18-3-R04

고급언어 기반의 대규모 네트워크 보안 관리 시스템 개발
(Development of High-Level Language based Security Management System for Large-Scale Network)

김장하·이강희·김상욱

경북대학교 컴퓨터과학과

jhkim@cs.knu.ac.kr khlee@cs.knu.ac.kr swkim@cs.knu.ac.kr

요 약

대규모 네트워크에서는 보안 관리를 위한 여러 가지 기법이 적용되고 있다. 네트워크를 통해 전파되는 바이러스와 인터넷 웜의 경우 자동화된 공격 기법을 포함하여 단시간에 수많은 호스트를 감염시킨다. 네트워크의 링크 계층을 지원하기 위한 여러 가지 장비는 이러한 문제점을 해결하기 위한 보안 관리 기능을 제공하고 있다. 그러나 최근 사용되는 대부분의 네트워크 장비가 접근 제어와 같은 보안 기능을 포함하고 있지만 위급한 상황에 이러한 기능을 적절히 활용하는 것은 쉽지 않다. 2002년 초에 발생한 Slammer웜은 대규모 네트워크를 통해 빠르게 전파되었지만 관리자의 보안 기능 활성화 이전에 전체 네트워크로 확산된 대표적인 예이다. 네트워크를 구성하고 있는 장비들이 다양하고 그것들의 보안 설정이 서로 상이하기 때문에 긴급 상황에서 일괄적인 관리가 어렵다는 단점 때문이다.

이러한 문제점을 해결하기 위해 본 논문에서는 대규모 네트워크를 구성하는 다양한 하부 노드들의 보안 설정을 위한 공통적인 속성을 분석하고 이러한 정보를 추상적으로 기술할 수 있는 고급언어를 제안한다. 그리고 중간단계의 자료구조를 설계하여 고급 언어와 실제 하부 노드에 적용될 명령어의 변환을 돕는다. 전체 관리 시스템은 고급언어 컴파일러가 포함된 관리자 인터페이스, 네트워크 도메인을 관리하기 위한 도메인 관리 서버와 하부 노드 인터프리터 에이전트로 구성된다. 관리자가 고급언어로 기술한 보안 정책은 컴파일러에 의해 중간단계의 자료구조로 변환되고 도메인 관리 서버에서 연산되고 인터프리터에 의해 하부 노드에 최종 적용된다. 기존의 네트워크 보안 관리를 위해 서로 다른 하부 노드를 순차적으로 설정하였던 것에 비해 본 논문에서 제안하는 보안 관리 시스템은 고급 언어로 기술된 하나의 정책으로 전체 시스템의 관리를 받고 있는 하부노드들의 보안 설정을 일괄적으로 제어할 수 있다.

1. 서 론

1.1. 연구 배경

네트워크 환경이 발전함에 따라, 악성 코드와 유해 트래픽을 차단하기 위한 다양한 대책이 요구된다. 대규모 네트워크를 통해서 전파되는 유해 트래픽은 침해 사고 발생에서 주요한 과정으로 수행된다[1][2][3]. 빠르게 전파되고 있는 트래픽을 차단하기 위해서 최근 개발되고 있는 다양한 네트워크 장비에서는 접근 제어 목록 기능을 기본적으로 포함하고 있다. 접근 제어 목록 기능을 포함한 네트워크 장비는 유해 트래픽의

특성을 기술하기 위한 각각의 기술 문법을 가지고 있다. 장비 제조사와 내장 운영체제 버전에 따라 접근 제어 기술 문법은 다르기 때문에 접근 제어 목록을 기술하기 위한 다양한 형태의 명령어가 필요하다. 수많은 네트워크 노드로 구성된 대규모 네트워크에서 긴박한 대처를 필요로 하는 상황이 발생할 경우, 전체 네트워크 노드를 일괄적으로 제어할 수 있는 방법이 없다.

본 논문에서는 이러한 문제를 해결하기 위해 접근 제어 목록 통합 기능을 가진 고급 접근 제어 정책 언어, 중간 형태의 자료 구조와 도메인 관리 서버, 인터프리터 에이전트를 포함한 대규모 네트워크 통합 보안 관리 시스템을 제안한다. 제안된 시스템의 고급 언어 컴파일러는 기존의 프로그래밍 언어에 유사한 문법을 제공하여 접근 제어 정책에 직관적이고 친숙한 인터페이스를 제공한다. 고급 언어로 작성된 접근 제어 정책이 컴파일러에 의해 실제 네트워크 노드의 접근 제어 목록으로 적용되기 위한 중간 형태의 자료 구조로 변환된다[4]. 대규모 네트워크 보안 관리 시스템은 컴파일러가 포함된 관리자 인터페이스와 각 네트워크 도메인을 관리하는 도메인 관리 서버, 하부노드 설정을 제어하는 인터프리터 에이전트로 구성된다. 도메인 관리 서버는 중간 형태의 자료 구조로 전달되는 접근 제어 정책의 충돌 탐지 및 순응 기능을 포함한 정책 관리 서버이다. 인터프리터 에이전트는 도메인 관리 서버에서 전달되는 정책을 하부 노드에 직접적으로 반영하기 위한 도메인 관리 서버와 관리 대상 하부 노드 사이의 브리지 역할을 한다[5].

고급 언어는 다양한 네트워크 장비에서 지원하고 있는 접근 제어 목록의 공통적 특성을 포함하고 있다[6]. 각 장비에 대한 구체적인 정보 기술을 피하고 추상화된 접근 제어 목록을 기술함으로써 상이한 접근 제어 문법을 가진 네트워크 장비에 공통적으로 적용할 수 있는 고급의 접근 제어 정책을 기술할 수 있다. 제안한 시스템을 이용하여 유입되는 유해 트래픽을 제어하기 위한 접근 제어 목록을 추상화 단계에서 기술하여 일괄적으로 관리 할 수 있다[7].

본 장에서는 대규모 네트워크 환경에서의 보안 관리 시스템에 관한 연구 배경을 설명하고, 기존에 존재하는 대규모 네트워크 보안 관리에 관한 관련 연구를 소개한다. 또한 제시하는 대규모 네트워크 환경에 적합한 보안 관리 시스템에 대하여 설명한다.

1.2. 관련 연구

현재 대규모 네트워크 보안 관리 기술로 가장 많이 응용되고 있는 분야는 ESM(Enterprise Security Management)[8]이며 응용 기술로 CSPM (Cisco Security Policy Management)[9]와 PDL(Policy Description Language)[10]등이 사용되고 있다.

1.2.1. STRONGMAN 프로젝트

DARPA(Defense Advanced Research Project Agency) DC(Dynamic Coalition)[11] 프로그램 중 다차원 보안 정책 관리(Multi-Dimensional Security Policy Management) 분야의 일부로서 University of Pennsylvania와 AT&T Labs - Research가 공동으로 연구 중이다. 고급 언어를 이용하여 대규모 네트워크를 위한 보안 정책을 표현하고 행위 중심의 인증을 적용하는 메커니즘을 연구, 개발한다[12].

