2001년 한국정보처리학회 춘계 학술발표논문집 제8권 제1호

# IPsec 지원을 위한 보안 정책 시스템에서의 연관성 제거 알고리즘 구현

박명찬*, 신동명*, 인소란**, 최용락*
*대전대학교 컴퓨터 공학과
**(주)니츠
e-mail:mcpark@zeus.taejon.ac.kr

# Design and Implementation of Policy De-correlation Algorithm on The Security Policy System for IPsec support.

Myung-Chan Park*, Dong-Myung Shin*, So-Ran Ine**,
Yong-Rak Choi*
*Dept of Computer Engineering, Daejeon University
**Nitz, Inc

요약

IPsec 프로토콜에서는 각 보안 영역에 따라 각기 다른 보안 정책을 내부적으로 정의하여 사용한다. 각 보안 정책들 간에는 상호 연관성을 가진 정책들이 존재할 수 있다. 이때 상호 연관성을 가진 정책들로 인하여 정책 요청에 의한 정책 협상과정에서 뜻하지 않은 정책정보를 적용할 수 있다. 본 논문에서는 이와 같이 각 보안 정책들간에 연관성을 가진 정책으로 인하여 발생할 수 있는 문제점을 분석하여, 이들 정책들간의 연관성을 제거함으로써 신뢰성 높은 정책정보를 제공하려고 한다.

## 1. 서론

최근 인터넷의 폭발적인 성장은 인터넷을 통한 다양한 형태의 서비스를 제공할 수 있게 되었다. 또한, 광역적인 인터넷 환경을 통해 B2B, B2C등의 상업적 서비스가 증가하게 되었다. 이러한 다양한 서비스의 제공을 위해서는 이들 서비스간의 안전한 데이터 전송이 필수적인 요소로 떠올랐다. 현재의 IPv4망은 인터넷의 성장으로 인한 IP 부족 현상이 예상되어지고, 데이터 전송에 대한 안전성도 문제가 되고 있다. 이를 위하여 IETF(The Internet Engineering Task Force)의 IP Security 작업반에서는 기존의 망(IPv4) 뿐 아니라 새로 정의된 IPv6망에 대해 보다 체계적인 보안 서비스를 정의하고 있다. IPv6는 주소지정 능력 및 데이터 보안을 지원하는 확장헤더를 정의하여 보다 안전성있는 데이터 전송을 제공한다[1].

현재의 IPsec 프로토콜은 각각의 보안 영역들간에 독자적인 보안 정책(Security Policy)을 정의하여 사용할 수 있으며, 이를 이용하여 안전한 데이터 전송 서비스를 제공한다. 그러나 IPsec에서는 독자적인 보안 정책으로 인하여 몇 가지 문제점이 발생하였다. 먼저 보안 영역들간의 정책 요구사항이 서로 달라 데이터의 전송 상에 패킷이 목적지까지 전달되지 않는 경우를 발생시킬 수 있다. 이와 같이 패킷의 전송에 영향을 미치는 경우에 대하여 보안 영역들간에 정책 정보를 협상할 필요성이 있다. 또한 현재의 복잡한 토폴로지로 인하여 IP 패킷이 목적지까지 도달하는 데 다양한 경로가 존재할 수 있으며, 다양한 경로에 대하여 다른 정책이 적용될 수 있고, 이것은 패킷 전송에 대한 신뢰도를 떨어뜨리는 요인이 된다. 따라서, 다양한 경로상의 안전한 패킷전송을 위한 보안 게이트웨이의 발견이 필요하다. 이와 같은 문제점들을 해결하기 위하여 각각의 보안 영역별로 독자적으로 정의된 정책 정보들에 대한 중앙 집중적인 관리와 협상을 지원하는 보안정책 시스템(SPS: Security Policy System)이 있다[2][3].

