

취약성 정보를 활용한 정책 기반 보안 시스템 모델링

서희석* · 김동수** · 김희완***

Policy-based Security System Modeling using Vulnerable Information

Hee Suk Seo* · Dong Soo Kim** · Hee Wan Kim***

■ Abstract ■

As the importance and the need for network security is increased, many organization uses the various security systems. They enable to construct the consistent integrated security environment by sharing the vulnerable information among firewall, intrusion detection system, and vulnerable scanner. And Policy-based network provides a means by which the management process can be simplified and largely automated. In this article we build a foundation of policy-based network modeling environment. The procedure and structure for policy rule induction from vulnerabilities stored in SVDB (Simulation based Vulnerability Data Based) is conducted. It also transforms the policy rules into PCIM (Policy Core Information Model).

Keyword : Policy-based Framework, Vulnerability Database, Network Security, Security System Modeling

1. 서 론

컴퓨터를 사용한 업무의 처리가 늘어남에 따라 컴퓨터의 성능이 급격하게 증가하고 있으며, 인터

넷의 사용과 같은 통신의 발전으로 인해 현재의 네트워크 장비는 다양한 종류의 서버, 라우터, 스위치, 게이트웨이 등으로 구성되고 있다. 이 장비들은 다양한 제조자로부터 생산되고 있으며 네트워

* 성균관대학교 정보통신공학부

** 국민대학교 BIT 대학원

*** 삼육대학교 컴퓨터과학과

크 구성 또한 복잡하게 이루어진다. 이러한 시스템의 기술적 복잡도 증가는 서비스를 위한 대역폭 확보로 인한 장점도 있지만 끊임없이 새로운 기술을 배워야 하며 이로 인한 인적 자원 비용의 상승을 발생시킨다. 또한 다양한 보안 제품의 출시와 이들 간의 상이한 특성으로 인해 효율적인 운용과 유지에 어려움이 있다. 이로 인해 이들 간의 체계적이며 일관적인 보안 관리 체계의 필요성이 증가하고 있다[1]. 이러한 어려움에 직면한 네트워크 관리자는 더욱 많은 양의 전문적 지식이 필요하게 되었으며 네트워크 관리를 위한 수작업이 증가하게 되었다. 이러한 문제를 해결하기 위한 한 방편이 정책 기반 프레임워크를 사용하는 것이다. 정책 기반 프레임워크에서의 네트워크 관리자는 자원이나 서비스가 어떻게 사용되는지를 정책으로 정의하고, 정책 기반 관리 시스템은 이렇게 정의된 정책이 네트워크에 적용될 수 있는 형태로 변형하며 이러한 설정을 네트워크에 적용하게 된다. 정책 기반 프레임워크를 통한 가장 중요한 이점은 네트워크 관리 프로세스의 단순화와 자동화이다[2].

정책 기반 프레임워크를 구성하는 다양한 네트워크의 구성 요소들을 모델링하게 위하여 DEVS(Discrete Event system Specification) 방법론[3, 4]을 사용하여 모델을 구축하였다. 또한 본 논문에서는 취약점 데이터 베이스를 이용한 보안 정책의 유도와 취약점 데이터 베이스의 정책을 정책 기반 프레임워크에 적용하는 방법을 소개할 것이다. 이를 위하여 본 연구진은 여러 보안 시스템 모델들이 사용 할 수 있는 취약성 정보들을 집약시킴으로써 보안 시스템간의 정보 공유를 쉽게 할 수 있는 SVDB(Simulation based Vulnerability Data Base)를 구축하였다. 이로부터 보안 정책을 유도하였으며 이 정책을 정책 기반 프레임워크에 적용을 위한 인터페이스를 설계하고 구현하였다. 본 연구진은 구성된 모델들의 구조와 문제 해결을 위한 알고리즘을 소개할 것이다.

정책 기반 프레임워크에서 취약성 데이터 베이스를 이용한 정책 유도와 적용을 검증하기 위해 서

비스 거부 공격(Denial of Service) 공격 중의 하나인 jolt2 공격을 사용하여 모델의 동작을 설명할 것이다. 서비스 거부 공격은 많은 양의 트래픽을 발생시켜 이를 공격 대상 시스템에 보냄으로써 대상 시스템이 정상적인 동작을 수행할 수 없게 하거나 시스템을 전복시키는 공격이다. jolt2 공격은 상위 계층에서 만들어진 데이터를 많은 IP(Internet Protocol) 데이터그램으로 쪼개 공격 대상 시스템에 보내는 공격이다. 이러한 패킷을 수신한 시스템은 이를 수신하고 제조립하는데 많은 자원을 소비하게 되며 결국은 CPU(Central Process Unit)의 사용량이 거의 100%에 도달하도록 하는 공격이다.

본 논문의 구성은 다음과 같다. 2장에서는 정책 기반 시뮬레이션 환경 구축을 위해 사용된 이론 및 시스템에 관해 설명할 것이고, 3장에서 보안 모델링을 위해 구성된 각 모델들에 대해 설명할 것이다. 4장에서는 SVDB와 정책 기반 프레임워크의 통합을 위한 모듈에 대해서 설명할 것이다. 마지막으로 5장에서는 결론에 대하여 설명할 것이다.

2. 배경 이론 및 정책 기반 프레임워크

이 장에서는 논문의 배경 이론 및 시스템에 대해서 설명한다. 2.1에서는 네트워크 구성 요소들을 구성하는데 사용된 모델링 이론인 DEVS 형식론에 대해서 설명하고, 2.2에서는 취약성 데이터 베이스에 대해 각각 설명한다. 2.3에서는 정책 기반 프레임워크에 대해서 설명하고 2.4에서는 정책 표현에 대해서 설명할 것이다.

2.1 DEVS formalism

Zeigler에 의해 정립된 DEVS 방법론은 연속적인 시간상에서 발생되는 이산 사건을 처리하는 시스템을 시뮬레이션 하기 위해 이론적으로 정립된 모델링 방법론이다[3, 4]. 이는 모델의 구조와 행동을 시뮬레이션 수행으로부터 추상화시키기 위해 모델을

집합 이론적 방법으로 이용한 것으로, 시스템을 계층적(hierarchical)이고 모듈화(modular)된 형식으로 기술한다. 본 논문에서 구성된 모델들은 이러한 DEVS 모델을 사용하여 계층적으로 구성하였다.

