네트워크 보안 정책 정보 모델에 기반한 정책 관리 도구의 구현

김 건 량[†] · 장 종 수[††] · 손 승 원[†††]

요 약

본 논문은 정책 서버를 이용한 네트워크 보안 시스템에서 정책 정보 모델에 기반하여 구현한 정책 관리 도구를 소개한다. 본 논문이 기술하는 네트워크 보안 시스템은 특정 도메인을 보호하기 위해 정책을 관리하고 정책을 내리는 정책 서버와 정책 서버로부터 받은 정책들을 기반으로 침입을 탐지하고 대응하는 정책 클라이언트로 구성된다. 정책 서버와 정책 클라이언트간에 송수신되는 정책은 NSPIM(Network Security Policy Information Model)을 기반으로 구축된 정책 관리 도구를 이용하여 LDAP을 통해 디렉토리 형태로 데이터베이스에 저장된다. NSPIM은 본 네트워크 보안 시스템에서 사용하는 정책에 맞게 IETF의 PCIM과 PCIMe를 확장하여 구축한 정책 정보 모델이다. NSPIM 기반의 정책 관리 도구는 정책 생성, 수정, 삭제 등의 기능을 제공하여 정책을 관리할뿐 아니라 재사용 객체 개념을 이용한 편집 기능, 객체 이름의 자동 생성 기능, 차단 정책의 자동 생성 기능 등을 통해 사용자에게 편리성을 제공한다.

The Implementation of Policy Management Tool Based on Network Security Policy Information Model

Geon Lyang Kim[†] · Jong Soo Jang[††] · Sung Won Sohn[†††]

ABSTRACT

This paper introduces Policy Management Tool which was implemented based on Policy Information Model in network security system. Network security system consists of policy server managing and sending policies to keep a specific domain from attackers and policy clients detecting and responding intrusion by using policies that policy server sends. Policies exchanged between policy server and policy client are saved in database in the form of directory through LDAP by using Policy Management Tool based on network security policy information model. NSPIM is an extended policy information model of IETF's PCIM and PCIMe, which enables network administrator to describe network security policies. Policy Management Tool based on NSPIM provides not only policy management function but also editing function using reusable object, automatic generation function of object name and blocking policy, and other convenient functions to user.

키워드 : 정책 관리 도구(policy management tool), 정책 정보 모델(policy information model), 정책 서버(policy server), 네트워크 보안(network security)

1. 서 론

인터넷이 활성화되고 대규모 네트워크가 활발히 구축되면서 더불어 네트워크 기반 시스템의 해킹 사례가 급증하고 있어 네트워크 보안을 위한 필요성이 증가하고 있다. 최근에는 DDOS(Distributed Denial of Service) 공격과 같은 외부의 공격을 원천 봉쇄하기 위한 방법이 필요한데, 네트워크 보안을 위한 솔루션으로 IETF(The Internet Engineering Task Force)의 정책 프레임워크 워킹 그룹에서 제시한 정책 프레임워크를 적용하여 정책기반네트워크 넷안 시스템을 구축한다[1].

정책 기반 네트워크 보안 시스템은 여러 정책 클라이언트들에게 정책을 전송하고 클라이언트들을 제어하는 정책 서버와 전송된 정책들을 기반으로 침입을 탐지하고 대응 처리한 결과를 다시 정책 서버에 반환하는 여러 정책 클라이언트들로 구성되어 있다. 이때 정책을 저장하는 정책 저장소는 LDAP (Lightweight Directory Access Protocol) 프로토콜을 이용하여 접근하도록 하였으며, 정책 클라이언트로 정책을 전송할 때는 PIB(Policy Information Base)의 형태로 전송하도록 하였다. 이러한 시스템에서 정책 저장소인 LDAP 서버의 스키마와 PIB과 MIB(Management Information Base)의 구조를 정의하고, 정책 관리 도구의 설계를 위해 정책 정보 모델의 구축이 필요하다. 정책 정보 모델을 구축함으로써 시스템의

† 정 회 원 : 한국전자통신연구원 정보보호연구본부 보안게이트웨이연구팀
†† 정 회 원 : 한국전자통신연구원 책임연구원, 정보보호연구본부 보안게이트웨이연구팀 팀장
††† 정 회 원 : 한국전자통신연구원 책임연구원, 정보보호연구본부 네트워크 보안연구부장
논문접수 : 2002년 3월 13일, 심사완료 : 2002년 7월 30일

구성 요소들은 정책 정보를 표현, 관리하고 공유할 수 있으므로 여러 가지 작업들을 효율적으로 처리할 수 있다.

정책을 기술하는데 사용될 수 있는 정보 모델은 DMTF(The Distributed Management Task Force)의 CIM(Common Information Model)이 있으며, IETF는 이러한 정보 모델을 확장한 PCIM(Policy Core Information Model)을 정의하여 RFC3060으로 채택하였고, PCIM을 확장한 PCIMe(Policy Core Information Model extensions)의 연구가 진행중이다. 현재 네트워크 보안을 위한 정책 정보 모델은 표준화되지 않았기 때문에, 본 논문이 소개하는 네트워크 보안 시스템은 정책 서버나 정책 클라이언트 등 시스템의 여러 장치에서 침입 탐지와 대응을 위해 사용되는 시그니처(signature) 정책 정보를 표현하기 위해 PCIM을 확장하여 NSPIM(Network Security Policy Information Model)을 정의하였다[2-5].

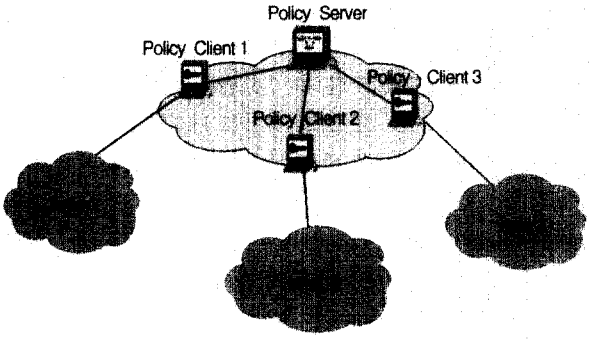
네트워크 보안 시스템에서 사용되는 정책들은 NSPIM에 따라 정의하였는데, 이러한 정책들을 관리하고 제어하기 위해서는 NSPIM을 기반으로 설계한 정책 관리 도구가 필요하다. 본 논문에서는 정책 정보를 표현하는 NSPIM과 정책 기반 네트워크 보안 시스템에서 네트워크 보안에 사용될 정책들을 관리하는 정책 관리 도구를 중점적으로 기술한다.