본 논문에서는 보안 정책 시스템에서의 보안 영역간 보안 정책 연관성을 가진 정책들에 대하여 보안 정책의 적용시 발생할 수 있는 문제점을 분석하고

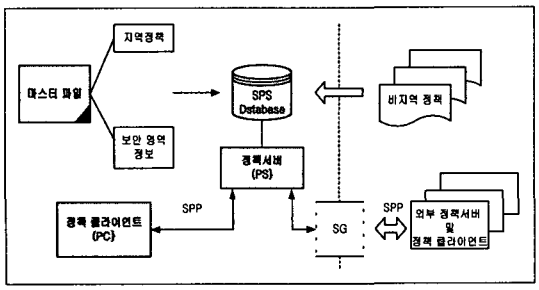
정책들간의 연관성을 제거함으로써 신뢰성있는 정책 정보를 제공하고자 한다. 이를 위하여 연관된 정책들간의 연관성을 제거하는 정책 연관성 제거 알고리즘을 설계 구현하였다.

## 2. 보안 정책 시스템

### 가. 보안 정책 시스템 구성요소

보안 정책 시스템은 호스트와 보안 게이트웨이에게 안전한 통신 설정을 위해 필요한 보안 정책 정보를 제공하는 분산 시스템이다.

보안 정책 시스템의 구성은 (그림 1)과 같다.



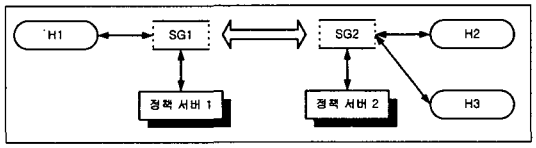

(그림 1) 보안 정책 시스템 구성요소

정책서버는 정책 클라이언트와 다른 정책 서버들로부터 정책 정보 요청을 받아서 요청자의 정책 신원에 따라 정책 정보를 제공해 준다. 정책 클라이언트는 정책 서버에게 정책 정보를 요청하고 정책 서버로부터의 응답을 수신하여 정책 정보를 요청한 응용에게 전달한다. 마스터 파일은 특정 보안 영역의 지역 정책 정보와 보안 영역에 관한 정보가 저장된다. 이때 저장된 정책 정보의 순서가 매우 중요하다. 보안 정책 시스템에서는 정책 정보를 유지하기 위하여 세가지의 데이터 베이스를 유지한다. 보안 영역의 마스터 파일 표현되어있는 정책들의 연관성을 제거한 정책들을 저장하고 있는 지역 정책(Local DB)이 있다. 그리고 보안 정책 시스템의 학습을 통해 얻어진 정책들을 포함하는 캐쉬 데이터베이스(Cache DB), 각 보안 영역의 멤버인 노드들의 신원을 나타내는 엔트리 및 보안 영역 주변에서 정책을 실행하는 각 SG의 신원을 포함한 리스트 정보를 포함한다(정책 서버는 그들이 호스트에 관한 권한이 있음을 결정하기 위해 이 정보를 사용한다). 마지막으로 정책 클라이언트와 정책 서버가 보안 정책 정보를 교환하기 위한 보안 정책 프로토콜이 정의되었다. 보안 정책 프로토콜은 6가지의(SPP Query, SPP Reply, SPP Policy, SPP Policy Acknowledge, SPP Transfer, SPP Keep Alive) 메시지로 구성되어있다[2][3][4].

## 3. 정책 연관성

### 가. 정책 연관성의 문제점 분석

보안 정책 시스템내의 마스터 파일에는 보안 영역의 지역 정책들이 정의되어있다. 각 지역 정책들은 SA정보로 구성되어 있으며 이들 SA는 실렉터(Selector)로 나타내어진다. 실렉터들은 정책 협상시 필요한 정보들로 구성되어 지고, 이 값들이 상호 연관성이 있으면 정책들의 사용에 문제가 발생할 수 있다. (그림 2)은 보안 정책 시스템에서의 정책 제공의 예를 나타낸다[2].