DEVS에서는 기본(Basic) 모델과 결합(Coupled) 모델을 정의한다. 기본 모델은 시스템의 동적인 특성을 표현하기 위한 모델이고, 결합 모델은 시스템의 구성 요소간의 상호작용을 표현하기 위한 모델이다. 실제 모델의 동작을 나타내는 것이 기본 모델이며 결합 모델은 기본 모델의 결합을 이루도록 도와주는 모델이다. 이 모델들은 다음의 항들로 명세 할 수 있다.

$$M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, t_a \rangle$$

기본 모델을 나타내는 M에서 X는 입력 사건의 집합이고 S는 상태들의 집합이다. Y는 출력 사건의 집합, δ_{int} 는 내부 상태 변이 함수이고 δ_{ext} 는 외부 상태 변이 함수이다. λ 는 출력 함수이고 t_a 는 시간 갱신 함수이다.

$$DN = \langle D, \{M_i\}, \{I_i\}, \{Z_{i,j}\}, select \rangle$$

결합 모델을 나타내는 DN은 다음과 같이 명세 할 수 있다. D는 구성 요소 이름의 집합이고 $\{M_i\}$ 는 구성 모델을 나타낸다. $\{I_i\}$ 는 모델 i와 연관된 모델의 집합을 의미하며 $\{Z_{i,j}\}$ 는 모델 i와 모델 j간의 연결 함수를 나타낸다. select은 tie-breaking selection 함수를 나타낸다.

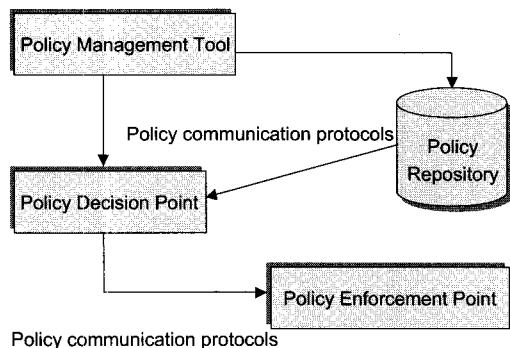
2.2 취약점 데이터 베이스

취약성이란 위협 요소에 의해 침해될 수 있는 보안 절차, 기술적 통제, 물리적 통제, 기타 다른 통제들 내의 어떤 조건이나 결점이다[5]. 취약성 분석의 목적은 분류의 방법이나 분류의 집합을 마련하는 것이다. 또한 취약성의 집합으로부터 원하는 정보를 추상화하는 것을 가능하게 한다. 이러한 정보들은 침입 탐지 시스템의 시그너처(signature), 공격자가 다른 취약성들을 이용하기 위한 시스템

환경 등으로 이루어진다[6].

각 국의 CERT(Computer Emergency Response Team), 보안 회사의 게시판 및 운영체제와 응용 프로그램의 개발 회사에서는 이러한 취약성들을 분석, 보고하여 취약성들로 인한 피해를 최소화하려는 노력을 한다. ISS(Internet Security Systems, Inc.), SecurityFocus.com에서는 데이터베이스를 운영하고 있으며, NIST(National Institute of Standards and Technology)에서는 산재해있는 취약점 데이터 베이스를 일괄적으로 참조할 수 있는 취약점 메타베이스를 운영하고 있다. 이러한 취약성 정보들은 같은 취약성이라 할지라도 서로 다른 이름을 가지고 있다. 보안 시스템 모델들이 취약성 정보를 공유하기 위해, 취약성 정보의 유일성을 보장하기 위해 여러 보안 관련 기관들이 참여해 만든 CVE(Common Vulnerability and Exposures)라는 이름을 사용한다[7].

2.3 정책 기반 프레임워크



[그림 1] IETF의 정책 기반 프레임워크

본 논문에서는 네트워크 영역의 정책 분배와 설정에 사용되는 표준인 IETF(Internet Engineering Task Force) 정책 프레임워크를 적용한다. 정책 프레임워크는 [그림 1]과 같이 네 개의 요소로 이루어져 있다[8, 9].

- 정책 관리 툴(PMT : Policy Management Tool)
정책 관리 툴은 정책을 구성하고, 정책을 배치하고, 그리고 정책 관리 환경의 상태 모니터를

위한 사용자 인터페이스이다. 정책 관리 툴의 핵심적인 기능은 인간이 이해 가능한 추상적인 형태의 규칙을 정책 저장소의 정책 정보 모델에 맞게 변환하여 정책 저장소에 저장하는 것이다. 이를 위해 정책 관리 도구는 단순한 문장 검사 뿐만 아니라 규칙의 충돌, 규칙의 실행 가능성 등 의미론적인 검사와 같은 검증 작업을 시행한다.

- 정책 결정 지점(PDP : Policy Decision Point)
정책 결정 지점은 정책 해석과 배치에 관한 작업을 한다. 정책 저장소에 저장된 정책과 관련 데이터로부터 PEP(Policy Enforcement Point)가 받아들일 수 있는 형태와 구문으로 변환한다. 또한 실행 감지와 조작, 규칙의 네트워크와 자원에 따른 정책 검증, 조정과 정책의 실행 감지, 조작, 적용 분석을 위해 PEP와 통신한다.
- 정책 시행 지점(PEP)
정책 시행 지점은 정책을 시행하고 적용하는 작업을 한다. PEP는 에이전트로서 장치 안에서 동작하거나, 응용의 형태로 존재할 수도 있다. 또한 정책을 수행한 결과나 PEP 내의 동적인 정보들을 PDP에 보고한다.
- 정책 저장소(Policy Repository)
정책과 관련된 정보를 저장하기 위한 저장소이다. 디렉토리 또는 관계형 데이터 베이스의 형태로 저장된다. 서로 다른 제조자의 제품간의 상호 연동을 보장하기 위해서 저장된 정보는 정책 프레임 WG(Working Group)에서 정의한 정보 모델에 맞춰 저장되어야 한다.
- 정책 통신 프로토콜(PCT : Policy Communication Protocols)
정책 저장소로부터 읽고 쓰기 위한 프로토콜(LD-AP)과 PDP와 PEP간의 통신하기 위한 프로토콜(COPS, SNMP)이 사용된다.

