通信情報保護學會學會誌
第 9 券 第 4 號, 1999. 12

VPN - swIPe

천 광철*

요 약

VPN 또는 침입차단시스템 간 비밀 통신의 기본 구조로 사용되는 IP 네트워크 보안(IP network-layer security) 프로토콜인 swIPe 프로토콜에 대하여 설명한다. 우선 swIPe 프로토콜에 대하여 분석하고, 침입차단시스템에서 이를 실제로 적용하는 방법에 대하여 서술한다. swIPe 프로토콜에서는 swIPe 개요, 개념적 구조, swIPe를 이용한 통신 절차, 패킷의 포맷, encryption과 decryption 프로세스 등을 서술하며, 실제 적용에서는 Host-to-Host 터널링, 침입차단시스템간 비밀통신 절차, swIPe driver에 대하여 설명한다.

Ⅰ. 서 론

SP3(Security Protocol 3, NSA & NIST), NLSP(Network Layer Security Protocol, ISO) 등과 같은 네트워크계층 보안 프로토콜들은 특정한 보안정책에만 적용 가능하며, 불필요한 복잡성으로 인해 효과적인 구현이 어렵다.[1]

또한, Kerberos, ADP(Authenticated Datagram Protocol) 등과 같은 비 네트워크 계층 프로토콜들은 일관적인 보안정책의 적용이 어렵고, 응용프로그램에 수정을 가해야 하며, 통신 경로 상에 존재하는 라우터나 게이트웨이 등에 대한 보안기능(Intermediate-hop security)을 수행하기가 어렵다.[2]

이러한 문제를 해결하기 위해 개발된 네트워크 계층 보안 프로토콜의 하나로서, swIPe(Software IP encryption)는 최소한의 구조만을 규정하며 또한 구현이 용이하다.

Ⅱ. swIPe(Software IP encryption)

1. swIPe 개요

swIPe는 IP(Internet Protocol) 네트워크 계층의 보안 기능을 제공하기 위한 프로토콜로서, IP의 구조를 그대로 유지하면서 필요한 보안 기능만을 추가함으로써 IP의 기능을 보완하도록 설계되었다.

swIPe 프로토콜은 IP 계층에서 생성된 IP 패킷(Inner IP 패킷)을 새로운 IP 패킷으로 encapsulation 하거나 또는 그렇게 encapsulation된 패킷을 de-capsulation 하여 원래의 IP 패킷(Inner IP 패킷)을 복원하는 과정에 대한 규정이다. 또한 encapsulation된 IP 패킷, 즉 swIPe 패킷의 포맷에 대해서도 정의하고 있다. 〈그림 2 swIPe 패킷 포맷 참조〉

즉, 송신 시에는 IP 패킷을 새로운 IP 프로토콜 타입 IPPROTO_IPIP (프로토콜 number 94)의 IP 패킷 또는 swIPe 패킷으로 encapsulation하고, 수신시에는 수신된 swIPe 패킷에서 원래의 IP 패킷 또는 Inner IP 패킷을 추출해내는 de-capsulation 과정을 수행하게 함으로써, IP의 보안기능을 제공

* (주)시큐어소프트 보안연구소 (kc1000@securesoft.co.kr)

한다.

Encapsulation 과정에서 패킷에 대한 인증 정보 및 패킷 sequence number를 생성하여 암호화를 수행한다. 반면 de-capsulation 과정에서는 복호화를 한 후, 인증 정보와 패킷 sequence number를 검사하게 한다. 이러한 과정을 통해 비밀성, 무결성 및 Replay attack 방지 등의 보안 서비스 제공이 가능해진다.

swIPe 프로토콜을 사용함으로써 얻을 수 있는 이점은 다음과 같다.

