보안 시스템의 자동 관리를 위한 보안 네트워크 관리 구조의 설계 및 성능 분석

정회원 안 개 일*

Design and Performance Analysis of Security Network Management Architecture for Auto-managing Security Systems

Gae-Il Ahn* ***Regular Member***

요 약

본 논문에서는 정책 기반의 네트워크 관리 구조를 확장하여, 보안 네트워크의 구성의 자동화를 지원할 수 있는 보안 네트워크 관리 구조와 방법을 제안한다. 제안하는 구조 및 방법은 보안 관리 서버가 보안 시스템의 역할과 능력 그리고 보안 정책의 역할과 시간 정보를 기반으로 하여 보안 시스템이 필요로 하는 최적의 보안 정책을 자동 결정하고 보안 정책 룰을 가장 효과적이고 효율적으로 실행할 수 있는 보안 시스템을 자동 결정할 수 있게 한다. 본 논문에서는 기존의 네트워크 시스템과 보안 시스템을 통합 제어할 수 있는 SNMP 프로토콜기반의 보안 네트워크 토폴러지 맵 자동 생성기도 제안한다. 본 논문에서 제안된 구조와 방법의 우수성을 보여주기 위하여 공격에 대한 자동 대응 기능을 시뮬레이션하고 그 성능을 평가한다.

Key Words : Security network management, Security system, Automation, Topology map generator.

ABSTRACT

This paper proposes the architecture and the methods of security network management for auto-configuration of security systems by extending the existing policy-based network management architecture. The architecture and the methods proposed in this paper enable a security management server to automatically decide the best-suited security policy to apply to a security system and the most effective and efficient security system to perform security policy rule, based on the role and capability information of security systems and the role and time information of security policy. For integrated control of network system and security system, this paper also proposes SNMP protocol based security network topology map generator. To show the excellence of the proposed architecture and methods, we simulate and evaluate the automatic response against attacks.

Ⅰ. 서 론

시스템이나 네트워크를 불법적으로 이용하거나 파괴하는 공격을 탐지 및 방어하기 위하여 방화벽, 침입탐지 시스템, 그리고 침입방지 시스템과 같은 보안 시스템이 제안되었다. 기존의 보안 시스템은 사용자 망에서 독자적으로 운용되었지만, 현재는 ESM (Enterprise Security/System Management)의 형태로 각기 다른 형태의 보안 시스템들이 서로 통합 관리되고 있다[1]. 이제는 한발 더 나아가 보안 기능을 네트워크 장치에 추가하면서[2], 그 보안 영역을 사용자 망에서 인터넷 서비스 제공자 망으로 확대하

* 한국전자통신연구원 네트워크보안 연구부 (fogone@etri.re.kr)
논문번호 : KICS2005-03-121, 접수일자 : 2005년 3월 24일

고 있는 추세이다[3,4]. 이처럼 관리해야 할 보안 영역이 확대되면 그에 비례하여 보안 시스템의 효율적이고 자동화된 관리가 요구된다.

최근에 역할(role) 개념을 이용하여 네트워크 구성(configuration)의 자동화를 제공할 수 있는 정책 기반의 네트워크 관리 구조[5-7]가 제안되었다. 이 구조는 통신 서비스 품질을 고려하면서 대규모 통신 네트워크 자원을 자동적으로 관리하는 것을 목적으로 한다.

본 논문에서는 기존의 정책 기반의 네트워크 관리 구조를 확장하여, 보안 시스템 관리를 위한 보안 네트워크 관리 구조와 방법을 제안한다. 제안하는 구조와 방법의 목적은 사용자의 공격을 효과적이고 효율적으로 탐지하고 차단할 수 있도록 보안 네트워크의 구성을 자동화하는 것이다. 본 논문에서는 보안 네트워크 관리 구조를 설계하고, 보안 네트워크 구성의 자동화를 지원하기 위한 여러 가지 알고리즘을 제안한다. 또한 기존의 네트워크 시스템과 보안 시스템을 통합 제어할 수 있도록 SNMP 기반의 보안 네트워크 토폴러지 맵(topology map) 자동 생성기를 제안한다.

본 논문은 다음과 같이 구성된다. 2장에서는 기존의 네트워크 관리 구조를 살펴보고, 3장에서는 본 논문에서 제안하는 보안 네트워크 관리 구조를 설계한다. 4장에서는 보안 네트워크 구성의 자동화를 위한 몇 가지 알고리즘을 제안하고, 5장에서는 시뮬레이션을 통하여 제안하는 구조의 성능을 평가한다. 6장에서는 보안 네트워크 토폴러지 맵 자동 생성기를 설계하고, 마지막으로 7장에서 결론을 맺는다.

Ⅱ. 관련 연구

2.1 SNMP 기반의 네트워크 관리

현재의 TCP/IP 환경에서는 SNMP(Simple Network Management Protocol) 프로토콜[8]을 기반으로 한 네트워크 관리가 가장 보편적이다. SNMP 프로토콜은 전송 장비와 단말 장치들을 관리하기 위한 메니저와 에이전트간 프로토콜이다. SNMP 프로토콜에서 관리 정보들은 MIB(Management Information Base) 객체로 표현된다. 각 MIB 객체는 OID (Object Identifier)가 할당되어 유일하게 식별 된다. MIB은 ASN.1(Abstract Syntax Notation One) 표현 방식을 사용하여 SMI(Structure of Management Information)로 기술된다. SMI는 MIB 객체들을 기술하는데 필요한 규칙을 정의하고 있다.

SNMP는 현재 세 개의 버전이 있다. 1988년에 버전 1이 제안되었고, 1993년에 버전 2, 그리고 1997년에 버전 3이 마지막으로 제안되었다[9].

SNMP 프로토콜에서 가장 많이 사용되고 있는 오퍼레이션은 Get, GetNext, Set, 그리고 Trap이다. Get은 관리 장치의 특정 MIB 정보를 검색할 때 사용하며, GetNext는 관리 장치의 특정 MIB 정보의 다음 MIB 정보를 검색할 때 사용하고, 그리고 Set은 관리 장치의 특정 MIB 값을 설정할 때 사용한다. Trap은 관리 장치에서 어떤 등록된 이벤트가 발생했을 때 이를 SNMP 메니저에게 알려줄 때 사용되는 오퍼레이션이다.