2.4 정책 표현 방법

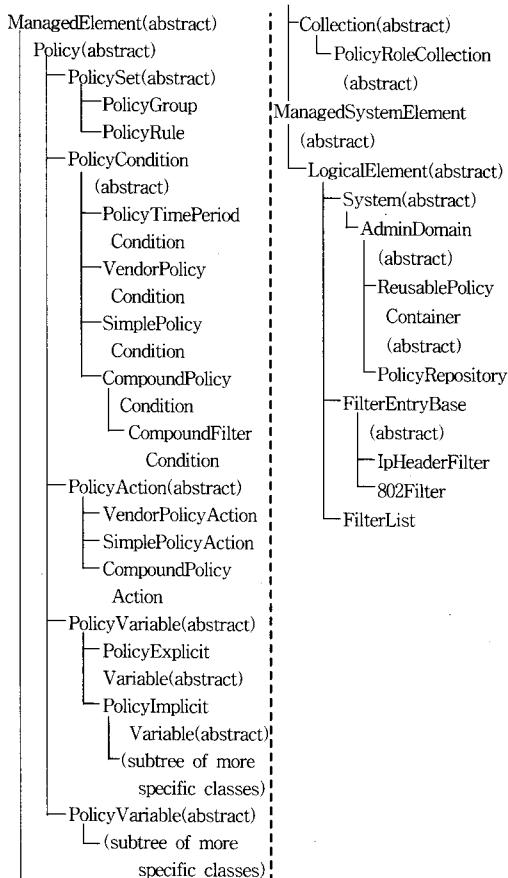
네트워크 관리를 위해 필요한 상위 계층과 하위

계층 정책은 다양한 방법으로 정의될 수 있다[2].

인간을 위한 관점에서 상위 계층 정책을 명세하는 가장 좋은 방법은 자연어로 기술하는 것이다. 하지만 이러한 접근법으로 현재 자연어 처리 기술의 한계로 정책 기술에는 사용되지 않는다. 다음 접근법으로는 컴퓨터에서 번역되고 처리될 수 있는 특별한 언어로 정책을 표현할 수 있다. 정책은 컴퓨터에서 번역될 수 있는 프로그램처럼 기술되었을 때, 실행하는 것이 가능하다. 하지만 서로 다른 두 개의 프로그램에 의해 기술된 정책이 서로 일치 여부를 결정하는 것은 매우 어렵다는 단점이 있다. 가장 간단한 접근법은 정책을 규칙 기반으로 표현하는 것이다. 각 규칙은 간단한 컨디션과 액션의 쌍으로 구성되어 있다. 정책 규칙은 컨디션(condition)의 집합과 대응하는 액션(action)의 집합의 결합으로 구성된다. 이러한 정책 형태는 “IF condition THEN action”의 구조를 갖는다. 또한 규칙의 충돌을 해결하기 위해서 정책은 우선순위를 가질 수 있다. IETF에서는 규칙 기반 정책 표현을 선택하여 표준화를 진행하였다. 정책 정보를 표현하기 위한 방법으로는 IETF와 DMTF/Desktop Management Task Force)에서 제시된 PCIM(Policy Core Information Model)을 사용한다. PCIM은 정책 정보 모델을 표현하기 위한 객체 지향 정보 모델이다[9]. 정책 정보는 정책의 제어와 정책 정보를 표현하는 구조 클래스와 구조 클래스의 상호 연관성을 나타내는 연관 클래스를 사용하여 정의된다. 정책은 정책 규칙들의 집합을 사용하여 적용되고, 각 정책 규칙은 조건들의 집합과 반응들의 집합으로 구성된다. 여러 정책 규칙들은 정책 그룹들과 결합되고, 이러한 그룹들은 또 다른 그룹을 구성할 수 있다. [그림 2]는 PICMe(PCIM extensions)의 구조 클래스의 상속 계층을 나타낸다[10].

마지막으로 테이블 표현 접근법이 있다. 이는 정책을 테이블 안의 엔터티처럼 표현하는 것이다. 애드리뷰트의 일부는 컨디션 파트를 구성한다. 그리고 다른 일부는 액션 파트를 구성한다. 이러한 테이블 표현은 규칙 기반 표현이 가능한 대부분의 정

책을 표현할 수 있다. 이 방법은 일치성 검사와 같은 정책의 검증 작업에 효율적이다.



[그림 2] PCI-Me의 구조 클래스 상속도

3. 네트워크 보안 모델링

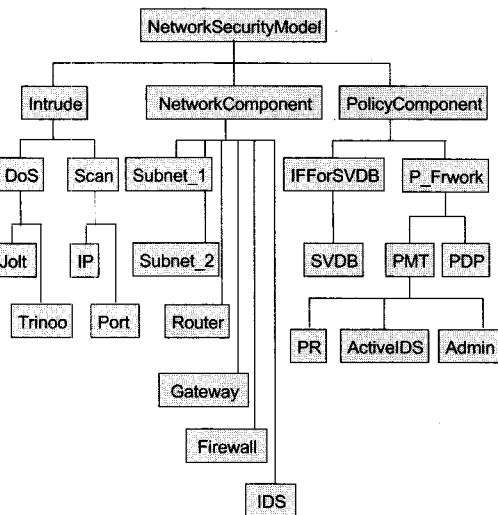
이번 장에서는 모델링된 네트워크 구성 요소들에 대해서 설명한다. 3.1에서는 설정된 네트워크 구조를 설명하고, 3.2에서는 SVDB에 대해서 설명한다. 3.3에서는 구성된 세부 모델을 살펴본다.

3.1 네트워크 구조

SES(System Entity Structure)[3, 4]는 시스템의 구조적인 지식을 표현할 수 있는 방법을 제공한다. 이 장에서는 SES를 사용하여 구성된 네트워크를

계층적으로 설명한다. SES는 분해(decomposition), 분류(taxonomy)와 연결 관계(coupling relationship)가 결합된 지식 표현 방법이다. 각 개체(entity)와 개체와의 관계는 분해과 세분화(specialization)의 관계로 표현된다.

