정책기반 네트워크 관리 시스템의 정책 충돌 탐지 및 복구*

이 규 웅**

Detection and Recovery of Policy Conflicts in Policy-based Network Management Systems*

Kyu Woong Lee**

■ Abstract ■

Policy-based Network Management (PBNM) has been presented as a paradigm for efficient and customizable management systems. The approach chosen is based on PBNM systems, which are a promising and novel approach to network management. These systems have the potential to improve the automation of network management processes. The Internet Engineering Task Force (IETF) has also used policy concepts and provided a framework to describe the concept as the Policy Core Information Model (PCIM) and its extensions. There are policy conflicts among the policies that are defined as the policy information model and they are not easily and effectively detected and resolved.

In this paper, we present the brief description of PBNM and illustrate the concepts of policy core information model and its policy implementation for a network security. Especially we describe our framework for detecting and resolving the policy conflicts for network security.

Keyword : PBNM, Network, Security, Policy, Conflict

* 본 연구는 2005년도 상지대학교 교내연구비 지원에 의한 것임.
** 상지대학교 컴퓨터정보공학부

1. 서 론

인터넷 환경에서 통신망 자원관리, 정보보호 기술, 실시간 멀티미디어 서비스 제공 등은 최근 논의되는 첨단 기술로서 많은 연구개발이 되고 있으며, 특히 정보보호 기능을 추가한 차세대 인터넷 기술의 전개와 능동형 통신망 운영기술 적용에 관한 많은 연구들이 수행되고 있다.

정책기반의 네트워크 보안 관리는 보안 관련 이벤트 모니터링을 통해 수집된 데이터를 통해 이를 분석하고, 공격 및 이상 징후의 탐지 및 대응에 이르는 일련의 보안 행위에 있어 해당 도메인이 추구하는 보안 정책을 강제하여 수행할 수 있도록 관련 기능을 제공하는 보안 프레임워크이다[6, 9]. 정책 모델의 발전에 따라 네트워크 보안 관리는 기존의 요소기술을 통합하여 실질적이고, 운영이 간편한 통합관리(ESM, Enterprise System Management) 구조로 구축되고 있다. 또한 네트워크 보안에 대한 정책(policy)를 NE(Network Element)에 능동적으로 적용하는 정책 기반의 네트워크 관리기술(PBNM : Policy Based Network Management)이 연구되고 있다[5, 6, 9].

정책기반의 네트워크 관리 기법은 IETF, DMTF (distributed management task force), CERT, The Parlay Group 등의 표준화 기관을 중심으로 표준화 규격 연구가 활발하게 진행되고 있다. 또한 네트워크 시스템 생산업자와 소프트웨어 벤더 간의 협력으로 PBNM 기능의 소프트웨어를 개발하여, 인터넷상에서 정책 기반의 QoS와 정보보호를 관리하는 관련 상용 시스템을 시장에 내놓고 있다. 현재 PBNM 기술은 초기단계로서 상용제품 개발에 대한 전반적인 기능 향상과 정책 기반의 통신망 관리구조를 유지하면서, 정보보호와 QoS를 비롯하여 시스템 관리 서비스 등을 정책 기반의 관리에 따른 통신망의 진화와 인프라 구조에 따른 방향 정립을 위한 단계로 예상된다.

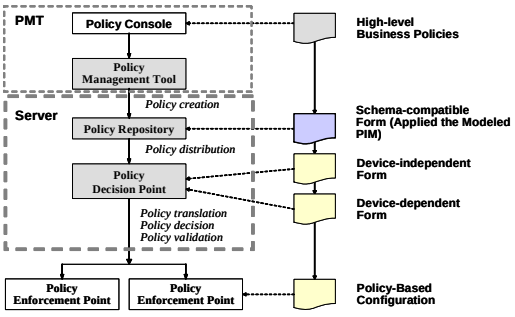
정책기반의 네트워크 관리환경에서 보안 정책은 네트워크 및 이를 통하여 제공되는 응용 서비스에 대한 접근, 침해 요인의 탐지와 그에 대한 대응과 관련되어 해당 도메인에서 강제되어야 할 운용 규칙이다.

본 논문은 정책 정보 모델의 네트워크 보안 유지를 위한 정책 설립을 위한 정보 모델의 프레임 워크를 조사하고, 네트워크 보안 정책의 충돌을 탐지 및 복구 하는 방법을 제안한다. 네트워크 보안 정책의 충돌로 인하여 네트워크 관리 시스템의 비효율적 운영을 방지하고 비정상적 장비 운용을 사전에 예방할 수 있다.

2. 정책기반 보안 정책 모델

2.1 정책 정보 모델링

정책기반의 정책 정보모델은 PMT(Policy Management Tool), 정책서버와 PEP(Policy Enforcement Point)로 [그림 1]과 같이 구성된다. 정책기반의 통신망을 관리하기 위해 정책 콘솔의 GUI 로 구축된 PMT를 이용하여 비즈니스 레벨로 정해진 정책 혹은 정책 규칙을 설정한다. PMT에서 설정된 정책규칙을 정책 정보모델(PIM ; Policy Information Model)에 따라 변환하여 객체 레퍼런스 형태로 정책 레포지터리에 저장한다.