2.2 정책기반의 네트워크 관리 구조

IETF(Internet Engineering Task Force)에서는 정책 기반의 네트워크 관리 구조를 제안하였다. 이 구조의 가장 큰 장점은 역할 개념을 이용하여 네트워크 구성의 자동화를 가능하게 한다는 것이다. 여기서, 정책이란 조건과 액션으로 구성된 룰(rule)들의 집합을 말하며, 룰의 조건이 만족될 때 그 룰의 액션이 실행된다. 역할이란 관리되는 시스템의 고유한 관리적 특성을 말하며, 정책 룰에 대한 선택자로서 행동한다.

예를 들어, SNMP 프로토콜 기반의 관리 구조에서는 관리자가 각 네트워크 시스템의 유형을 고려하면서 파라미터 값을 설정하므로 수동적인 관리 구조이다. 그러나 정책기반의 관리 구조에서는 네트워크 시스템의 역할이 정책 룰에 명시된 역할과 일치하면 관리 서버가 그 정책 룰을 해당 네트워크 시스템에 자동적으로 설정하기 때문에 네트워크 구성의 자동화를 제공할 수 있다.

그림 1은 기존에 제안된 정책 기반의 네트워크 관리 구조이다. 이 구조는 정책을 관리하는 PMT (Policy Management Tool), 정책을 결정하는 PDP (Policy Decision Point), 정책을 실행하는 PEP (Policy Enforcement Point), 그리고 정책 저장소로 구성되어 있다.

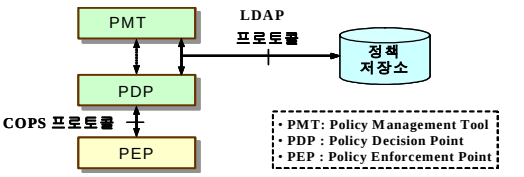


그림 1. 정책기반의 네트워크 관리 구조.
Fig. 1. Policy-based network management architecture.

정책 저장소 접근을 위하여 LDAP(Lightweight Directory Access Protocol) 프로토콜[10]이, 그리고 PDP와 PEP간의 정책 전달을 위하여 COPS(Common Open Policy Service) 프로토콜[11]이 정의되었다. 정책 그룹, 룰, 조건, 액션 등 의 정보 및 관계를 표현하기 위하여 PCIM(Policy Common Information Model)[12]이란 정보 모델이 정의되었다.

COPS 프로토콜에서는 Outsourcing모델과 Provisioning 모델을 정의하고 있다. Outsourcing 모델은 PEP가 정책 결정을 PDP에게 요구하면 PDP가 그것을 결정한 후 PEP에게 그 결과를 알려주는 모델이며, Provisioning 모델은 PEP가 정책을 요구하면 PDP가 해당 정책을 결정하여 PEP에게 전달하는 모델이다.

본 논문에서는 보안 네트워크의 구성 문제를 다루므로 Provisioning 모델을 따른다[13]. 이 모델에서 각 구성 요소들은 다음과 같이 동작한다. PMT는 사용자의 입력을 통하여 정책을 생성하고 정책 저장소에 저장한다. PDP는 PEP가 정책을 요구하면 정책 저장소로부터 PEP에 해당되는 정책을 결정하여 PEP에게 그 정책을 전달한다.

정책 기반의 네트워크 관리 구조는 네트워크 분야뿐만 아니라 네트워크 보안 분야에도 사용될 수 있도록 일반적인 모델로 정의되었지만, 보안 시스템 관리를 위해서는 정책뿐만 아니라 구조와 알고리즘 등 많은 부분이 확장되어야 한다.

Ⅲ. 보안 네트워크 관리 구조의 설계

본 장에서는 보안정책을 침입 탐지/대응 관점에서 효과적이고 효율적으로 보안 시스템에 자동 적용할 수 있는 보안 네트워크 관리 구조를 제안한다. 본 논문에서는 보안 네트워크상에서 관리될 보안정책으로써 공격탐지를 위한 침입탐지정책과 공격차단을 위한 침입대응정책만을 고려대상으로 한다.

본 논문에서 제안하는 보안 네트워크 관리 구조는 그림 2에 도시 되어 있다. 제안하는 구조는 사용자 인터페이스, 보안정책 서버, 보안망 정보수집 서버 그리고 보안 시스템으로 구성된다.

사용자 인터페이스는 보안정책 룰에 대한 생성/수정/삭제를 보안정책 서버에게 명령하며, 또한 보안 네트워크 상태 정보를 보안정책 서버 및 보안망 정보수집 서버로부터 보고 받는다.

보안정책 서버는 보안정책관리 모듈, 룰타이머관리 모듈, 보안정책결정 모듈, 그리고 경보정보분석

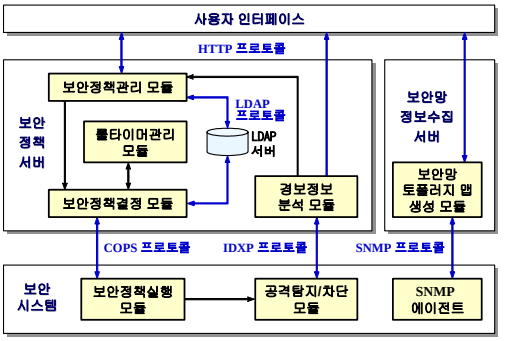


그림 2. 보안 네트워크 관리 구조.
Fig. 2. Security network management architecture.

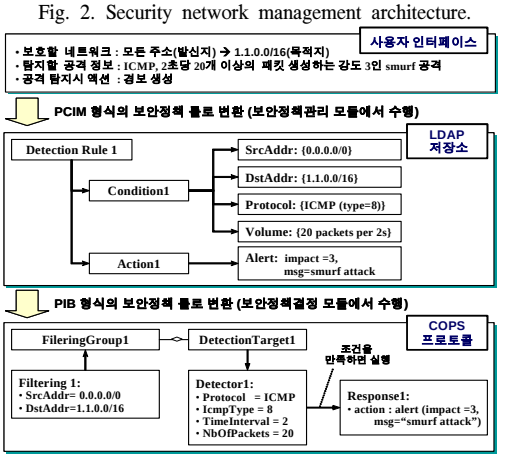


그림 3. 보안정책 룰의 표현과 변환.
Fig. 3. Representation and translation of security policy rule.