1) IP 패킷이 새로운 IP 패킷에 encapsulated되므로, swIPe 프로토콜을 사용하지 않는 네트워크를 경유할 수 있다.
2) 트랜스포트계층(Transport layer) 및 그 상위 계층의 프로토콜들을 수정할 필요가 없으며, 따라서 기존의 네트워크 기반구조 및 인터페이스를 사용할 수 있다.
3) IP 패킷을 처리하는 모든 곳(예: 호스트, 라우터 등)에서 encapsulation 및 de-capsulation을 처리할 수 있으므로, 단대단 통신에 있어서의 보안 및 intermediate-hop에서의 보안 기능을 제공할 수 있다.
4) IP 패킷 전체를 암호화하여 전송하는 터널 모드(tunnel mode)를 지원한다.
5) 다양한 보안정책의 구현이 가능하다.
6) 다양한 유닉스 플랫폼에서 구현이 가능하다.

2 swIPe의 개념적 구조

swIPe는 개념적으로 Policy 엔진과 Security processing 엔진으로 구성된다 — 실제로 침입차단시스템에서 암호모듈의 swIPe 드라이버 구현 시에는, 암호알고리즘 모듈 및 해쉬알고리즘 모듈까지 포함될 수 있다.

2.1 Policy 엔진

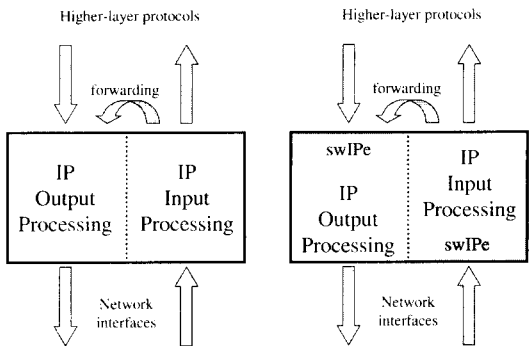
송신될 IP 패킷이 swIPe 프로세싱을 거쳐야 하는지 검사하며, 또한 수신된 IP 패킷을 상위 프로토콜 계층으로 전송할 것인지의 여부와 수신된 IP 패킷에 적용해야할 swIPe 프로세싱의 종류를 확인한다. 또한 보안정책에 목적지/출발지 주소별로 정의된 암호화 키를 참조한다.

2.2 Security processing 엔진

Policy 엔진에서 패킷에 적용될 swIPe 프로세싱의 종류를 검사한 결과에 따라, 패킷별로 인증, 무결성 검사 또는 암/복호화 프로세싱을 한다. 이 과정에서 암호 알고리즘과 해쉬 알고리즘을 사용한다.

swIPe의 Security processing은 암호 알고리즘 및 해쉬 알고리즘에 독립적이다. 이는, swIPe 프로토콜이 특정 알고리즘의 사용을 규정하지 않으며, 따라서 암호 알고리즘 및 해쉬 알고리즘으로 어떠한 것을 사용하여도 무방하다는 의미이다.

3 swIPe를 이용한 통신 절차



〔그림 1〕 Inbound, Outbound Processing

swIPe 프로토콜을 사용하여 통신하는 과정을 기술하면 다음과 같다.

3.1 송신측 — Outbound 프로세싱 (그림 1(b)의 IP Output Processing)

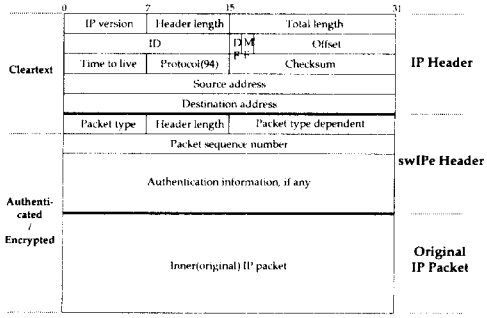
① Policy 엔진이 IP 패킷(Inner IP 패킷)의 출발지와 도착지 주소에 따라 해당 보안정책을 확인

하여, swIPe 프로세싱을 거치지 않아도 되면 IP 패킷을 그대로 전송하고, 그렇지 않으면 다음 과정을 거친다.
② Policy 엔진을 사용하여 보안정책에 정의된 암호화 키를 읽어온다.
③ swIPe 헤더를 생성한다.
④ swIPe 헤더의 패킷 sequence number 필드를 채운다.
⑤ Security processing 엔진에서 위의 키를 사용하여, swIPe 헤더와 Inner IP 패킷에 대해 적절한 인증 또는 암호화 작업을 수행한다.
⑥ 새로운 IP 패킷(= encapsulated IP 패킷 = swIPe 패킷)을 생성하여 전송한다.