1.2.2. NetSPoC 컴파일러

NetSPoC 컴파일러는 독일 Fraunhofer 연구소에서 Berlios 오픈 소스 포탈을 통해 개발을 추진하고 있는 네트워크 보안 정책 컴파일러이다. 시스코 라우터의 규칙과 PIX 방화벽 규칙, 리눅스 iptable 과 라우터 설정을 지원한다. 정책 기술을 위한 언어를 입력하면 네 가지 형태의 규

칙을 생성한다. 이 컴파일러는 기존의 시스코 보안 정책 관리자를 대체하기 위해 개발되었으며 펄 언어를 지원하는 플랫폼으로 이식가능하다[13].

1.2.3. Cisco Security Policy Management

CSPM은 시스코사에서 독자적으로 개발한 보안 정책 관리 시스템이다. 시스코사에서 개발된 라우터와 VPN(Virtual Private Network) 장비를 통합 관리하고 시스코 IDS 센서를 모니터링 할 수 있는 시스템이다. 시스코사 제품에서 사용되는 PIX 방화벽과 IOS 운영체제를 대상으로 개발되어 타사 제품과 연동이 불가능한 단점이 있다[9].

1.2.4. Enterprise Security Management

ESM이란 침입차단시스템, IDS(Intrusion Detection System), VPN 등 다양한 종류의 보안 솔루션을 하나로 모은 통합 보안 관리 시스템으로 보안 관리보다는 통합 시스템 관리(Enterprise System Management)의 형태로 시스템 관리의 영역에서 먼저 출발하였다. 보안관리 측면에서의 ESM은 Firewall, VPN, 바이러스 검사, 콘텐츠 필터링, URL 모니터링/필터링, 침입탐지 등 별개의 보안 구성 요소를 일관적인 전체로 결합하여, 인증과 감시, 허가에서 네트워크 관리에 이르기까지 모든 것들을 망라하는 통합관리로 연구되고 있으나 중앙집중식 관리의 문제점과 새로운 구성요소 추가에 어려움이 있는 단점이 있다[8].

1.2.5. Policy Description Language

PDL은 Bell-Labs에서 개발된 이벤트 방식의 언어이다. 발생한 이벤트에 대한 행위를 정의하는 함수로서 방어 메커니즘을 정의하기 위하여 데이터베이스의 이벤트-조건-행위 패러다임을 사용한다. 가장 기본적인 형태는 "event cause action if condition" 의 형태로 특정 이벤트가 발생하고 이 이벤트가 발생한 조건이 참이면 행위를 수행하라는 구문이다. PDL 은 접근 제어를 위한 방어 메커니즘의 표현과 복합 방어 메커니즘의 표현은 지원하지 않는다. 하지만 다양한 네트워크 동작이나 관리 시나리오의 표현에 있어서 장점을 가진다[10].

1.3. 논문 구성

본 논문은 제 2장에서 대규모 네트워크에서 보안 관리 시스템을 기술하고, 제 3장에서는 제안하는 보안 정책 언어 기술을 위한 고급 언어의 기능과 특성을 설명한다. 제 4장에서는 제안한 고급언어 기반의 보안 관리 시스템의 기능을 분석하여 다른 보안 관리 시스템과 비교한다. 제 5장에서는 구현 결과를 제 6장에서는 결론과 문제점을 기술한다.

2. 대규모 네트워크 보안 관리 시스템

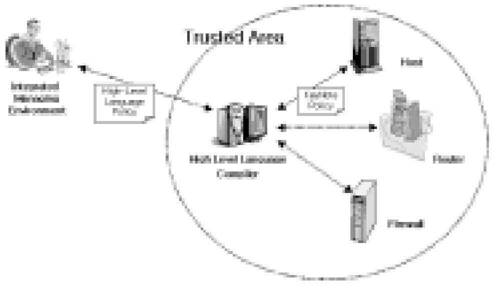
2.1 대규모 네트워크 환경

대규모 네트워크에서 관리하고자 하는 구성 요소는 기본적으로 라우터, 일반 호스트, 방화벽 등 여러 가지가 있다. 이들 요소는 일반적인 공통된 역할과 요소마다 각각의 특수한 역할을 가지고 있다. 대규모 네트워크 관리에서 주목하는 것은 이들의 공통된 역할과 기능이다. 대규모 네트워크에서 이들을 일일이 통제할 수 없기 때문에 공통된 추상적 기능을 이용한 일괄적 관리가 가능할 경우, 절차적, 시간적으로 매우 효과적으로 긴급한 상황에 대처할 수 있기 때문이다.

그러나 그러한 구성 요소가 공통된 기능을 가지고 있긴 하지만, 설정하기 위한 인터페이스, 형식, 인자 등은 매우 다양한 형식을 가지고 있다. 때문에, 일괄적 관리를 위해서는 각각의 기

능을 아우를 수 있는 형식이 필요하며, 그러한 형식을 구성 요소 각각의 형식에 맞게 변형하기 위한 메커니즘과 도구가 필요하다.

이와 같은 도구로는 대규모 네트워크의 제어하는데 필요한 적절한 수준의 정보를 저장한 데이터베이스, 네트워크 구성 요소의 접근 방식과 세부적인 사항에 독립적인 추상적 네트워크 모델을 기반으로 하는 고급의 언어, 그리고 네트워크 구조와 구성 요소의 변동에 영향을 받지 않는 인증 메커니즘이 요구된다. 그림1은 이러한 요소들로 구성된 대규모 네트워크 보안 관리 구조를 그림으로 나타낸 것이다.


Trusted Area
Integrated Networking Environment
High-Level Language Policy
Host
Router
High Level Language Compiler
Firewall

(그림1) 대규모 네트워크 관리 구조

대규모 네트워크에는 다양한 구성 요소와 이들 사이의 복잡한 관계로 구성되어 있다. 때문에 그것을 관리하고 제어하기 위해서는 자동화된 관리 메커니즘이 요구되며, 그러한 메커니즘에 의한 실제적인 구성 요소에 대한 접근과 제어를 위해서는 일정 수준의 세부 정보가 필요하다.

대규모 네트워크를 구성하는 다양한 요소는 빠르고 효과적으로 접근하기 위해서는 여러 가지 구성 요소를 통합할 수 있는 고급의 언어가 필요하다. 이러한 언어는 구성 요소의 특징이 효과적으로 반영되어야 하며, 네트워크 보안 제어를 위한 직관적이면서도 효율적인 기능과 구조가 요구된다.

