정책 기반의 프레임워크에서의 보안 시스템 모델링 및 시뮬레이션*

이원영**, 조대호**

Modeling and Simulation of Security System on Policy-based Framework

Won Young Lee, Tae Ho Cho

Abstract

현재의 네트워크는 다양한 서버, 라우터, 스위치 등 여러 장치들로 구성된 복잡한 구조를 가지고 있다. 정책 기반의 프레임워크는 복잡한 네트워크를 단순화하고 자동화하여 관리할 수 있는 도구를 제공한다. 본 논문에서는 여러 보안 시스템 모델들이 사용할 수 있는 취약성 정보들을 집약시킴으로써 보안 시스템간의 정보 공유를 쉽게 할 수 있는 SVDB (Simulation based Vulnerability Data Base)를 구축하였다. 또한 SVDB를 활용하여 보안 규칙을 유도하고 유도된 보안 규칙을 정책 기반 프레임워크에 적용할 수 있는 환경을 구성하였다. 정책 기반의 프레임워크에서 취약점 데이터베이스를 이용한 정책 유도와 적용을 검증하기 위해 서비스 거부 공격 (Denial of Service)과 Probing 공격을 사용하여 시뮬레이션을 수행하였다. 정책 기반의 프레임워크에서 보안 시뮬레이션을 수행함으로써 적용될 보안 정책이 기대되는 대로 동작하는지 검증할 수 있는 환경을 제공할 수 있을 것이다.

Key Words: Security Policy, PBNM (Policy-based Network Mangement), network security, DEVS formalism, simulation

* 본 연구는 한국과학재단 목적기초연구(R05-2002-000-00107-0)지원으로 수행되었음
** 성균관대학교 정보통신공학부

1. 서론

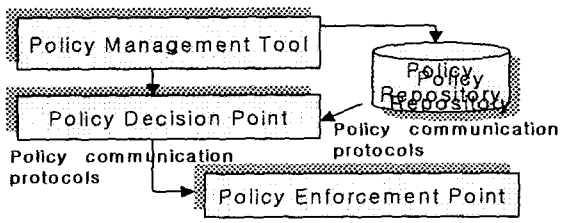
근래의 네트워크는 다양한 장치들로 구성된 복잡한 구조를 갖고 있다. 이러한 시스템의 기술적 복잡도 증가는 서비스를 위한 대역폭 확보로 인한 장점도 있지만 새로운 기술을 배우고 관리하는 인적 자원 비용의 상승을 일으킨다. 그리고 음성 및 비디오와 같은 멀티미디어 서비스의 확대와 이의 응용에 따른 다양한 서비스 요구가 증가하고 있다. 또한 다양한 보안 제품의 출시와 이들 간의 상이한 특성으로 인해 효율적인 운용과 유지에 어려움이 있다. 또한 이들 간의 체계적이며 일괄적인 보안 관리 체계의 필요성이 증가하고 있다[1]. 이러한 문제를 해결하기 위해 정책 기반 프레임워크에서는 네트워크 관리자가 자원이나 서비스가 어떻게 사용되는지를 정책으로 정의하고, 정책 기반 관리 시스템은 이렇게 정의된 정책을 네트워크 장치가 인식 가능한 형태로 변형하여 네트워크에 적용하게 된다. 정책 기반 프레임워크를 통한 가장 중요한 이점은 네트워크 관리 프로세스의 단순화와 자동화이다[2].