모듈로 구성되어 있다. 보안정책관리 모듈은 관리자가 사용자 인터페이스를 통하여 입력한 보안정책 정보를 LDAP 형식으로 변환하는 정책 변환, 보안정책 룰의 저장/삭제/검색을 위한 LDAP 서버 관리, 그리고 적용/해제할 보안정책 룰을 보안정책결정 모듈에게 알려주는 기능을 수행한다. 룰타이머관리 모듈은 보안 정책 룰의 유효 시간을 검사하는 기능을 수행한다.

보안정책결정 모듈은 LDAP 서버에 저장된 보안정책 룰을 PIB(Policy Information Base) 객체로 변환하는 기능, 보안 시스템의 역할과 위치 정보를 기반으로 한 보안정책 결정 기능, 그리고 또한 COPS 프로토콜을 사용하여 보안 시스템에게 보안정책을 전달하는 기능을 수행한다.

보안정책결정 모듈이 정책 표현 변환 기능을 수행하는 이유는 LDAP 프로토콜에서는 정책 룰을 PCIM 모델로 표현하고 COPS 프로토콜에서는 PIB

객체로 표현하기 때문이다. 그림 3은 사용자 인터페이스에서 입력한 공격 탐지(스머프 공격 탐지) 시그니처에 대한 PCIM 및 PIB 형식의 보안정책 룰 표현과 변환을 도시한 그림이다.

경보정보분석 모듈은 보안 시스템이 탐지한 경보정보를 분석하는 모듈이며, 보안정책관리 모듈에게 요구하여 침입대응정책을 자동 생성하기도 한다. 경보정보 전달용 프로토콜로는 IDXP(Intrusion Detection Exchange Protocol)[14]가 사용된다.

보안 시스템은 보안정책실행 모듈, 공격탐지/차단 모듈, 그리고 SNMP 에이전트로 구성된다. 보안정책실행 모듈은 보안 시스템의 역할 및 위치(보호할 망의 주소) 정보를 보안정책 서버에게 전달하는 기능과 보안정책 서버로부터 전달받은 PIB 객체를 내부 데이터 형식으로 변환하여 공격탐지/차단 모듈에 적용하는 기능을 수행한다.

본 논문에서 제안하는 보안정책 서버와 보안 시스템간의 기본적인 동작 시나리오는 다음과 같다.

1) 보안 시스템이 COPS 프로토콜의 OPEN과 REQUEST 메시지를 사용하여 그의 역할과 위치 정보를 보안정책 서버에게 제공하면서 보안 정책을 요구한다.
2) 보안정책 서버는 보안 시스템에 적용할 보안정책 (즉, 침입탐지정책 및 침입대응정책)을 결정한 후에 COPS 프로토콜의 DECISION 메시지를 사용하여 해당하는 보안정책을 내려준다.
3) 보안 시스템은 침입탐지정책을 사용하여 공격을 탐지하고, IDXP 프로토콜을 사용하여 보안정책 서버에게 경보정보(공격탐지정보)를 전달한다.
4) 보안정책 서버는 전달 받은 경보정보를 분석하여 침입대응정책을 자동으로 생성하고 또한 가장 효과적이고 효율적으로 차단할 수 있는 보안 시스템을 선택한 후, 그 보안 시스템에게 COPS 프로토콜의 DECISION 메시지를 사용하여 그 침입대응정책을 내려준다.

보안망 정보수집 서버는 SNMP 프로토콜을 사용하여 보안 시스템의 구성 및 성능 정보를 수집하는 기능을 수행한다. 본 논문에서는 기존의 네트워크 시스템과 보안 시스템을 통합 제어할 수 있도록 보안 네트워크 토폴러지 맵 자동 생성 기능을 특히 강조하며, 6장에서 자세히 설명한다.

Ⅳ. 보안 네트워크 구성의 자동화 방안

본 논문에서는 보안 네트워크 구성의 자동화를 위하여 다음과 같은 네 가지 보안 정책 결정 기능을 정의한다.

- 침입대응정책의 자동 생성 기능: 경보정보를 분석하여 침입대응정책을 자동 생성하는 기능.
- 시간정보기반의 보안정책 적용 기능: 시간정보에 따라서 보안정책 룰을 자동 적용/해제하는 기능.
- 위치기반의 대응할 보안시스템 선정 기능: 네트워크상의 보안 시스템 위치를 고려하여 대응할 보안 시스템을 선정하는 기능.
- 역할기반의 보안정책 결정 기능: 보안 시스템의 역할정보를 고려하여 보안 시스템에 적용할 보안정책을 결정하는 기능.

본 논문에서 제안하는 자동화된 보안정책 결정 기능은 각 보안 시스템이 필요로 하는 침입탐지정책을 자동으로 결정하고, 공격을 탐지하면 그에 대한 침입대응정책을 자동 생성하고, 그리고 대응할 보안 시스템을 자동 선정하여 특정 시간동안 공격을 자동으로 차단할 수 있는 능력을 제공한다.

4.1 침입대응정책의 자동 생성 기능

그림 2에서의 보안정책 서버는 다음과 같은 시나리오로 침입대응정책을 자동 생성한다.

1) 보안 시스템의 공격탐지/차단 모듈은 경보 정보를 보안정책 서버의 경보정보분석 모듈에게 제공한다.
2) 경보정보분석 모듈은 경보정보를 분석하여 위험한 공격으로 판단하면 새로운 침입대응정책을 생성할 수 있도록 보안정책관리 모듈에게 공격자의 정보(공격자의 IP 주소)와 대응방법(차단 또는 대역폭제한)을 제공한다.
3) 보안정책관리 모듈은 LDAP 서버에 침입대응정책을 저장한 후 보안정책결정 모듈에게 정책적용 명령을 한다.
4) 보안정책결정 모듈은 생성된 침입대응정책을 LDAP 서버로부터 검색하여 보안 시스템에게 전달한다.