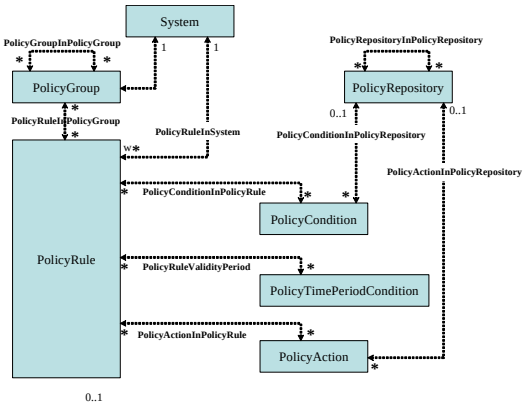


[그림 1] 정책정보 모델의 시스템 구성

2.2 정책 정보모델

정책 정보모델 PCIM(policy core information mo-

del)은 정책 정보를 표현하기 위한 모델이다. 표준화 기구인 DMTF의 정보 모델 CIM(common information model)의 확장 형태와 IETF의 정책 프레임워크 WG와 공동 개발된 객체 지향적 정보 모델이다[2, 8]. 정책 정보모델 PCIM은 두 가지의 객체 클래스 계층을 정의하고 있다. 정책 정보와 정책 제어를 표현할 수 있는 구조적 클래스(structural class)와 구조적 클래스 인스턴스들 간의 상호 관계를 나타낼 수 있는 연관 클래스(association class)를 정의한다. [그림 2]는 정책 정보모델 PCIM의 주요 클래스와 이들 간의 연관 클래스의 관계를 도식화한 그림이다.



[그림 2] PCIM의 주요 클래스와 관계

정책 정보모델의 지속적 산업체 요구사항을 반영하기 위하여 PCIM의 확장형태인 정책정보모델 PCIMe를 제안하였다. PCIMe는 정책 정보모델을 설정하는 기본 프레임워크로 정책 기반의 서비스를 관리하는 경우에 해당 서비스에 따라 세부기능을 분석하여 이에 따른 조건부와 실행부를 원활하게 표현할 수 있도록 PolicyCondition과 PolicyAction 클래스에 여러 속성의 클래스를 추가하여 정책 정보모델을 설정할 수 있도록 하였다.

정보모델 LDAP 스키마는 IETF의 네트워크 워킹 그룹에서 제안한 RFC 3703와 RFC 4104에서 권고하고 있는 PCELS(Policy Core Light Weight Access Protocol Schema)에 따른다[10, 12]. 이 권고안들은 PCIM과 PCIMe에서 사용하는 표현 클래스들을 접근 프로토콜로서 LDAP(Light Weight Access Protocol)을 사용하는 디렉터리 내에서 구현될 수 있도록 사상관계를 정의하고 있다. 즉, PCIM과 PCIMe에서 표현하는 정보모델의 클래스들을 LDAP 스키마 내에서 표현할 수 있도록 관련된 클래스와 사상관계를 정의한다.

2.3 정책 정보모델 표준화 동향

객체지향 모델을 사용하여 네트워크 정보를 모델링 하는 DMTF 표준화 기구는 정책 기반의 네트워크 관리에 관련된 표준으로 CIM Policy 버전 2.9를 정의하고 있다. DMTF CIM에서 정립하고 있는 각각의 주요 표준은 DMI(Desktop Management Interface), 즉 데스크톱 개인용 컴퓨터를 관리하기 위한 표준, DEN(Directory Enable Network), 즉 디렉터리 안에 네트워크 요소 및 서비스를 표시하기 위한 표준 정보모델 및 WBEM(Web Based Enterprise Management), 즉 인터넷 상에서 웹을 기반으로 하는 네트워크 관리를 위한 정보모델과 이를 전달하는 메시지 전달방식을 제정하는 표준을 수립한다. 특히, CIM은 XML 기반의 정보모델로 손쉽게 접속이 가능하도록 연구를 진행하고 있으며, 정책 객체를 사용하는 사용자 또는 운영자의 접속에 대한 인증과 접속범위 설정을 위한 클래스, 정책 저장소로의 질의 조건 클래스, 외부 저장소에서 접속하는 방법을 설정하는 클래스가 정의되어 있다.

IETF는 정책 정보모델링을 위해 정책 프레임 네트워크 워크그룹을 운영하고 있으며 시스템에 독립적인 정책 모델 PCIM을 RFC 3060에서 제정하였다[5]. PCIM은 DMTF의 CIM 버전 2.5를 기반으로 제정되었다. 이후 PCIM에서 정의한 정책 정보모델의 미흡한 부분을 보완하여 PCIMe을 정립하여 RFC 3460으로 규격을 제시하였으며, IETF의 정책 프레임 워크그룹과 DMTF에서 함께 개발하였고, 정책정보를 표현하기 위한 객체지향 정보모델을 기술하고 있다[8, 7]. IETF의 정책정보모델은 정책모

델의 핵심을 정의하였고, 특정한 시스템에 적용하기 위해서는 정책모델의 핵심을 기반으로 확장하여 사용하도록 하고 있다.

3. 정책정보모델의 보안정책 충돌

3.1 정책 충돌 정의 및 문제점

정책정보모델에서 보안정책은 네트워크 내의 패킷을 제어하는 것이 목적이다. 네트워크 패킷은 라우터 내에서 제어되기 위한 원천지 주소, 목적지 주소를 기본적으로 포함하고 있으며, 물리적 계층 구조에 따라 추가적인 정보를 패킷 내에 포함한다. 예를 들어 응용 계층의 패킷은 응용 계층의 네트워크 제어를 받기 위해 원천지 IP 주소, 목적지 IP 주소, 상위 레벨 프로토콜(TCP 또는 UDP), 프로토콜의 원천지 및 목적지 포트 번호로 구성되는 5-튜플 구조를 갖는다[1, 4, 11]. 정책정보모델에 의하여 네트워크 장비에 부과될 네트워크 보안 정책은 대체적으로 패킷을 필터링 하기위한 주소를 기반으로 구성되며, 이러한 정책은 정책 정보모델에 의하여 충돌여부를 탐지할 수 없다.