본 논문은 제 2장에서 정책 기반 네트워크 보안 시스템의 구조와 모듈별 기능 및 상호 작용을 기술하고, 제 3장에서는 NSPIM의 정의와 예제, 정책 관리 도구의 기능과 개발 환경, 그리고 정책 관리 도구를 이용한 정책 데이터 입력 과정을 기술하며 제 4장에서 결론으로 마무리한다.

2. 정책 기반 네트워크 보안 시스템의 구조

본 논문이 소개하는 정책 기반 네트워크 보안 시스템은 하나의 정책 서버와 여러 정책 클라이언트로 하나의 도메인을 구성하며, 도메인이 광범위할 경우 계층 구조로서 여러 정책 서버를 관제하는 상위 계층의 정책 서버를 구축하여 확장할 수 있다.

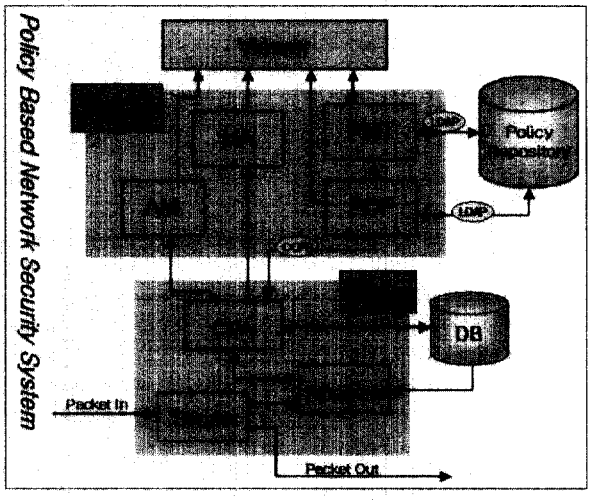
정책 기반 네트워크 보안 시스템의 구조는 (그림 1)과 같다.



(그림 1) 정책 기반 네트워크 보안 시스템

정책 클라이언트는 내부 네트워크로 접근하는 패킷들을 분석하고 공격을 탐지하여 정책 서버로 경보를 올려주며, 정책 서버가 정책을 생성하는데 기반이 되는 트래픽 정보와 로그 정보를 알려준다. 정책 클라이언트는 정책을 기반으로 실시간 탐지 및 대응을 하는데 반해, 정책 서버는 여러 정책 클라이언트가 전달하는 트래픽 정보, 로그 정보, 경보 정보를 이용하여 통계 분석, 이상행위 분석(Anomaly Analysis) 등의 종합적인 분석을 통해 앞으로 발생할 공격에 대한 체계적인 대응 정책을 생성한다. 정책 저장소에 저장된 정책들과 정책 클라이언트에서 탐지하여 정책 서버에 전송하는 경보 메시지 등은 정책 서버와 정책 클라이언트와의 연결에서 IPSec(Internet Protocol Security) Protocol을 이용하여 안전한 경로를 설정하여 전송한다[6].

(그림 2)는 정책 기반 네트워크 보안 시스템의 기능 모듈이다. (그림 2)와 같이 정책 서버는 정책 관리 도구 모듈(Policy Management Tool-PMT), 정책 결정 모듈(Policy Decision Point-PDP), 경보 관리 모듈(Alert Manager-AM), 상위 레벨 분석 모듈(High level Analyzer-HA)의 네 모듈로 구성되며, 정책 클라이언트 시스템은 사이버 순찰 에이전트 모듈(Cyber Patrol Agent-CPA), 분석 모듈(Analyzer), 센서 모듈(Sensor)로 구성된다.



(그림 2) 시스템 모듈 구성도

초기화는 정책 클라이언트가 정책 서버에 접속하면 정책 결정 모듈이 정책 저장소에 저장된 정책들을 정책 클라이언트에게 전달하고 정책 클라이언트는 전달된 정책들을 DB에 저장하여 DB에 저장된 정책 정보를 기반으로 침입을 탐지하도록 하는 과정으로 구성된다.

정책을 변경하는 경우, 정책 관리 뷰어를 통해서 관리자가 변경 사항을 입력하면 정책 관리 도구 모듈은 일관성을 확인한 후 LDIF(LDAP Data Interchange Format) 형태로 정책을 변환하고, LDAP을 이용하여 정책 저장소에 접근하여 정책 정보를 변경한다. 그리고 정책 결정 모듈에 정책의 변경 정보를 전송한다. 정책 결정 모듈은 정책 관리 도구 모듈의 변경 정보나 정책 클라이언트의 정책 요청을 수신하면 정책 저장소에서 해당하는 정책을 ASN.1으로 인코딩한 PIB(Policy

Information Base)의 형태로 COPS(Common Open Policy Service) 프로토콜을 이용하여 정책 클라이언트에 전달한다.

차단 정책을 생성할 때는 경보 뷰어를 통해 사용자가 차단 정책 생성을 요청하면 정책 관리 도구 모듈은 차단 정책을 생성하여 정책 저장소에 저장하고 경보 뷰어와 정책 관리 뷰어에 생성한 차단 정책을 디스플레이한다. 차단 정책 생성이 완료되면 정책 결정 모듈에 정책이 생성되었음을 알리고, 정책 결정 모듈은 정책 저장소에서 차단 정책을 검색하여 정책 클라이언트에 전송한다.

정책을 변경하였거나 새로 생성하였을 경우, 정책 클라이언트의 사이버 순찰 에이전트 모듈은 정책 서버로부터 전송된 정책을 데이터베이스에 저장하고 분석 모듈에 변경을 알리면, 분석 모듈은 데이터베이스의 변경된 정책 정보를 참조하여 센서 모듈에서 전송된 데이터와 비교분석하고 침입을 탐지하고 대응한다.

이와 같이 정책 기반 네트워크 보안 시스템에서 모듈들 간의 통신 데이터들은 정책인데 정책들을 효율적으로 관리하고 제어하기 위해서는 정책 정보 모델을 기반으로 설계된 정책 관리 도구가 필요하다.

