 사용을 의미하며, SOCK_RAW는 생성되는 소켓의 유형이 로 소켓임을, IPPROTO_ICMP는 사용되는 프로토콜이 ICMP임을 의미한다. 이렇게 생성되는 로 소켓은 루트 사용자만이 사용 권한을 갖는다.



(그림 15) 로 소켓의 동작 위치

4.3 ICMP 메시지의 송신과 수신

마스터 쓰레드는 ICMP 메시지의 송신과 수신을 담당한다. 마스터 쓰레드에서 ICMP 에코우 요청 메시지를 전송하기 위해, 다음과 같은 sendto 시스템 콜이 호출된다.

sendto(sockfd, sendbuf, h_len, 0,
pointer_host_info -> h_add,
pointer_host_info -> h_length)

둘째 인수와 셋째 인수는 전송될 ICMP 메시지의 위치와 길이를 가리키며, 마지막 두 인수는 상대방 주소와 그 길이를 표시한다. 유사하게, 로 소켓을 통한 에코우 응답 메시지의 수신에는 recvfrom 소켓 시스템 콜을 이용한다.

마스터 프로세스는 사용자가 지정한 개수의 에코우 요청 메시지와 응답 메시지를 수신한 후, 이 결과를 슬레이브 프로세스에서 캡쳐한 패킷 분석 결과와 비교하여 암호화 여부를 판정한다. 즉, 마스터 프로세스에서 송수신한 ICMP 메시지의 개수와 슬레이브 프로세스에서 캡쳐한 AH나 ESP로 암호화된 패킷의 개수를 비교하여 일치 여부를 비교한다.

4.4 패킷 캡쳐

슬레이브 프로세스는 네트워크 카드의 데이터 링크 계층에서의 패킷 캡쳐를 수행한다. 데이터 링크 계층에서의 패킷 캡쳐는 리눅스 표준 패킷 캡쳐 라이브러리인 pcap 라이브러리[12]를 사용하여 수행하였다.

슬레이브 프로세서는 pcap 라이브러리를 이용하여 네트워크 카드에서 송수신되는 모든 패킷을 캡쳐하고, 송수신 주소와 프로토콜 식별자를 비교하여, 이 패킷들이 마스터 프로세스에서 발생되어 송신되는 패킷들이 적절한 보안 프로토콜로 암호화된 패킷들인지를 확인한다. 즉, 사용자가 원하는 보안 프로토콜이 사용되었는지와 프로토콜 적용 모드가 일치하는지를 확인한다.

슬레이브 프로세스는 시스템의 모든 인터페이스에 한 개씩 적용된다. 즉, 여러 네트워크 카드를 갖는 보안 게이트웨이의 경우, 통상 인터페이스는 복수개로 이러한 경우 하나의 인터페이스에 하나의 슬레이브 프로세스가 생성되어 패킷 캡쳐와 분석이 실행된다. 멀티 인터페이스의 경우, 소스 주소의 결정은 라우팅 테이블을 참조하여 결정되어야 하며, 각 인터페이스가 패킷 캡쳐링을 지원하는지에 대한 확인이 선행되어야 한다. 결국, 앞의 기술과 같이, 마스터 쓰레드는 ICMP 메시지를 생성하여 발송하고, 자식 쓰레드들은 해당 호스트의 모든 네트워크 인터페이스를 통과하는 모든 패킷을 캡쳐하여 AH 또는 ESP 패킷 여부를 판별하고, 이를 마스터 쓰레드에서 발생시킨 패킷들과 비교하여 기대된 바와 같이 패킷이 과연 암호화되었는지의 여부를 판단하게 된다.

4.5 암호화 여부의 판별

캡쳐된 패킷의 암호화 여부 판별은 다음과 같이 수행된다. 트랜스포트 모드의 적용시, AH 또는 ESP 패킷의 확인은 캡쳐한 패킷의 IP 헤더와 상위 프로토콜 헤더, 즉, TCP 또는 UDP 헤더 사이의 AH 또는 ESP 헤더의 존재 여부로 판단하게 된다. 터널 모드의 경우는 IP 터널을 위해 새로 부착된 IP 헤더와 원 IP 패킷 사이에 위치한 AH 또는

ESP 헤더의 존재 여부로 암호화 여부를 판단하게 된다. 보안 헤더의 존재는 다시 IP 헤더의 protocol 필드를 이용하여 확인한다.

즉, 캡쳐된 IP 패킷의 암호화 여부 판별은 IP 헤더의 protocol 필드의 코드 값으로 식별한다. ESP와 AH 프로토콜의 경우 이 코드 값이 각각 십진수 값으로 50과 51으로 정해져 있다. 이 코드 값은 RFC 1700에 명시되어 있다. 본 프로그램에서는 함수를 이용해 IP 패킷의 암호화 여부를 판단하였다. 이를 위해 먼저 IP 헤더에 대한 구조체를 선언하고, protocol 멤버변수의 값이 ESP 또는 AH 프로토콜 코드 값과 일치하는지를 판단한다. 캡쳐된 패킷의 protocol 필드 값은 사용자가 이 프로그램을 구동시 입력 인수로 제시한 프로토콜 코드 값과 비교된다. 사용자는 IPSEC 설정이 완료된 후 자기가 기대하는 보안 프로토콜 값을 명령어의 입력 인수로 제시한다.