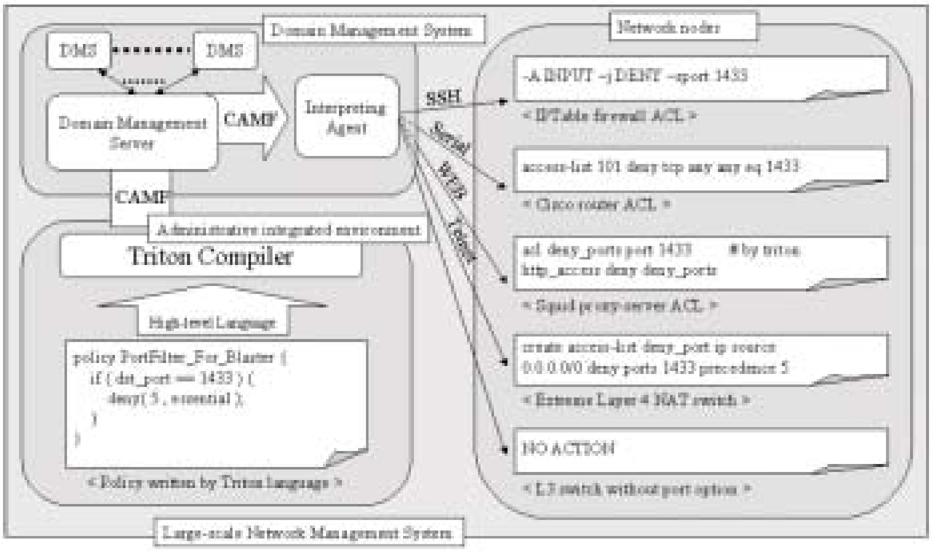
네트워크에 포함되어 있는 구성 요소를 제어하기 위해서는 그에 합당한 권한을 획득해야 하며, 그것을 위해서는 인증 정보가 서로 필요하다. 그러나 대규모 네트워크는 구조와 구성 요소를 일일이 파악하기 어려우며, 변동이 심하고 제어의 종류와 계층도 다양하기 때문에 모든 구성 요소가 인증 정보를 충분히 소유하기 어렵다. 때문에, 접근 가능한 모든 사용자의 정보를 필요로 하는 기존의 인증 방식을 벗어나 접근 권한을 위임 받아서 인증 정보가 없어도 인증 절차가 가능한 방식이 요구된다.

이러한 요구사항을 반영하기 위해 보안 관리 시스템은 KeyNote를 이용한다[14]. KeyNote는 접근 의도를 가진 모든 사용자에 대한 정보 대신 접근에 필요로 하는 실질적인 권한의 소유 또는 위임을 검사함으로써, 간단하고 효율적인 인증 정보 관리를 가능하게 한다.

2.2 하부노드의 추상화된 표현

다양한 하부 노드를 제어하기 위해서 언어는 추상적 측면에서 일관된 정책을 기술한다. 최근 네트워크 노드는 기본적으로 3계층에서의 필터링을 제공하고 있다. 네트워크 관리자는 악성 트래픽을 차단하기 위해서 필터링 기능을 사용한다. 서로 다른 운영체제와 설정과정을 가지고 있는 노드는 급변하는 네트워크 상황에 유연하게 대처하기 어렵다. 자동화된 악성 트래픽을 차단하기 위해서는 초기 진압이 가장 중요하다. 대규모 네트워크 관리 관점에서 일관된 정책을 설정하는 것은 이질적인 하부 노드를 일괄적으로 통합 관리 가능한 것이다. 특정 상황에서 단한번의 설정으로 전체 네트워크의 구성요소들의 설정에 반영되어야 한다.

네트워크 노드마다 특성과 설정이 다르므로 이러한 정보를 일괄적으로 수집하고 설정하기 위해서는 추상화 과정이 필요하다. 대부분의 네트워크 노드가 포함하고 있는 보안 기능에 대한 일반화된 정보가 필요하다. 하부 노드 설정을 추상적으로 기술함으로써 하나의 명령으로 전체 네트워크를 제어할 수 있다. 특정 네트워크 노


Domain Management System
DMS
DMS
Domain Management Server
CAMF
Interpreting Agent
CAMF
Administrative integrated environment
Triton Compiler
High-level Language
policy PortFilter_For_Blaster {
if (dst_port == 1433) {
deny(5 , essential);
}
}
< Policy written by Triton language >
SSH
Serial
WEB
Telnet
Network nodes
-A INPUT -j DENY --sport 1433
< IPTable firewall ACL >
access-list 101 deny tcp any any eq 1433
< Cisco router ACL >
acl deny_ports port 1433 #by triton
http_access deny deny_ports
< Squid proxy-server ACL >
create access-list deny_port ip source
0.0.0.0/0 deny ports 1433 precedence 5
< Extreme Layer 4 NAT switch >
NO ACTION
< L3 switch without port option >
Large-scale Network Management System

(그림2) 추상적 정책의 하부노드 적용

드의 기능에 의존적이지 않고 다양한 네트워크 노드의 기능을 포괄적으로 포함할 수 있는 추상화 특성이 필요하다. 고급 언어는 이러한 추상화 기능을 통해 다양한 네트워크 노드를 일괄적으로 제어할 수 있다. 그림2는 추상적인 정책이 실제 하부노드 설정으로 변환되는 과정을 나타내고 있다.

시스템은 고급 언어를 실제 명령어로 변환하기 위해 중간 단계의 보안 접근 제어 정보를 포함하는 구조체를 이용한다. 본 시스템에서는 중간 단계 구조체를 CAMF(Common Access Management Form)이라고 명한다.

중간 형태 정보가 인터프리터 에이전트로 전달된다. 인터프리터 에이전트는 하부 노드에 대한 정보를 이용한다. 각 하부노드의 특성에 따른 ACL 명령어를 생성한다. 그러나 고급 언어로 기술한 정책이 네트워크 장비 특성상 적용 불가능할 경우 동작을 취하지 않는다. 다양한 네트워크 장비를 지원하기 위해 다양한 형태의 설정 정보를 확보해야한다. 인터프리터 에이전트는 새로운 하부 노드 설정 정보를 획득하기 위한 CAMF을 가진다. CAMF를 통하여 빠르고 쉽게 새로운 하부 노드 설정 정보를 추가 확장할 수 있다.

3. 보안 정책 기술을 위한 고급 언어

3.1. 고급언어

고급 언어는 세부적으로 알 수 없는 네트워크를 일괄적으로 가장 효과적으로 제어하기 위한 것으로, 아래와 같은 특징을 가진다.

- 네트워크를 구성하는 요소, 장비의 공통적 특징과 관계, 전체 네트워크 구성을 관리의 대상으로 한다.
- 세부적인 네트워크 구조 및 속성보다는 일반적인 추상적 네트워크 모델을 기반으로 정책 설정을 한다.
- 네트워크 관리에 대한 방법(HOW)의 기술보다는 관리자가 결과적인 상황으로 요구하는 목적(WHAT)의 기술에 초점을 맞춘다.

그리고 아래와 같은 일반적인 관점에서의 언어적 특징을 가진다.

- 일반적 프로그램 언어 형태에 가깝게 한다.
- 네트워크 모델과 직관적으로 연결이 가능하게 한다.

고급 언어의 기본 목적은 네트워크 트래픽의 효과적인 제어다. 특정 범위 안에서 특정 조건에 따라 네트워크 구성 요소 또는 트래픽을 제어하기 위한 것이다. 따라서 고급 언어는 아래와 같은 형식적 특징을 가진다.