보안정책 서버는 생성한 침입대응정책을 직접 적용할 수도 있지만, 관리자에게 그 침입대응정책을

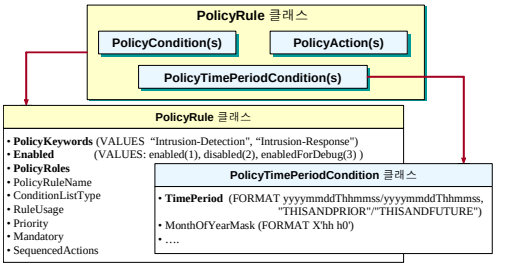


그림 4. PCIM에서 정의된 PolicyRule 클래스.
Fig. 4. PolicyRule class defined in PCIM.

추천하여 적용여부를 결정하게 하는 것도 가능하다. 이 경우에는 경보정보분석 모듈은 관리자에게 사용자 인터페이스를 통하여 공격자의 정보와 대응방법을 제공한다.

생성된 침입대응정책에서 각 룰은 그림 4에서 도시된 PCIM 모델의 *PolicyRule* 클래스로 표현된다. *PolicyRule*은 조건정보를 나타내는 *PolicyCondition* 클래스와 액션정보를 나타내는 *PolicyAction* 클래스, 그리고 보안정책 룰의 적용 및 해제 시간을 가리키는 *PolicyTimePeriodCondition* 클래스로 구성된다. *PolicyRule* 클래스의 대표적인 룰 속성 정보는 *PolicyKeywords*, *Enabled*, 그리고 *PolicyRoles* 이며 각각은 정책 종류(침입탐지 또는 침입대응), 활성화(보안 시스템에 적용) 여부, 그리고 역할 정보를 가리킨다.

예를 들어, 비활성화인 침입대응정책 룰의 경우 *PolicyRule* 클래스의 *PolicyKeywrods*와 *Enabled*는 각각 "*Intrusion-Response*"과 "disabled(0)"로 설정되며, 공격자 정보와 대응방법은 각각 *PolicyCondition* 클래스와 *PolicyAction* 클래스에 명시한다.

PolicyTimePeriod Condition 클래스의 이용 방법은 4.2절에서, *PolicyRoles* 룰 속성은 4.3과 4.4절에서 자세히 설명한다.

보안정책 룰은 생성이 되서 *Enabled* 룰 속성이 활성(*enabled*)/비활성(*disabled*)되다가, 결국 소멸된다. 그림 5는 룰의 상태가 변경될 때마다 처리해야 할 일련의 오퍼레이션을 정의한 룰 상태전이 알고리즘이다.

보안정책결정 모듈은 룰 상태전이 알고리즘을 수행하면서, 보안 시스템에 정책을 적용하거나 해제할 경우가 발생하면 보안정책결정 모듈에게 룰 번호와 룰에 대한 처리 ("*Install*" 또는 "*Remove*")를 명령한다. "*Install*"은 보안 시스템에게 룰 번호에 해당하는 보안정책 룰을 적용하라는 의미이고, "*Remove*"

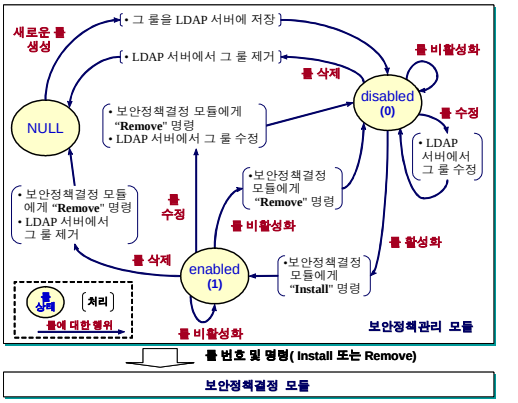


그림 5. 보안정책관리 모듈에서 룰 상태전이 알고리즘.
Fig. 5. Rule state transition algorithm in security policy management module.

는 룰 번호에 해당하는 보안정책 룰을 보안 시스템으로부터 제거하라는 의미이다. 보안정책관리 모듈은 보안정책결정 모듈에게 비활성화 상태인 룰이 활성화되었을 때 "*Install*" 명령을 내리고, 활성화 상태인 룰이 비 활성화되거나 수정 또는 삭제되었을 때 "*Remove*" 명령을 내린다.

4.2 시간정보기반의 보안정책 적용 기능

시간정보기반의 보안정책 적용 기능은 그림 4에 도시된 *PolicyTimePeriodCondition* 클래스의 *Time Period* 속성을 사용한다. 만약 *TimePeriod* 속성 값이 "*20050101T102030/20050202T090000*"라면, 그 룰은 2005.1.1일 10:20:30 초에 활성화되고, 2005.2.2일 09:00:00 초에 비 활성화된다.

시간정보를 가진 보안정책 룰은 룰타이머관리 모듈에서 처리되며, 적용/해제 시나리오는 그림 6에 도시 되어있다. 룰타이머관리 모듈은 두개의 타이머 이벤트 큐를 가지고 있다. 하나는 활성화를 기다리는 룰 리스트이고 또 다른 하나는 비활성화를 기다

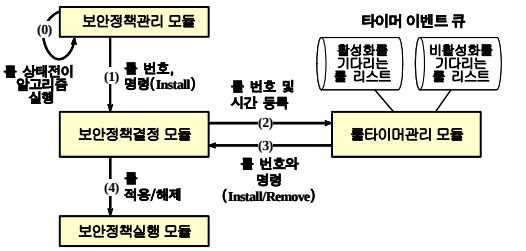


그림 6. 시간정보를 가진 보안정책 룰 적용/해제 시나리오.
Fig. 6. Scenario for installation/removal of security policy rule with time information.