3. 네트워크 보안 정책 정보 모델에 기반한 정책 관리 도구

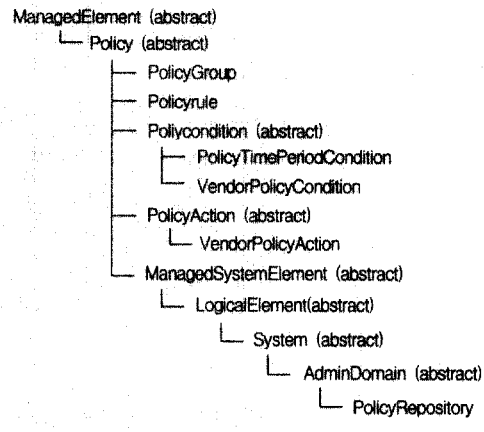
본 절에서 언급하는 정책 관리 도구는 정책 서버에 속한 모듈로서 관리자와 정책 관리 도구를 연결해주는 정책 관리 뷰어를 포함한다. 정책 관리 도구는 정책 정보 모델에 기반하여 설계되었기 때문에 정책 관리 도구를 설명하기에 앞서 PCIM과 PCIMe를 기반으로 확장한 네트워크 보안 정책 정보 모델인 NSPIM(Network Security Policy Information Model)을 기술한다. 그리고, 정책 관리 뷰어를 포함한 정책 관리 도구의 개발 환경과 정책 관리 도구가 제공하는 기능, 정책 관리 도구를 이용하여 정책 저장소에 데이터를 입력하는 방법을 기술한다.

3.1 네트워크 보안 정책 정보 모델(NSPIM)

정보 모델은 구현 전에 이해할 수 있도록 지식을 추상화한 것으로 사용자, 애플리케이션, 네트워크에 대한 지식뿐만 아니라, 여러 사용자들이 그 지식들을 사용할 수 있도록 여러 지식 도메인들 간의 상호 작용하는 방법들에 대한 지식까지도 구조화한다[7].

PCIM은 IETF와 DMTF에서 진행되고 있는 연구를 종합하여 정책 정보 모델을 표현하기 위해 제시하는 객체 지향 정보 모델로서, 수정 보완을 위한 연구도 진행되고 있다. PCIM은 정책 핵심 정보 모델로서 애플리케이션들과 연관된 어떤 정책이든지 표현할 수 있도록 일반적이고 핵심적인 클래스들을 정의하였다. PCIM은 정책의 제어와 정책 정보를 표현하는 구조 클래스와 구조 클래스의 상호 연관성을 나타내는 연관 클래스를 정의하고 있다. PCIM에 정의된 구조 클래스와 연관 클래스들은 어떠한 정책들도 모두 표현할 수 있도록 일반적으로 구축하였지만, IETF에서는 먼저 QoS(Quality of Service)와 IPSec(Internet Protocol Security Protocol) 애플리케이션들을 대상으로 연구를 활발히 진행하고 있다. 이렇게 특정 애플리케이션을 위해 정책 정보 모델을 구축할 경우 여러 가지 방법으로 핵심 모델을 확장할 수 있다[2-4, 8, 9].

(그림 3)은 PCIM을 구성하고 있는 구조 클래스들의 상속 계층 구조이다.

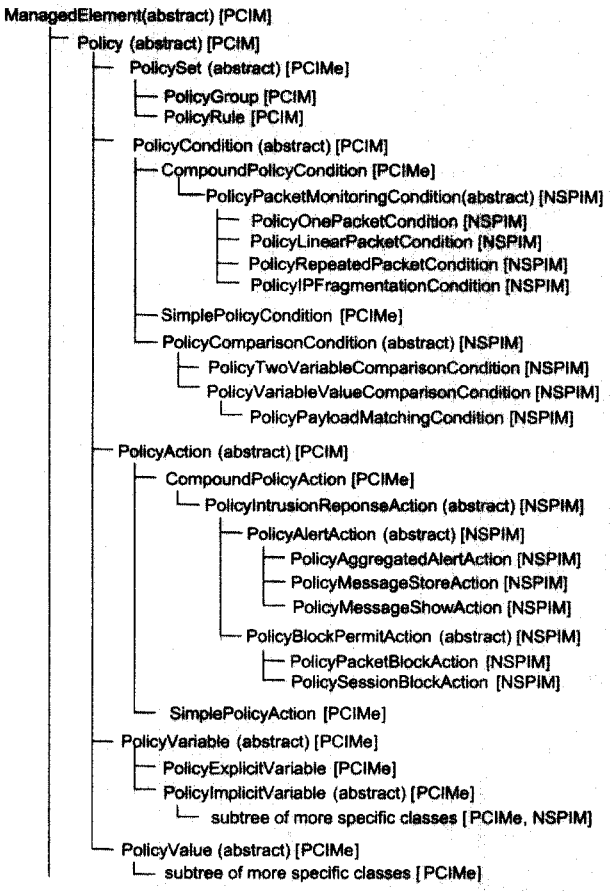


(그림 3) PCIM의 클래스 상속 계층 구조

정책은 정책 규칙들의 집합을 사용하여 적용되고, 각 정책 규칙은 조건들의 집합과 동작들의 집합으로 구성된다. 여러 정책 규칙들은 정책 그룹들과 결합되고, 이러한 그룹들은 또 다른 그룹을 구성할 수 있다. PolicyGroup은 연관된 PolicyRule들의 집합이나, 연관된 PolicyGroup들의 집합을 위한 컨테이너를 의미하는 클래스이다. PolicyRule은 "주어진 조건을 만족하면 지정된 동작을 취한다"와 같은 의미를 표현하기 위한 클래스이고, PolicyCondition은 정책 규칙에서 정책 조건을 나타내는 클래스, PolicyAction은 정책 규칙에서 조건을 만족하면 수행되는 동작을 표현하는 클래스이다. PolicyTimePeriodCondition은 미리 정해진 스케줄에 따라 정책 규칙을 활성화시키거나 비활성화시킬 수 있는 기능을 제공하며, PolicyRepository는 정책과 관련된 정보 저장을 위해 관리적인 측면에서 정의된 컨테이너를 나타낸다. VendorPolicyCondition, VendorPolicyAction은 특정 벤더의 확장을 위해 제공하고 있다[2].