3.2 수신측 — Inbound 프로세싱 (그림 1(b)의 IP Input Processing)

① Policy 엔진이 수신된 IP 패킷이 swIPe 패킷인지 확인하여, 그렇지 않으면 보안정책에 따라 상위 프로토콜 계층으로 전송할 것인지 아니면 수신을 거부할 것인지 결정한다. swIPe 패킷이 수신된 경우에는 다음의 과정을 거친다.
② De-capsulation: Security processing 엔진이 복호화, 인증/무결성 검사 및 패킷 sequence number 검사를 수행하고, 이상이 있을 경우 예외 처리를 한다.
③ 이상이 없을 경우에는 Policy 엔진이 원래의 IP 패킷(Inner IP 패킷)의 헤더를 검사하여 보안정책에 따라 처리를 한다.

4. swIPe 패킷 포맷



〔그림 2〕 swIPe Packet Format

IP 패킷은 IP-inside-IP(IPIP) 프로토콜의 확장 필드를 사용하여 swIPe 패킷으로 encapsulated된다. swIPe 패킷은 자체로서 IP 패킷이며, IP 프로토콜 필드가 94(IPIP_SWIPE)로 고정된 IP 헤더를 갖는다. swIPe 패킷의 payload는 swIPe 헤더를 포함하며, 여기에 수신측의 패킷 처리에 필요한 정보와 인증정보 및 replay attack을 감지하기 위한 패킷 순차번호(Packet sequence number) 등이 저장된다. 또한 원래의 IP 패킷이 swIPe 패킷의 payload에 포함된다.

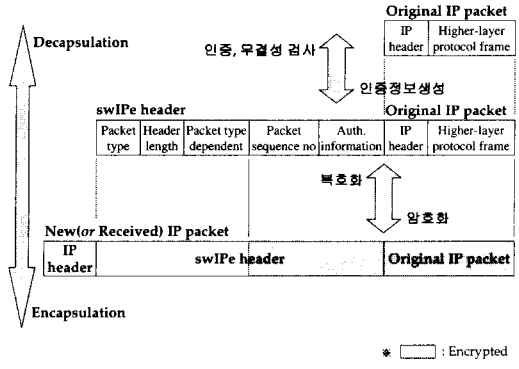
swIPe 헤더의 인증정보와 패킷 순차번호, 원래의 IP 패킷은 보안정책에 따라 암호화될 수 있다.

5. 패킷 encryption/decryption 프로세스

swIPe 드라이버는 IP 네트워크계층(IP network-layer)의 보안 프로토콜인 swIPe(Software IP encryption)를 구현한 것으로서, 송신 측에서는 상위계층에서 생성된 IP 패킷에 swIPe 헤더를 붙이고, 그 결과를 새로운 IP 패킷으로 encapsulation한다.

즉, swIPe 드라이버의 입력은 IP 패킷이며, 출력은 그 IP 패킷이 encapsulated된 새로운 IP 패킷(swIPe 패킷)이다.