- 관리 범위를 결정할 수 있다.
- 관리를 통해 최종적으로 나타나는 결과를 결정할 수 있다.

고급 정책 언어는 단일 정책 단위로 구조화되어 있으며, 정책을 적용할 도메인과 트래픽의 특성을 기술할 수 있다. 그림3은 고급 정책 언어의 문법 구조이다.

```
<program> ::= <policy_list>
<policy_list> ::= <policy>
                | <policy_list> <policy>
<policy> ::= 'policy' <id> <statement_block>
<statement_block> ::= '' <statement_list> ''
<statement_list> ::= <statement>
                   | <statement_list>
<statement>
<statement> ::= <direction_statement>
              | <for_statement>
              | <if_statement>
              | <action_statement>
              | <assign_statement>
<direction_statement> ::= 'incoming'
<statement_block>
                        | 'outgoing'
<statement_block>
<for_statement> ::= 'for' '(' <domain_list> ')'
<statement_block>
<if_statement> ::= 'if' '(' <condition> ')'
<statement_block>
<action_statement> ::= <accept_statement> ';'
                     | <deny_statement> ';'
                     | <report_statement>
';'
<assign_statement> ::= <id> '=' <expression>
';'
<domain_list> ::= <domain>
                | <domain_list> <domain>
<condition> ::= <condition> 'or' <condition>
              | <sub_condition>
<sub_condition> ::= <sub_condition> 'and'
<sub_condition>
                  | '(' <condition> ')'
                  | <expression>
<condition_operation> <expression>
<accept_statement> ::= 'accept'
<deny_statement> ::= 'deny'
<report_statement> ::= 'report'
<expression> ::= <expression>
<arithmetic_operation> <expression>
               | <number>
               | <domain>
               | <id>
<domain> ::= '"' <string> '"'
<number> ::= <digit>
           | <number> <digit>
<id> ::= <string>
<string> ::= <letter>
           | <string> <letter>
<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
8 | 9
<letter> ::= A .. Z | a .. z
```

(그림 3) 제안한 고급 언어 문법 구조

3.2. 중간단계 자료구조 CAMF

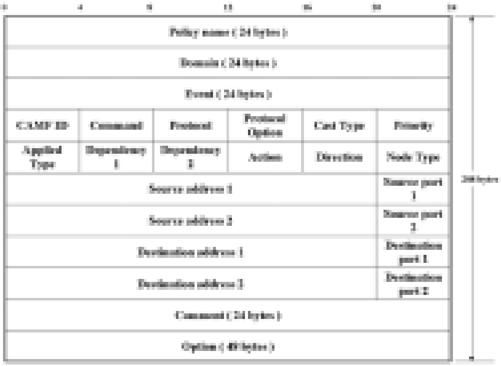
고급 언어로 작성된 정책은 네트워크 노드에 적용되기 위해서 저수준 명령으로 변환되어야 한다. 그러나 이 과정에서 다양한 네트워크 노드의 설정 정보를 컴파일러가 동적으로 포함할 수 없다. 시스템에서 제안하는 언어는 추상적인 정책을 저수준 명령으로 변환하기 앞서, 도메인 간 전송, 정책 순응, 저수준 명령 변환을 쉽게 하기 위하여 중간 단계에 해당하는 구조로 변환된다. 컴파일러에 의해 중간단계로 변환된 정책 정보는 각 도메인으로 전송된 후, 순응과정을 거친다. 검증된 중간단계 정보는 하부 노드에 적용될 저수준 명령으로 변환된다.

고급 언어로 작성된 보안 정책을 실제 하부 노드의 적용 규칙으로 변환하기 위하여, 이질적이고 다양한 적용 규칙의 공통적인 요소를 추상

화시켜, 대규모 네트워크의 구성 요소에 일괄적으로 적용할 수 있는 중간 단계의 형식을 정의한다. 중간 단계 형식 CAMF는 아래 그림4와 나타난다.

CAMF를 이용하여 네트워크에 적용되는 일반적인 하부 노드의 공통적인 트래픽 제어 조건을 기술함으로서 고급 정책을 위한 추상화된 네트워크 환경을 제공한다. CAMF의 구조는 어느 하부 노드에서도 수용할 수 있게 각 노드의 공통적인 요소를 추출하여 정의한 것이다. 단순하지만 각 노드들이 충분히 수용할 만한 정보의 항목만 추출하였으므로 쉽게 적용이 가능하다. 하지만 모든 노드에 적용할 수 있게 설계되었으므로 각각의 노드들마다의 특징적인 기능에 대한 정보를 담기가 어렵고, 그에 따라 세부적인 설정 정보를 담을 수가 없다.

중간 수준의 형식은 각각의 구성 요소가 요구하는 접근 형식에서 공통적 요소를 추출하여 일반적인 접근 형식을 나타내기 위한 추상적 형식이다. CAMF 구조는 정책 이름, 적용 대상 도메인, 이벤트, 고유 식별자, 정책수행명령, 프로토콜, 우선순위, 필수성, 의존성, 액션, 방향성, 하부노드종류, 트래픽을 기술하기 위한 출발지와 도착지의 주소와 포트, 코멘트와 옵션 정보가 포함한다.



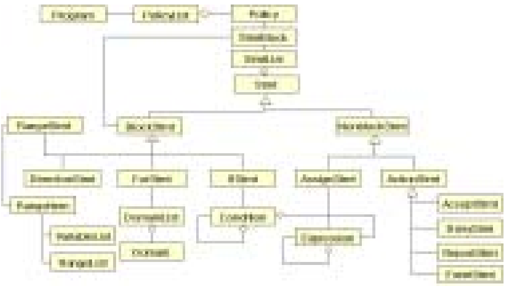
0
4
8
12
16
20
24
208 bytes

Policy name (24 bytes)					
Domain (24 bytes)					
Event (24 bytes)					
CAMF ID	Command	Protocol	Protocol Option	Cast Type	Priority
Applied Type	Dependency 1	Dependency 2	Action	Direction	Node Type
Source address 1					Source port 1
Source address 2					Source port 2
Destination address 1					Destination port 1
Destination address 2					Destination port 2
Comment (24 bytes)					
Option (40 bytes)					

(그림4) CAMF 구조

3.3. CAMF 생성 기법

파스 트리를 구성하기 위한 클래스는 고급 언어의 BNF에 표현되는 비단말(nonterminal)과 일치한다. 따라서 BNF의 전체 구성은 클래스 사이의 상속과 참조 구조로 표현될 수 있다. 그림5는 클래스 사이의 관계를 표현한 것이다.