NSPIM은 침입을 탐지하고 대응하는 시그니처 정책들을 위한 모델로서, NSPIM을 구성하고 있는 구조 클래스들의 상속 계층 구조는 (그림 4)와 같다.

NSPIM에서는 PolicyGroup, PolicyRule과 같이 PCIM에서 정의한 기본적인 핵심 클래스를 그대로 사용하였으나, 적용 목적에 따라 PolicyPacketMonitoringCondition, PolicyComparisonCondition, PolicyIntrusionResponseAction 등과 같은 클래스들을 새로 추가 정의하였다. PolicyPacketMonitoring-



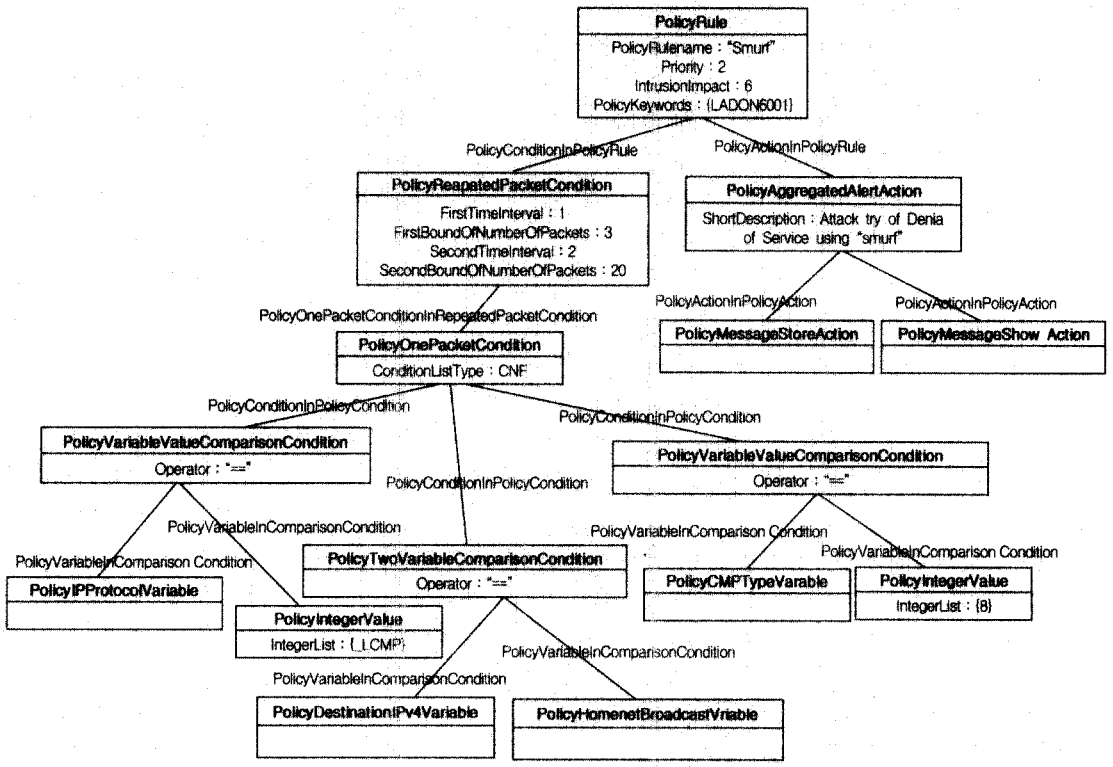
(그림 4) NSPIM의 클래스 상속 계층 구조

Condition은 패킷 모니터링에 대한 조건으로서, CompoundPolicyCondition에서 상속되었기 때문에 자식 클래스로 여러 조건 클래스들을 가질 수 있다. PolicyOnePacketCondition은 하나의 패킷에 대한 조건을 의미하고, PolicyRepeatedPacketCondition은 특정 시간 동안 특정 개수의 패킷에 대한 반복되는 조건을 의미한다. PolicyLinearPacketCondition은 특정 개수의 연속되는 패킷에 대한 조건을 의미하고, PolicyIPFragmentationCondition은 IP 헤더의 Identification 필드 값과 근원지 주소가 동일한 패킷들에 대한 조건을 의미한다. PolicyComparisonCondition은 두 객체의 비교에 대한 조건을 의미하며, 두 개의 변수에 대한 비교 조건, 변수와 값에 대한 비교 조건이 포함되어 있다. PolicyIntrusionResponse Action은 공격에 대한 대응 동작을 정의한 추상화 클래스로서, 경보 동작과 차단 동작에 대한 클래스를 자식 클래스로 정의하였다. 이렇게 기본 골격이 되는 것은 표준을 따랐으나 네트워크 보안 분야에 적용하기 위해서는 기본 골격에서 많은 부분을 확장하여 사용하였다[5].

NSPIM을 이용한 정책 규칙에 대한 클래스 정의를 설명하기 위한 예로써 "Smurf" 정책 규칙에 대해 기술한다.

```
6001 Pattern [Smurf : IcmpAnomaly ; 2 ; 6 ; MStore|MShow]
  while(3 : 1~20 : 2) {
            icmp any > _homenet_br (CTYPE : 8) }
(MESSAGE : "smurf를 이용한 서비스 거부 공격이 시도됨")
```

(그림 5) Smurf 정책 규칙



(그림 6) UML을 이용한 Smurf 정책 규칙의 모델링

(그림 5)는 IcmpAnomaly 그룹에 속하는 Smurf 정책 규칙이다. Smurf는 패턴 아이디는 6001이고, 우선순위는 2, 침입 영향은 6인 정책 규칙인데, 패킷 헤더의 프로토콜이 icmp인 조건과 목적지 주소가 homenet broadcast인 조건, ICMP type이 8인 조건을 1초 동안 3개의 패킷이 만족하고 2초 동안 20개의 패킷이 반복하여 만족하면 "Smurf를 이용한 서비스 거부 공격이 시도됨"이란 경보 메시지를 저장하고 경보 메시지를 디스플레이하는 행동을 취하라는것을 의미한다. 이와 같은 Smurf 정책 규칙은 (그림 6)과 같이 구조 클래스와 연관 클래스로 구성된 UML(Unified Modeling Language)로 표현할 수 있다.