반면, 수신 측에서는 수신된 IP 패킷(swIPe 패킷)을 복호화화고, 인증과정을 거쳐 decapsulation한다. 이 경우, 암호모듈의 입력은 swIPe 패킷이며, 출력은 원래의 IP 패킷(Inner IP 패킷)이 된다. 다음의 〔그림 3〕은 송신 및 수신시의 swIPe 드라이버의 입출력을 보여준다

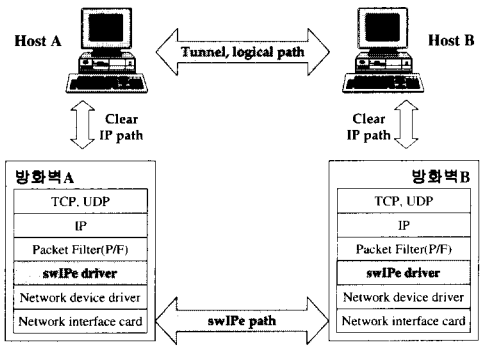


〔그림 3〕 Encapsulation/De-capsulation Process

Ⅲ. swIPe의 실제 적용

1. Host-to-host(End-to-end) 비밀통신 — 터널링(Tunneling)

다음 〔그림 4〕는 침입차단시스템을 통한 Host-to-host 터널링을 형성하는 과정을 나타낸다.



〔그림 4〕 침입차단시스템을 통한 Host-to Host 터널링

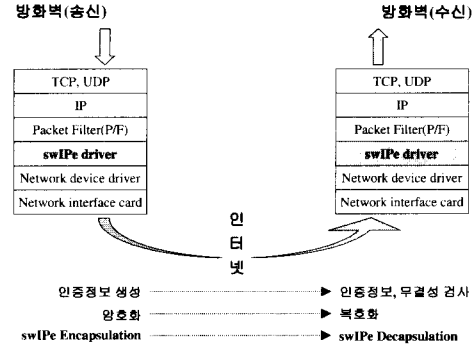
위의 그림에서 Host A와 침입차단시스템 A 간, Host B와 침입차단시스템 B간의 통신에서는 일반적인 IP 프로토콜을 이용하며, 침입차단시스템 A와 침입차단시스템 B 사이에서는 swIPe 프로토콜을 사용한다.

Host A와 Host B 사이의 물리적인 실제 경로는 Host A ⇔ 침입차단시스템 A ⇔ 침입차단시스템 B ⇔ Host B이지만, 논리적으로는 침입차단시스템 A와 침입차단시스템 B 사이의 swIPe 프로토콜로 인하여, swIPe tunnel을 사용하는 효과가 생긴다.

이로서 Host A와 Host B는 자체로 보안 메커니즘을 구현하지 않더라도, 비밀통신을 할 수 있게 된다.

2. 침입차단시스템간 비밀통신 절차

침입차단시스템간의 비밀통신 절차는 다음 그림 4과 같다.



〔그림 5〕 침입차단시스템간 비밀통신

송신측 침입차단시스템의 IP 계층에서 생성된 IP 패킷은, 두 침입차단시스템간 통신에 있어서의 보안정책에 따라 swIPe 패킷으로 encapsulated되며, 이 swIPe 패킷이 수신측

침입차단시스템으로 전송된다. 수신측 침입차단시스템에서는 수신된 swIPe 패킷을 decapsulation한 후, 원래의 IP 패킷을 보안정책에 따라 처리한다.

위의 [그림 5]에서 인증 정보 생성/검사, 암호화/복호화 및 encapsulation/decapsulation 과정은 각각 송신측, 수신측 침입차단시스템의 swIPe 드라이버에서 수행된다.

다음은 각각의 과정에 대한 상세 설명이다.

2.1 인증정보의 생성 / 인증정보, 무결성 검사

(1) 인증정보의 생성(송신측)

① 패킷 필터는 IP 계층에서 생성된 IP 패킷의 헤더를 조사하여 보안정책 테이블을 참조한다.
② IP 패킷에 대한 인증을 거쳐야 할 경우, swIPe 드라이버는 IP 패킷에 대한 MD5 해쉬값을 계산하여 swIPe 헤더의 Authentication Information 필드에 저장한다.
③ swIPe 패킷을 수신측 침입차단시스템에 전송한다.