네트워크의 속도가 급속하게 증가하고 발전하는 상황에서 많은 양의 데이터를 처리해야 하는 전체 네트워크 인프라를 직접 사용하여 성능을 평가하는 것은 효율적이지 못하다[3]. 이를 위하여 DEVS (Discrete EVent system Specification) 방법론에 의거한 시뮬레이션 모델을 구축하고, 이러한 모델들을 사용하여 시뮬레이션 환경을 구축할 것이다. 또한 본 논문에서는 취약점 데이터베이스를 이용한 보안 정책의 유도와 이 정책을 정책 기반 프레임워크에 적용하는 시뮬레이션을 수행할 것이다. 이를 위하여 본 연구진은 여러 보안 시스템 모델들이 사용할 수 있는 취약점 정보들을 집약시킴으로써 보안 시스템 간의 정보 공유를 쉽게 할 수 있는 SVDB (Simulation based Vulnerability Data Base)를 구축하고, 보안 정책의 유도와 이 정책의 정책 기반 프레임워크에 적용을 위한 인터페이스를 설계하고 이를 구현하였다. 정책 기반 프레임워크에서 취약점 데이터베이스를 이용한 정책 유도와 적용을 검증하기 위해 서비스 거부 공격 (Denial of Service) 공격과 Probing 공격을 사용하여 시뮬레이션을 수행하였다.

2. 배경 이론

2.1 정책 기반 프레임워크

본 논문에서는 네트워크 영역의 정책 분배와 설정에 사용되는 표준인 IETF (Internet Engineering Task Force) 정책 프레임워크을 적용한다. 정책 프레임워크는 <그림 1>과 같이 네 개의 요소로 이루어져 있다[4-6].



<그림 1> IETF의 정책 기반 프레임워크

- *정책 관리 툴 (PMT: Policy Management Tool)*

정책 관리 툴은 정책을 구성하고, 정책을 배치하고, 그리고 정책 관리 환경의 상태 모니터를 위한 사용자 인터페이스이다.

- *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.

- *정책 시행 지점 (PEP)*

정책 시행 지점은 정책을 시행하고 적용하는

작업을 한다. PEP는 에이전트로서 장치 안에서 동작하거나, 응용의 형태로 존재할 수 도 있다. 또한 정책을 수행한 결과나 PEP내의 동적인 정보들을 PDP에 보고한다.

- *정책 저장소 (Policy Repository)*

정책과 관련된 정보를 저장하기위한 저장소이다. 디렉토리 또는 관계형 데이터베이스의 형태로 저장된다.

- *정책 통신 프로토콜 (Policy Communication Protocols)*

정책 저장소로부터 읽고 쓰기위한 프로토콜(예: LDAP)과 PDP와 PEP간의 통신하기 위한 프로토콜(예: COPS, SNMP)이 사용된다.

2.2 정책 표현 방법

네트워크 관리를 위해 필요한 상위 계층과 하위 계층 정책은 다양한 방법으로 정의될 수 있다[2]. 가장 간단한 접근법은 정책을 규칙 기반으로 표현하는 것이다. 각 규칙은 간단한 컨디션 (condition)과 액션 (action)의 쌍으로 구성되어 있다. 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[7,8]. 정책 정보는 정책의 제어와 정책 정보를 표현하는 구조 클래스와 구조 클래스의 상호 연관성을 나타내는 연관 클래스를 사용하여 정의된다.

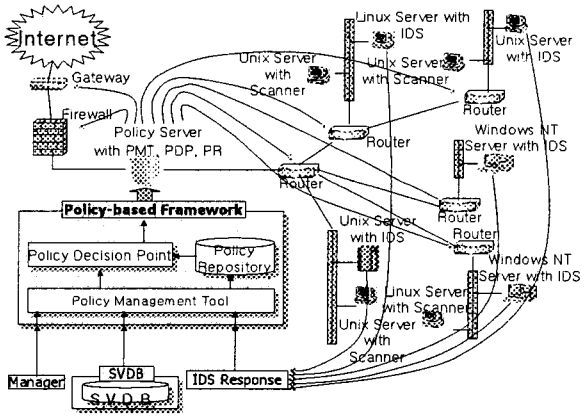
2.3 취약점 데이터베이스

취약점이란 위협 요소에 의해 침해될 수 있는 보안 절차, 기술적 통제, 물리적 통제, 기타 다른 통제들 내의 어떤 조건이나 결점이다[9].