3.2 정책 관리 도구의 기능

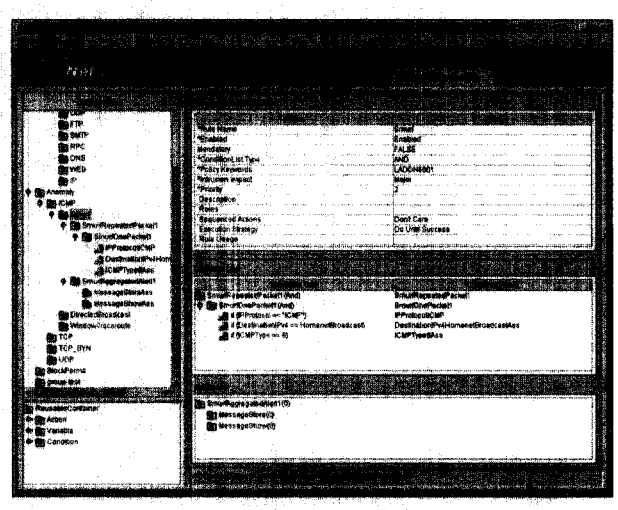
정책 관리 뷰어를 포함하여 정책 관리 도구의 기능은 다음과 같다.

3.2.1 정책 검색 기능

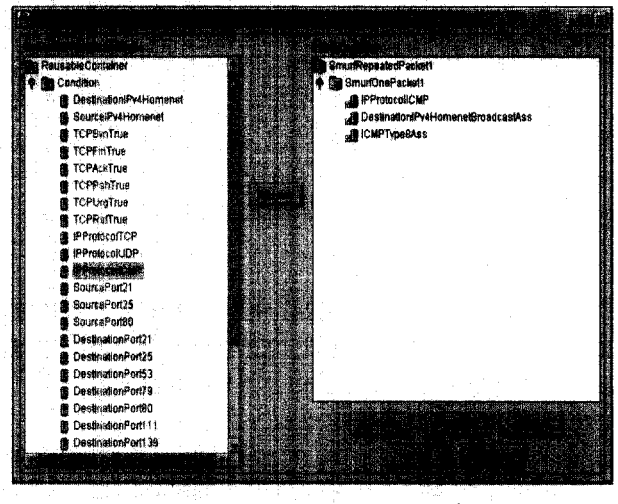
정책 관리 도구는 정책 저장소(Policy Repository)에 저장된 정책들을 검색하여 디스플레이한다. (그림 7)과 같이 정책 저장소에 존재하는 정책들은 그들이 가진 특성에 따라 그룹으로 묶을 수 있고 그룹들을 다시 상위 그룹으로 묶어서 관리할 수 있다. 왼쪽 화면은 정책을 구성하고 있는 객체를 트리 구조로 보여주고 있다. 정책을 구성하고 있는 객체들은 정책에 직접 연관된 것이 아니라 계층 구조로 되어 있는데 이러한 구조를 한 눈에 볼 수 있도록 하였다. 또한 오른쪽 화면은 왼쪽 화면에서 선택한 객체들의 속성 값과 그 객체와 연관된 하위 객체들의 정보를 계층 구조로 보여주고 있다. 이렇게 객체들이 소유하는 속성값 등 정책의 모든 정보를 한 눈에 볼 수 있으며, "Search by Name", "Search by Properties"의 탭과 같이 정책의 이름이나 키워드 등 객체 이름이나 속성에 따라 객체들을 검색할 수 있다.

3.2.2 재사용 객체를 이용한 정책 편집 기능

정책 관리 도구는 정책 생성시 재사용 객체 개념을 사용하여 편집 시 편리함을 제공한다. 객체는 특정 규칙을 구성하기 위해 사용되는 규칙에 특정한 객체(Rule-specific object)와 여러 규칙의 구성에 사용되는 재사용 가능한 객체(Reusable object)로 나누어 진다. 규칙에 특정한 객체는 연관된 특정 규칙을 위해 한번 사용되는 객체를 의미하고, 재사용 가능한 객체는 생성하여 재사용 용기에 저장하면 여러 규칙에서 연관시킴으로써 여러 규칙에서 반복적으로 사용될 수 있는 객체를 의미한다. (그림 8)을 보면 왼쪽에서 재사용 가능한 조건 객체들을 선택하여 연관시키고, 재사용 가능한 객체에 없는 객체들은 오른쪽 아래 "Create" 버튼을 사용하여 규칙에 특정한 객체를 생성함으로써 조건을 구성하도록 한 것을 볼 수 있다. 하나의 정책 규칙을 구성하는 객체들은 규칙 객체, 조건 객체, 동작 객체, 변수 객체, 값 객체인데, 이 중에서 조건 객체와 동작 객체의 일부, 그리고 모든 변수 객체들을 재사용 가능한 객체로 저장하였다. 조건 객체는 빈발하게 나타나는 객체들을 등록하였고, 동작 객체는 경보 메시지를 속성으로 가지는 AggregatedAlertAction을 제외하고 모두 재사용 가능한 객체로 등록하였다. 예를 들어 Smurf 정책의 경우 재사용 객체를 사용하는 객체들은 프로토콜이 icmp인 조건 객체, icmp type이 8인 조건 객체, 목적지 주소가 homenet broadcast인 조건 객체와 경보 메시지를 저장하는 동작 객체, 경보 메시지를 디스플레이하는 동작 객체와 모든 변수 객체이다. 이렇게 재사용 객체 개념을 사용하면 각 객체들을 생성할 필요없이 재사용 용기에서 연관만 시키면 되기 때문에 관리자가 정책을 생성할 때 편리함을 제공해 준다.



(그림 7) 정책 관리 뷰어의 첫 화면



(그림 8) 재사용 객체를 이용한 정책 조건 생성 화면

3.2.3 정책 변경 및 무결성과 일관성 확인 기능

정책 저장소의 스키마를 검증하여 정책 저장소의 정책들을 변경할 수 있는 기능과 정책 변경시 무결성 및 일관성 확인 기능을 제공한다. 정책 변경이란 삽입, 수정, 삭제를 말하며, 무결성 및 일관성 확인 기능은 정책 변경시 변경되는 객체의

속성에 입력한 데이터 값이 올바른 값인지, 생성될 수 있는 객체들은 어떤 것들이 있는지를 확인하는 것이다. 예를 들어 source port의 경우 입력되는 데이터가 0~65535 범위에 속하는지 확인하고, Protocol이 UDP라 하면 ICMP의 패킷 헤더 필드인 sequence number가 0인 조건이나 type이 8인 조건들은 생성되지 않고, UDP의 패킷 헤더 필드에 관한 조건만을 생성하도록 하는 것이다. 또한 재사용 가능한 객체를 이용한 정책 생성, 수정, 삭제시 올바른 연산에 대한 방법을 제공한다. 자세한 설명은 3.3절에서 기술한다.