정보모델의 보안정책 충돌을 정의하기 위해 패킷 필터 정책의 필터 규칙을 활용하여 정책 충돌의 문제를 정형화 한다. 보안정책 패킷 필터 규칙의 필터 F는 k개의 튜플(F[1], F[2], …, F[k])로 구성되며 각 필드 F[i]는 전위 비트열(prefix bit string)이다[10]. 각 전위 비트열 x.*는 주소의 범위 또는 수치 값의 범위를 결정지으며, [x0 … 0, x1 … 1]과 같이 표현된다. 여기서 x 뒤에 따라오는 비트의 수는 각 필드가 갖는 최대 비트 길이와 x에 명시된 비트의 수와의 차이이다. 예를 들어 4비트로 구성되는 필드 F[i]가 전위 비트열 10* 를 갖는다면 수치 값의 범위 [1000, 1011]을 결정지으며 십진수 값으로[4, 10] 사이의 범위를 결정짓는 필드 F[i]이다. 따라서 임의의 수치값 또는 주소값 X가 x.*와 일치한다면 X는 [x0 … 0, x1 … 1] 사이에 위치하게 된다. 예를 들어, 원천지 주소와 목적지 주소 (220.67. 180.67, 220.66.159.1)를 갖는 IP 패킷은 2-튜플로 구성되는 필터 F1(220.67.*, 220.66.159.*)와 일치하게 된다.

정책 정보모델은 IPv6 패킷 정보 수용을 위해 패킷 필터 정책 정의시 9개의 조건 변수 값을 사용하므로 k의 값이 9로 정의될 수 있으며 IPv4 주소를 수용하기 위해 원천지와 목적지 주소 값을 전위 비트로 하는 2-튜플 패킷 필터 규칙을 정의할 수 있다.

각 필터 F는 실행부 A(F)를 가지며, 보안정책의 필터 F와 일치하는 패킷 P는 실행부 A(F)에 따라 처리되어져야 한다. 이 때, 다수의 필터와 일치되는 패킷 P는 실행부 A(F)를 결정짓는데 모호성을 갖게 되어 보안정책의 문제를 야기하게 된다. 예를 들어, IPv4의 원천지 및 목적지 주소를 갖는 2-튜플 필터인 경우에 네트워크 x의 전위 비트열로 220.67.*를 갖고 네트워크 y의 전위 비트열로 220.66.*를 갖는 경우를 가정한다. 이 때 보안정책 필터 F1은 (220.67.*, *)이고 A(F1)은 전송 대역폭을 100Mbps로 설정하는 것이며, 보안정책 필터 F2는(*, 200.66.*)이고 A(F2)는 전송 대역폭을 1Mbps로 설정하는 두 개의 보안정책 필터 규칙을 가정한다. 이러한 가정하에서 네트워크 x에서 네트워크 y로 전송되는 패킷의 경우에 보안정책 규칙 필터 F1과 F2가 모두 조건 만족되므로 실행부 A(F1)과 A(F2)중에서 어떠한 실행부가 실행되어야 하는지 모호성을 갖게 된다[1, 3, 4, 11]. 이러한 문제를 해결하기 위한 몇 가지 대안 책으로 다음과 같은 것을 생각할 수 있다. 먼저, 가장 간단한 방법으로 첫 번째 일치되는 규칙을 적용하는 방법이다. 이 방법은 보안정책 규칙 데이터베이스의 저장 순서에 따라, 보안정책 규칙 F1이 F2보다 선행되어 저장되어 있다면, F1의 실행부 A(F1)을 실행하게 하는 방법이다. 이 방법은 현재 상용화 되어 있는 방화벽 시스템에서 일반적으로 사용되는 방법이다. 두 번째로, 각각의 보안정책 규칙에 우선순위를 부여하는 방법이다. 각각 서로 다른 규칙에 우선순위를 부여하여 패킷 처리 시 우선순위가 높은 보안정책 필터에 따라 처리하는 방법이다. 그러나 이 방법은 우선순위를 기준으

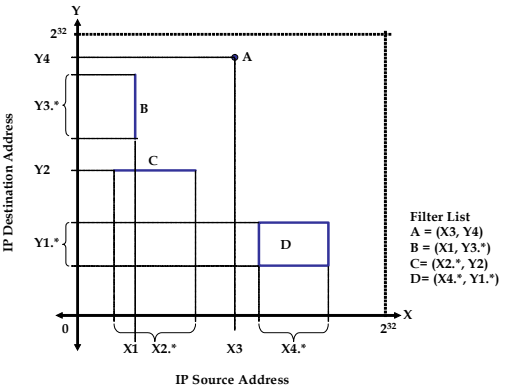
로 정렬하여 저장하면 첫 번째 방법과 동일한 방법이다. 마지막으로 각 보안정책 규칙 필터의 필드마다 우선순위를 부여하는 방법이다. 다중 일치 규칙 필터가 존재하는 경우 각 필터의 필드마다 우선순위를 설정하여 가장 높은 우선순위의 필드를 만족하는 규칙 필터를 선택하게 하는 방법이다. 예를 들어 원천 주소 필드의 우선순위를 가장 높게 한 경우, 목적지 주소가 일치하는 필터보다 원천지 주소가 일치하는 보안정책 규칙을 실행하게 하는 방법이다. 앞의 예에서 네트워크 x에서 네트워크 y로 패킷을 전송하는 경우 원천지 주소 필드 우선순위가 목적지 주소 필드의 우선순위보다 높은 경우 보안정책 규칙 F1의 필터가 적용되게 된다.