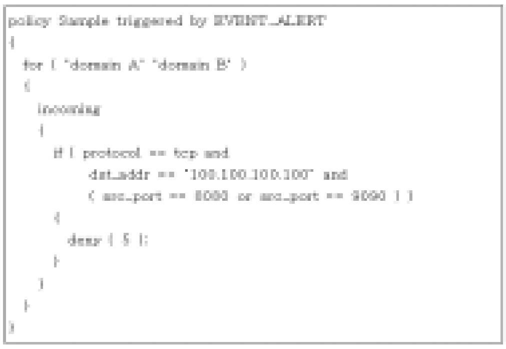
(그림5) 고급 언어 컴파일러 클래스

BNF에서 하나의 비단말이 다른 비단말로 치환되는 형태는 클래스에서의 참조 구조로 표현되며, 동일한 속성을 가지는 여러 개의 비단말을 하나로 묶어주는 비단말은 상속의 구조로 표현되어 있다. 그리고 동일한 비단말의 하나 이상의 집합으로 표현되는 비단말은 컴포지션 관계로 표현된다.

LEX를 이용하여 기본적인 스캐닝과정을 수행한 후 YACC을 이용하여 파스 트리를 생성하기 위한 탐색과정에서 CAMF 트리를 생성한다. 파스 트리 생성 과정 중에 단말(terminal)을 따라 전체 문장을 탐색한다. 탐색 과정에서 CAMF 트리의 각 노드를 생성하고 파싱과 함께 CAMF 트리가 생성된다. 생성된 CAMF를 깊이 우선 탐색(Depth First Search)하면 탐색 과정에서 각 노드의 의미적 필요성에 따라 트리의 변형과정을 거친다. 탐색 과정에서는 트리의 단말에 도달할 때 마다 하나의 CAMF가 생성된다.

그림6은 고급 언어로 기술된 보안 정책의 예이다. 예를 통해 파스 트리로부터 CAMF 트리를 생성하고 CAMF 정보를 생성하는 과정을 설

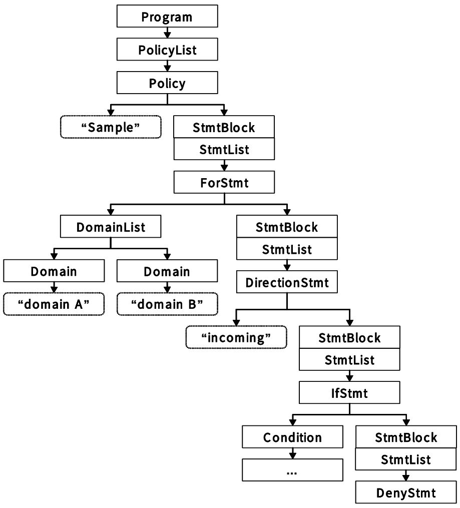
명할 수 있다.



```
policy Sample triggered by EVENT_ALERT
{
  for ( "domain A" "domain B" )
  {
    incoming
    {
      if ( protocol == tcp and
           dst_addr == "100.100.100.100" and
           ( src_port == 8080 or src_port == 9090 ) )
      {
        deny ( 5 );
      }
    }
  }
}
```

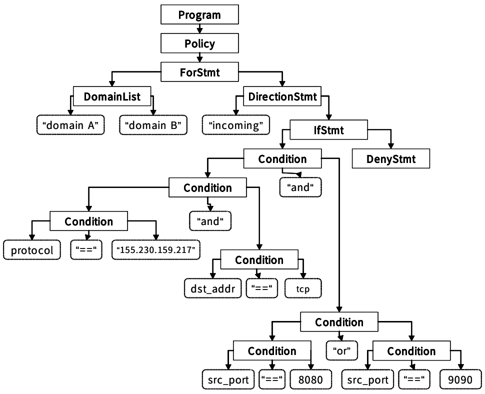
(그림6) 고급 언어로 기술된 보안 정책의 예

고급 언어로 작성된 정책을 CAMF 형태로 변환하는 과정을 설명한다. 그림3-4와 같은 고급 언어 정책은 앞서 언급한 스캐너와 파서를 거쳐 그림7과 같은 형태의 파스 트리로 파싱된다.



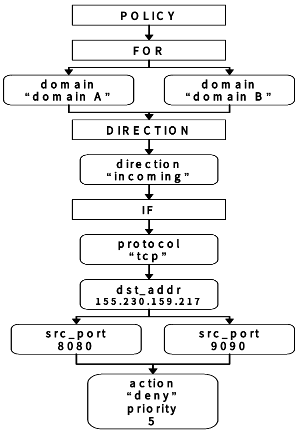
(그림7) 파스 트리의 예

파스 트리에 최단말에 위치한 Condition 노드의 경우 다양한 조건의 논리 접속사에 따라 탐색과정에서 다양한 형태로 변형되어야 한다. 그림8은 그림7의 파스 트리를 최적화한 트리이다.



(그림8) 최적화된 파스 트리

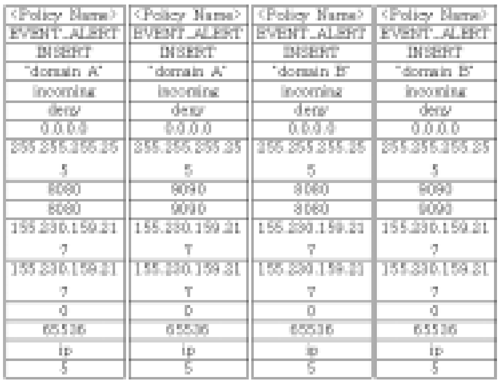
최적화된 파스 트리를 기준으로 깊이 우선 탐색을 시도하며 각 노드에서 필요한 정보를 추출한다. 그림9의 트리는 최종 생성되는 CAMF 트리를 그림으로 나타낸 것이다. 파스 트리의 변형으로 생성된 CAMF 트리는 파스 트리가 내포하고 있는 도메인의 범위와 트래픽 제어의 조건의 직렬과 병렬의 연결로 나타낸 것이다. 즉, 파스 트리에서 발생할 수 있는 모든 의미적인 경우의 수를 CAMF 트리에서는 루트 노드와 말단 노드 사이에 나타나는 패스의 종류로 표현한 것이다. 따라서, 파스 트리가 의미하는 작업의 종류를 CAMF 트리에서는 루트 노드에서 말단 노드까지의 정보를 누적시킨 것으로 표현하게 된다.



(그림9) CAMF 트리의 예

고급 정책에서 표현하려는 여러 작업 중 하나를 CAMF 노드로 표현하며, 정책의 수행에 필요한 모든 정보와 적업 내용은 CAMF 트리의 전체 탐색의 결과로 나타낼 수 있다. 따라서 고급 정책의 최종 결과물인 CAMF 리스트는 CAMF 트리의 깊이 우선 탐색으로 발생하는 모든 CAMF 노드의 집합인 것이다.

그림10은 그림9의 CAMF 트리를 깊이 우선 탐색했을 경우 발생은 모든 CAMF이다.