4.6 실행 결과

```
[root@localhost root]#./crypton eth0 ESP IP_addr
10.0.0.1 → 192.0.0.1
Raw socket Create...
Detect ESP header! ESP protocol code value is 50.
IP packets are encrypted as expected!
```

(그림 16) 프로그램 실행결과

본 프로그램의 명령은 crypton이며, 총 4개의 인수를 가진다. 이 인수들은 명령어, 네트워크 인터페이스 장치(기본적으로 eth0으로 설정되어 있음), 프로토콜 코드 값, 그리고 상대방 주소로서 사용자로부터 입력된다. 컴파일 후 실행 모습은 (그림 16)과 같다.

V. 결 론

본 논문은 당 연구실에서 수행한 IPSec VPN의 구성 및 개발된 암호화 식별 도구 프로그램을 기술하였다. 본 연구에서 개발된 IPSec 암호화 식별 도구는 불특정 다수의 많은 VPN connection을 유지 관리해야 하는 관리자에 편리한 관리 기능을 제공하게 될 것이다. IPSec VPN 구축 확산의 가장 큰 걸림돌은 현IKE 프로토콜의 복잡성에 기인한 상호 연동성 부족이었다. 다행히, 최근 기존 IKE 프로토콜보다는 훨씬 간단한 IKEv2 프로토콜이 곧 표준화 될 전망이다[14]. IKEv2 프로토콜은 멀티벤더로 구축될 IPSec VPN의 상호연동성을 크게 향상시킬 것으로 보이며, 또한, 이동단말기와 같은 소용량 컴퓨팅 능력을 갖는 소형 단말기에도 IPSec 프로토콜의 설치를 가능하게 하여 IPSec VPN의 모바일 원격 접속 서비스를 부축일 것으로 전망된다. 현재 당 연구실에 구축된 IPSec VPN 테스트 베드는 기존 IKE 프로토콜을 기반으로 하고 있으나, 향후 IKEv2 프로토콜을 기반으로 교체할 예정이다. 향후의 테스트 베드 실험에서는 ISAKMP 메시지의 생성에 필요한 암호화 시간 측정 및 IPSec SA 생성 까지의 제반 성능 등에 대한 연구를 수행할 예정이다.

참 고 문 헌

[1] S. Kent, et. al., "Security architecture for the Internet Protocol," RFC 2401, IETF. 1998. 11

[2] www.freeswan.org

[3] S. Kent, et. al., "IP Authentication Header," RFC2402, IETF, 1998.11

[4] S. Kent, et. al., "Encapsulting Security Payload," RFC2406, IETF, 1998.11

[5] D. Harkins, et. al., "The Internet Key Exchange," RFC2409, IETF, 1998. 11

[6] D. Maughan, et. al., " Internet Security Association and Key Management Protocol," RFC2408, IETF, 1998.11

[7] H. Orman, et. al., " The Oakley Key Determination Protocol," RFC2412, IETF, 1998.11

[8] H. Krawczyk, "SKEME: A versatile secure key exchange mechanism for internet," IEEE, Symposium on Network and Distributed Systems Security, 1996

[9] D. piper, "The Internet IP Security Domain of Interpretation for ISAKMP." RFC2407, IETF, 1998.11

[10] C. Kaufman, "Internet Key Exchange(IKEv2) Protocol," draft-ietf-ipsec-ikev2-08.txt, 2003.6

[11] W. Richard Stevens, "TCP/IP Illustrated, Volume 1," Addison Wesley, 1994

[12] Network Research Group at the Lawrence Berkeley National Laboratory, " LIBPCAP 04:

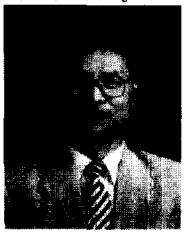
Packet Capture Library," ftp.ee.lbl.gov, 1997
[13] W. R. Stevens, "UNIX Network Programming, Volume 1," Prentice Hall, 1997.7
[14]http://www.ietf.org/html.charters/ipsec-charter.html

김 윤 희(Yun-Hee Kim) 준회원

2002년 2월 : 동의대학교 전자통신공학과 졸업
2002년 3월~ 현재 : 동의대학교 정보통신공학과 석사과정

<주관심분야> 네트워크 보안, 윈도우 프로그래밍, IPv6 프로토콜, 리눅스 보안

이 계 상(Kye-Sang Lee) 정회원



1979.2: 서울대학교 공대(공학사)
1981.2: 서울대학교 대학원 전자공학과 (공학석사)
1997.2: KAIST 전기전자공학과 (공학박사)
1981.10 ~ 1997.8: ETRI 재직
1997.9 ~ 현재: 동의대학교 정보통신공학과 재직중

<주관심분야> 인터넷 구조, 프로토콜, 보안, 표준