主題

분산 침입 탐지 시스템을 위한 통신 모델

대구가톨릭대학교 공과대학 컴퓨터정보통신공학부 장정숙, 전용희
한국전자통신연구원 네트워크보안연구부 장종수, 손승원

차 례

Ⅰ. 서론

최근에 네트워크를 통한 분산 공격의 증가로 인하여 분산 침입 탐지 기술에 대하여 관심이 고조되고 있다. 이에 따라 침입 탐지 시스템은 집중 및 단일 프레임워크에서 분산 시스템으로 이동하는 추세이다. 분산 침입 탐지 시스템은 데이터의 분석이 감시되는 호스트 수에 비례하여 다수의 위치에서 수행되는 시스템으로 정의될 수 있다[1]. 이 정의에 의하면, 단순히 분산 데이터 수집을 하는 것으로는 분산 침입 탐지라 말할 수 없으며, 분석(analysis) 컴포넌트가 감시되는 호스트 수에 비례하고, 위치가 분산되어야 한다.

세계적으로 분산 침입 탐지 시스템에 대한 연구가 많이 진행되고 있지만, 국내에서는 아직 분산 침입 탐지 시스템 컴포넌트 사이의 통신 메커니즘과 일반적인 통신 모델에 대하여 발표된 연구 결과는 거의 없는 실정이다. 따라서 본 논문에서 기존 침입 탐지 시스템에서의 통신 메커니즘을 분석하고, 분산 침입 탐지 시스템을 위한 통신 메커니즘에 대하여 고찰하고 비교 분석하고자 한다.

분산 침입 탐지 시스템은 컴퓨팅 노드들 사이의 통신으로 외부와 내부의 불법적인 침입을 탐지하고 대응하는 보안 메커니즘이다. 분산 침입 탐지 시스템의 다른 컴포넌트 사이의 통신은 시스템 기능성의 한 중요한 부분이다. 컴포넌트들은 통신 메시지를 통하여 시스템의 전반적인 상태를 얻을 수 있기 때문에, 통신의 붕괴는 시스템으로 하여금 오동작을 유발하거나 실패하게 만들 수 있다.

분산 침입 탐지 시스템 구성 컴포넌트 사이 탐지 정보 분배에 대한 통신 량의 최소화는 분산 소프트웨어 설계의 성능 목표 중의 하나이다. 즉 컴포넌트 사이 필요한 최소 정보의 통신으로 보다 높은 확장이 가능하여진다. 따라서 침입 탐지를 위하여 구성된 컴포넌트 사이 효율적인 정보 분배는 분산 침입 탐지 시스템에서의 중요한 문제가 된다.

본 논문에서는 침입 탐지 시스템의 컴포넌트 사이 효율적인 침입 탐지 정보의 분배와 컴포넌트 사이의

침입 탐지 정보에 대하여 상호 작용하는 분산 통신 모델에 대하여 분석한다. 이를 위하여 먼저 기존의 침입 탐지 시스템인 DIDS[2], AAFID[3], EMERALD[4], 그리고 GrIDS[5]의 통신모델에 대하여 분석한다. 그리고 폭발적인 통신의 증가로 인한 기존의 클라이언트-서버 모델과는 대비되는 피어 투 피어 통신 모델에서 이벤트 기반과 푸시(push) 기반 분산 통신 모델에 대하여 분석한다. 이런 분석을 통하여 차세대 침입 탐지 시스템을 위한 효과적인 통신 모델을 도출하고자 한다.

본 논문은 다음과 같은 순서로 구성된다. Ⅱ절에서는 분산 침입 탐지 통신 모델, Ⅲ절에서는 기존 침입 탐지 시스템의 통신 메커니즘에 대하여 살펴보고, Ⅳ절에서는 분산 통신 모델을 분류하고 각각의 특징에 대하여 기술하고, Ⅴ절에서는 모델들을 비교 분석하며, 마지막으로 Ⅵ절에서 요약과 결론으로 끝을 맺는다.

Ⅱ. 분산 침입 탐지 통신 모델

침입 탐지 시스템(IDS: Intrusion Detection System)은 사용자에 대한 데이터 조직과 특성을 묘사하고 관심 있는 활동을 식별하는 시스템 활동으로서, 외부와 내부의 불법적인 침입을 탐지하고 대응하는 보안 메커니즘이다[6]. 분산 침입 탐지 시스템에서는 분산되어 있는 컴퓨팅 노드에 침입 탐지 에이전트를 설치하고 호스트나 네트워크에서 실시간으로 외부 또는 내부의 침입을 탐지한 후에 관리자 컴포넌트에게 보고와 대응에 관한 탐지 정보를 효과적으로 분배하는 통신 메커니즘이 중요하다. 기존 분산 침입 탐지 시스템인 DIDS, AAFID, EMERALD, 그리고 GrIDS는 TCP/IP 통신 기반으로 메시지를 이용하여 에이전트들과 관리자 컴포넌트 사이 침입 탐지 정보를 분배한다. 분산 침입 탐지 시스템의 통신 메커니즘을 결정하는 요인은 다음과 같다[1]:

- 컴포넌트의 수
- 컴포넌트의 위치
- 고려되는 데이터의 형태
- 데이터 양
- 데이터 생성 빈도
- 데이터 표현 방법
- 데이터의 민감성

침입 탐지 시스템을 위한 통신 기법에서 바람직한 특징은 다음과 같다: 신뢰성, 보안, 인증, 무결성, 기밀성, 부인 봉쇄, 복제 봉쇄, 서비스 거부 공격에 대한 저항, 확장성, 속도.