리는 룰 리스트이다. 보안정책결정 모듈은 보안정책 관리 모듈로부터 전달 받은 룰이 시간정보를 포함하고 있으면 룰타이머관리 모듈에게 그 룰 번호와 시간을 등록한다. 룰타이머관리 모듈은 룰이 활성화/비활성화 할 시간이 되면 그 사실을 보안정책결정 모듈에게 알려준다. 보안정책결정 모듈은 그 명령에 따라서 해당 룰을 보안 시스템에 적용/해제 한다.

4.3 위치기반의 대응할 보안시스템 선정 기능

어떤 공격을 탐지했을 때, 그 공격을 가장 효과적이고 효율적으로 대응할 수 있는 보안 시스템을 선정하는 것은 매우 중요한 일이다. 이러한 결정을 관리자가 직접 하려면 보안 네트워크상의 모든 보안 시스템에 대한 정보를 미리 알고 있어야 하므로 쉽지 않고 또한 매우 귀찮은 작업이다.

본 논문에서는 위치기반의 대응할 보안시스템 자동 선정 기능을 지원하기 위하여, *"Ingress"*, *"Egress"*, *"Both"*, *"All"*, 그리고 *"보안시스템 주소"* 등 5 종류의 위치요구 정보를 정의한다. 이 정보는 그림 4에 도시 되어있는 *PolicyRule* 클래스의 *Policy Roles* 속성에 명시된다.

보안관리 서버는 보안정책 룰의 위치요구 정보와 그 룰의 조건정보에 있는 주소 정보, 그리고 각 보안 시스템의 위치 정보를 고려하여 정책을 적용할 최적의 보안 시스템을 선정한다. 즉, 보안정책 룰의 위치요구가 *"Ingress"* 이면 룰의 조건정보에 명시된 출발지 주소 정보와 가장 가까운 위치에 있는 보안 시스템을 선택한다. *"Egress"* 이면 룰의 조건정보에 명시된 목적지 주소와 가장 가까운 위치에 있는 보안 시스템을 선택한다. *"Both"* 인 경우에는 *"Ingress"* 역할 및 *"Egress"* 역할을 만족시키는 보안 시스템을, 그리고 *"All"* 인 경우는 모든 보안 시스템을 선택한다. *"보안시스템 주소"*는 적용할 보안시스템의 주소로써 관리자가 명시적으로 보안시스템을 지정할 때 사용한다.

침입대응정책 룰의 유형에 따라서 서로 다른 위치요구 정보를 설정함으로써 공격을 더 효과적이고 효율적으로 차단할 수 있다. 예를 들어, 공격탐지 정확도가 떨어지는 경우에는 목적지 망으로 들어오는 공격 트래픽만 차단할 수 있도록 침입대응정책 룰의 역할을 *"Egress"*로 설정한다. 그러나 공격자임이 분명한 경우에는 그 공격자를 원천 차단할 수 있도록 침입대응정책 룰의 역할을 *"Ingress"*로 설정한다. 분산 서비스 거부 공격과 같이 공격자가 다수이고 또한 네트워크 자원을 고갈시키는 공격의 경우에는 공격자 차단 및 네트워크 서비스 보호를 위하여 *"Both"*로 설정한다. 마지막으로 웜과 같이 전파가 빠른 공격의 경우에는 웜 공격 자체를 무력화시킬 수 있도록 침입대응정책 룰을 *"All"*로 설정하는 것이 효과적일 것이다.

```
Procedure is-ss-target ( ss,  rule )
// ss       : 보안 시스템의 정보 구조체를 가리키는 변수
// ss.addr : 보안 시스템의 IP 주소
// ss.pdAddr: 보안 시스템이 책임지고 있는 보호영역을 나타내는 IP 주소
// rule     : 적용할 룰의 정보 구조체를 가리키고 있는 변수
// rule.role: 룰의 역할 정보
// rule.src : 룰의 조건 절에 명시된 출발지 IP 주소
// rule.dst : 룰의 조건 절에 명시된 목적지 IP 주소
 case
    : rule.role includes "ingress" :
      if ( rule.src = ss.addr or rule.src in ss.pdAddr or rule.src = null )
            then return (TRUE)
      end
    : rule.role includes "egress" :
      if ( rule.dst = ss.addr or rule.dst in ss.pdAddr or rule.dst = null )
            then return (TRUE)
      end
    : rule.role includes "both" :
      if ( rule.src = ss.addr or rule.src in ss.pdAddr or rule.src = null ||
           rule.dst = ss.addr or rule.dst in ss.pdAddr or rule.dst = null )
            then return (TRUE)
      end
    : rule.role includes "all" :      return (TRUE)
    : rule.role includes ss.addr :    return (TRUE)
    : else :                          return (FALSE)
 end
end is-ss-target
```

그림 7. 정책 룰에 대한 보안시스템 적용여부 검사 알고리즘.
Fig. 7. Algorithm for checking whether or not this is the right security system to apply a security policy rule.

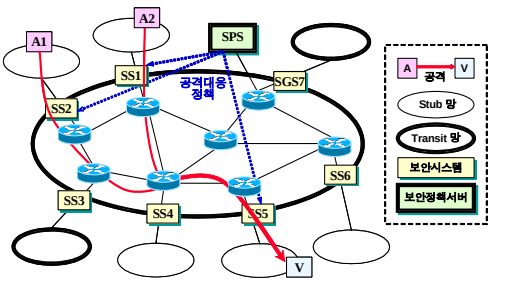


그림 8. 보안정책 서버의 대응 시나리오.
Fig. 8. Reaction scenario of security policy server.

그림 7은 주어진 한 보안 시스템이 보안정책 룰을 적용할 최적의 보안 시스템인지를 검사하는 알고리즘이다. 이 알고리즘에서 입력은 보안 시스템과 보안정책 룰이며, 만약 그 보안 시스템이 그 룰을 적용할 시스템으로 판단되면 *"TRUE"*를, 그렇지 않다면 *"FALSE"*를 리턴한다.

그림 8은 보안 시스템 SS5로부터 공격 보고를 받은 보안정책 서버가 *"Both"* 역할을 갖는 침입대응정책 룰을 자동 생성하고, 해당 보안 시스템에게 그 정책을 분배하는 상황을 도시한 그림이다. 공격탐지를 보고 받은 보안정책 서버 SPS는 그 공격을

차단하기 위하여 SS5뿐만 아니라 SS1과 SS2 에게도 침입대응정책 룰을 전달하고 있다.