[그림 3]은 구성된 전체 네트워크 모델의 SES를 나타낸다. SES를 사용하여 모델간의 관계와 계층을 파악하기 용이하도록 하였다.



[그림 3] 대상 네트워크의 구조

NetworkSecurityModel의 구성은 Intrude, NetworkComponent, Policy Component 모델로 구성된다. Intrude 모델은 DoS 모델과 Scan 모델로 구성되며, DoS 모델은 Jolt 모델과 Trinoo 모델로 세분화된다. Scan 모델은 다시 IP 모델과 Port 모델로 세분화된다. 네트워크의 일반적인 요소들을 나타내는 NetworkComponent 모델은 Subnet_1, Subnet_2, Router, Gateway, Firewall과 IDS 모델로 구성된다. 정책 관련 처리를 담당하는 PolicyComponent 모델은 IFForSVDB 모델과 P_Frwork 모델로 구성된다. 정책 기반의 처리를 위한 P_Frwork 모델은 PMT와 PDP로 구성된다. 정책을 설정하기 위해 존재하는 PMT 모델은 PR, ActiveIDS, Admin 모델로 구성된다.

3.2 SVDB의 구성

본 논문에서 구성한 SVDB는 일반적으로 취약성 데이터 베이스가 갖는 취약성 정보뿐만 아니라 보안 시스템 모델이 사용할 수 있고 보안 툴이 가지고 있는 패킷 수준의 상세한 정보까지 포함한다. 우선 CVE 이름, 취약성에 대한 요약 기술, 공격의 범위, 손실의 유형 등의 일반적인 취약성 정보를 기술한다[11]. 그리고 취약점 스캐너와 같이 내부의 취약성 정보 리스트를 가지고 대상 시스템을 점검하는 도구를 위해 취약한 시스템과 소프트웨어 및 버전을 기술한다[7]. 그리고 침입 차단 시스템과 침입 탐지 시스템이 사용할 수 있는 패킷 정보를 기술한다.

〈표 1〉 SVDB의 구성

테이블 (Table)	필드(Field)
취약성 정 보	CVE(Common Vulnerabilities and Exposures) 이름, 취약성에 대한 요약 기술, 취약성 공개 날짜, 취약성 유형, 공격의 범위, 손실의 유형, 취약한 소프트웨어 및 버전
패 킷 정 보	IP 프래그 비트, TTL 값, 프로토콜, 발신자 IP 주소, 목적지 IP 주소, IP 옵션, ICMP 코드, ICMP 타입, 발신자 포트 번호, 목적지 포트 번호, 순서 번호, TCP 프래그, 오프셋, Payload 크기, URL 내용(Contents), 내용, CVE 이름
시스템 정 보	소프트웨어 개발 업체, 소프트웨어 이름, 소프트웨어 버전
참 고 정 보	제공자, 타입, 이름, 링크 주소

일반적인 패킷내의 정보뿐만 아니라 취약성 중 빈도가 많은 웹은 URL(Uniform Resource Locator)의 내용(Contents)을 따로 기술하고, 전체 패킷 크기, 패킷 내에서의 위치 등 패킷에 대한 규칙을 적용할 때 정확성과 효율성을 높일 수 있는 정보를 기술한다. 보안 시스템 모델의 성능을 평가하는 시뮬레이션 환경을 위해 구성한 SVDB의 내용은 〈표 1〉과 같다.

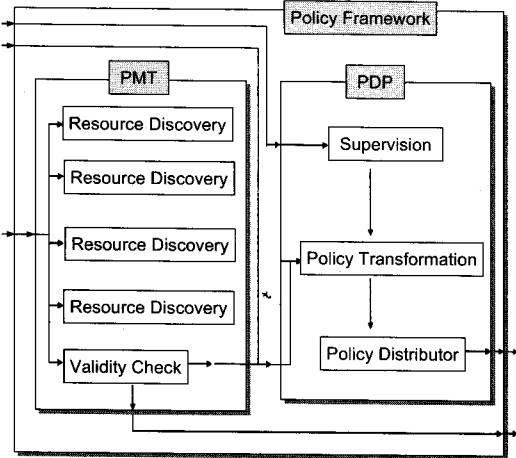
3.3 보안 모델 명세

대상 네트워크를 구성하는 대표적인 모델들에 대해 설명한다.

3.3.1 Policy Framework 모델

Policy Framework(P_Frwork) 모델은 PMT 모델과 PDP 모델로 구성된다. PMT 모델은 SVDB에서 추출된 규칙을 받고 PDP로 정책 실행을 요청하거나 정책 저장소에 저장한다. PDP 모델은 PEP와 통신하며 실행을 제어하거나 정책 저장소나 PMT로부터 받은 정책을 해석하여 네트워크로 분배하는 역할을 한다. [그림 4]에서와 같이 PMT 모델은 Resource Discovery 모델과 Validity Check 모델 등으로 구성된다. Resource Discovery 모델은 네트워크의 용량과 토폴로지, 운용되는 응용 프로그램, 사용자들의 정보 등의 내용을 바탕으로 정책을 검증하는 작업을 한다. Validity Check 모델은 속성에 따라 주어진 값이 유효한지를 검사하는 범위 검사, 기존의 정책과 충돌 여부를 검사하는 일치성 검사, 현재 네트워크 토폴로지에서 실행 가능한지를 검사하는 실현성 검사 등의 과정을 실행한다. P_Frwork 모델이 갖는 PR, ActiveIDS, Admin 모델은 인터페이스를 담당하는 모델로서 PR은 Policy Resource 모델과의 통신을 위해 사용되며 Active ID는 IDS로부터 탐지된 침입 상황을 관리자의 개입 없이 바로 네트워크에 적용하기 위해서 필요한 인터페이스이다. Admin 모델은 원래 정책 기반 프레임워크에서 관리자가 설정할 수 있는 요소값들을 반영하기 위해 필요한 인터페이스이다.