많은 침입 탐지 시스템들이 분산 시스템이기 때문에, 분산 통신 모델을 분석할 필요가 있다. 분산 통신 모델은 포함된 개체들 사이의 통신 형태를 나타낸다. 참가 개체의 역할, 개체의 정보 요구, 정보의 이용가능성, 정보의 분배 등을 결정한다. 통신 모델은 정보를 전달하기 위하여 실제로 사용되는 하부 통신 메커니즘과는 독립적이다. 분산 통신 모델로 기본적으로 클라이언트-서버 모델과 피어-투-피어 모델의 두 가지가 있다. 클라이언트-서버 모델에서는 두 참가 개체만 취급하지만, 피어-투-피어 모델은 많은 피어 개체들을 다룬다. 분산 침입 탐지 시스템은 시스템 상태 정보를 생산하고 소비하는 다수의 개체들로 구성되기 때문에, 피어-투-피어 모델이 더욱 적절한 것으로 판단된다.

이와 같이, 좀 더 효율적인 침입 탐지에 대한 탐지 정보 분배를 위해서 피어 투 피어 통신 모델을 사용할 수 있다. 피어-투-피어 통신 모델은 이벤트 기반과 푸시 기반 통신 모델로 다시 분류할 수 있다. 침입을 탐지하는 컴포넌트들이 피어 투 피어 통신 모델을 기반으로 탐지 정보를 분배하여 차세대 침입 탐지 보안 시스템에 이용 될 수 있을 것이다. 이들에 대하여 보다 자세하게 Ⅳ절에서 다룬다.

Ⅲ. 기존 침입 탐지 시스템의 통신 메커니즘

3.1 DIDS

U.C. at Davis에서 개발한 분산 침입 탐지 시스템(Distributed Intrusion Detection System)은 이기종 망을 감시하기 위하여 중앙 데이터 분석과 함께 분산 모니터링과 데이터 감축을 결합하고 있다[2].

3.1.1 DIDS 구조

DIDS는 호스트 이벤트 데이터를 수집하는 호스트 에이전트 모듈과 분산 데이터 모니터링을 수행하는 LAN 모니터 에이전트 모듈 그리고 이들 이벤트를 분석하는 중앙 관리자 모듈 컴포넌트들로 구성된다. 이들 세 가지 주요 컴포넌트의 기능은 다음과 같다:

- 호스트 에이전트 모듈: 호스트 보안에 관련된 이벤트 데이터를 수집 한 후 중앙 관리자 모듈에게 보고한다.
- LAN 모니터 모듈: LAN 트래픽 분석 후에 결과를 중앙 관리자 모듈에게 보고한다.
- 중앙 관리자 모듈: LAN 모니터와 호스트 에이전트로부터의 보고를 수신하고 보고 결과로서 이벤트를 분석하여 시스템의 보안상태를 결정한다.

DIDS 구조에서 중앙 관리자 모듈은 분산 침입 탐지에서 명백한 병목현상을 야기한다. DIDS에서의 모든 호스트는 하나의 계층 단계이고 LAN 모니터 모듈은 단일 중앙 관리자 모듈에게 보고하는 구조로서 확장성이 결여된다. 그림 1은 DIDS의 구조를 보여준다.

호스트 에이전트 모듈 혹은 LAN 모니터 모듈에서 의심스런 행위가 탐지되었을 때, alert을 중앙 관리자에게로 보내면, 관리자 모듈은 침입을 추론하여 처리한다.

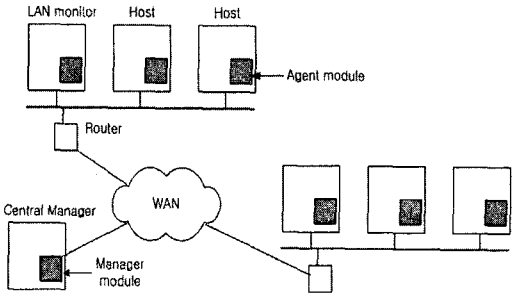


그림 1. DIDS 구조

3.1.2 DIDS 통신 메커니즘

호스트 에이전트 모듈과 LAN 모니터 모듈에서 중앙 관리자 모듈까지 탐지 정보를 보고하는 통신 메커니즘은 통신 하부구조와는 독립적이고 비동기적인 통신으로 이루어진다. 컴포넌트 사이 통신 프로토콜은 ISO CMIP(Common Management Information Protocol)를 기반으로 한다[7]. CMIP는 그림 2처럼 OSI 통신 모델에 근거한 네트워크 관리 프로토콜이다. 이와 관련된 CMIP는 네트워크 객체나 장치들에 대한 액세스 정보와 그들을 제어하고 또 상태 보고를 수신하기 위한 서비스를 정의한다.

CMIP는 관리정보를 전송하는 절차 즉 CMISE (Common Management Information Service Element) 사이에서 CMIS(Common Management Information Service) 서비스를 완성시키기 위해서 교환하는 CMIP PDU (Protocol Data Unit)를 만들고 전송하는 것에 대해 정의해 놓은 것이다. 다시 말하면 CMIP는 관리자와 에이전트사이 메시지 교환을 위한 통신 프로토콜이다. CMIP는 TCP/IP 환경의 인터넷 망 관리를 위해서 사용하는 SNMP에 비해 세심한 작업을 정의하고 있다. 그리하여 각종 통신환경에서 효율적이고 투명한 파일 접근 능력을 제공한다.