4.4. 역할기반의 보안정책 결정 기능

보안 네트워크 관리자는 보안 네트워크을 구성하는데 있어서 서로 다른 요구사항을 가질 수 있다. 예를 들어, 보안보다 네트워크 서비스를 더 중요시하는 관리자는 오탐으로 인하여 네트워크 서비스 품질이 떨어지지 않도록 치명적인 공격만 탐지/차단하는 보안 네트워크 구성을 요구할 수 있다. 그 반대로, 보안을 더 중요시하는 관리자는 공격의 강도에 상관없이 모든 종류의 공격을 탐지/차단하는 보안 네트워크 구성을 요구할 수 있다.

한편, 보안 시스템도 서로 다른 탐지/차단 능력을 갖는다. 예를 들어, 방화벽은 차단 기능만 수행하고, 침입 방지 시스템은 침입 탐지와 차단 기능을 동시에 수행한다. 또한 같은 침입 탐지 시스템이더라도, 그 탐지 방법에 따라서 시스템 로그 파일을 분석하여 침입을 탐지하는 호스트기반의 침입 탐지 시스템과 네트워크 트래픽을 분석하여 침입을 탐지하는 네트워크기반의 침입 탐지 시스템이 있다.

본 논문에서는 이러한 보안 네트워크 구성에 대한 다양한 요구사항을 만족시킬 목적으로, 공격 위험도 (*severity*) 정보와 적용할 보안 시스템 유형정보 (*detectionType*)를 보안정책 룰의 역할 속성 (즉 그림 4의 *PolicyRoles* 룰 속성)에 명시하고 또한 각 보안 시스템의 보안 요구사항 (*severityRequest*) 및 탐지 요구사항 (*detectionType*)을 보안 시스템의 역할/능력 정보에 명시한다. 보안 시스템은 이러한 보안 시스템의 역할/능력 정보를 보안정책 서버와 처음 접속할 때 알려주어야 한다. 그림 9는 보안정책 서버가 보안정책을 요구하는 보안 시스템에게 그 역할과 능력정보를 고려하여 해당하는 보안정책을 결정하여 전달하는 알고리즘이다.

Procedure ***policy-decision-triggered-by-policy-request (ss)***

```
// ss : 보안정책을 요구하는 보안 시스템
// ss.severityRequest: 침명적인 공격 탐지 (값=2), 모든 공격 탐지(값=1)
// ss.detectionType: 호스트기반 공격탐지 또는 네트워크기반 공격탐지

if ( ss.IntrusionDetectionCapability = TRUE ) then
    LDAP 서버에서 침입탐지정책을 검색.
    그 검색결과는 rulePtr이 가리킴.
    while ( rulePtr ≠ null ) do
        if ( ss.severityRequest ≥ rulePtr.severity and
             ss.detectionType in rulePtr.role and
             [call is-ss-target(ss, rulePtr)] = TRUE ) then
                call translate-rule-into-pib (rulePtr, pibInstance)
                COPS 프로토콜을 사용하여 룰을 ss에 적용함.
        end
        rulePtr ← rulePtr.next
    end
end

if ( ss has IntrusionResponseCapability ) then
    LDAP 서버에서 침입대응정책을 검색.
    ……..
```

그림 9. 역할 및 능력기반의 보안정책결정 알고리즘.
Fig. 9. Role & capability-based security policy decision algorithm.

Ⅴ. 자동 침입 대응 성능 평가

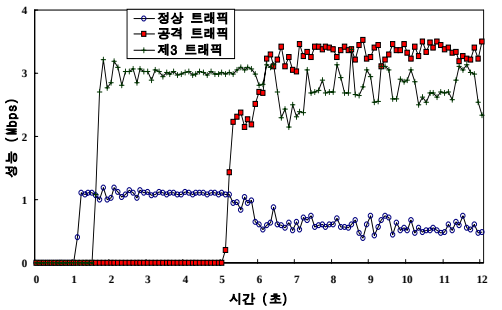
본 장에서는 본 논문에서 제안하는 보안 네트워크 관리 구조에서 자동 대응에 대한 성능을 평가한다. 이를 위하여 ns(network simulator) 시뮬레이터[15]를 확장하여 침입 탐지/차단 기능과 COPS 프로토콜을 구현하였다.

본 논문에서는 그림 8을 실험 망으로 시뮬레이션하고, 또한 세 종류의 트래픽(정상, 공격, 그리고 제 3 트래픽)을 정의한다. 정상 트래픽은 시스템 V의 서비스를 이용하는 트래픽이고, 공격 트래픽은 시스템 V를 서비스 거부 공격[16] 하는 트래픽이다. 제 3 트래픽은 시스템 V이외의 다른 시스템의 서비스를 이용하지만 공격 트래픽과 라우팅 경로가 같은 트래픽을 말한다. 본 시뮬레이션에서 각 보안 시스템은 보안정책 서버로부터 특정 대역폭이상인 혼잡 유발 플로우를 공격 트래픽으로 간주하는 침입탐지정책을 전달받아 실행한다. 보안정책 서버는 보안 시스템으로부터 경보를 받으면 자동으로 침입대응정책 룰을 생성하여 해당 보안 시스템에게 전달한다.

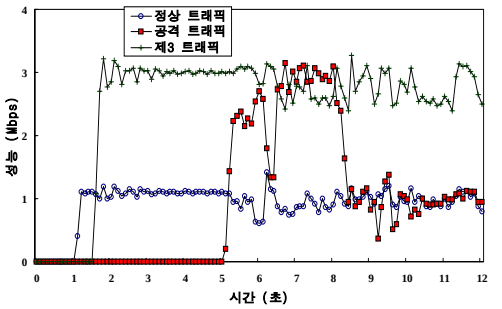
그림 10-(a)는 공격에 대하여 어떠한 대응도 하지 않는 경우에 대한 실험 결과이다. 서비스 거부 공격이 시작된 약 5초부터 정상 트래픽뿐만 아니라 제 3 트래픽도 그 성능이 감소하는 것을 볼 수 있다.