이러한 방법들은 구현하기 쉽고, 실제 상용 시스템에서 적용하는 방법이지만, 보안정책의 임의 저장순서에 따라 패킷이 전송되어 지거나 버려지게 되는 중대한 결점을 갖게 되어, 임의성 및 무작위성을 갖게 된다. 두 번째 방법 역시 필터 우선순위에 따라 정렬하게 되면 임의성을 갖게 된다. 세 번째 방법은 보안정책 규칙 F2의 실행부 A(F2)를 실행시킬 방법이 없게 되는 단점을 갖게 된다. 즉, 임의성 및 무작위성의 단점이 비효율성이라는 단점으로 전환되어질 뿐이다. 위의 예에서 새로운 보안정책 규칙 필터 F3(220.67.*, 220.66.*), A(F3) = {대역폭 10Mbps}를 추가하고 가장 많은 필드를 만족하는 보안정책 필터를 수행하는 기준에 따른다면, 네트워크 x에서 y로 전송되는 패킷은 모두 보안정책 규칙 F3에 따라 실행되어진다는 중요한 사항을 관찰 할 수 있다. 따라서 네트워크 x에서 y로 전송되는 패킷은 보안정책 규칙 F1이나 F2가 아닌 F3에 의해 제어되고, 보안정책의 충돌이 최적화되지 못한 방법으로 문제점을 갖고 있음을 알 수 있다[1, 4].

3.2 정책 충돌의 기하학적 접근 분석

네트워크 보안정책 정보모델의 정책 규칙 충돌을 정형화 하고 분석하기 위하여 기하학적 표현을 통하여 충돌 문제의 범위를 살펴본다. 문제를 정형화하고 일반화하기 위하여 k-튜플로 구성되는 보안정책 규칙 필터를 2-튜플로 단순화하여 정의하고, 2차원 좌표 상에서 충돌 범위를 정의한다. 2-튜플 필터는 원천지 주소와 목적지 주소 튜플로 구성되므로 2차원 좌표의 각 축은 원천지 주소 축과 목적지 주소 축으로 구성되며, 32비트 주소 값에 대하여 각 축의 값은 0에서 2^{32}의 값을 갖는다.

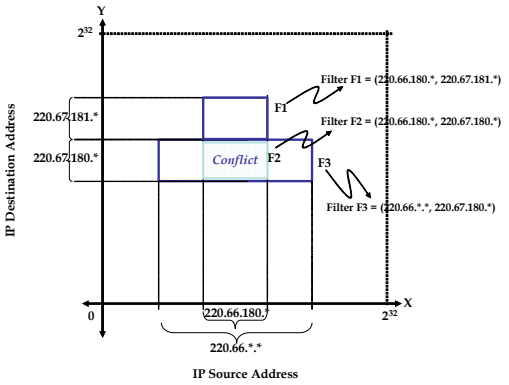
예를 들어, 보안정책 규칙 필터 리스트가 A = (X3, Y4), B = (X1, Y3.*), C = (X2.*, Y2), D = (X4.*, Y1.*)와 같이 주어졌을 때, 이 보안정책 규칙 필터 리스트의 2차원 좌표상 표현은 다음 [그림 3]과 같이 구성될 수 있다.



[그림 3] 보안정책 규칙의 2차원적 표현

[그림 3]에서 필터 A는 원천지 주소와 목적지 주소가 완전히 기술되어 있으므로, 하나의 점으로 표현되며 필터 B와 필터 C는 각각 원천지 주소 또는 목적지 주소가 하나씩 범위 값으로 주어졌으므로 하나의 선으로 표현된다. 필터 D는 원천지 및 목적지 주소가 모두 범위 값으로 주어졌으므로 2차원 좌표 상에서 사각형으로 표현되게 된다. 2차원 좌표상 표현을 보안정책 규칙 필터의 실제 범위 값으로 정하고, 각 필터간의 충돌 상황을 살펴보면 [그림 4]와 같이 표현할 수 있다.

[그림 4]에서 보안정책 규칙 필터 F1과 F2는 원천지 주소가 일치하지만 목적지 주소의 범위가 일



[그림 4] 보안정책 충돌의 2차원 표현

치하지 않아 정책간 충돌이 발생하지 않으며, F1과 F3도 목적지 주소의 범위가 서로 달라 정책간 충돌이 발생하지 않는다. 반면에 F2와 F3은 목적지 주소의 범위가 같으며 원천지 주소 범위는 서로 포함 관계에 위치하여 중복된 영역이 발생하고 이에 따른 규칙 충돌이 발생함을 알 수 있다. 즉 원천지 주소가 220.66.180.*이고 목적지 주소가 220.67.180.*인 패킷들에 대하여 보안정책 규칙 필터 F2와 F3이 동시에 만족하여 각 규칙 필터의 실행부 A(F2)와 A(F3)의 처리에 혼란이 발생하게 된다. 특히, 두 실행부의 실행 의미가 상반되는 경우 즉, 패킷 전송 허가와 거부의 실행 의미를 갖고 있을 때, 어떤 보안 정책이 적용되는가에 따라 매우 큰 파급 효과를 갖게 될 수 있다[3, 11].

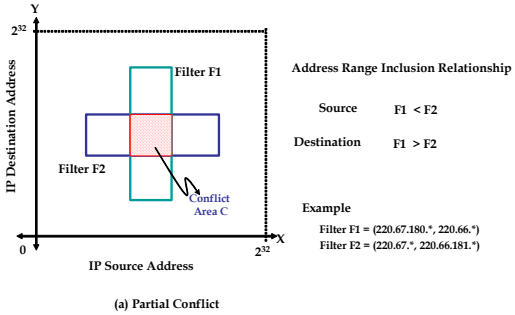
3.3 정책 부분 충돌 및 완전 충돌

보안정책 규칙의 충돌 여부를 정형화 하고 분석하기 위하여 충돌의 유형을 [그림 5]와 같이 부분 충돌과 완전 충돌 두 가지 유형으로 분류한다.