• CMISE(Common Management Informa-

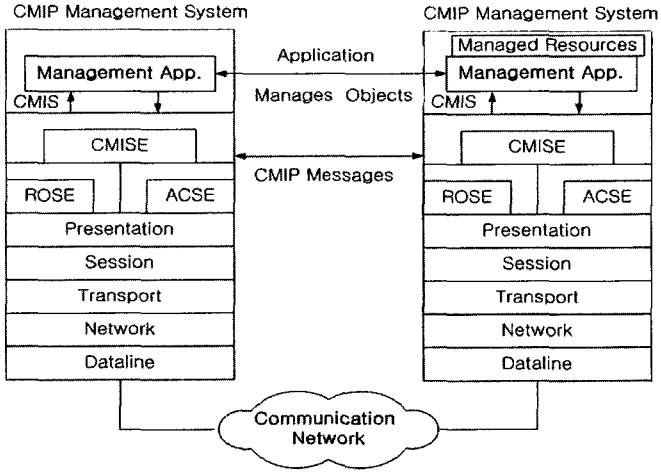


그림 2. DIDS 통신 메커니즘

tion Service Element)

CMISE는 사용자 응용과 인터페이스 서비스를 담당하는 CMIS와 PDU 및 관련 절차를 규정한 CMIP로 이루어진다. CMIP는 CMIP 서비스를 처리하기 위해서 11개의 PDU를 정의해 놓았다. 양단의 CMISE 사용자(관리자와 에이전트 혹은 CMISE 응용)들이 정보교환을 위해서 시스템을 연결 혹은 종료하는데 ACSE(Association Control Service Element)를 이용하며, CMIP PDU 전송을 위해서 ROSE(Remote Operation Service Element)를 이용한다.

3.2 AAFID

Purdue 대학에서 수행한 DARPA 프로젝트 "Enhanced Intrusion and Misuse Detection Technology"에서 AAFID에 대하여 연구하였다. AAFID는 Autonomous Agents for Intrusion Detection의 약자로 간단하고 경량의 여러 프로세스가 서로 협력하여 침입탐지를 수행한다. 이것은 분산 IDS로써 에이전트 기반 해결책을 제공한다.

3.2.1 구조

AAFID는 침입 탐지를 지향하는 분산 호스트 모니터를 위한 프레임워크로서, 개체의 계층적인 구조를 사용한다. 가장 낮은 계층에서 AAFID 에이전트는 호스트를 모니터하고 데이터 감축을 수행하는 높은 계층의 단계로 침입 발견을 보고한다. AAFID는 데이터 수집 및 분석을 위한 가장 하위 요소로서, 에이전트를 사용하여 확장성을 고려한 계층적 구조이다. AAFID는 에이전트와 트랜시버, 모니터의 세 가지 구성 요소와 사용자 인터페이스로 이루어진다(그림 3참조).

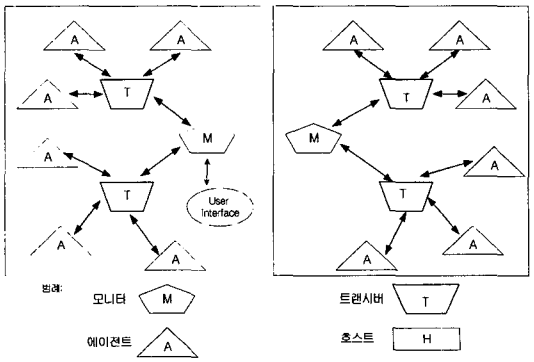


그림 3. AAFID의 구조

에이전트들은 호스트 상에 거주하며, 호스트의 어떤 측면을 감시하는 독립적 실행 개체이다. 그리고 에이전트는 비정상적인 혹은 관심 있는 행위의 발견을 단일 트랜시버로 보고한다. 트랜시버는 호스트 상에 모든 에이전트의 동작을 감시한다. 트랜시버는 또한 에이전트에 의해 보고된 정보의 축약을 수행하고 다음 단계 계층인 모니터에게 결과를 보고한다. 모니터는 AAFID에서 가장 최상의 계층이며 AAFID 개체들을 제어하고 모든 트랜시버로부터 축약된 정보를 수신 받아 데이터를 처리하는 역할을 수행한다. 또한 전체 구조에서는 확장과 구성변경이 용이한 CIDF[8], 네트워크에 대한 관리정보의 교환이 용이하도록 설계된 SNMP 프로토콜, 그리고 감사 자료들을 효율적으로 분산하는 메커니즘으로 Audit router의 적용을 고려하고 있다[3].

3.2.2 통신 메커니즘

AAFID는 계층적인 통신 모델로서, 개체들 사이 정보 교환을 위하여 표준 메시지 형태를 정의하고 있으며 메시지 전송이 AAFID 시스템 기능의 핵심 부분이다[1]. AAFID 모든 개체들은 TCP 연결에서 그들의 제어 개체들과 함께 메시지를 교환하며 제어 개체들로부터 수신되는 메시지들은 가능한 빨리 처리되어야 한다. 그림 4처럼 계층의 정상에 단일 루트 노드가 있으며 개체는 아래 계층과 위 계층으로 즉각 통신이 가능하다. 단 에이전트와 필터 사이 통신은 그들 스스로 메시지 전송이 가능하다. AAFID 통신 메커니즘은 AAFID 통신에서의 오버헤드 감소를 위한 효율적인 통신 메커니즘이며 다른 침입 탐지 시스템보다 공격에 대한 저항이 강하고 몇 가지 인증과 기밀성을 제공한다.

AAFID의 호스트 내 통신은 intra 통신과 inter 통신으로 분류한다. intra 통신에서 트랜시버가 여러 에이전트에게 메시지를 보내는 것은 일 대 다 통신이며 에이전트가 트랜시버에게 메시지를 보내는 것은 다 대 일 통신이다. 또한 intra 통신에서는 메시

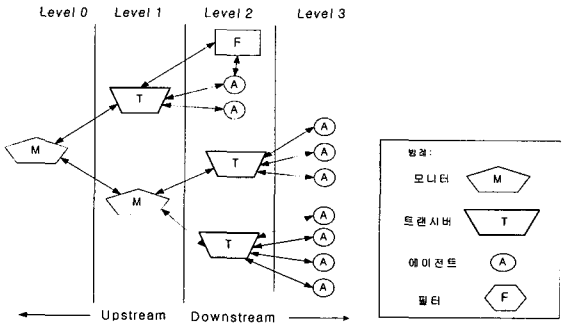


그림 4. AAFID 통신 모델

지 큐(message queues)와 공유메모리(shared memory) 그리고 파이프(pipes) 스킴들이 고려되고 있으며, inter 통신에서는 IDS에서의 성능과 신뢰성 그리고 보안에 대한 사항을 주로 고려한다. 표 1은 AAFID에서 정의하고 있는 표준 메시지의 종류를 보여준다.