그림 10-(b)는 자동 생성된 대응정책 룰의 역할을 "*egress*" 로 설정하여, 공격을 탐지한 보안 시스템, SS5에게만 대응 정책을 적용했을 때의 실험 결과이다. 공격 트래픽의 양을 반 정도 차단함으로써 정상 트래픽은 어느 정도 보호되었지만, 제 3 트래픽은 여전히 공격 트래픽의 영향을 받고 있다. 그 이유는 공격 트래픽이 목적지 stub 망에서는 SS5에 의하여 제어되었지만, transit 망에서는 여전히 혼잡을 발생시키고 있기 때문이다.

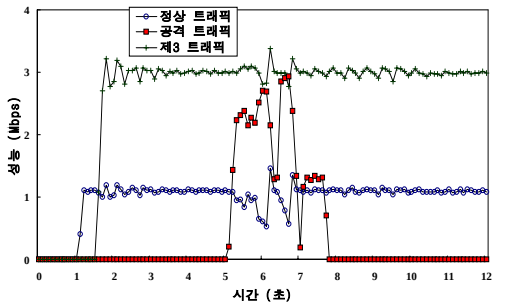
그림 10-(c)는 자동 생성된 대응정책 룰의 역할을 "*both*"로 설정하여, 공격을 탐지한 SS5 뿐만 아니라 SS1과 SS2에게도 대응 정책을 적용하였을 때의 실험 결과이다. 그림 10-(c)에서 보여지듯이 공격 트래픽을 완전히 차단하여 정상 트래픽과 제 3 트래픽 모두를 보호하고 있다.



(a) 침입대응정책을 보안시스템에 적용하지 않은 경우.



(b) "*egress*" 인 역할을 갖는 침입대응정책 적용.



(c) "*both*" 인 역할을 갖는 침입대응정책 적용.

그림 10. 보안 네트워크 관리 구조에서의 자동 침입 대응.
Fig. 10. Automatic intrusion response in security network management architecture.

본 실험을 통하여, 제안하는 보안 네트워크 관리 구조의 자동 침입 대응 방안이 성공적으로 수행함을 검증하였고, 또한 분산 서비스 거부 공격을 완벽하게 차단하기 위해서는 침입대응정책 룰을 "*Both*"로 설정하는 것이 효과적이라는 사실을 확인하였다.

Ⅵ. 보안 네트워크 토폴러지 맵 자동 생성기

본 논문에서는 기존의 네트워크 시스템과 보안 시스템을 통합 제어할 수 있도록 SNMP 기반의 보

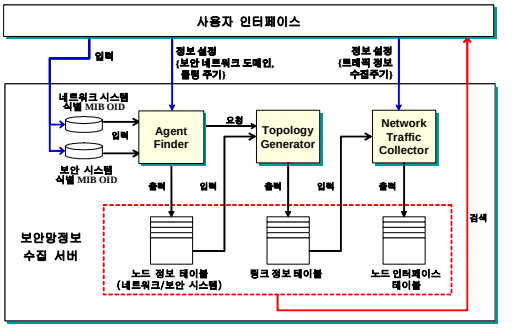


그림 11. 보안 네트워크 토폴러지 맵 생성 모듈의 구조.
Fig. 11. Architecture of security network topology map generation module.

안 네트워크 토폴러지 맵 자동 생성기를 제안한다. 제안하는 보안 네트워크 토폴러지 맵 생성기는 보안 네트워크에 대한 토폴러지 맵을 자동 생성하고 성능 정보를 주기적으로 수집하여 사용자에게 보여주는 기능을 제공한다.

보안 네트워크 토폴러지 맵 생성 모듈의 구조는 그림 11에 도시 되어 있다. 모두 세 개의 컴포넌트로 구성되어 있으며, 각각의 기능은 다음과 같다.

먼저, Agent-Finder는 관리자로부터 네트워크 시스템과 보안 시스템을 식별할 수 있는 MIB 객체 식별자, 탐색할 보안 네트워크 도메인, 폴링 주기 시간 그리고 트래픽 정보 수집 주기 등을 입력 받아 보안 네트워크 도메인에 있는 모든 시스템을 폴링하고, 그 결과로서 탐색된 네트워크/보안 시스템들에 대한 시스템 속성 정보 (즉, SNMP MIB 정보인 mib-2.system 정보)를 노드정보 테이블에 저장한다. 노드정보 테이블은 [노드 주소, 노드 위치(보안 네트워크 내부 또는 외부), 노드 유형(보안 시스템 또는 네트워크 시스템)]으로 구성된다.

Topology-Generator는 탐색된 노드들에 대한 시스템 인터페이스 주소 정보(즉, mib-2.ip. ipAddrTable. ipAddrEntry 정보), 시스템 인터페이스 속성 정보 (즉, mib-2.interfaces.ifTable.ifEntry 정보), 그리고 시스템 라우팅 정보 (즉, mib-2.ip. ipRouteTable. ipRouteEntry 정보)를 분석하여 링크 정보를 생성하고 그것을 링크정보 테이블에 저장한다. 링크정보 테이블은 [노드-a 주소, 인터페이스-a 주소, 노드-b 주소, 인터페이스-b 주소]로 구성된다. Topology-Generator가 링크정보 테이블을 생성하는 알고리즘은 다음과 같다.

1) 노드정보 테이블에 등록된 노드의 시스템 인

터페이스 주소 정보를 수집하여 [노드 주소, 인터페이스 주소, 인터페이스 인덱스]로 구성된 노드주소정보 임시테이블을 생성.

2) 노드주소정보 임시테이블에 등록된 인터페이스 주소별로 시스템 라우팅 정보를 수집하여 [노드 주소, 인터페이스 인덱스, 다음 노드 주소]로 구성된 라우팅정보 임시테이블을 생성.
3) 두 임시 테이블 결합하여 링크정보 테이블 생성.