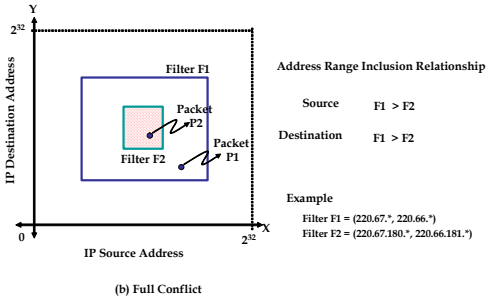
보안정책 규칙의 부분 충돌 상황은 원천지 주소의 포함 관계와 목적지 주소의 포함관계가 서로 상반되는 관계이다. 즉, [그림 5]의 (a)와 같이 원천지 주소의 범위가 규칙 필터 F1이 F2보다 작은 반면 목적지 주소 범위는 F1이 F2보다 큰 상황이므로 보안 정책 규칙 F1과 F2는 상호 부분 충돌 영역 C가 발생한다.

보안정책 규칙의 완전 충돌 상황은 [그림 5]의 (b)와 같이 보안정책 규칙 필터 F1이 F2를 완전히 포함하는 현상이며, 패킷 필터의 주소 범위를 기준으로 볼 때, F1의 원천지 및 목적지 주소가 F2의 원천지 및 목적지 주소를 완전히 포함하여 보안정책 규칙 필터 F2의 영역은 완전 충돌 영역과 일치한다. 이 때, 패킷 P1은 보안정책 규칙 필터 F1만을 만족하므로, 이에 해당하는 실행부 A(F1)를 실행하면 된다. 그러나 패킷 P2는 완전충돌 영역에 포함되므로 A(F1)과 A(F2) 두 개의 실행부 중 어떤 것을 실행해야 하는 결정에 모호성을 갖는다. 면밀한 기준에서 살펴보면 패킷 P2는 두 개의 규칙 중에 세부적으로 명시된 필터 F2에 보다 더 충실하게 만족하는 것으로 볼 수 있으며, 따라서 실행부 A(F2)를 실행하는 것으로 충돌을 쉽게 해결할 수 있다.

따라서 [그림 5]의 보안정책 규칙 부분 충돌과 완



(a) Partial Conflic



(b) Full Conflict

[그림 5] 보안정책 규칙의 부분 충돌 및 완전 충돌

전 충돌 상황 중에서 완전 충돌의 경우에는 해당 규칙을 쉽게 결정지을 수 있으므로 충돌 탐지 및 해결이 쉽게 수행 될 수 있으나, 부분 충돌의 경우에는 규칙 충돌 탐지 및 해결에 대해 연구가 필요한 상황이다. 따라서 보안정책 규칙에 대하여 다음과 같은 정의를 내릴 수 있다.

[정의 1] 보안정책 규칙 필터의 충돌 1

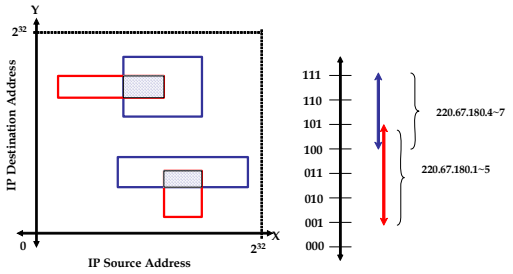
두 개의 보안정책 규칙의 필터가 부분적으로 중복되어 있는 경우 두 필터는 상호 규칙 충돌을 갖는다.

3.4 정책 부분 충돌의 분석

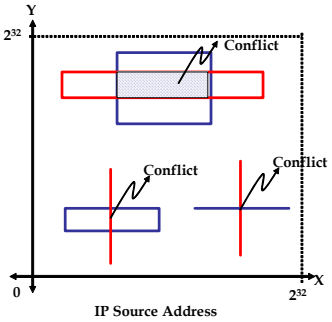
완전 중복에 의한 규칙 충돌은 쉽게 해결 가능하므로 부분 중복에 의한 규칙 충돌 상황을 좀 더 자세히 살펴보도록 한다. 부분 중복에 의한 규칙 충돌 발생시, 각 규칙이 가지고 있는 필터는 k-튜플 필터인 경우 k개의 필터로 구성된다. 각 필드는 수치 값 또는 주소 값의 범위로 표현되거나, 완전히 명시된 수치 값을 갖게 된다. 먼저 완전히 명시된 수치 값으로 구성된 필드인 경우를 고려해 보면, 중복 구간은 한 점에서만 발생하거나, 다른 필터가 범위 값을 갖는 경우 하나의 선으로 발생하게 된다. [그림 6]의 경우는 두 개의 규칙이 정의하는 범위가 부분 중복되어 발생하지만 서로 상호 교차 중복이 아닌 일부 중복인 경우이다. [그림 6]와 같이 상호 교차 중복이 아닌 일부 포함 중복인 경우에는 규칙 필터의 필드에 전위 비트열을 사용한 범위 값을 사용하는 경우에는 발생하지 않는 경우이다. 즉, 필드의 값을 범위 값으로 표현하는 경우에는 다음과 같은 두 가지 경우만 발행하게 된다.

- 1차원 좌표 상으로 분리하여 표현하였을 때, 하나의 규칙 범위 값이 다른 한 규칙의 범위 값을 완전히 포함하는 경우
- 두 규칙의 범위 값이 완전히 다른 경우

따라서 [그림 6] 우측의 1차원 표현과 같이 일부의 구간에서 중복되는 규칙 충돌은 범위 값을 표현할 수 없는 경우이므로 규칙 충돌 상황에서 발생하지 않는 경우로 판단할 수 있다. 전위 비트열을 사용하여 범위 값을 표현하는 필드를 사용하는 경우에는 오직 [그림 7]과 같은 경우만 발생할 수 있을 알 수 있다.