<Policy Name>	<Policy Name>	<Policy Name>	<Policy Name>
EVENT_ALERT	EVENT_ALERT	EVENT_ALERT	EVENT_ALERT
INSERT	INSERT	INSERT	INSERT
"domain A"	"domain A"	"domain B"	"domain B"
incoming	incoming	incoming	incoming
deny	deny	deny	deny
0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0
255.255.255.255	255.255.255.255	255.255.255.255	255.255.255.255
8080	9090	8080	9090
8080	9090	8080	9090
155.230.159.217	155.230.159.217	155.230.159.217	155.230.159.217
155.230.159.217	155.230.159.217	155.230.159.217	155.230.159.217
0	0	0	0
65536	65536	65536	65536
ip	ip	ip	ip
5	5	5	5

(그림10) CAMF 리스트의 예

4. 언어 기반의 보안 관리 시스템

대규모 네트워크 보안 관리 시스템은 관리자 인터페이스에서 작성된 정책을 정책 관리 서버에서 CAMF로 변환하여 하부 노드 인터프리터로 전달하기 위한 것이다. 정책 관리 서버는 고급 정책 언어로 작성된 보안 정책을 컴파일하고, 그 결과로 생성된 CAMF 리스트를 정책에 명시되어 있는 각 도메인별로 분류하여 각각의 도메인 인터프리터로 전달한다.

4.1. 관리자 인터페이스

관리자 인터페이스에서는 하부 노드 보안 설정을 위한 고급 언어 작성 기능과 CAMF 생성을 위한 컴파일 기능을 포함하고 있다. 그 외에도 네트워크 상태 감시와 기존에 작성된 정책의 관리 기능과 관리 도메인 간의 등록과 협상 과정을 위한 등록 협상 인터페이스를 가지고 있다.

4.2. 도메인 관리 서버

대규모 네트워크의 특성상 관리자 인터페이스는 하나 이상이 될 수 있으나, 정책 관리 서버는 네트워크 정보의 관리 일원화를 위해 하나만 존재하게 된다. 따라서 관리자 인터페이스에서 정책 관리 서버의 접근은 원격으로 이루어지게 되며, 정책 관리 서버에서 각 도메인의 인터프리터로의 접근 또한 원격을 기본으로 한다. 때문에, 각 시스템 사이의 정보 전달 과정의 오류 가능성을 줄이고, 접근에 따른 인증 과정을 포함하고 있다.

대규모 네트워크는 관리 개념적으로 소규모 네트워크를 계층적으로 구성하고 있다. 계층적 구조는 최상위에서 전달되는 명령을 빠르게 전체 네트워크로 전달할 수 있는 장점을 가진다. 본 시스템에서 도메인 관리 서버는 재귀적 구조를 수용하여 각 소규모 네트워크에서 독립적인 관리 주체의 역할을 수행하면서 상위 네트워크의 명령을 수신하고 하위 네트워크로 명령을 전달하는 기능을 포함한다.

4.3. 인터프리터 에이전트

고급 언어로 작성된 정책은 네트워크를 구성하는 다수의 하부 노드에 직접 적용하기 위해서 다양한 하부 노드의 공통적 특성을 정리한 중간 형태의 CAMF를 이용한다. CAMF를 기반으로 실제 하부 노드에 보안 설정 정보를 제어하기 위한 인터프리터 에이전트가 필요하다. 인터프리터는 CAMF를 분석하여 하부노드에 맞는 Rule-Set으로 변환을 하게 되며, 라우터나 웹 서버, 방화벽과 같은 제어 대상 네트워크 노드와 도메인 관리 서버의 사이에 위치한다.

인터프리터 에이전트는 정책 변환 기능과 처리 결과 통보 기능을 기본적으로 가진다. 정책 변환 기능은 도메인 서버로부터 전송받은 CAMF를 해당 하부노드 형식에 맞는 Rule-Set으로 변환하는 기능이다. 일반적인 네트워크 노드들의 Rule-set 형태는 조금씩 다르지만 CAMF가 포함하는 액션, 프로토콜, 출발지, 도착지, 방향성 등을 파라메타로 가진다. 따라서 CAMF를 거치는 고급 언어로부터 네트워크 노드의 Rule-set을 변환하는 과정은 그렇게 어렵지 않다. 처리결과 통보 기능은 하부노드에 Rule-Set을 적용하거나 Rule-Set 적용 결과 또는 하부노드의 정책 상태 등을 관리자에게 통보하며 관련 정보를 DB 기록하는 역할을 한다. 인터프리터 에이전트는 다양한 하부 노드의 특성에 따라 각자의 설정 정보를 포함한다. 하부 노드에 접근하기 위해 직렬 및 병렬 통신, 텔넷, SSH, HTTP 등의 다양한 프로토콜을 활용한다.

4.4. KeyNote 신뢰 관리 시스템

PolicyMaker[15]부터 진행되어온 신뢰 관리 시스템에 관한 연구는 인증서와 관계 기간의 보안 정책 기술 기법과 인터프린팅 기법으로 발전되어 통합 연구되어왔다. KeyNote는 이러한 발전에서 행위기반의 인증을 가능하도록 고안된 신뢰 관리 시스템 기술 언어이다.

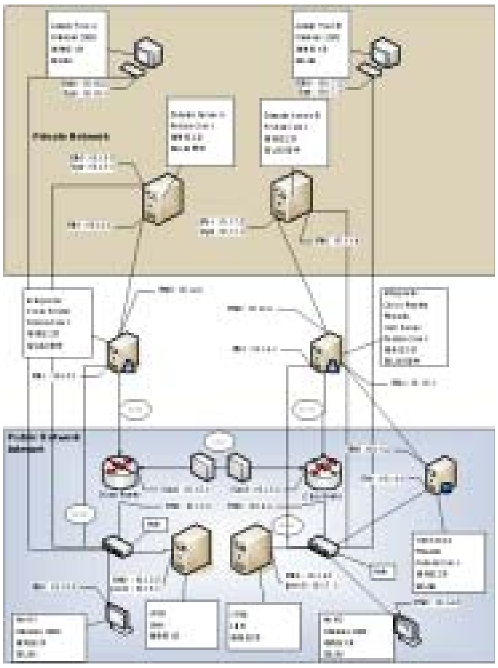
본 시스템의 도메인 관리 서버간의 인증을 위해 KeyNote 기술 방법을 적용한다. KeyNote는 표준화되고 정량적인 목적을 수행하는 메커니즘으로 특정 어플리케이션이나 도메인을 보안 정책과 인증서로 기술한다. 신뢰 관리 인증서는 공개키 인증서를 이용하여 특정 대상에 대한 신뢰 관계를 위임하는 형식으로 기술된다. 기존의 인증서와는 달리 키와 이름, 인증서가 하나로 연결되어 있으며 실제 행위를 수행할 특정 업무에 대한 권한자의 직접적인 키가 기술되어있는 형식으로 되어있다.

5. 구현 및 실험 결과

5.1. 실험 환경

그림 11은 고급 언어 기반의 대규모 네트워크 보안 관리 시스템의 테스트 환경이다. 인터넷과 같은 공중망으로 구성된 3COM 네트워크와 CISCO 네트워크를 관리할 수 있는 관리 시스템을 가상 사설망으로 구성했다. 전체 투입된 시스템 구성요소는 도메인 관리 서버 2대, 인터프리터 에이전트 2대, 관리자 인터페이스 2대, 일반 단말 윈도우 클라이언트 2대, 아파치 웹서버 1대와 라우터(CISCO, 3COM) 2대이다.