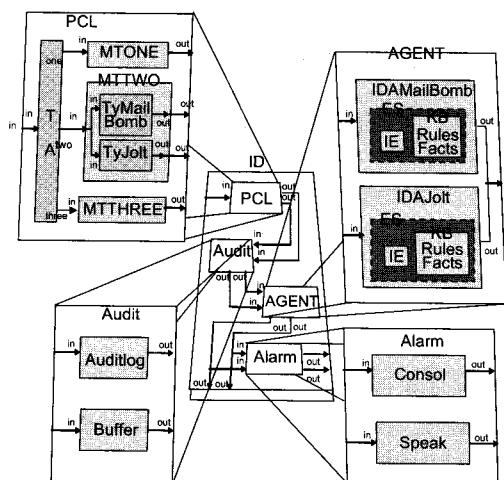
PDP 모델은 Supervision 모델, Policy Transformation 모델과 Policy Distributor 모델로 구성된다. Supervision 모델은 PEP와의 통신을 담당하며 실행 감지와 에러보고 등 PEP로부터 동적인 정보를 받고 제어 기능을 수행한다. Policy Transformation 모델은 상위 레벨의 정책을 네트워크 디바이스 설정이 가능한 하위 레벨의 형태로 변환한다. Policy Distributor 모델은 PEP의 IETF 정책 구조 지원 여부를 검사한 후 정책을 분배하는 역할을 한다.



[그림 4] Policy Framework 모델의 구성

3.3.2 IDS 모델

IDS(Intrusion Detection System)[12]는 침입을 탐지하는 모델로서 [그림 5]와 같이 구성된다. ID 모델의 처리량을 줄이기 위해 필요 없는 트래픽을 필터링하는 PCL(Packet Classify Library) 모델, 실제로 침입을 탐지하는 AGENT 모델과 여러 감사 정보들을 저장하는 Audit 모델, 침입을 탐지했을 때 이를 관리자에게 알리기 위한 Alarm 모델이 존재한다.



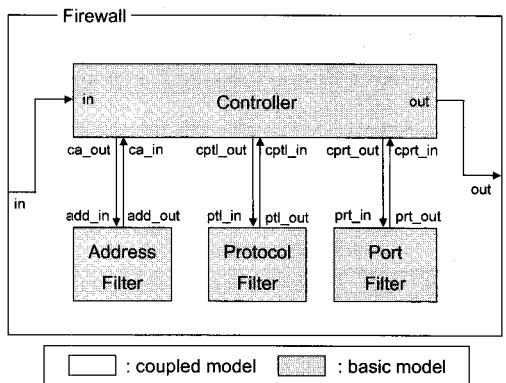
[그림 5] IDS 모델의 구성

[그림 5]의 각 모델은 DEVS 모델의 기본 모델과

결합 모델로 구성되어 있다. 결합 모델은 기본 모델을 내부에 포함하며 기본 모델이 동작할 수 있는 링크를 제공하여 준다.

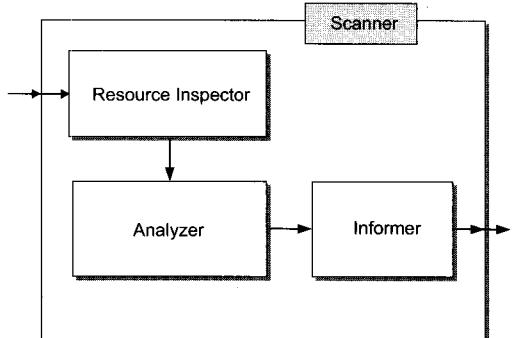
3.3.3 Firewall 모델

[그림 6]은 방화벽 모델의 구성을 나타낸다. 방화벽[13] 모델은 Controller 모델과 Address Filter, Protocol Filter, Port Filter 모델로 구성된다. Firewall 모델은 Inbound 패킷과 Outbound 패킷을 처리해야 하므로 양쪽 방향으로 패킷을 전달할 수 있도록 구성되었으며 이는 Controller 모델이 조정하게 된다. Controller 모델은 Address Filter 모델, Protocol Filter 모델, Port Filter 모델과 서로 정보를 주고 받으면 정해진 정책에 의해서 패킷을 필터링하게 된다.



[그림 6] Firewall 모델의 구성

3.3.4 Scanner 모델

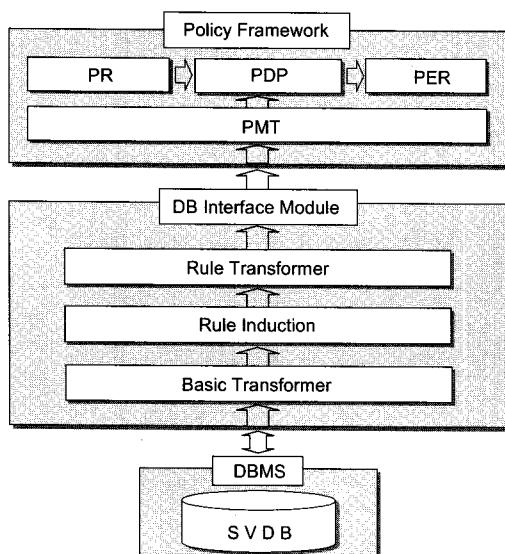


[그림 7] Scanner 모델의 구성

Scanner 모델은 [그림 7]과 같은 구조로 대상 시스템을 조사하는 Resource Inspector 모델로부터 얻은 정보를 Analyzer 모델이 내부 취약성 정보 리스트의 소프트웨어 버전 정보와 설정 정보 등을 비교하여 취약성을 판단하여 Informer 모델을 통하여 보고한다.

4. 보안 모델과 정책 기반 프레임워크의 연동

이번 장에서는 SVDB와 정책 기반 프레임워크의 연동에 대해서 설명한다. SVDB를 활용하여 보안 규칙을 유도하여 정책 기반 프레임워크에 적용하기 위한 기본적인 구성은 [그림 8]과 같다.



[그림 8] 인터페이스의 구성

Basic Transformer : DB 접속과 기본적인 데이터형 변환과 검사를 수행한다.