[그림 6] 부분 충돌에 의한 규칙 충돌 중 발생하지 않는 상황



[그림 7] 발생 가능한 규칙 충돌

부분 충돌에 의한 규칙 충돌은 [그림 6]의 경우가 발생하지 않고, [그림 7]의 경우만 발생한다. 따라서 [정의 1]에서 정의한 규칙 충돌 정의는 2-튜플(원천지 주소, 목적지 주소)로 구성된 보안정책 규칙 필터에 대하여 [정의 2]와 같이 세분화될 수 있다.

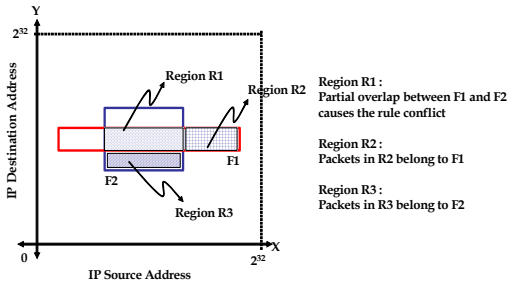
[정의 2] 보안 정책 규칙의 충돌 2

두 개의 보안 정책 규칙 필터 F1, F2가 충돌이 발생하려면 다음과 같은 두 개의 조건중 하나를 만족해야 한다.

1) F1의 필드 f1 > F2의 필드 f1이고
F1의 필드 f2 < F2의 필드 f2

2) F1의 필드 f1 < F2의 필드 f1이고
F1의 필드 f2 > F2의 필드 f2

부분 충돌에 의한 규칙 충돌의 유형을 분석 해 본 결과, 대부분의 경우가 발생하지 않는 경우임을 알 수 있었으며, 완전 충돌의 경우 규칙 충돌을 간주하지 않고도 처리할 수 있는 상황임을 분석하였다. 따라서 상호 교차 중복에 의한 규칙 충돌 부분을 좀 더 세밀하게 관찰하면 [그림 8]과 같이 충돌 상황에서 패킷 처리 여부를 쉽게 결정할 수 있다.



[그림 8] 부분 충돌에 의한 규칙 충돌 영역 분석

[그림 8]에서 다음과 같은 사항을 관찰할 수 있다. 만약 영역 R1에 해당하는 새로운 보안정책 규칙 필터 F3을 추가하게 된다면, 규칙 필터 F3은 규칙 필터 F1, F2와 모두 완전 충돌되는 영역 R3을 갖게 되며, 완전 중복에 의한 규칙 충돌은 영역 접근성을 기준으로 살펴 볼 때, 가장 가깝게 만족되는 규칙 F3을 선택하게 되어 규칙 충돌을 배제함을 앞의 논의를 통해 알 수 있다. 그러므로 보안정책 규칙 충돌의 해결 방법으로 새로운 보안정책 규칙의 추가로 인하여 규칙 충돌을 해결할 수 있는 방안을 제안할 수 있다.

4. 보안정책 충돌 탐지 및 복구 기법

4.1 보안정책 충돌 탐지 기법

앞 장에서 분석한 보안정책 규칙 충돌 상황에 따라, 보안정책 규칙의 필터가 k개의 튜플로 구성되는 F = (F[1], F[2], …, F[k]) 필터에 대하여 규칙사이의 충돌을 정의할 수 있다. 여기서 각 필드 F[i]는 범위 값을 지정하는 전위 비트열이다. 두 필드 x.*와 y.* 가 서로 교집합이 발생하지 하는 경우 공통 주소 부분을 갖지 않게 된다. 그러므로 k-튜플로 구성되는 보안정책 규칙 필터에 대해 규칙 충돌 정의를 다음과 같이 내릴 수 있다.

[정의 3] k-튜플로 구성되는 보안정책 규칙의 충돌
보안정책 규칙 필터 F와 G는 다음의 조건 중 하나를 만족하면 충돌이 발생하지 않는다.
(1) 임의의 전위 비트열 F[i]와 G[i] (1 ≤ i ≤ k)에 대하여, 서로 소인 경우
(2) 모든 전위 비트열에 대하여 F[i]가 G[i]의 전위 접두 비트열인 경우 또는 G[i]가 F[i]의 전위 접두 비트열인 경우

[정의 3]의 조건 (1)은 보안정책 규칙 필터의 한 필드라도 서로 소인 경우에 충돌이 발생하지 않음을 표현하는 것이며 조건 (2)는 두 규칙 F와 G가 완전 중복에 의한 포함관계를 갖는 상황을 표현하는 것이다. 완전 중복에 의한 규칙 충돌은 근접조건을 만족하는 규칙을 따르기로 앞 절에서 논의하였으므로 규칙 충돌로 간주하지 않는다. [정의 3]에 따른 규칙 충돌을 탐지하기 위하여 두 규칙이 조건 (1)과 조건 (2)를 만족하는 지 조사하는 프로시듀어를 작성하고, 이를 통하여 규칙간 충돌 탐지를 수행할 수 있다. 규칙 충돌 탐지 프로시듀어는 [그림 9]에 의사코드로 기술되어 있다.