취약점 정보들은 침입 탐지 시스템의 시그너쳐(signature), 공격자가 다른 취약성들을 이용하기 위한 시스템 환경 등으로 이루어진다[10]. 보안 시스템 시뮬레이션 환경에서는 일반적인 취약점 정보뿐만 아니라 보안 시스템 모델이 사용할 수 있는, 보안 툴이 가지고 있는 패킷 수준의 상세한 정보까지 포함해야 된다. 우선 CVE 이름, 취약점에 대한 요약 기술, 공격의 범위, 손실의 유형 등의 일반적인 취약점 정보를 기술한다[11]. 그리고 취약점 스캐너와 같이 내부의 취약점 정보 리스트를 가지고 대상 시스템을 점검하는 도구를 위해 취약한 시스템과 소프트웨어 및 버전을 기술한다[12]. 그리고 침입 차단 시스템과 침입 탐지 시스템이 사용할 수 있는 패킷 정보를 기술한다. 일반적인 패킷내의 정보뿐만 아니라 패킷에 대한 규칙을 적용할 때 정확성과 효율성을 높일 수 있는 정보를 기술한다.

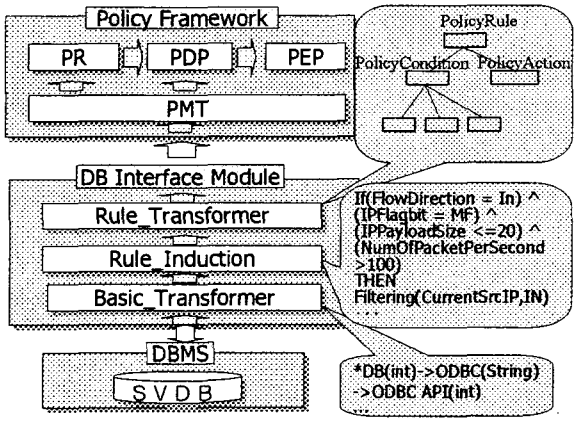
3. SVDB와 정책 기반 프레임워크 결합

본 연구에서는 취약성 데이터베이스 보안 정책을 유도하고 침입 탐지 시스템의 침입 대응으로부터 대응 정책을 생산하여 이 정책을 정책 기반 네트워크에 적용한다. <그림 2>는 관리자와 SVDB 및 침입 탐지 시스템의 대응에 의한 정책 설정을 나타내고 있다.



<그림 2> 정책 기반의 네트워크 보안 시뮬레이션

SVDB를 활용하여 보안 규칙을 유도하여 정책 기반 프레임워크에 적용하기 위한 기본적인 구성은 <그림 3>과 같고 구성요소의 주요 기능은 다음과 같다.

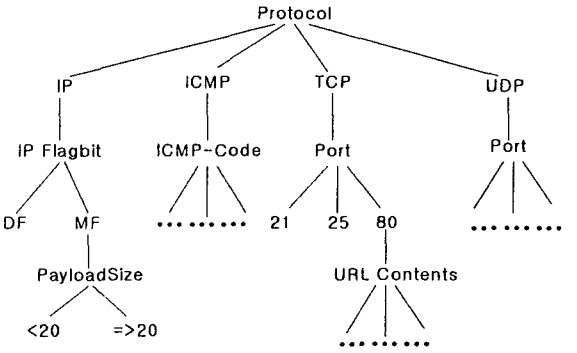


<그림 3> 인터페이스의 구성

• 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 PCIM에 기술된 객체 형태로 사상시킨다.

3.1 SVDB에서의 규칙 유도

데이터 베이스에서 규칙을 유도하기 위하여 데이터 마이닝 분류 알고리즘의 하나인 ID3 알고리즘을 이용하였다. ID3 알고리즘은 훈련 집합 (training set)으로부터 정보 획득량 (information gain)을 측정하여 루트 노드를 결정하고, 다시 재귀호출을 통하여 결정 트리를 구축하는 하향식(top-down) 방법을 사용하는 알고리즘이다[13]. <그림 4>은 ID3 분류 알고리즘을 사용하여 생성한 결정 트리의 일부분이다. 여기에서 마지막 분류되는 클래스는 공격 이름이다. 생성된 분류 트리에서 어느 레벨까지 유효하게 규칙으로 결정할 것인지는 고려하지 않고 마지막 레벨까지 트리를 생성하였다.