3.2.4 객체 이름 생성 기능

정책 관리 도구는 객체 생성시 객체 이름의 자동 생성 기능을 제공한다. 객체 이름의 자동 생성 기능은 객체를 생성하기 위해 메뉴에서 원하는 객체의 이름을 선택하였을 때 실행되고, 자동 생성된 객체 이름을 포함하여 속성을 입력할 수 있는 창이 팝업되면서 사용자에게 제공된다. 예를 들어 "IPSpoofing"라는 정책 규칙의 경우, 근원지 주소와 목적지 주소가 Homenet이고 근원지 MACAddress와 목적지 MACAddress가 동일한 조건의 경우, 경보 메시지를 저장하고 관리자에게 경보 메시지를 보여주는 동작으로 구성된다. 그러므로 필요한 객체는 하나의 패킷에 대한 조건이므로 원패킷조건객체, 근원지 주소와 목적지 주소가 Homenet인 조건 객체, 근원지 MACAddress와 목적지 MACAddress가 동일한 조건 객체, 경보 메시지를 저장하고 경보 메시지를 관리자에게 보여주는 동작 객체이다. 이러한 객체들의 이름은 각각 IPSpoofingOnePacket, SourceIPv4Homenet, DestinationIPv4Homenet, SourceMACDestinationMAC, IPSpoofingAggregatedAlert, MessageStore, MessageShow이다. 원패킷조건객체는 규칙 객체의 이름과 조건 클래스의 이름으로 구성되었고, 비교 조건 객체는 변수 클래스와 값 이름으로, 동작 객체는 동작 클래스의 이름으로 구성된다. 객체 이름은 사용자가 임의로 정할 수도 있지만 정책 관리 도구에서 객체 이름의 자동 생성 기능은 사용자가 그 객체의 특성을 잘 표현하는 이름을 생각하고 입력해야 하는 번거로움을 덜어주기 때문에 편리함을 제공한다.

3.2.5 정책 변경 알림 기능

정책 관리 도구는 정책 변경시 변경된 정보를 정책 결정 모듈에 알리는 기능을 제공한다. 사용자가 정책 관리 뷰어를 통해 정책 변경을 요구하고 정책 저장소의 정책 변경이 확인되면 정책 관리 도구는 정책 결정 모듈에 정책 변경을 알린다. 이렇게 변경된 정책들을 정책 결정 모듈이 정책 클라이언트에 전달함으로써 수정된 정책을 기반으로 침입을 탐지하고 대응할 수 있다.

3.2.6 차단 정책 생성 기능

정책 관리 도구는 차단 정책을 자동으로 생성하는 기능을 제공한다. 메인 뷰어에는 정책 클라이언트에서 전송되는 경보 메시지가 실시간으로 전달되고, 경보량이 한계 값을 초과할 경우 블랙 리스트에 불량 사용자나 불량 호스트로 기록된다. 관리자가 블랙 리스트를 보고 차단 대상이 되는 불량 IP를 클릭하면 정책 관리 도구는 해당하는 객체들을 생성하여 정책 저장소에 저장하고 차단 리스트에 기록한다. 현재 공격을 받고 있는 특정 IP를 차단할 필요성이 있을 경우, 차단 정책을 생성하기 위해서는 정책을 편집하는 시간이 소요되나, 차단 정책을 자동으로 생성하는 기능을 제공함으로써 공격에 대해 빠른 대응을 할 수 있어 심각한 피해를 줄일 수 있다.

3.2.7 개발 환경

본 논문에서 기술하는 정책 관리 뷰어는 웹을 이용함으로써 관리자가 원격지에서도 장소에 구애받지 않고 접속할 수 있도록 환경을 구축하였으며, 이때 Java Web Start를 설치하였다. 보안을 위해 사용자 인증을 사용하였고, OpenSSL을 설치하였다. 자바 스윙과 서블릿을 이용하여 사용자에게 친근한 화면을 구축하였고, 정책 저장소는 LDAP을 이용하여 디렉토리 구조로 구축하였으며, 정책 저장소를 접근하기 위해 JNDI를 이용하였다.

요컨대, 정책 클라이언트에는 JDK, LDAP Client API, JNDI Client API를 위해 J2SE 1.3을 설치하였고, 정책 서버에는 gcc, J2SE 1.3, OpenLDAP, OpenSSL, 웹 서버인 Apache와 Jakarta Tomcat 4.0을 설치하였다.

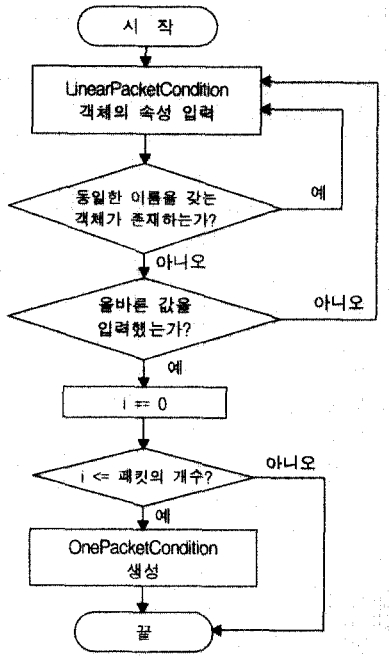
3.3 정책 관리 도구를 이용한 정책 데이터 입력 과정

객체를 생성할 경우 속성을 입력하였을 때 동일한 이름이나 키워드를 가진 규칙 객체가 존재하는지 확인하고, 속성이 정의한 범위에 속하는 값을 입력하였는지 확인한 후 생성하며, 재사용 객체를 추가할 것인지를 확인한다.