표 1. AAFID 표준 메시지

NOTYPE	STOP
CONNECT	EVAL
DISCONNECT	SET_PARAMS
STATUS_UPDATE	GET_PARAMS
COMMAND	DUMP_YOURSELF

다음은 AAFID 에이전트가 서로간의 정보교환을 위한 메시지 패싱으로 에이전트의 일반적인 행위인 STOP이란 메시지를 통신하는 간단한 예이다.

```
Do initialization
loop
        process input
        if STOP message was received then
        Cleanup and exit
        endif
        Perform checks
        if abnormal condition then
        generate STATUS_UPDATE message with
        status information
        endif
        Sleep for a certain amount if time(inter-
        check period)
end loop
```

3.3 EMERALD

3.3.1 EMERALD 구조

SRI International은 DARPA 프로젝트 "Analysis and Response for Intrusion Detection in Large Networks"에서 NIDES를 업그레이드한 EMERALD(Event Monitoring Enabling Responses to Anomalous Live Disturbances)에 대하여 연구하였다. EMERALD는 분산 침입 탐지를 수행하는 프레임워크로서, 대규모 네트워크를 통하여 악성 행위를 추적하기 위한 분산 확장 도구이다. 일반적인 EMERALD 모니터 구조는 AAFID 구조와 유사하며 그림 5처럼 EMERALD monitor, EMERALD engine 그리고 EMERALD resolver로서 구성된다.

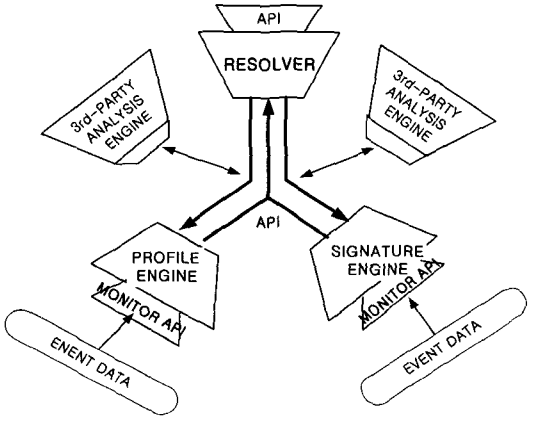


그림 5. EMERALD 구조

EMERALD 모니터는 특정 목표 이벤트 스트림을 선택한다. 이벤트 스트림은 감사 데이터, 네트워크 데이터그램, SNMP 트래픽, 응용 로그, 그리고 다른 침입 탐지 장치로부터의 분석 결과를 포함하는 다양한 근원지로부터 유도된다. 이벤트 스트림은 자원 객체 정의에서 제공되는 특정한 대상 이벤트 수집 방법에 의해 분석되고 여과되며 형식화된다. 이벤트 기록은 처리를 위해 모니터의 분석 엔진으로 전달된다. 그리고 자원 객체에 따라 독립적인 코드 분석을 수행하고 resolver에서 전체 분석을 수행하는 계층적인 구조이다.

EMERALD의 중요 개념은 이종환경의 특징 기반과 통계 기반 분석을 제공한다는 것이다. 또한, 호환성을 지원하는 스크립트 기반 메시지 인터페이스를 제공한다. 이것은 시스템과 네트워크 오용을 분석하고 해석하는 능력을 갖게 되었으며, 상호운용성(interoperability), 재사용성, 빠른 통합, CIDF로 협력하는 방법을 제공한다. 그리고, 다른 응용 도메인으로 크게 확장할 수 있는 분석 기술(즉, 생존성, 결함 허용, 신뢰성)을 지원하게 되었으며 무한한 확장성을 제공한다.

EMERALD는 네트워크에 독립적으로 분산되어 있는 모니터를 사용하여 대규모 네트워크에서의 비정상 활동을 감지한다. EMERALD는 분산 환경에서 실용적인 시스템 구현을 위해 기존의 침입 탐지 시스템에서보다 작고 분산적이며 상호 협력적인 구성요소를 채택하고 계층적인 구조를 이루어 광범위한 네트워크를 관찰한다. 나아가 EMERALD는 이질적인 대상 호스트와 통합하여 융통성 있는 응용 프로그래머 인터페이스를 도입하고 제 3자 분석 도구와 상호동작을 제공한다.

3.3.2 EMERALD 통신 메커니즘

EMERALD 통신은 이벤트 스트림을 대상으로 하는 메시지 시스템을 이용한다. 컴포넌트 상호동작은 클라이언트/서버 기반에서 이루어진다. EMERALD 모니터의 구독(subscription) 기반 상호 작용은 푸시(push)와 풀(pull) 데이터 교환 능력에서 EMERALD 메시지 시스템을 제공한다. EMERALD 클라이언트 모듈들은 서버에 의해 생산된 이벤트 스트림 분석 결과를 구독하여 수신한다. 구독 요청이 서버에 의하여 수용되면, 서버 모듈은 데이터가 이용 가능할 때 자동적으로 클라이언트 구독자

에게 구독 기반 메시징 패싱에 의해 비동기적으로 이들 이벤트 혹은 분석 결과를 전달하며, 클라이언트의 제어 요청에 따라 동적으로 재구성될 수 있다.

다음에 EMERALD의 인터페이스 명세와 전송 계층 설계를 간단하게 요약한다[4].

• **Interface Specification(인터페이스 명세)**

인터페이스 명세는 메시지의 정의와 메시지가 처리되는 방법을 나타낸다. EMERALD 모니터는 대상 이벤트 스트림을 수집하고 여과하는 이벤트 수집 방법, 여과된 이벤트를 처리하는 분석 엔진 그리고 분석 엔진 결과를 처리하고 대응하는 해결자(resolver)의 세 가지 일반적인 모듈 형태를 포함한다.