[예제 1] 3-튜플 (원천지 주소, 목적지 주소, 프로토콜)로 구성되는 보안정책 규칙 F와 G가 다음과 같이 구성된다고 가정한다.
F = (220.67.180.*, 200.66.*, 21)
G = (220.67.*, 200.66.181.*, 21)

두 개의 규칙 F와 G에 대하여, [그림 9]의 알고리즘에 따라 조건 (1)의 만족 여부를 수행하면 3개의 각 필드가 서로 소가 아니므로 조건 (1)을 만족하지

못한다. 조건(2)의 완전 중복 여부를 조사하면 두 규칙 F와 G가 모든 필드에 대해 서로 전위 접두어로 사용하지 않으므로 역시 조건(2)를 만족하지 못한다. 결국 두 개의 규칙 F와 G는 규칙 충돌로 탐지되게 된다.

```
Procedure NSPIM_Rule_Conflict_Check (Rule F, Rule G)
{
        k = the number of fields in packer of rule ;

        /* Check the Condition 1 of Rule Conflick
           Definition */
        for (i = 1 ; i <= k ; i++)
        {
                if (F[i] and G[i] are disjoint)
                        return("Not Conflict") ;
        }

        /* Check the Condition 2 of Rule Conflict
           Definition */
        Flag = 1 ;
        for (i = 1 ; i <= k ; i++)
        {
                if (F[i] is not a prefix of G[i])
                        Flag = 0 ;
        }
        if (Flag == 1) retrun ("Not Conflict") ;
        Flag = 1 ;
        for (i = 1 ; i <= k ; i++)
        {
                if (G[i] is not a prefix of F[i])
                        Flag = 0 ;
        }
        if (Flag == 1) retrun ("Not Conflict") ;

        return("Conflict") ;
}
```

[그림 9] 보안정책 규칙 충돌 탐지 알고리즘 의사코드

[예제 2] 3-튜플(원천지 주소, 목적지 주소, 프로토콜)로 구성되는 보안정책 규칙 F와 G가 다음과 같이 구성된다고 가정한다.

F = (220.67.180.*, 200.66.180.*, 21)

G = (220.67.*, 200.66.*, 21)

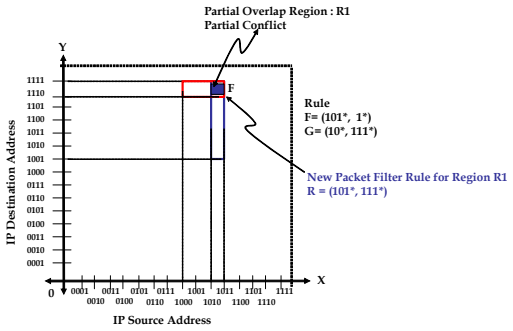
두 개의 규칙 F와 G에 대하여, [그림 9]의 알고리즘에 따라 조건 (1)의 만족 여부를 수행하면 3개의 각 필드가 서로 소가 아니므로 조건 (1)을 만족하지 못한다. 조건(2)의 완전 중복 여부를 조사하면 두 규칙 F와 G가 모든 필드에 대해 완전 중복을 갖는 포함관계를 보이므로 서로 전위 접두어로 사용하고 있음을 알 수 있다. 즉 모든 G[i]는 F[i]의 전위 접두 비트열이므로 조건 (2)를 만족하여 결국 두 개의 규칙 F와 G는 규칙 충돌이 아님을 탐지하게 된다.

4.2 보안정책 충돌 복구 기법

보안정책 규칙 탐지 알고리즘에 의해 두 규칙이 충돌임을 탐지하게 되면, 이들 규칙간의 충돌을 복구해야 한다. 즉 두 규칙이 충돌을 갖는다는 것은 임의의 패킷에 대해 두 가지 규칙이 모두 만족되어 어떤 규칙의 실행부가 실행되어야 할지 모호해지므로, 이를 해결하기 위한 방법으로 다음 두 가지 사항을 고려해본다.

(1) 상호 교차가 아닌 일부 중복에 의한 규칙 충돌은 발생하지 않으며, 완전 중복에 의한 충돌은 규칙 충돌로 간주되지 않는다.

(2) 상호 교차 중복에 의한 규칙 충돌은 중복 부분을 정의하는 새로운 규칙의 추가로서 완전 중복에 의한 규칙 충돌로 전환할 수 있고, 이에 따라 규칙 충돌을 복구할 수 있다.



[그림 10] 충돌 복구를 위한 새로운 규칙의 영역 생성

[그림 10]의 예와 같이 규칙 F와 규칙 G가 상호 교차 중복에 의한 규칙 충돌을 갖는 상황을 고려해 보자. 보안정책 규칙 F와 G는 부분 중복을 갖는 영역 R1을 정의하고 있으며, 이 영역에 의하여 규칙

충돌을 발생하게 된다. 이 때 영역 R1에 해당하는 새로운 규칙 R=(101*, 111*)를 추가하면, 규칙 F와 G는 규칙 R과 완전 중복에 의한 충돌을 갖게 되고, 따라서 완전 중복에 만족하는 패킷은 규칙 적용시 모호성을 갖지 않고, 신설된 규칙 R을 수행할 수 있도록 규칙 충돌이 복구 된다.

규칙 충돌 발생시 복구 알고리즘은 단순하게 구현할 수 있다. 부분 충돌 영역에 의한 규칙에 대해 중복 영역 R을 정의하고 영역 R1을 커버하는 새로운 규칙을 신설 추가하므로써 규칙을 복구 할 수 있다. [그림 10]의 예는 2-튜플로 구성된 상황에서 중복 영역 R1을 갖고 있다. 먼저 좌표축 X에 대하여 중복 구간을 분석해 보면 규칙 F의 101*와 규칙 G의 10*에 대하여 규칙 F의 101* 구간에서 중복이 발생함을 알 수 있고, 유사한 방법으로 좌표축 Y상에서는 규칙 F의 1*와 규칙 G의 111*에 대하여 규칙 G의 111* 구간에서 중복이 발생하게 되므로, 결국 중복 영역 R1은 (101*, 111*) 영역임을 알 수 있게 된다.