실험 망에는 POSEIDON 시스템이 배치되어 있으며 라우터와 웹서버와 같은 하부 노드 제어를 수행하고 있다. 각 라우터 하부의 네트워크는 VPN 게이트웨이를 통해 가상사설망으로 구성되어 도메인 관리 서버와 관리자 인터페이스 간의 안전하고 독립적인 네트워크 통로를 확보하게 된다.



(그림11) 테스트 시스템 구성도

윈도우 기반의 관리자 인터페이스는 윈도우 2000 운영체제에서 MFC 라이브러리를 사용하여 개발되었으며 포함된 컴파일러는 MKS Lex & Yacc 기반의 Visual C++ 컴파일러로 동적 링크 라이브러리로 개발되었다.

리눅스 기반의 도메인 관리 서버와 인터프리터 에이전트는 Fedora 리눅스 운영체제에서 g++ 3.2 에서 개발되었으며 통신 API의 효율성을 위해 ACE 라이브러리를 사용했으며 데이터베이스는 MySQL 4.0 을 사용했다.

5.2. 실험 결과

실험을 통하여 고급 언어에 기반을 둔 본 시스템이 타 정책 기술 언어와 기존의 네트워크 보안 제어를 위한 방법들을 비교하였다. 결과는 전체 시스템을 비교하기 위한 고급 언어 기능 요소 비교와 하부 노드 제어 과정에 소요되는 명령어 개수를 비교로 나타난다.

5.2.1. 고급 언어 기능 비교

표 5-1은 본 논문에서 제시하는 고급 언어와 유사한 목적으로 개발된 보안 정책 기술 언어의 기능비교를 나타낸다[16]. 본 논문에서 제시하는 고급 언어를 Triton 이라고 한다. Triton은 기존의 보안 정책 기술 언어와 비교하여 아래의 장점을 가진다.

첫째, 정책 기술 문법이 기존의 프로그램 언어와 유사하고 다양한 하부 노드 보안 설정을 추상화하여 정책 작성자가 작성한 하나의 정책으로 여러 개의 중간 형태의 세부 정책을 생성할 수 있는 장점이 있다[17]. 둘째, 의미적으로 상반된 정책이 적용되는 상황이 발생하였을 경우 기존의 정책 언어는 상반되는 정책의 하나를 제거하는 방법을 선택하였다. 본 시스템의 고급 언어는 정책이 의미적으로 상반되는 경우에 효과적으로 조율하기 위해 사전협상 메커니즘, 우선순위, 필수성 및 기적용성을 활용하고 정책을 순응하여 정책의 의미를 최대한 보존할 수 있는 장점이 있다. 셋째, 기존의 정책 기술언어가 호스트 기반의 그룹을 기술할 수 있다. 호스트 기반의 그룹에서 포트와 노드 타입을 함께 그룹으로 지정할 수 있는 장점이 있다.

(표 1) 다른 정책 기술 언어와의 비교

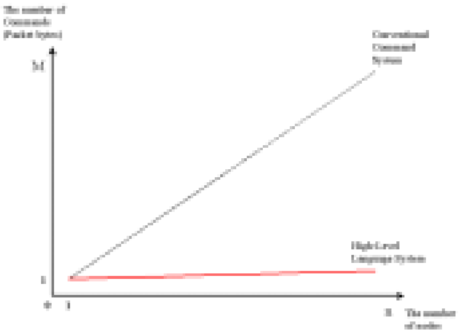
정책언어 비교항목	Policy Term	PFDL	RPSL	SRL	NetSPoC	Triton (고급 언어)
대상 네트워크 환경	분산	중앙 집중	중앙 집중	분산	중앙집중	분산
순응검사	X	O	X	X	X	O
그룹	X	X	O	X	O	O
이벤트	X	O	X	X	X	O
하부노드 상세표현	X	X	X	X	O	X

본 논문에서 제시한 고급언어의 단점으로는 기존의 정책 언어에서 지원하고 있는 각 하부 노드에 대한 상세 설정을 못하는 것이다. 그렇지만 고급언어의 조건문법에 하부 노드 종류를 추가함으로써 하부 노드를 기술할 수 있는 표현력을 향상시켰다.

5.2.2. 하부 노드 제어를 위한 정책 작성

실험망을 기준으로 다수의 하부 노드를 관리하기 위해 필요한 명령어의 크기와 고급 언어로 작성한 명령의 수를 비교하였다. 기존 관리 체계에서는 직접 하부 노드 장비로 접속하여 설정 환경으로 진입하고 접근 제어 목록을 수정하고 반영하기까지 반복적인 입력이 필요하다. 고급 언어로 작성된 정책은 제안한 시스템에 의해 다양한 하부 노드의 설정으로 변환되어 반영된다. 그림12는 이 과정에서 관리자에 의해 작성되는

입력 명령 혹은 정책 수의 변화를 나타낸 그림이다.



(그림 12) 기존 관리 체계와 고급 언어를 이용한 관리의 비교

기존 관리 체계가 하부 노드의 수가 증가함에 따라 더 많은 양의 명령을 작성해야 하는 반면 동일한 기능을 수행하기 위한 고급 언어로 작성한 명령은 최소한의 크기를 유지한다. 관리를 위하여 필요한 명령어의 수가 노드의 수에 비례하여 증가하는 반면에 고급 언어로 기술할 경우 1기 노드를 기술하기 위한 명령에서 적용 대상의 네트워크 도메인이 증가할 때 도메인명만 추가된다. 도메인명은 CAMF의 도메인명 필드 크기에 따라 최대 24 바이트까지 증가된다.

실험망에서 적용된 하부노드 수가 증가함에 따라 지속적으로 기존 관리 체계의 명령어 크기가 증가하는 것을 알 수 있다. 고급언어로 기술할 경우 적용 대상 도메인 확장에 따른 간헐적으로 최소한의 명령어 확장을 요구함으로 전체 네트워크 하부 노드의 보안 설정을 일괄적으로 관리하는 일반적인 상황에서 기존 관리 체계의 명령 방식보다 더욱 신속하고 간단하다.

이러한 정책 설정의 간소함은 기존 관리 체계의 네트워크 관리자가 가지는 부담을 최소화 할 수 있는 장점이 있다. 그러나 인터프리터 에이전트 개발과 세밀한 보안 설정이 어렵다는 단점이 단점으로 지적된다.

6. 결론 및 향후 연구 과제

6.1. 결론

본 논문에서는 대규모 네트워크에서 고급 언어 기반의 보안 관리를 위해 보안 정책을 기술하기 위한 고급 언어 Triton과 이를 포함하고 있는 POSEIDON 시스템을 제시하였다.