내부모니터와 모니터간 통신은 구독 기반 클라이언트 서버 모델을 채택하고 있다. EMERALD의 내부모니터와 모니터간 프로그래밍 인터페이스는 같다. 이들 인터페이스는 5가지 범주의 상호동작으로 나눈다: 채널 초기화 및 종료, 채널 동기화, 동적 구성, 서버 조사 그리고 보고/이벤트 분배.

• **Transport Layer(전송 계층)**

메시지 시스템의 프레임워크는 모니터 혹은 모니터와 제 3 자 보안 모듈 사이 통신 채널을 설립하는 전송 메커니즘을 명세 한다. EMERALD 전송 계층에서 통신의 보안, 무결성과 신뢰성의 문제를 해결한다. EMERALD 메시지 시스템 설계에서 필수적인 요소는 EMERALD 컴포넌트와 다른 협력 분석 장치 사이에서 어느 정도의 내부 보안을 제공하고 공개키/개인키 인증 프로토콜과 세션 키 교환을 사용하여 외부적으로도 안전한 분석결과를 교환하는 것이다.

3.4 GrIDS

3.4.1 GrIDS의 구조

UCDavis에서 연구한 GrIDS(Graph Based Intrusion Detection System for Large Networks)는 호스트들의 행위와 호스트들 사이의 트래픽에 대한 정보를 수집하며, 이러한 정보를 행위 그래프(Activity Graphs)로 모은다. 행위 그래프에서 노드는 시스템에서 호스트와 일치하며 에지는 호스트간의 네트워크 활동을 나타낸다. 이 시스템의 메커니즘은 대규모 분산된 활동으로부터 행위 그래프를 만드는 것이다. 이는 대규모의 자동화된 또는 협동 공격을 거의 실시간으로 탐지하는 것을 가능하게 한다. GrIDS는 네트워크 관리자들로 하여금 사용자들이 호스트들의 특정 서비스를 사용하는 것에 대한 정책 기술을 허용하며, 이에 따른 행위 그래프의 특성들을 분석함으로써 기술된 정책의 위반을 탐지하거

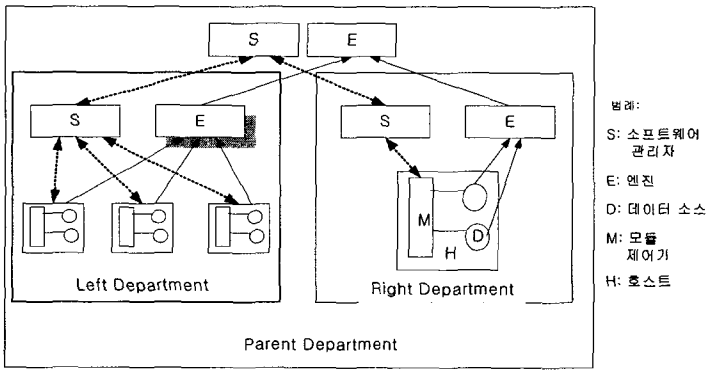


그림 6. GrIDS 구조

나 보고한다. GrIDS는 대규모 네트워크에 적용될 수 있으며, 초기 프로토타입은 웜 공격을 탐지하는데 성공적이었다.

그림 6은 세 개의 부분(department)으로 구성된 단순한 계위를 묘사한다. 모든 GrIDS 소프트웨어는 표준화된 인터페이스를 지닌 모듈들의 형태이며, 이러한 모듈들은 각 호스트에 위치한 모듈 제어 프로세스에 의해 시동되거나 멈추며 제어된다. 각 부분은 소프트웨어 관리자와 그래프 엔진으로 구성된다. 소프트웨어 관리자는 계위의 상태와 분산 모듈들을 관리하며, 계위는 사용자 인터페이스에서 동적으로 재배열된다. 특정 모듈을 시동하고 정지시키는 것도 비슷하게 자동화된다.

GrIDS 데이터 소스들은 호스트와 네트워크상의 행위를 모니터 하는 모듈들이며, 탐지된 행위를 엔진에게 보고한다. 그래프 엔진은 데이터 소스 모듈들로부터의 입력을 받아 그래프를 구축하며, 상위 부분으로 이러한 그래프의 개요(summaries)를 보낸다. 상위 엔진은 차례로 보다 조악한 선명도를 가진 그래프를 구축한다. 이러한 구성 요소 이외에도 사용자와 시스템과의 상호작용, 관리 기능 및 경보 디스플레이를 위한 사용자 인터페이스 모듈이 있다.

3.4.2 GrIDS의 통신 메커니즘

GrIDS의 세 가지 컴포넌트들은 메시지 기반 통신 메커니즘을 사용하여 메시지를 교환한다. GrIDS 컴포넌트들은 TCP/IP 네트워크에서 다음을 포함하여 서로 다른 종류들의 메시지들을 보낸다[9].

- 이벤트의 보고 혹은 이벤트의 부분적인 수집(aggregations)
- 그래프에 대한 질의
- GrIDS 컴포넌트의 동작을 제어하는 Get/Set 메시지