- Rule Induction : 규칙 유도를 위해서 필요한 정보를 DB에서 가져온다. 분류 알고리즘을 거쳐 결정 트리를 구성하고 결정 트리에서 규칙 형태로 변환한다.
- Rule Transformer : Rule Induction에서 유도된

규칙을 정책 정보 모델(PCIMe)형태로 변환 작업을 한다. 규칙에서 각 컨디션과 액션 및 변수를 PCIMe에 기술된 객체 형태로 사상시킨다.

4.1 SVDB에서의 규칙 유도

데이터 베이스에서 규칙을 유도하기 위하여 데이터 마이닝 분류 알고리즘의 하나인 ID3 알고리즘을 이용하였다. 훈련 집합(training set)으로부터 정보 획득량을 측정하여 루트 노드를 결정하고, 다시 재귀호출을 통하여 결정 트리를 구축하는 하향식 방법을 사용하는 알고리즘이다. ID3 알고리즘은 다음과 같다[14].

Algorithm ID3

Input : a set of example

Output : a decision tree

Method :

ID3_tree (examples, properties)

If all entries in examples are in the same

Category of
decision variable

Return a leaf node labeled with that category
Else

Calculate information gain ;

Select a property P with highest information
gain ;

Assign root of the current tree = P ;

Assign properties = properties - P ;

for each value V of P

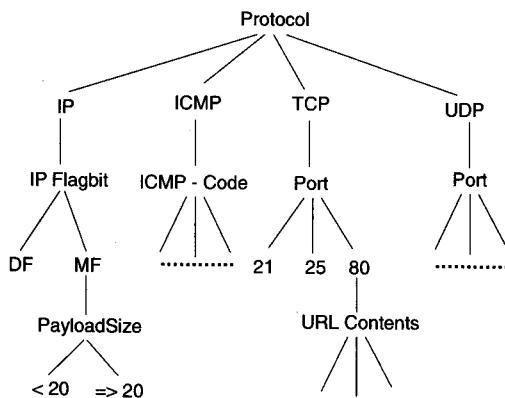
Create a branch of the tree labeled with V ;

Assign examples_V = subset of examples
with values V for

property P ;

Append ID3_tree (example_V, properties) to
branch V

본 연구를 위해 구성된 SVDB의 내용으로 분류 알고리즘을 거쳐 생성된 결정 트리는 [그림 9]와 같다. 여기에서 마지막 분류되는 클래스는 공격 이름이다. 생성된 분류 트리에서 어느 레벨까지 유효하게 규칙으로 결정할 것인지는 고려하지 않고 마지막 레벨까지 트리를 생성하였다.



[그림 9] 구성된 ID3 트리

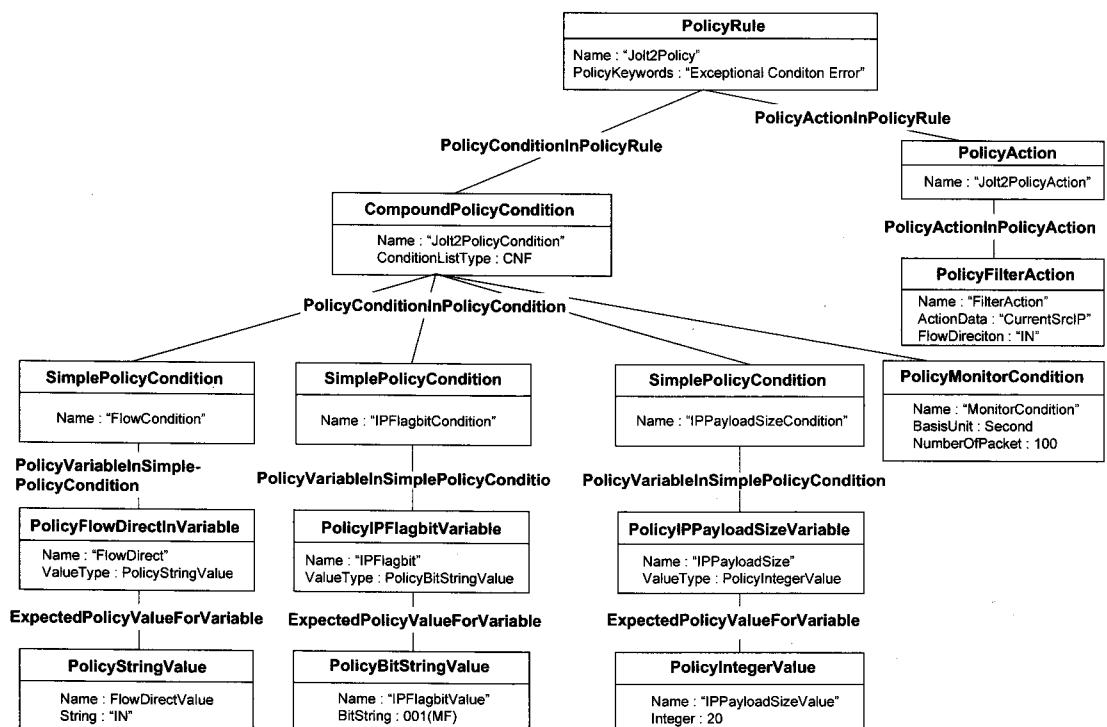
4.2 PCI Me 형식의 규칙 변환

유도된 규칙은 PCI Me 형식의 객체 정보들로 변형하는 과정을 거친다. 규칙은 PCI Me에서 정한 각 클래스에 맞게 변형 과정을 거친다. 규칙의 PCI Me 형태로의 변환을 나타내기 위해 jolt2 공격에 대한

정책 규칙을 기술한다. jolt2 공격은 IP 패킷을 작은 조각으로 나누고, 많은 수의 조각으로 된 패킷을 공격 대상 시스템에 전송하여 CPU의 과부하를 유도하는 서비스 거부 공격의 한 형태로 CVE-2000-0305로 분류되고 있다[15]. 아래는 jolt2 공격에 대한 보안 정책 규칙이다.

```
IF ( FlowDirection = IN ) ∧ ( IPFlagbit = MF ) ∧
( IPPayloadSize <= 20 ) ∧ ( NumOfPacketPer
Second > 100 )
THEN Filtering ( CurrentSrcIP, IN )
```

[그림 10]은 위의 jolt2에 대한 규칙을 객체로 나타낸 것이다. 패킷 크기나 IP 프래그 비트와 필터 액션을 나타내기 위하여 PCI Me 클래스를 상속하여 나타내었다. 이렇게 객체 형식으로 변형된 규칙은 정책 기반 프레임워크의 정책 검증 과정을 거쳐 네트워크에 적용된다.