(그림 2) 보안 정책 시스템 예

[표 1] 정책 서버2 정책 연관성을 내포한 정책

| 실렉터 / 정책 | 소스 주소 | 목적지 주소 | 프로토콜 | Direction | Action |
|---|---|---|---|---|---|
| 정책 #1 | * | H2 | * | Inbound | Permit |
| 정책 #2 | * | [illegible] | * | Inbound | Deny |

[표 1]은 정책 서버 2의 마스터 파일에 포함된 정책들을 나타내며, 이들 두 개의 정책들은 모든 실렉터(소스 주소, 목적지 주소, 프로토콜, Direction)에 대하여 연관성을 가지고 있음을 알 수 있다. 정책 정보의 교환 과정은 다음과 같다. 먼저 정책 서버1은 H3에 대한 정책정보를 요청한다. 그러면 정책 서버2는 "정책 #2"를 정책 서버1에 제공하고, 정책 서버1은 "정책 #2"를 획득하고 이 정책 정보를 캐쉬 데이터베이스에 저장한다. 이때 정책 서버1은 다시 H2에 대한 정책 정보를 획득하기 위해 자신의 캐쉬 데이터베이스를 검색한다. 이때, 캐쉬 데이터베이스에는 이미 획득한 정책중에 요청한 정책에 대한 대응 정책 "정책 #2"를 발견하여 이 정책을 이용하게된다. 이 결과는 원래 정책 서버2의 정책이 H2에 대해 허용함을 의미하고 있는데 반해, H2에 대한 통신이 거절되어야 함을 나타내므로 이 정책은 명확히 틀렸음을 나타낸다. 이와 같이 두 개의 정책의 연관성에 의해 발생할 수 있는 문제를 해결하기 위해 정책들간 연관성제거 모듈을 필요하게 되었다. [표 2]는 정책 연관성을 제거 후의 정책 리스트를 나타내고 있

다[2][3].

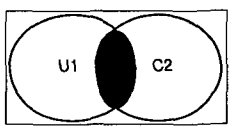
[표 2] 정책 서버2 연관성 제거 후의 정책 리스트

| 실렉터 / 정책 | 소스 주소 | 목적지 주소 | 프로토콜 | Direction | Action |
|---|---|---|---|---|---|
| 정책 #1 | * | H2 | * | Inbound | Permit |
| 정책 #2 | * | not H2 | * | Inbound | Deny |

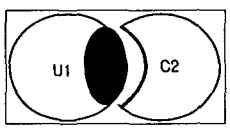
정책 정보 교환 과정을 살펴보면 먼저, 정책 서버1은 H3에 대한 정책을 요청한다. 정책 서버2는 "정책 #2"를 정책 서버1에게 전달하고, 정책 서버1은 "정책 #2"를 캐쉬 데이터베이스에 저장한다. 이때 정책 서버1이 H2에 대한 정책을 요청한다. 정책 서버1은 캐쉬 데이터베이스에서 대응되는 정책을 검색하고 대응되는 정책이 없으므로 H2에 대한 정책을 정책 서버2에게 요청한다. 정책 서버2는 요청에 대해 [표 2]에 대응하는 "정책 #1"을 정책 서버2에게 전달하고, 정책 서버1은 H2에 대한 바른 정책인 "정책 #1"을 캐쉬한다.

나. 정책 연관성 제거 알고리즘

(그림 3)는 집합 U1과 C2사이에 정책들간의 연관성이 존재하는 경우이다. (그림 4)는 (그림 3)에서 정책 연관성을 제거후의 그림이다. U1은 Un-correlation set이고 C2은 Correlation set을 나타낸다.




(그림 3) 연관성 존재




(그림 4) 연관성 제거

(그림 3)에서 U1과 C2사이의 교집합을 Scjn이라고 한다면 (그림 4)는 U1은 Scjn과 C2의 ~Scjn으로 분리되어지고 Scjn ∪ ~Scjn = Tcjn이 된다. 여기에서 Scjn은 정책들간의 교집합을 나타내며, ~Scjn은 정책 연관성을 제거한 정책들을 나타낸다. 또한 Tcjn은 초기에 전체집합을 의미하며 C는 정책 연관성을 존재하는 집합을 나타낸다. (그림 3)는 U1과 C2 사이에 그림과 같이 상호 연관성을 가진다고 했을 때 이를 제거하기 위한 절차는 다음과 같다. 먼저 실렉터를 선택(Optimization)한다.