모든 GrIDS 통신이 비록 이산적으로 발생하더라도 때로는 TCP 전송으로, 때로는 단일 UDP 패킷으로 다중의 GrIDS 패킷들을 보낸다. GrIDS 패킷들은 UDP 혹은 TCP 헤더와 내용(body)을 포함한다. 패킷 헤더는 이것을 수신하는 GrIDS 소프트웨어 패킷의 형태를 식별한다. 내용은 실제적인 정보를 담고 있다. GrIDS 컴포넌트 통신에서는 메시지 API와 그래프 언어를 이용한다. 메시지 API에서 공통의 기능 인터페이스 라이브러리는 모든 GrIDS 컴포넌트에서 이용 가능하다. 메시지 API는 송신 메시지를 위해서 API를 문서화하며 init, tcp_send, tcp_recv, tcp_close, shutdown 같은 기능을 포함한다. 그래프 언어는 그래프가 GrIDS 패킷으로 표현되는 방법을 묘사한다. 데이터 소스로부터 보고는 연결(link)의 보고이거나 연결 또는 노드들의 속성의 보고이다. 몇 가지 경우에서 노드들의 연결 대신에 그래프를 보내는 것이 필요하다. 그것은 첫 번째, 사용자가 aggregator에 의해 현재 유지하는 그래프의 가시화를 원할 때, 둘째, 사용자가 aggregator 이동 혹은 재시작을 원할 때, 세 번째, aggregator가 상위 aggregator에게 다중의 보고를 보낼 때이다. 그래프는 결합되어 한가지 보고처럼 보내어진다. 그러므로 요구되는 통신 대역폭이 감소된다. 그래프 표현으로는 DOT[10] 그래프 문법을 수정하여 사용하였다.

Ⅳ. 분산 통신 모델의 분류 및 특징

대표적인 분산 통신 모델로 클라이언트 서버 모델과 피어 투 피어 모델이 있다. 이 절에서는 클라이언트 서버 통신 모델[11]과 피어 투 피어 통신 모델[12]에 대하여 기술한다.

4.1 클라이언트 서버 통신 모델

클라이언트 서버 통신 모델은 인터넷의 출현으로 분산 응용에 대한 플랫폼처럼 사용된다. 클라이언트-서버 모델에서 분산 플랫폼 소프트웨어를 구성하는 구

조적인 접근 방법은 서버들의 작은 수에서 클라이언트의 적절한 수를 추측하여, 클라이언트는 사용자들과 상호작용하고 서버들은 컴퓨터작업 혹은 데이터 서비스 요청에 대한 응답으로 접촉한다. 그림 7처럼 클라이언트 서버 통신 모델은 두 참여자로서 처리된다.

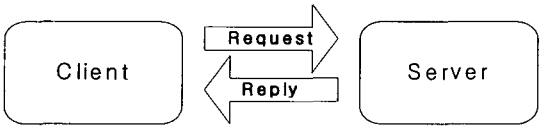


그림 7. 클라이언트 서버 통신 모델

클라이언트 서버 시스템의 통신 모델은 서버와 클라이언트가 세션 동안에 상태 정보를 공유하는 연결 접근의 세션 기반과 상태정보를 공유하지 않는 비 연결 접근의 웹 기반 모델이 있다. 세션 기반 통신은 한 쪽 측 서버로 연결을 유지하여 HTTP 클라이언트 요청을 연속적으로 제공하므로 확장성이 결여된다. 세션 기반의 확장성에 대한 단점을 해결하고자하는 웹 기반 통신 모델은 비 연결 HTTP 프로토콜 기반이다. 서버와 클라이언트 사이 비-영구적 연결 확립으로 서버는 클라이언트에 대해 상태 정보를 유지하지 않는다. 이 스킴은 확장성에 유용하지만 상태 정보를 유지하기가 어렵다. 웹 기반 응용들은 작은 서버 수에서 큰 수의 클라이언트로 확장한다. 응용 설계자들은 두 가지 모델의 상호보완관계에서 선택한다.

4.2 피어 투 피어 통신 모델

분산 침입 탐지 시스템은 시스템 상태 정보를 생산하고 소비하는 다수의 개체들로 구성되기 때문에, 피어-투-피어 모델이 더욱 적절한 것으로 판단된다.

피어 투 피어(Peer-to-peer) 통신은 각 컴퓨터가 동등한 능력을 가지고 있어, 어떤 컴퓨터에서라도 통신 세션을 시작할 수 있는 통신 모델을 지칭한다. 피어 투 피어 동등 계층 통신이라고도 부르는데, 네트워크에 연결되어 있는 모든 컴퓨터들이 서로 대등한 동료의 입장에서 데이터나 주변장치 등을 공유할 수 있다는 의미를 담고 있다. 이 개념과 대비되는 다른 모델로는 클라이언트/서버 모델이나 마스터/슬레이브 모델 등이 있다

피어 투 피어 모델에서, 통신은 소비자가 생산자의 관심사에 등록하는 구독단계로부터 시작한다. 이 시점에서, 피어 투 피어 모델은 두 가지 분류로 나눌 수 있다: 이벤트 기반 모델[13,14]과 푸시 기반 모델[15,16].

4.2.1 이벤트 기반 통신

이벤트 기반 시스템은 이벤트의 개념을 기반으로 분산 응용의 구축을 위한 새로운 스타일을 정의한다. 이벤트는 메시지의 특별한 형태로서 어떤 개체라도 이벤트를 생산하고 그리고 어떤 개체라도 이벤트를 소비할 수 있다. 느슨한 연결에서 컴포넌트사이의 결합이 거의 없으므로 높은 확장이 가능하다.

이벤트 시스템에서 컴포넌트는 이벤트의 생산과 수신으로 상호 작용한다. 컴포넌트들은 특정한 이벤트를 수신하는데 관심을 선언하고 그러한 이벤트 발생에 대하여 통지를 받는다. 이 모델에서는 이벤트 통지 프로토콜(event notification protocol: ENP)을 사용하다. ENP를 사용하여 개체들은 그들이 생산한 이벤트를 광고하면 개체들은 다른 개체로부터 특정한 이벤트 수신에 관심을 선언하고 그리고 이들 이벤트의 발생에 대하여 통지를 받는다. 이것을 광고라고 한다. 이벤트 기반 통신 모델에서는 두 가지 방법이 있다: 이벤트 기반 구독 전달과 이벤트 기반 광고 전달.

이벤트 기반 구독 통신에서 소비자 구독은 모든 서버에게로 전달된다. 통지는 구독에 의해 설정된 경로를 역(backward)으로 따른다. 그림 8은 이벤트를 구독하는 과정이다.