복구 알고리즘에서 가장 핵심이 되는 부분은 부분 중복 영역을 찾는 것이므로 각 축에 대하여 전위 접두 비트열이 길게 표현된 구간에서 중복 영역이 발생함 을 알 수 있다. 이 방법을 이용하여 규칙 충돌이 복구 알고리즘은 [그림 11]과 같이 표현할 수 있다.

```
Procedure NSPIM_Rule_Conflict_Check (Rule F, Rule G)
{
        k = the number of fields in packer of rule ;
        for (i = 1 ; i < = k ; i++)
        {
                x[i] = the longer of the two prefixes F[i]
                        and G[i]
        }

        return (x1, x2, ···, Xk) ;
}
```

[그림 11] 보안정책 규칙의 충돌 복구 알고리즘

앞의 예에 표현된 규칙 F와 G에 대하여 복구 알고리즘을 수행하여 새로운 부분 충돌 영역 R1에 대한 추가 규칙을 생성할 수 있다. 충돌이 발생한 규칙 F와 G에 대하여, 각 규칙의 필드 k의 범위 표현 비트 스트링 중에서 가장 길게 표현된 비트 스트링을 찾아, 새로운 규칙의 필드로 사용하면, 임의의 충돌이 생긴 두 규칙의 충돌 영역을 규정하는 새로운 규칙을 만들 수 있고, 이 규칙을 활용하여 충돌을 복구할 수 있게 된다.

6. 결 론

본 논문은 보안 정책을 운용하기 위한 보안 정책 정보 모델에서 정책 충돌을 탐지하는 보안 정책 일치성 기법을 연구하고, 부과된 정책 수행을 일관된 상태로 유지하기 위한 보안정책 복구 방법을 제안하였다. 정책 정보 모델에서 적용 규칙의 사례를 통하여 네트워크 정보 모델로 표현된 정책의 충돌 가능성을 조사하였으며 클래스 계층 구조상에서의 정적인 정책 충돌 탐지를 연구하였다. 충돌 가능한 네트워크 정책의 동적 해결 방법 및 정책 실행 부분의 트랜잭션 처리 부분에 대한 연구를 향후 연구과제로 남겨두고 있으며, 규칙의 수가 많아지는 경우 충돌 탐지를 효율적으로 수행하기 위하여 충돌 검색 비교 방법의 개선을 추가적으로 연구할 필요가 있다. 또한, 정책 규칙의 충돌을 실행 부분과 같이 함께 고려 할 때, 더 많은 충돌의 가능성 범위를 규명할 수 있고, 이러한 상황에서 규칙 충돌 탐지 방법을 확장에 관한 연구가 진행 중이다.

참 고 문 헌

[1] Adiseshu Hari, Subhash Suri, Guru Parulkar, *Packet Filter Management for Layer 4 Switching*, Proceedings of IEEE INFOCOM, 1999.

[2] Distributed Management Task Force, Inc., *Common Information Model (CIM) Specification*, version 2.7, Apr. 2003.

[3] Florin Baboescu and George Varghese. *Scalable packet classification.* Proceedings of SIGCOMM, 2001.

[4] Hari, A., S. Suri, and G. Parulkar, *Detecting and Resolving Packet Filter Conflicts*, Proc. of the Internaltion Conference on INFOCOM, 2000.

[5] Kanada, Y. and O'Keefe, B. J., *Diffserv Policies and Their Combinations in a Policy Server Called PolicyXpert*, IEICE SIG on Information Networks & SIG on Network Systems, March 2002.

[6] Kanada, Y., *Taxonomy and Description of Policy Combination Methods*, Workshop on Policies for Distributed Systems and Networks, Lecture Notes in Computer Science, No.1995, (Springer, January 2001), pp.171-184.

[7] Moore, B. Ed. IBM, *Policy Core Information Model (PCIM) Extensions*, The IETF Network Working Group, RFC 3460, Jan. 2003.

[8] Moore, B., E. Ellesson, and J. Strassner, *Policy Core Information Model - Version 1 Specification*, The IETF Network Working Group, RFC 3060, Feb. 2001.

[9] Nigel Sheridan-Smith, *A Distributed Policy-based Network Management for Enriched Experience Networks*, Ph. D Dissertation, University of Technology, Sydney, 2003.

[10] Pana, M., A. Reyes, A. Moron, M. Brunner, *Policy Core Extension Lightweight Directory Access Protocol Schema*, The IETF Network Working Group, RFC4104, Jun. 2005.

[11] Srinivasan, V., S. Suri, and G. Varghese, *Packet Classification using Tuple Space Search,* Proceedings of ACM SIGCOMM, Sept. 1999.

[12] Strassner, J., B. Moore, R. Moats, E. Ellesson, *Policy Core Lightweight Directory Access Protocol (LDAP) Schema*, The IETF Network Working Group, RFC3703, Feb. 2004.

◈ 저 자 소 개 ◈

이 규 웅 (leekw@sangji.ac.kr)

서강대학교 대학원에서 공학석사를 하고 동 대학원에서 데이터베이스 시스템 전공으로 공학박사를 받았다. 한국전자통신연구원 선임연구원을 지냈고, 현재 상지대학교 컴퓨터정보공학부 부교수로 재직중이며 주요 연구관심 분야로 클러스터 데이터베이스 시스템, 클러스터 기반 데이터 분산 트랜잭션 관리, 주기억장치 데이터베이스 시스템 등을 연구하고 있으며, 정보통신부 및 중소기업청 주관의 클러스터 시스템 데이터 관리 분야의 연구를 수행하고 있다.