(2) 인증 및 무결성 검사(수신측)

① 수신된 swIPe 패킷을 decapsulation한다.
② 원래의 IP 패킷(Inner IP 패킷)에 대한 MD5 해쉬값을 계산한다.
③ 위의 과정에서 계산된 해쉬값과 수신된 swIPe 패킷의 swIPe 헤더에 포함된 Authentication Information 필드값과 비교한다.
④ 비교 결과가 같다면, 데이터의 무결성이 입증된다. 또한, 수신된 인증정보(Authentication Information 필드)는 수신처의 정보(주소)를 포함한 상태로 암호화되었으므로, 데이터 인증도 입증된다.

2.2 암호화/복호화

(1) 암호화(송신측)

① 패킷 필터는 IP 계층에서 생성된 IP 패킷의 헤더를 조사하여 보안정책 테이블을 참조한다.
② swIPe 프로세싱을 거쳐야 할 경우, swIPe 드라이버는 두 침입차단시스템간에 공유된 키를 보안정책 테이블에서 참조한다.
③ swIPe 드라이버는 위의 과정에서 참조한 키와 암호알고리즘을 사용하여, IP 패킷과 swIPe 헤더의 일부를 암호화한다.
④ swIPe 패킷을 수신측 침입차단시스템에 전송한다.

(2) 복호화(수신측)

① 수신된 swIPe 패킷을 decapsulation한다.
② swIPe 드라이버는 두 침입차단시스템간에 공유된 키를 보안정책 테이블에서 참조한다.
③ swIPe 드라이버는 위의 과정에서 참조한 키와 암호알고리즘을 사용하여, 복호화를 한다.

2.3 Encapsulation / Decapsulation

(1) Encapsulation(송신측)

① 패킷 필터는 IP 계층에서 생성된 IP 패킷의 헤더를 조사하여 보안정책 테이블을 참조한다.
② swIPe 프로세싱을 거쳐야 할 경우, swIPe 드라이버는 swIPe 헤더를 생성한다. 이 때, IP 패킷에 대한 MD5 해쉬값이 Authentication Information 필드에 저장된다.
③ swIPe 드라이버는 두 침입차단시스템간에 공유된 키를 사용하여, swIPe 헤더의 일부와 IP 패킷을 암호화한다.
④ swIPe payload(swIPe 헤더 + IP 패킷)을 생성하고, 여기에 새로운 IP 헤더를

첨가한다.
⑤ swIPe 패킷을 수신측 침입차단시스템에 전송한다.

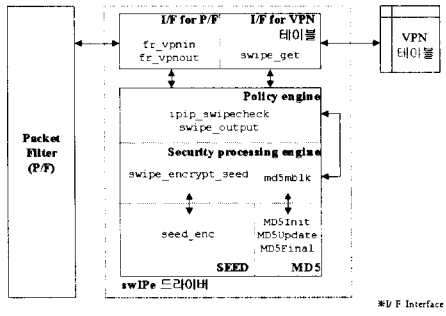
(2) Decapsulation(수신측)

① 수신된 swIPe 패킷의 IP 헤더를 떼어낸다(swIPe payload 추출).
② 두 침입차단시스템간에 공유된 키를 사용하여 암호화된 부분을 복호화한다.
③ swIPe 헤더를 떼어내고, 원래의 IP 패킷(Inner IP 패킷)을 추출한다.

3. swIPe Driver

swIPe 드라이버는 Policy 엔진, Security processing 엔진, 데이터 암호화를 위한 암호 알고리즘 모듈 및 MD5 모듈(데이터 무결성 검사)로 구성되며, 패킷필터 및 VPN 정책 테이블과 swIPe 드라이버 사이의 입출력은 인터페이스 모듈을 통해서 이루어진다.

침입차단시스템에서 구현할 때 swIPe 드라이버의 개념적인 구성과 그 인터페이스를 도시하면 다음 [그림 6]과 같다.