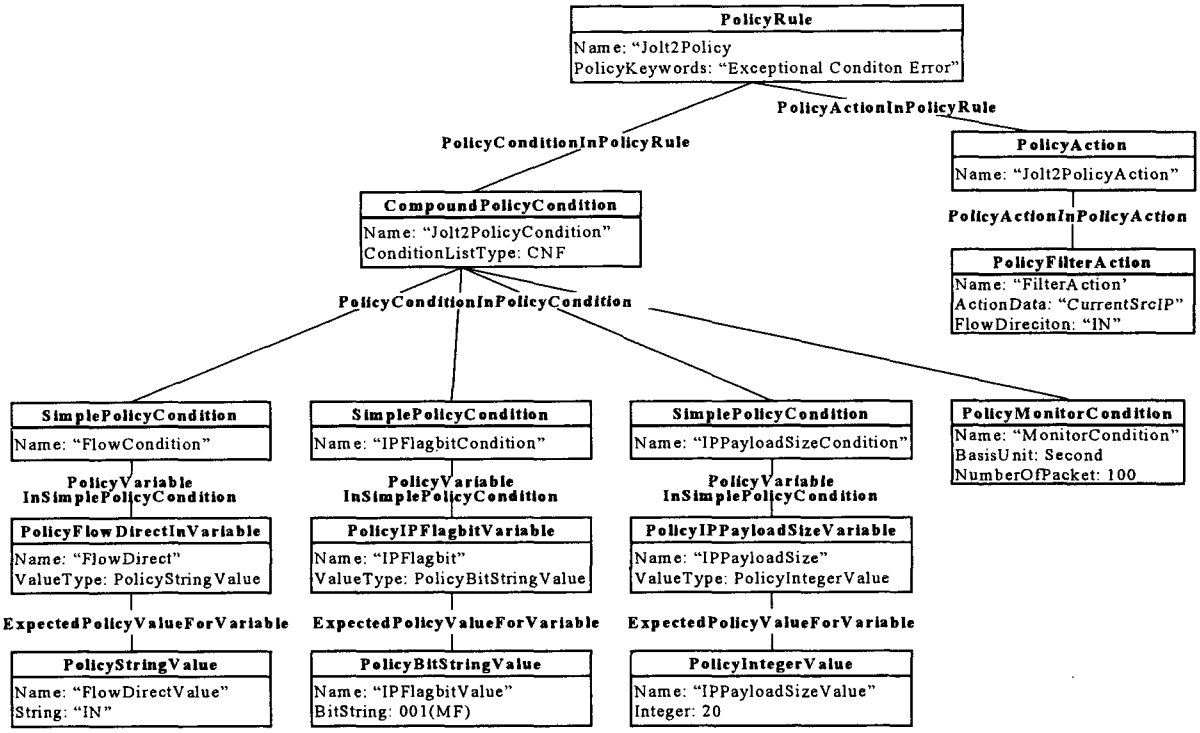
<그림 4> 구성된 ID3 트리

3.2 PCIMe 형식의 규칙 변환

유도된 규칙은 PCIMe 형식의 객체 정보들로 변형하는 과정을 거친다. 규칙은 PCIMe에서 정한 각 클래스에 맞게 변형 과정을 거친다. 규칙의 PCIMe 형태로의 변환을 나타내기위해 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. 아래는 jolt2 공격에 대한 보안 정책 규칙이다.