[그림 10] jolt2 공격에 대한 정책 규칙 표현

5. 시뮬레이션 결과

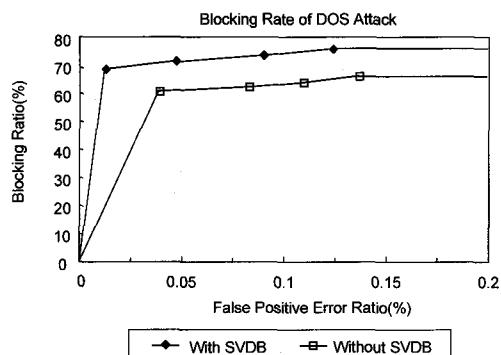
본 연구를 위한 시뮬레이션은 두 가지 형태의 공격에 대해서 수행하였다. 한 가지 경우는 smurf, ping-of-death, jolt2와 같은 서비스 거부 공격에 대한 시뮬레이션 결과이고, 다른 한 가지는 port-scan, ping-sweep과 같은 Probing 공격에 대한 결과이다. Smurf 공격은 출발지 주소를 특정 호스트의 주소로 위장해 목적지 주소를 브로드캐스트 주소로 ICMP ECHO(Type 8) 패킷을 보내 특정 호스트의 과도한 부하를 유도하는 공격이다. Ping-of-death 공격은 공격 대상 호스트에 최대 패킷 사이즈를 초과하는 패킷을 보내 대상 호스트가 패킷을 처리하지 못하고 정지하거나 재부팅 되는 등의 이상현상을 유도하는 서비스 거부 공격이다.

jolt2 공격은 IP 패킷을 작은 조각으로 나누고, 많은 수의 조각으로 된 패킷을 공격 대상 시스템에 전송하여 CPU의 과부하를 유도하는 공격이다. Port-scan은 해당 시스템의 활성화 여부와 접속 가능한 모든 포트를 찾는 공격이다. Port-scan을 위한 방법에는 TCP 연결 스캔, TCP SYN 스캔, TCP FIN 스캔, TCP ACK 스캔, UDP 스캔 등 다양한 방법들이 존재한다. ping-sweep은 대상 호스트들을 향해 ICMP ECHO 패킷을 보내어 ICMP ECHO_REPLY(Type 0)의 응답을 유도하여 현재 시스템이 활성되어 있는지 확인하는 공격이다.

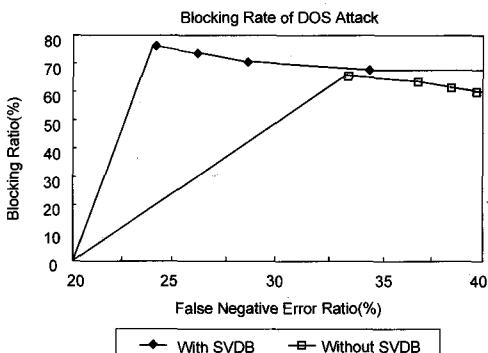
시뮬레이션을 수행하기 위한 시뮬레이션 환경은 본 연구진이 개발한 DEVS-ObjC를 사용하였다. 시뮬레이션을 위한 성능 지표로는 침입 차단 시스템의 유해 트래픽의 차단 비율, False Positive 에러 비율과 False Negative 에러 비율을 측정하였다.

[그림 11], [그림 12]는 DOS 공격에 대한 시뮬레이션 결과이고, [그림 13], [그림 14]는 Probing 공격에 대한 시뮬레이션 결과를 나타낸다. 그림에서 보이듯이 두 가지 경우 SVDB를 이용해서 공격을 차단하는 비율이 기존의 정책 기반 네트워크에서 차단하는 비율에 비해 높게 나타난다. 또한 False Positive 에러 비율과 False Negative 에러 비율

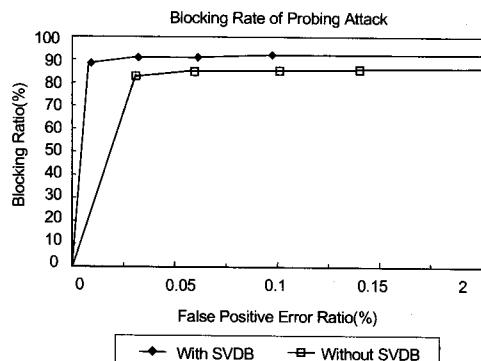
값이 낮게 나타난다. 이는 취약점 데이터베이스에 있는 추가적인 시스템 정보를 이용하여 공격 탐지의 효율성이 높아졌기 때문이다. DOS 공격의 경우 침입의 차단 비율이 높아지면서 False Positive 에러 비율 또한 증가하고 있다. False Positive 에러 비율의 이러한 증가는 시스템의 보안 수준을 강화하면 침입 탐지의 오류가 증가함을 나타낸다. 하지만 Probing 공격의 경우는 False Positive 에러 비율의 증가와 상관없이 거의 일정한 차단 비율을 보이고 있다. 또한 Probing 공격에 대한 차단 비율은 DOS 공격에 비해 상대적으로 높다. 이는 Probing 공격은 주어진 시간 안에 많은 수의 포트나 호스트의 연결을 설정하기 때문에 공격의 변화가 상대적으로 제한되어 있어 탐지가 용이하기 때문이다. 이에 반해 DOS 공격은 다양한 공격의 특성을 갖기 때문에 비교적 낮은 차단 비율을 보인다.