이벤트 기반 광고 통신에서 생산자 광고는 모든 서버에게로 전달된다. 구독은 광고에 의해 설정되어 역으로 활성화된 경로를 따른다. 그림 9는 이벤트를 광고하는 과정이다.

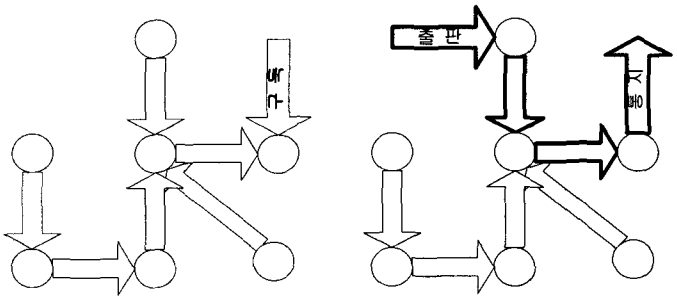


그림 8. 이벤트 기반 구독 전달

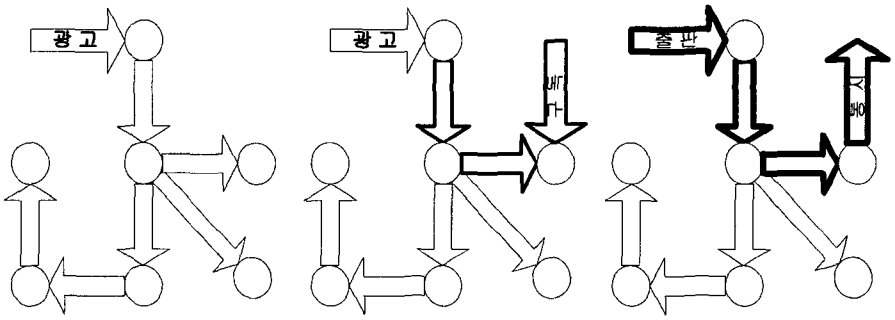


그림 9. 이벤트 기반 광고 전달

효과적인 이벤트 통지 서비스(ENS: Event Notification Service)는 이벤트 통지 프로토콜과 이벤트 통지 하부구조로서 가능하다. 이벤트 통지 프로토콜의 목표는 단순하고, 확장 가능한, 고도의 효율적인 통지 프로토콜을 제공하는 것이다. 인터넷과 기업 환경 모두 필요로 하는 적절한 실현성을 제공해야한다.

Ⅲ 절에서 기술한 침입 탐지 시스템들은 계층적인 구조이다. 지역 IDS 컴포넌트는 지역적인 침입을 조사하고 그들의 상위 계층 단계로 분석 결과를 보고한다. 언급한 침입 탐지 시스템들은 데이터 분석에서 완전한 분산 처리는 아니다. 계층적인 더 높은 단계에서 집중화된 데이터 분석이 수행되기 때문이다.

[17]에서는 이벤트 기반 통신 모델에서 특정한 관심사로 협력하는 에이전트를 사용하여 침입 탐지 정보가 분산되도록 하는 분산 침입 탐지에 대한 통신 프레임워크를 구체화하였다. 분산 IDS의 컴포넌트 사이 통신은 시스템 기능의 핵심 부분이며 요구되는 특성으로는 신뢰성, 보안, 확장성, 속도 등이 있다. 침입 탐지에서 연동하는 데이터 모델과 통신 모델은 침입 탐지 기술에 적합하게 선택되어져야 한다. 이벤트 기반 IDS를 통하여 관심사를 등록하고 분산 전파한다. 이벤트 기반은 에이전트화가 가능하다. 이벤트 기반 IDS의 관심 전파로는 세 가지 단계(level)로 나누어 계층적인 프레임워크를 사용한다: 지역적(locally), 도메인(domain) 그리고 기업(enterprise) 관심.

- 지역적 관심: 에이전트가 지역적 호스트에서만 데이터 획득에 관심이 있을 때
- 도메인 관심: 에이전트가 도메인에서만 데이터 획득에 관심이 있을 때
- 기업 관심: 에이전트가 기업에서 어떤 장소로부터 데이터 획득에 관심이 있을 때

4.2.2 푸시 기반 통신

푸시 시스템은 데이터 중심에서 정보의 효율적인 분배를 요점으로 한다(그림 10 참조). 통신은 소비자가 생산자의 관점에서 등록하는 구독단계로부터 시작한다.

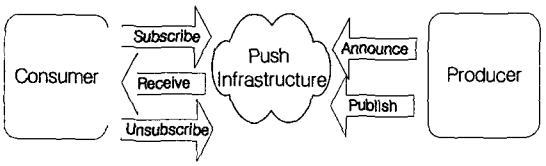


그림 10. 푸시 기반 통신 모델

푸시 컴포넌트 모델은 생산자와 소비자들, 채널 그리고 전송 시스템으로 구성된다. 푸시 기반 시스템에서 정보 분배는 특정한 채널 기반에서 수행된다. 소비자는 채널을 구독하고 그리고 보내지는 정보를 수신한다. 생산자는 채널을 통하여 데이터를 보낸다.

그림 11은 푸시 시스템의 컴포넌트들을 보여준다. 푸시 시스템은 데이터 분산 장소 그리고 규칙과 함께 방송자에게 정보를 공급하고 방송자는 채널을 따라 정보를 보내며 채널을 관리하는 책임이 있다. 방송자는 데이터에 필터를 적용하여 확실한 채널의 내용을 수신 받아 구독한 소비자들에게 채널을 경유해서 데이터를 보급한다. 수신자들 역시 필터를 적용하여 단지 내용만 수용한다. 푸시 시스템 분산 처리에서 많은 사용자들의 수로 확장 가능한 것은 방송자, 수신자 그리고 채널에 대하여 개념적으로 투명한 전송시스템을 포함하기 때문이다. 푸시 시스템의 확장성을 위해서는 특별한 방송자 하부구조가 필요하다.