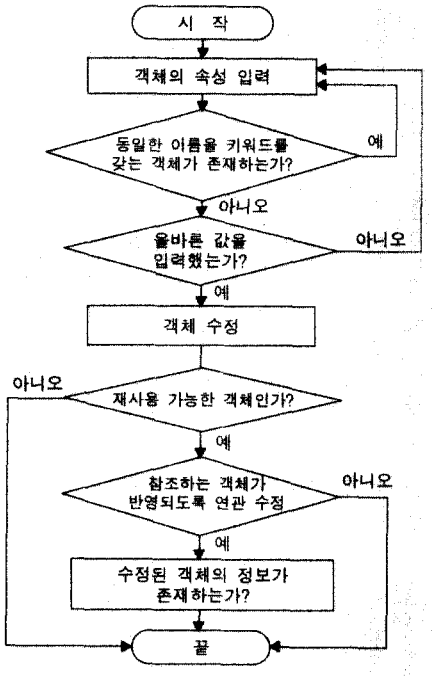
객체 생성 시 정책 관리 도구는 NSPIM에 기반하여 설계되었기 때문에 정책 규칙과 연관될 수 있는 객체들은 조건 객체나 동작 객체이고, 다른 객체들과는 연관될 수 없다. 조건 객체의 경우 모든 정책 규칙들은 하나의 패킷에 대한 조건, 반복되는 패킷에 대한 조건, 연속되는 패킷에 대한 조건, 또는 동일한 Identification 값과 근원지 주소 값을 가지는 패킷에 대한 조건 중에 하나를 가지기 때문에 최상위 조건 객체가 될 수 있는 객체는 OnePacketCondition객체, RepeatedPacketCondition 객체, LinearPacketCondition객체, IPFragmentationCondition 객체이다. 또한 RepeatedPacketCondition객체와 LinearPacketCondition객체, IPFragmentationCondition객체는 하나 이상의 OnePacketCondition 객체와 연관될 수 있다. (그림 9)는 LinearPacketCondition객체의 생성을 위한 작업 흐름도 이다. 이와 같이 정책 관리 도구는 NSPIM에 기초한 인터페이스를 제공함으로 올바른 정책을 생성하도록 돕는다.

객체를 수정할 경우 (그림 10)과 같이 규칙에 특정한 객체와 재사용 가능한 객체가 다른 절차를 갖는다. 규칙에 특정한 객체는 수정할때 동일한 이름을 갖는 객체가 있는지 확인하고, 수정하는 객체의 속성 값이 속성이 정의한 범위에 속하는지 확인하여 수정하지만, 재사용 가능한 객체를 수정하는 경우 규칙을 구성하고 있는 객체들 중에서 수정하는 재사용 가능한 객체를 참조하는 연관이 존재하는지를 검사하고 참조하

는 객체가 존재할 경우 참조하고 있는 연관들도 수정되었는지 확인하는 절차가 추가된다.



(그림 9) LinearPacketCondition 객체 생성 흐름도

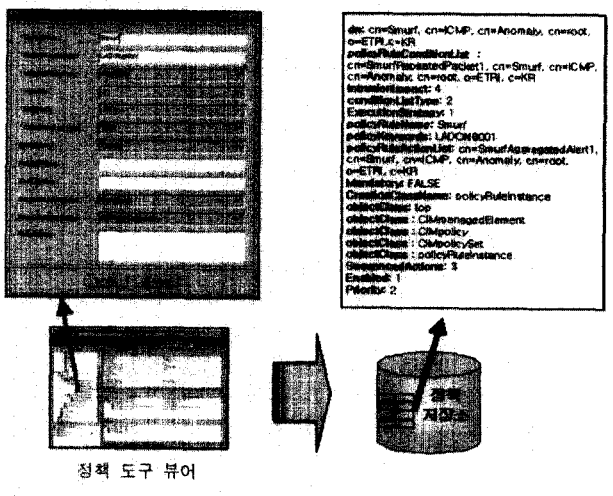


(그림 10) 객체 수정을 위한 작업 흐름도

객체를 삭제할 경우 또한 규칙에 특정한 객체와 재사용 가능한 객체가 다른 절차를 갖는데, 규칙에 특정한 객체에서 하위 객체가 존재하는 경우 하위 객체까지도 삭제할 건지 확인하고, 존재하지 않을 경우 삭제를 재확인하는 절차가 필요하다. 재사용 가능한 객체는 재사용 용기에 각각의 객체가 저장되어 있는 형태이므로 하위 객체가 존재하는 경우는 없다. 단지 삭제하는 재사용 가능한 객체를 참조하고 있는 객체가 존재할 경우는 삭제할수 없고 존재하지 않을 경우는 삭제를 재확인하고 삭제하는 절차가 존재한다.

정책을 데이터베이스에 저장하기 위해서는 LDIF(LDAP Data Interchange Format)를 관리자가 직접 입력하거나, 정책 정보들의 MOF(Managed Object Format)가 존재할 경우 MOF 컴파일러를 구축하여 LDIF로 변환시켜 데이터베이스에 저장해야 했다. 또한 관리자가 LDIF를 직접 입력할 때 NSPIM의 개념을 정확히 모두 알고 있어야 했다. 그러나 NSPIM에 기반한 정책 관리 도구를 사용함으로써 이러한 작업을 간소화하였다. 이와 같이 정책 관리 도구는 NSPIM을 기반한 설계를 통해 정책들을 정책 저장소의 스키마에 맞게 생성할 수 있으며, 정책 변경시 정책 저장소의 스키마를 검증하여 정책 변경 처리를 올바로 수행할 수 있다.

정책 관리 도구의 이러한 정책 편집 과정을 거친 후 생성되는 데이터는 정책 저장소에 저장된다. 정책 저장소는 데이터를 검색하기 쉬운 디렉토리 구조로써 사용자, 파일, 자원 등과 같은 객체들을 위한 정보 저장소이다. 정책 저장소는 LDAP과 같은 표준 디렉토리 접근 프로토콜을 사용함으로써 다른 형태의 클라이언트들도 디렉토리 데이터에 접근할 수 있도록 하며, 데이터가 여러 곳에 저장되어 있을지라도 하나의 저장소처럼 정보를 관리할 수 있다. 정책 저장소에서 디렉토리에 저장되는 가장 기본적인 정보의 단위는 엔트리(entry)인데, 새로운 엔트리가 디렉토리에 추가되면 그 객체의 속성들 뿐만 아니라 상위 객체의 속성들도 상속받게 된다. 엔트리들은 DIT(Directory Information Tree)라 불리는 트리 같은 계층 구조로 구성되며 DIT에서의 위치에 따라 상대적인 순서로 이름을 짓는데 이 때의 식별 가능한 이름을 DN(Distinguished Name)이라 한다.