POSEIDON 시스템은 인터넷과 같은 다양한 네트워크 도메인으로 구성된 대규모 네트워크에서 이질적인 네트워크 구성 노드들의 보안 설정을 일괄 기술할 수 있는 고급 언어와 기술된 정책의 충돌을 탐지하고 정책 순응을 수행할 수 있는 도메인 관리 서버, 하부 노드에 직접 접근하여 저급의 명령어를 입력하는 인터프리터 에이전트로 구성되어있다. 추상화된 고급언어로 기술한 네트워크 보안 정책은 일괄적인 하부 노드 제어를 지원한다. 일괄적인 하부 노드 제어를 통해서 네트워크를 통해서 유입되는 인터넷 웜과 같은 유해 트래픽을 일괄적으로 빠르게 제어할 수 있다. 하부 노드 인터프리터 에이전트는 다양한 통신 방법을 제공하여 기존의 하부 노드 보안 설정 기법을 그대로 활용할 수 있다.

기존의 대규모 네트워크를 대상으로 하는 보안 관리 시스템이 특정 하부 노드의 지원에만 국한되거나 특정 네트워크 도메인만을 대상으로 적용되었던 것에 비해 POSEIDON 시스템은 네트워크 도메인을 계층적으로 구분하여 각 도메인의 제어권을 최대한 보장함과 동시에 최상위 도메인으로부터 전달되는 정책을 적합한 협상과 순응 과정을 통해 정책 작성자의 의도를 최대한 반영한다. 그리고 각 도메인에 대한 스스로의 제어권을 보장할 수 있는 사전협상 메커니즘을 통해 다수의 네트워크 사이에서 중앙 집중 관리와 분산 관리의 장점과 단점을 적절히 조화하여 인터넷과 같은 대규모 네트워크 관리에서 활용될 수 있다.

6.2. 향후 연구 과제

향후 연구 과제로는 POSEIDON 시스템을 실제 대규모 네트워크에 적용하는 작업과 다양한 인터프리터 에이전트 개발의 효율성을 위한 인터프리터 프레임워크 개발이 필요하다. 현재 구현된 인터프리터 에이전트는 Cisco 1721 라우터, 3Com 5009 라우터, IPTables 방화벽, Orinoco AP-2000 용이 있고, 각각의 인터프리터 간에는 Rule-set를 변환하는 엔진의 일부분을 제외하고 관리 노드간의 통신, 네트워크 노드와의 통신 등 많은 모듈이 동일하게 구성되어 있다. 따라서 새로운 네트워크 노드에 대한 인터프리터 에이전트를 개발이 용이하다. 그러나 시스템 구성의 견고함과 코드의 재사용성을 높이며 인터프리터 에이전트 개발을 보다 자동화하기 위해서 반드시 일관성 있는 프레임워크가 요구된다.

그리고 KeyNote 신뢰 관리 시스템에서의 공용 인증서와의 호환성 문제, 하부 노드 정보의 동기화 문제, 고급 언어의 문법적 확장, 고급 정책과 CAMF간의 의존성에 대한 연구가 필요하다. 이러한 연구를 통하여 다양한 보안 장비들의 이질적인 설정 방법을 통합하고 일괄적으로 관리하게 된다면 대규모 네트워크에서 발생하는 보안 위협 상황에서 빠르고 효과적으로 대처하게 될 것이다.

참고문헌

[1] M. Lad, Z. Xiaoliang, Z. Beichuan, D. Massey and Z. Lixia, An Analysis of BGP Update Burst during Slammer Attack, *IWDC*, Calcutta, 2003

[2] N. Yialelis, Domain-based Security for Distributed Operating Systems, *Ph.D. Thesis, Dept of Computing Imperial College, London*, Aug, 1996

[3] G. G. Xie et al., SAAM : An Integrated Network Architecture for Integrated Services, *Proc. 6th IEEE/IFIP International Workshop. QoS*, Napa, Canada, 1998

[4] D. Marriott and M. Sloman, Implementation of a Management Agent for Interpreing Obligation Policy, *Proceeding IEEE/IFIP Distributed Systems Operations and Management Workshop (DSOM'96)*, L'Aquila, Italy, Oct. 1996

[5] R.S. Sandhu et al., Role-based Access Control Models, *Computer*, Vol. 29, No. 2, pp 38-47, 1996

[6] M. Sloman, Policy driven management for distributed systems, *Journal of Network and Systems Management*, Vol. 2, pp333, 1994

[7] T. Koch et al., Policy Definition Language for Automated Management of Distributed System, *Proceeding Second IEEE International Workshop Systems Management*, pp. 55-64, Toronto, Canada, June 1996

[8] A. Kern, M. Kuhlmann, A. Schaad, and J. Moffett, Observations on the Role Life-Cycle in the Context of Enterprise Security Management, *Proceedingof the 7th ACM Symposium on Access Control Models and Technologies 2002*, pp43-51, 2002.

[9] *Cisco Security Policy Management*, http://www.cisco.com/en/US/products/sw/cscowork/ps2330/, 2004

[10] J. Lobo, R. Bhatia, A Policy Description Language, *AAAI*, 1999.8

[11] W. Stephens, S. Narain, Service Grammars for Diagnosing Configuration Errors in Dynamic Coalition Networks, *3rd Information Operations Symposium*, San Diego, 2001.

[12] A. Keromytis, Sotiris Ioannidis, Michael B. Greenwald, and Jonathan M. Smith, The STRONGMAN Architecture, *In the 3rd*

DARPA Information Survivability Conference and Exposition, Washington D.C., 2003

[13] *NetSPoc*, http://netspoc.berlios.de/, 2004

[14] M. Blaze, J. Feigenbaum, J.Ioannidis, and Angelos D. Keromytis, The KeyNote Trust-Management System Version 2, *Request For Comments 2704*, 199

[15] M. Blaze, J. Feigenbaum, M. Strauss, Compliance-Checking in the PolicyMaker Trust-Management System, *Proceeding of2nd Financial Crypto Conference LNCS #1465*, pp251-265, 1998

[16] G. Stone, B.Lundy, and G. Xie, U.S Department of Defense, Network Policy Languages: A Survey and a New Approach, *IEEE Network*, 2001

[17] J. Kim, B. Song, K. Lee, S. Kim and E. Park, Design a High-Level Language for Large Network Security Management, *Cooperative Information Systems 2004*, LNCS Proceeding Vol 1, pp716-719, Oct 25-29, 2004

김장하

1999년~2003년 경북대학교 컴퓨터과학과(학사)
2003년~현재 경북대학교 컴퓨터과학과 석사 과정

관심분야: 프로그래밍 언어, 보안 정책, 사이버포렌식

이강희

1999년~2004년 경북대학교 컴퓨터과학과(학사)
2004년~현재 경북대학교 컴퓨터과학과 석사 과정

관심분야: 보안 정책 언어, 네트워크 및 시스템 보안

김상욱

1975년~1979년 경북대학교 컴퓨터과학과(학사)
1979년~1981년 서울대학교 컴퓨터과학과(석사)
1981년~1989년 서울대학교 컴퓨터과학과(박사)
1988년~현재 경북대학교 컴퓨터과학과 교수

관심분야: 멀티미디어 시스템, 멀티미디어 컨텐츠 보안 및 인간과 컴퓨터의 상호 작용