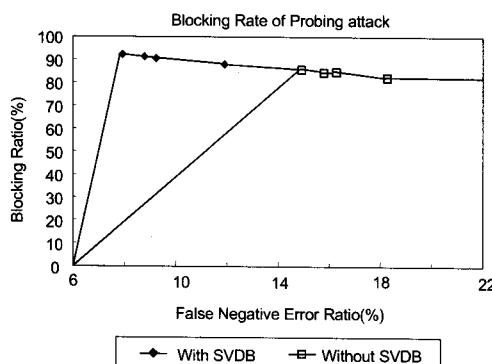
[그림 11] DOS 공격의 False Positive 에러 비율



[그림 12] DOS 공격의 False Negative 에러 비율



[그림 13] Probing 공격의 False Positive 에러 비율



[그림 14] Probing 공격의 False Negative 에러 비율

이상의 결과는 취약점 데이터베이스의 정보를 활용하여 효율적인 보안 정책을 유도하고 정책 기반 네트워크 프레임을 통한 정책 분배의 자동화를 통해서 네트워크를 효과적으로 보호할 수 있음을 보여준다.

6. 결 론

본 논문에서는 여러 보안 시스템 모델들이 사용 할 수 있는 취약성 정보들을 집약시킴으로써 보안 시스템간의 정보 공유를 쉽게 할 수 있는 SVDB를 구축하였다. 또한 IETF 정책 프레임워크에 적용할 수 있는 기초적인 환경을 만들고 SVDB를 활용하여 정책 프레임워크에 적용할 수 있는 보안 규칙을

유도하여 적용하였다. 정책기반 프레임워크를 사용 한 네트워크 관리의 장점을 활용하여, IDS를 통한 네트워크의 공격 탐지와 SVDB의 정보를 활용한 자동화된 네트워크 관리는 관리자의 수고를 덜 수 있을 뿐만 아니라 보다 수월하게 네트워크 자동 관리를 수행할 수 있다.

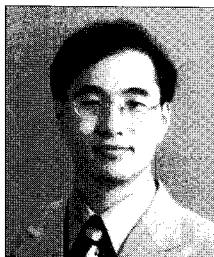
향후 과제로는 다양한 유형의 침입에 대한 시뮬레이션의 수행과 사용자의 입력으로 정책을 검증 할 수 있는 환경을 위한 사용자 인터페이스 개발이 이루어 질 것이다.

참 고 문 헌

- [1] Wang Changkun, "Policy-based Network Management," *Communication Technology Proceeding, WCC-ICCT 2000, International Conference on*, Vol.1(Aug. 2000), pp. 101-105.
- [2] Verma, D. C., "Simplifying Network Administration Using Policy-based Management," *Network, IEEE*, Vol.16(March-April 2002), pp.20-26.
- [3] B. P. Zeigler, "Object-Oriented Simulation with Hierarchical," *Modular Models*, US A : Academic Press, San Diego CA, 1990.
- [4] Seo, Hee Suk , Cho, Tae Ho and Chi, Sung Do, "Modeling and Simulation of Distributed Security Models," *Lecture Notes on Computer Science, Springer Verlag, LNCS 2660*, (Jun. 2003), pp.809-818.
- [5] NIST, *An Introduction to Computer Security : The NIST Handbook*, Technology Adminstration, U.S.A, 1995.
- [6] M. Bishop, "Vulnerabilities Analysis," *Proceedings of the Recent Advances in Intrusion Detection*, (Sep. 1999), pp.125-136.
- [7] Robert A. Martin, "Managing Vulnerabil-

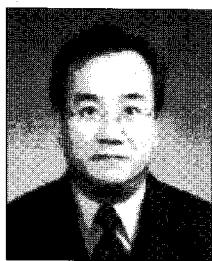
- ities in Networked Systems," *IEEE Computer*, Vol.34, No.11(Nov. 2001), pp.32-38.
- [8] M. Stevens. "Policy Framework," *Internet Draft*, *draft-ietf-policy-framework-05.txt*, Sep. 1999.
- [9] B. Moore, et al., "Policy Core Information Model-Version 1 Specification," *IETF RFC 3060*, Feb. 2000.
- [10] B. Moore, et al., "Policy Core Information Model (PCIM) Extensions," *IETF RFC 3460*, Jan. 2003.
- [11] <http://icat.nist.gov>, ICAT Metabase.
- [12] R. Bace, *Intrusion Detection*, Macmillan Technical Publishing, 2000.
- [13] E. D. Zwicky, S. Cooper and D. B. Chapman, *Building Internet Firewalls, second edition*, O'reilly & Associates, 2000.
- [14] Zhengxin Chen, John Wiley & Sons, *Data Mining And Uncertain Reasoning : An Integrated Approach*, 2001.
- [15] <http://icat.nist.gov/icat.cfm?cvename=CV-E-2000-0305>.

◆ 저자 소개 ◆



서희석 (hisstone@hanmail.net)

성균관대학교 산업공학과에서 학사, 성균관대학교에서 석사학위를 취득하고, 현재 성균관대학교에서 박사학위 과정중이다. 현재 모델링 방법론을 사용하여 침입 탐지 시스템, 침입 차단 시스템을 모델링 하는 방법을 연구중이다. 주요 관심분야는 네트워크 보안, 취약성 분석, 모델링 방법론 등이다.



김동수 (dskimm@dreamwiz.com)

광운대학교 공과대에서 학사, 서울산업대 산업대학원에서 컴퓨터공학 석사학위를 취득하고, 현재 국민대 BIT대학원에서 박사과정에 재학 중이다. 감리법인인 (주)키삭의 대표컨설턴트로 재직중이며, 신흥대 컴퓨터정보계열에서 겸임교수로 재직중이다. CISA, 전자계산기조작용용 기술사 및 한국전산원 정보시스템 감리인 자격을 가지고 있다. 현재 감리품질 향상을 위한 정량화 방안 및 자동화 방안을 연구 중이다. 주요 관심분야는 소프트웨어 품질, 감리, 개발방법론, 보안, 시스템 아키텍쳐 등이다.



김희완 (hwkim@syu.ac.kr)

광운대학교 전자계산학과에서 학사, 성균관대학교에서 석사와 박사학위를 취득하고, 삼육대학교 컴퓨터과학과 조교수로 재직 중이다. 정보관리 기술사 및 한국전산원 정보시스템 감리인 자격을 가지고 있다. 현재 컴퓨터 보안과 분산 데이터베이스를 연구 중이다. 주요 관심분야는 컴퓨터 및 네트워크 보안, 동시성 제어, 분산 DB, 보안 시뮬레이션 등이다.