(그림 11) 정책 관리 도구를 통해 입력되는 데이터

(그림 11)의 좌측은 Smurf 정책 규칙(PolicyRule) 객체의 속성을 입력하는 창이다. 정책 관리 도구를 이용하여 관리자

가 입력한 속성들은 LDAP을 통해 LDIF 형태로 정책 저장소에 저장된다. 정책 규칙 객체의 속성 값을 입력하였을 때 정책 저장소에 저장되는 정책 규칙 엔트리는 우측과 같다[10].

(그림 11)과 같이 정책 저장소에 저장되는 데이터는 그 객체의 속성 값과 상위 객체의 속성 값 뿐만 아니라, 디렉토리의 인스턴스들을 식별하는 DN값, 정책 규칙(PolicyRule)과 정책 규칙의 하위 객체인 조건(PolicyCodition) 객체와 행동(PolicyAction) 객체와의 연관성을 나타내는 엔트리의 식별 이름인 policyRuleConditionList, policyRuleActionList, 그리고 객체들의 클래스들을 기술하는 objectClass가 입력되었다[11].

4. 결 론

본 논문은 증가하는 침입에 대한 하나의 솔루션인 침입 탐지 및 대응을 위한 정책 서버 기반의 네트워크 보안 시스템에서 정책 정보 모델인 NSPIM을 기반으로 구축한 정책 관리 도구를 소개하였다. 본 논문이 소개한 정책 관리 도구는 정책을 검색, 변경하는 기능 뿐 아니라, 변경시 데이터 무결성 및 일관성 확인 기능이나 정책 결정 모듈로 정책 변경 정보를 알려주는 기능, 차단 정책 자동 생성 기능, 재사용 객체 개념 등을 이용한 편집 기능, 객체 이름의 자동 생성 기능 등을 제공하였고, 이러한 기능들을 통해 사용자에게 여러 가지 편의성을 제공하였다. 또한 NSPIM이란 정책 정보 모델을 기반으로 정책 관리 도구를 구축함으로써 정책 생성, 수정, 삭제시 정책 정보 모델에 맞는 정책 변경을 수행할 수 있었다.

향후 연구 과제는 NSPIM을 확장 구축하여 가능한 여러 종류의 정책 클라이언트를 포용할 수 있도록 하고, 침입을 탐지하고 대응하는 정책 뿐만 아니라 설정 정책이나 정책 클라이언트들 간의 제어를 위한 정책 그리고 이러한 정책들을 관리하는 정책 등 다양하고 계층적인 정책 정보 모델에 대한 연구가 필요하다.

참 고 문 헌

[1] http://www.ietf.org/html.charters/policy-charter.html.

[2] J. Strassner, E. Ellesson, B. Moore, and A. Westerinen, "Policy Core Information Model-Version 1 Specification," RFC 3060, February, 2001.

[3] B. Moore, L. Rafalow, Y. Ramberg, Y. Snir, J. Strassner, A. Westerinen, R. Chadha, M. Brunner, and R. Cohen, "Policy Core Information Model Extensions," work in progress, <draft-ietf-policy-pcim-ext-01>, April, 2001.

[4] Distributed Management Task Force, Inc., "Common Information Model(CIM) Specification," version 2.2, June, 1999.

[5] Sook-Yeon Kim, Myung-Eun Kim, Ki Young Kim, Jongsoo Jang, "Information Model for Policy-Based Network Security Management," The 16th International Conference on Information Networking, Session 8B - 4.1~11, 2002.

[6] Jinoh Kim, Kiyoung Kim, Jongsoo Jang, "Policy-Based Intrusion Detection and Automated Response Mechanism," The 16th International Conference on Information Networking, Session 2B-4.1~10, 2002.

[7] Steven Judd, John Strassner, "Directory Enabled Networks -Information Model and Base Schema," Draft v3.0c5, September, 1998.

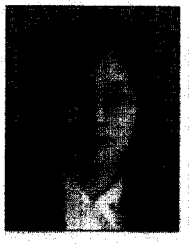
[8] Snir Y., Ramberg Y., Strassner J., Cohen R. : Policy Framework QoS Information Model, work in progress, <craft-ietf-policy-qos-info-model-03.txt>, April, 2001.

[9] Jason J., Rafalow L, Vyncke E. : IPsec Configuration Policy Model, work in progress, <draft-ietf-ipsp-config-policy-model-02.txt>, March, 2001.

[10] J. Strassner, A. Westerinen, E. Ellesson, B. Moore, R. Moats, "Policy Core LDAP Schema," work in progress, <draft-ietf-policy-core-schema-11.txt>, May, 2001.

[11] G. Good, "The LDAP Data Interchange Format (LDIF) Technical Specification," RFC 2849, June, 2000.

[12] A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog, A. Huynh, M. Carlson, J. Perry, S. Waldbusser, "Terminology for Policy-Based Management," November, 2001.



김 건 량

e-mail : glkim@etri.re.kr

1999년 전남대학교 전산학과 졸업(학사)

2001년 전남대학교 전산학과 졸업(이학석사)

2001년~현재 한국전자통신연구원 정보보호연구본부 보안게이트웨이연구팀

관심분야 : 네트워크 보안, PBNM, 데이터 마이닝

장 종 수

e-mail : jsjang@etri.re.kr

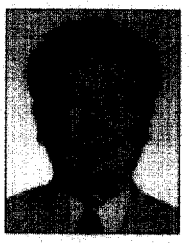
1984년 경북대학교 전자공학과 졸업(학사)

1986년 경북대학교 전자공학과 졸업(공학석사)

2000년 충북대학교 컴퓨터공학과 졸업(공학박사)

1989년~현재 한국전자통신연구원 책임연구원, 정보보호연구본부 보안게이트웨이연구팀 팀장

관심분야 : 네트워크 보안, PBNM, 능동네트워크



손 승 원

e-mail : swsohn@etri.re.kr

1984년 경북대학교 전자공학과 졸업(학사)

1994년 연세대학교 전자공학과 졸업(공학석사)

1999년 충북대학교 컴퓨터공학과 졸업(공학박사)

1991년~현재 한국전자통신연구원 책임연구원, 정보보호연구본부 네트워크보안연구부장

관심분야 : 네트워크 보안, 능동네트워크, 생체 인식