[표 3]은 최적의 실렉터 선택 방법을 나타낸다[2][3].

[표 3] 실렉터 선택 순서(Optimization)

1) T 실렉터 값이 모두 동일한 값 선택
2) T 실렉터 값 ⊃ Cj에 있는 Scjn의 값 선택
   /* Single branch(∵Complement가 nil) */
3) 하나의 Branch가 이미 연관성이 제거된 C의 상호 연관된 정책들 중 하나이면 branch를 조기 종료한다.
4) Node가 U의 모든 정책들과 상호 무관하면 종료
   /* T=U 전체집합(임시집합 T는 전체집합 [U]로 시작) */

branch 생성 순서는 다음과 같다.

[표 4] branch 생성 규칙

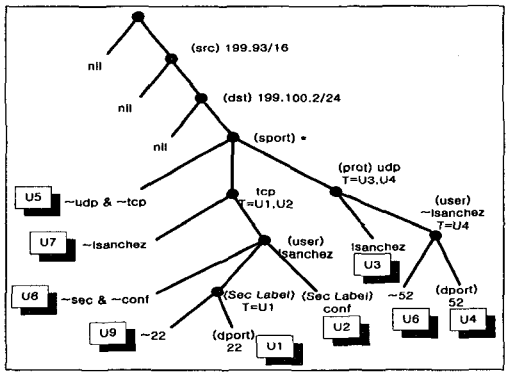
1) T의 임의의 정책들에 나타나는 Scjn 각각의 값에 대해 Tree의 branch에 추가한다.
   /* 그 값이 Cj의 Scjn의 값의 Super set 이면 Cj의 값을 사용한다(∵Cj의 값이 universal set 이므로) */
2) T의 Scjn의 모든 값은 합집합의 complement에 대해 branch 추가
   /* complement 취할 때 universal set은 Cj에 있는 Scjn의 값 */
3) nil : branch 종료

[표 5]는 정책 연관성이 제거된 예를 나타내고 있다.

[표 5] 예제

| [U] | src | dst | port | sport | dport | user | slabel |
|---|---|---|---|---|---|---|---|
| U1 | 199.93/16 | 199.100.2/24 | TCP | * | 22 | lsanche2 | sec |
| U2 | 199.93/16 | 199.100.2/24 | TCP | * | * | lsanche2 | conf |
| U3 | 199.93/16 | 199.100.2/24 | UDP | * | * | lsanche2 | * |
| U4 | 199.93/16 | 199.100.2/24 | UDP | * | 52 | * | * |

[표 5]를 연관성 제거 Tree로 나타내면 (그림 5)와 같다.




(그림 5) 정책 연관성 제거 Tree

4. 정책 연관성 제거 알고리즘 구현

가. 구현환경

정책 연관성 제거 알고리즘의 구현 환경은 [표 6]과 같다

[표 6] 구현 환경

| | |
|---|---|
| 운영체제 | - MIZI LinuxOS release 1.1<br>- Kernel 2.2.14 |
| 하드웨어 | - 펜티엄 III 650<br>- RAM 128<br>- LAN 100Base |
| 개발도구 | - Linux C(gcc 2.91.66, Make 3.77)<br>- MySQL 3.22.32 |

각 보안 영역을 중심으로 1개의 정책서버와, 2개 이상의 정책 클라이언트 및 1개의 보안 게이트웨이를 설치하였다. 보안 영역은 네트워크 단위로 분리하였다. 원시 정보는 정책서버의 마스터 데이터베이스에 저장되며 연관성 제거 후 지역 데이터베이스와, 보안영역 데이터베이스에 저장된다.

나. 구현

다음은 구현된 정책 연관성 제거 알고리즘에 대한 소스와 설명을 보여준다.