마지막으로 Network-Traffic-Collector는 링크정보 테이블에 있는 링크들에 대한 구성 및 성능 정보를 수집하여 노드인터페이스 테이블에 저장한다. 노드 인터페이스 테이블은 [노드 주소, 인터페이스 주소, 인터페이스 타입, 대역폭, 수신 옥텍 수, 평균 수신 속도, 발신 옥텍 수, 평균 발신 속도]로 구성된다.

그림 12는 보안 네트워크 토폴러지 맵 생성기를 구현한 GUI이다.

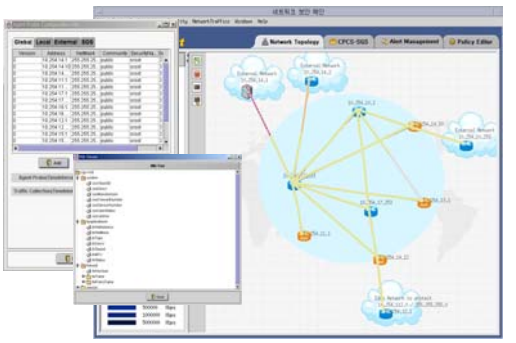


그림 12. 보안 네트워크 토폴러지 맵 생성기의 GUI.
Fig. 12. GUI of security network topology map generator.

Ⅶ. 결론 및 향후연구과제

본 논문에서는 대규모 망에서 보안 시스템 관리의 자동화를 위한 보안 네트워크 관리 구조와 방법을 제안하였다. 본 논문에서는 보안네트워크 구성의 자동화를 위한 보안 정책 결정 기능으로써 대응정책의 자동 생성 기능, 시간정보기반의 보안정책 적용 기능, 위치기반의 대응할 보안시스템 선정 기능, 그리고 역할기반의 보안정책 적용 기능 등 네 가지 기능을 정의하고, 각각에 대한 알고리즘을 제안하였다. 아울러, 본 논문에서는 기존의 네트워크 시스템과 보안 시스템을 통합 제어할 수 있도록 SNMP 기반의 보안 네트워크 토폴러지 맵 자동 생성기를 설계하였다.

본 논문에서 제안하는 보안 네트워크 관리 구조와 자동 대응 메커니즘에 대한 성능 평가를 위하여 서비스 거부 공격에 대한 차단 실험을 수행하였다. 실험을 통하여 네트워크 공격을 원천적으로 차단하기 위해서는 각 보안 시스템의 자동화된 협력이 요구되며, 본 논문에서 제안하는 보안 네트워크 관리 구조의 자동 침입 대응 방안이 그 요구 사항을 만족시킬 수 있음을 확인할 수 있었다.

본 논문에서 제안한 보안 네트워크 관리 구조 및 방법은 현재 구현이 완료된 상태이다. 실제의 네트워크 환경에서 제안된 구조 및 방법이 어떠한 성능 저하 없이 정상적으로 운용될 수 있는지를 시험하고 보완하는 것과 보안 관리의 자동화 관점에서 새로운 사용자의 요구 사항을 도출하는 것은 향후연구과제로서 남긴다.

참 고 문 헌

[1] Check Point, “Open Platform for Security,” Technical Note, 2000.
[2] Cisco, “Network Security - Embedded in the Network, Integrated in the Product”, white paper, 2002.
[3] R. Mahajan, S. M. Bellovin, S. Floyd, and et al., “Controlling High Bandwidth Aggregates in the Network,” ACM SIGCOMM Computer Communications Review, Vol. 32, No. 3, pp. 62-73, July 2002.
[4] D.K.Y. Yau, J.C.S. Lui, and Feng Liang, “Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles,” 10th IEEE International Workshop on Quality of Service, pp.35 -44, May 2002.
[5] Verma, Dinesh, “Policy Based Networking,” New Riders, November 2000.
[6] Raouf Boutaba, Salima Omari and Ajay Pal Singh Virk, “SELFCON: An Architecture for Self-Configuration of Networks”, IEEE Journal of communications and networks, VOL.3, NO.4, Dec. 2001.
[7] R. Sahita, S. Hahn, K. Chan, and K. McCloghrie, “Framework Policy Information Base,” IETF, RFC 3318, March, 2003.
[8] J. Case,M. Fedor,M. Schoffstall, and J.

Davin, "A Simple Network Management Protocol(SNMP)", IETF, RFC 1157, May 1990.
[9] William Stallings, "SNMPv3: A Security Enhancement for SNMP," IEEE Communications Surveys, Vol. 1 No. 1, Fourth Quarter 1998.
[10] Y. Yaacovi, M.Wahl and T. Genovese, "Lightweight directory access protocol(v3): Extensions for dynamic directory services," IETF, RFC 2589, May 1999.
[11] D. Durham, J. Boyle, R. Cohen, and et. al., "The COPS (Common Open Policy Service) Protocol," IETF, RFC 2748, Jan. 2000.
[12] B. Moore, E. Ellesson, J. Strassner, and A. Westerinen, Strassner, "Policy Core Information Model - Version 1 Specification," IETF, RFC 3060, Feb. 2001.
[13] K. Chan, J. Seligson, D. Durham, and et. al., "COPS Usage for Policy Provisioning (COPS-PR)," IETF, RFC 3084, March 2001.
[14] B. Feinstein, G. Matthews, and J. White, "The Intrusion Detection Exchange Protocol (IDXP)", IETF, draft-ietf-idwg-beep-idxp-07, Oct. 2002.
[15] UCB/LBNL/VINT, "ns Notes and Documentation," http://www.isi.edu/nsnam/ns.
[16] X. Geng and A. B. Whinston, "Defeating Distributed Denial of Service Attacks", IT Pro, pp 36-41, July 2000.

안 개 일 (Gae-Il Ahn) 정회원

1993년 2월 충남대학교 컴퓨터 공학과 졸업
1995년 2월 충남대학교 컴퓨터 공학과(석사)
2001년 8월 충남대학교 컴퓨터 공학과(박사)
2001년 8월~현재 한국전자통신연구원 선임연구원
<관심분야> 컴퓨터 네트워크, 네트워크 보안, 트래픽 엔지니어링, 네트워크 시뮬레이션