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

<그림 5>는 위의 jolt2에 대한 규칙을 객체로 나타낸 것이다. 패킷 크기나 IP 프래그 비



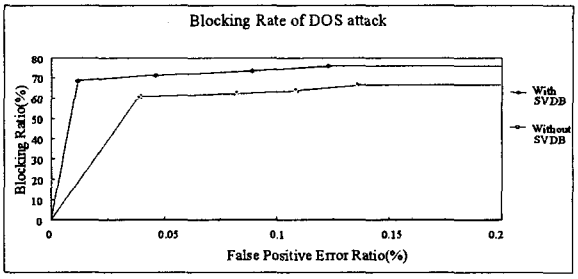
<그림 5> jolt2 공격에 대한 정책 규칙 표현

트와 필터 액션을 나타내기 위하여 PCIMe 클래스를 상속하여 나타내었다. 이렇게 객체 형식으로 변형된 규칙은 정책 기반 프레임워크의 정책 검증 과정을 거쳐 네트워크에 적용된다.

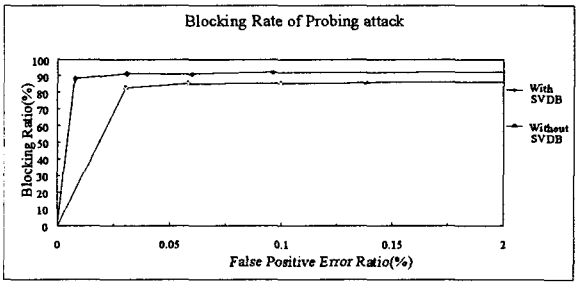
4. 시뮬레이션 결과

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.

<그림 6>은 DOS 공격에 대한 시뮬레이션 결과이고 <그림 7>은 Probing 공격에 대한 시뮬레이션 결과를 나타낸다.



<그림 6> DOS 공격의 False Positive 에러 비율



<그림 7> Probing 공격의 False Positive 에러 비율

그림에서 보이듯이 두 가지 경우 SVDB를 이용해서 공격을 차단하는 비율이 기존의 정책 기반 네트워크에서 차단하는 비율에 비해 높게 나타난다. 또한 False Positive 에러 비율 값과 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.

5. 결론

본 논문에서는 여러 보안 시스템 모델들이 사용할 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. 정책 기반의 프레임워크에서 보안 시뮬레이션을 수행함으로써 적용될 보안 정책이 기대되는 대로 동작하는지 검증할 수 있는 환경을 제공할 수 있고 나아가 현재 네트워크 인프라에 맞게 최적화 할 수 있을 것이다.

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

참고문헌

[1] Wang Changkun, "Policy-based network management," Communication Technology Proceeding, 2000. WCC-ICCT 2000, International Conference on, Vol. 1. pp. 101-105. Aug. 2000.

[2] Verma, D.C., "Simplifying network administration using policy-based management," Network, IEEE, Vol 16, pp 20-26, March-April. 2002.

[3] F. Cohen, "Simulating Cyber Attacks, Defences, and Consequences," *Computer & Security*, Vol.18, pp. 479-518, 1999.

[4] Dinesh C. Verma. *Policy-Based Networking: Architecture and Algorithm*, New Rider, 2001.

[5] Dave Kosiur. *Understanding Policy-Based Networking*, John Wiley & Sons, Inc. 2001.

[6] M. Stevens. "Policy Framework". Internet Draft, draft-ietf-policy-framework-05.txt. Sep. 1999.

[7] B. Moore, et al., "Policy Core Information Model-Version 1 Specification," IETF RFC 3060, Feb 2000.

[8] B. Moore, et al., "Policy Core Information Model (PCIM) Extensions," IETF RFC 3460, Jan 2003.

[9] NIST, "An Introduction to Computer Security : The NIST Handbook," Technology Adminstration, U.S.A, 1995.

[10] M. Bishop, "Vulnerablities Analysis," Proceedings of the Recent Advances in Intrusion Detection pp. 125-136 Sep. 1999.

[11] http://icat.nist.gov, ICAT Metabase

[12] Robert A. Martin, "Managing Vulnerabilities in Networked Systems," IEEE Computer, Vol.34, No.11, pp. 32-38, Nov. 2001.

[13] Zhengxin Chen, *Data Mining And Uncertain Reasoning: An Integrated Approach*, John Wiley & Sons, 2001