1) 파일설명

- Cal_DataStructure.h : 연관성 제거 모듈에서 사용하는 구조체와 함수 선언
- Cal_SubNode.c : 입력된 레코드와 동일한 레코드 추출, Cal_UnionSet.c 에 있는 함수 호출
- Cal_UnionSet.c : 전체집합, 여집합 계산, *(star) 기호가 있을 때의 처리

2) 함수설명

- 다음의 함수는 실렉터 선택을 위한 함수로, 먼저 각 실렉터들에 대한 전체집합을 구하고, 각각의 여집합을 구한다.

```
int Cal_UnionSet_src_addr(struct SELECTORS
**input, struct SELECTORS **UnionSet)
int Cal_UnionSet_dst_addr(struct SELECTORS
**input, struct SELECTORS **UnionSet)
.....
```

- 다음은 각 실렉터들 간의 연관된 레코드를 추출하기 위한 함수들이다(기본적으로8개의 실렉터 값을 대상으로한다. src, dst, port, s-protocol, d-protocol, user, slabel, direction이다).

```
int Cal_SubNode_src_addr(struct SELECTORS
**input, struct SELECTORS **UnionSet)
int Cal_SubNode_dst_addr(struct SELECTORS
**input, struct SELECTORS **UnionSet)
.....
```

- 다음은 추출된 실렉터의 연관성을 제거하는 함수이다. 각, 실렉터에 대하여 모두 적용한다.

```
int Cal_DeCorrSet(struct SELECTORS **input,
struct SELECTORS **output);
```

3) 알고리즘 적용 예

| src | dst | xport | port | user |
|---|---|---|---|---|
| 203.237.140.161 | 203.237.140.243 | tcp | 21 | user1 |
| * | 203.237.140.243 | tcp | 21 | user1 |

(그림 6) 정책 연관성 제거 전 정책리스트

| src | dst | xport | port | user |
|---|---|---|---|---|
| 203.237.140.161 | 203.237.140.243 | tcp | 21 | user1 |
| ~ 203.237.140.161 | 203.237.140.243 | tcp | 21 | user1 |

(그림 7) 정책 연관성 제거 알고리즘 적용 후

(그림 6)은 정책 연관성 제거 알고리즘 적용전의 마스터 파일에 정의 되어있는 정책을 나타내며 두 정책은 상호 연관성이 존재한다. (그림 7)은 정책 연관성 제거 알고리즘을 적용 후의 결과로 src에 대한 주소가 ~203.237.140.161로 변경되어 두 정책간의 연관성이 제거되었음을 확인할 수 있다.

5. 결론

IPsec에서는 각기 다른 보안 영역의 통신 상대와 통신을 하거나 다른 보안 영역을 거쳐서 통신하는 경우에 각각의 보안 영역들간에 정의된 정책 정보를 이용하여 정책을 협상한다. 이때 정책에 필요한 정책 정보는 마스터 데이터베이스에 정의 되어있으며 이들은 연관성을 가질 수 있다.

본 논문은 마스터 데이터베이스에서 정책들간의 연관성에 의해 발생할 수 있는 문제점을 분석하였고, 이들간의 연관성을 제거하는 정책 연관성 제거 알고리즘을 C 언어를 이용하여 구현하였다. 현재의 알고리즘은 연관성의 정도에 따라 정책의 종류 및 수행속도에 많은 차이를 가지고 있다.

향후 지속적인 연구를 통하여 보다 효율적인 연관성 제거 알고리즘을 연구할 계획이다.

참고문헌

[1] 박명찬, 인소란, 최용락, "IPsec 기반의 보안정책 협상 시스템 설계 및 구현", 한국통신정보보호학회, 추계학술발표논문집, VOL.10 NO.1 , 2000. 11. 25.

[2] L.A. Sanchez and M.N. Condell, "Security Policy Protocol",Internet Draft, draft-ietf-ipsec-spp-00, 1999.7.

[3] L.A. Sanchez and M.N. Condell, "Security Policy System", Internet Draft, draft-ietf-ipsec-sps-00, 1998.11.

[4] Atkinson, R.,and S. Kent, "Security Architecture for the Internet Protocol", RFC 2401, 1998. 11.