[그림 6] swIPe 드라이버의 구성 및 인터페이스

[그림 6]에서와 같이, swIPe 드라이버와 패킷필터 및 VPN 정책테이블과의 인터페이스는 인터페이스 모듈의 함수호출을 통해서만 이루어진다. 또한 swIPe 드라이버 내부의 구성요소 사이에서도 함수호출을 통한 인터페이스가 이루어진다. 각각의 기능을 상세하게 설명하면 다음과 같다.

구성 요소	관련 함수	기능
Policy 엔진	swipe_output	인자로 받은 키를 확인하여 적용해야 할 보안정책을 검사한다. 인증 정보 필드 및 Packet Sequence Number를 포함하는 swIPe 헤더를 생성한다. 적용할 보안정책에 따라 인증 정보 생성 또는 암호화를 위해 Security Processing 엔진을 호출한다. IP 패킷을 프로토콜이 IPPROTO_IPIP (프로토콜 number 94)인 IP 패킷(swIPe 패킷)으로 encapsulation
	ipip_swipecheck	swipe_get 함수를 호출하여 키를 참조하고, 패킷에 적용된 보안정책에 따라 무결성 검사 또는 복호화를 위해 Security Processing 엔진을 호출한다. 무결성 및 Packet Sequence number를 검사한다. swIPe 패킷에 포한된 Inner IP 패킷을 추출하기 위한 swIPe 패킷 de-capsulation을 수행한다.

구성 요소	관련 함수	기능
Security processing 엔진	swipe_encrypt_seed	암호화/복호화에 동일한 함수 사용 암호 알고리즘 모듈(seed_enc 함수)을 호출
	md5mblk	인자로 받은 패킷(swIPe 헤더 + Inner IP 패킷)dp 대한 해쉬값을 계산하기 위한 해쉬 알고리즘 모듈 호출

구성 요소	관련 함수	기능
암호 알고리즘 모듈	seed_enc	swIPe 헤더의 일부(인증 정보 및 Packet sequence number 필드)와 Inner IP 패킷을 암호화/복호화
해쉬 알고리즘 모듈	MD5Init, MD5Update, MD5Final	인자로 받은 패킷(swIPe 헤더 + Inner IP 패킷)에 대한 해쉬값을 계산

4. swIPe 패킷 Fragmentation/Reassembly

일반적으로 IP 계층에서 physical 인터페이스의 MTU(Maximum Transfer Unit)에 맞게 IP 패킷을 분할하여 내려보낸다. 이를 침입차단시스템의 암호모듈이 수신하여 보안정책에 따라 swIPe 헤더(최대 24바이트)를 붙이고, 그 결과를 새로운 IP헤더(20바이트)로 encapsulation하므로, 패킷의 크기는 최소 28바이트에서 최대 44바이트까지 증가할 수 있다. 크기가 증가한 패킷을 전송 시에 fragmentation 해주고, 수신시에는 reassembly 처리를 해줄 필요가 있다.

Ⅳ. 결론

네트워크 계층에서 보안을 적용하는데 있어서 swIPe 프로토콜은 최소한의 노력으로 다양한 플랫폼에 적용 가능하다는 장점이 있다. 다만 프로토콜 자체가 최소한의 구조만을 정의한 것이기 때문에 이것만으로는 완벽한 VPN을 구현하기는 어렵다. 특히 swIPe 프로토콜을 사용하여 개발할 때에는 안전한 키 관리 방법이 필요하다.

참 고 문 헌

[1] [AP87] M. Abrams and H. Podell, "Computer and Network Security", Los Alamitos, CA: IEEE Computer Society Press, 1987

[2] [IB] John Ioannidis and Matt Blaze. "The Architecture and Implementation of Network-Layer Security Under Unix", From AT&TBell Laboratories.

著者紹介

천 광 철 (Kwang-cheol Cheon)

1996년 2월 : 서울대 국사학과 졸업(학사)
1996~1998년 : (주)아이에스에스 보안연구소
1998~1999년: (주)시큐어소프트 보안연구소
1999년 11월 현재 : (주)시큐어소프트 보안연구소 개발실 선임연구원

<관심분야> 정보보호, 침입차단시스템, VPN