푸시 통신 모델에서 푸시 시스템의 컴포넌트들은 각각 다음과 같은 기능을 가지고 있다[18]:

• 채널

채널은 방송자와 수신자 사이 논리적인 커넥터이다. 채널이 이들 컴포넌트 사이의 프로토콜을 결정한다. 중요한 프로토콜은 채널 접근 프로토콜과 구독(subscription) 프로토콜이다. 채널은 방송자들과 수신자들 사이 다 대 다 연결을 제공한다. 방송자들은 채널 설정을 제공하고 수신자들은 구독한다. 채널

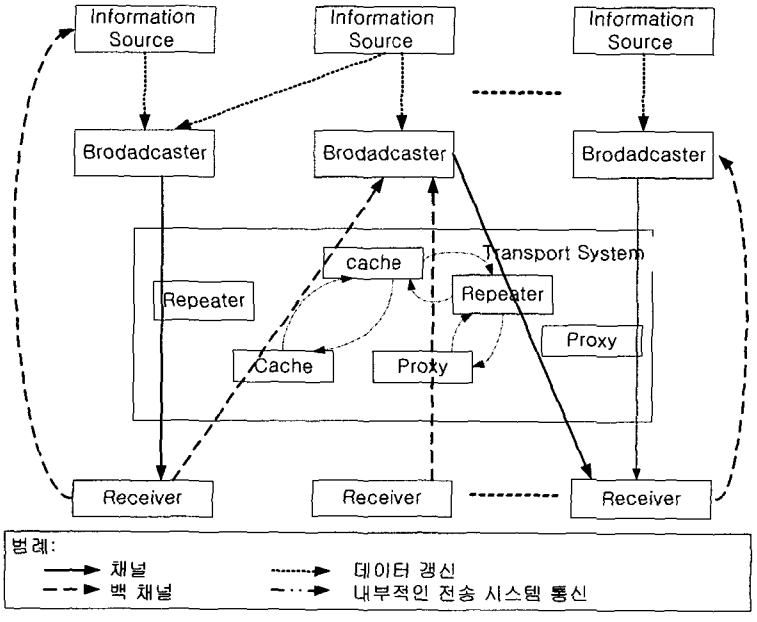


그림 11. 푸시 시스템의 컴포넌트

은 정보의 형태와 데이터 형식, 개인화-필터링, 채널 만기, 갱신 방법, 스케줄, 운영 형태, 지불에 대한 속성을 결정한다. 채널은 생산자와 소비자 사이에서는 일 대 다 관계이다. 푸시 시스템은 채널이 분산 실행 코드이므로 이동 코드 시스템과 관련 있으며 애플릿과 유사하다.

• 방송자

푸시 시스템은 채널을 제공하는 방송자 컴포넌트를 가지고 채널을 구독하는 데이터를 분산한다. 방송자의 주요한 목표는 수신자들이 방송형 컴포넌트로부터 채널로의 접근할 때 네트워크 트래픽 최소화, 지연 감소, 그리고 확장 가능한 시스템을 허가하는 것이다.

• 수신자

수신자들은 인간 사용자와 채널 사이 상호작용을 인터페이스 한다.

• 전송 시스템

전송 시스템은 확장이 가능하고 네트워크 대역폭 소비는 감소해야하며 이용성과 응답성은 향상되어야 한다. 전송 시스템의 전송 컴포넌트를 BDC(base distribution component)라 하며 BDC는 몇 가지로 구성된다: 재생기(Repeater), 캐쉬(Cache), 그리고 프록시(Proxy).

재생기는 채널의 문장을 미리 로드하고 방송자간에 같은 데이터를 제공하며, 캐쉬는 동적으로 로드하는 재생기와 같다. 그리고 프록시는 직접적으로 통신은 불가능하며 방송자와 수신자가 채널로의 접근을 용이하게 한다.

V. 분산 통신 모델 비교

표 2는 분산 시스템의 4가지 통신 모델에 대하여 결합(coupling)과 확장성(scalability)의 주요한 상호보완관계에 따른 4가지 통신 모델들을 비교한다. 분산 시스템에 대한 주요한 기초 설계는 확장성이다. 표 2의 결과를 분석하면, 푸시 시스템은 인터넷 규모에 사용되어지는 구조적인 모델로 적절하다[19,20].

표 3은 이벤트 기반과 푸시 기반 시스템을 적용한 개념에서 비교를 기술한다. 푸시 시스템에 대한 컴포넌트 통신 모델은 중간 전송 시스템이 수신자로부터 명백하게 분리되어 생산자에 의해 확장성을 지원한다. 푸시 시스템의 목적은 소비자들에게 데이터를 적절히 분산하는 것이고, 이벤트 기반 시스템은 이벤트의 통지를 요점으로 한다. 참여자들의 역할은 차이난다. 푸시 시스템은 두 개의 구별 그룹을 가진다. 즉, 이벤트 생산자(방송자)와 이벤트 소비자(수신자). 그러나 이벤트 시스템에서는 누구나 생산과 소비 이벤트가 가능하다[20].

정보 분류에 대한 채널 사용은 이벤트 기반 시스템과 푸시 시스템 사이의 주요한 구별을 제공한다. 이벤트 기반 시스템은 생산자와 소비자 사이 매우 느슨한 결합을 가지므로 이벤트 선택을 위한 이벤트 패턴을 그룹핑하는 메커니즘을 가져야 한다. 이 메커니즘이 유용하지만 서비스의 구현에는 복잡성을 추가한다. 이벤트 기반 시스템에서는 이벤트의 수를 분할하여 다운로드 하지만 푸시 시스템에서는 필터의 사용으로 데이터 전송을 감소시킨다.

표 2. 분산 통신 모델 비교

	클라이언트-서버		피어 투 피어	
	세션 기반	웹 기반	이벤트 기반	푸시 기반
결합	밀접	느슨	매우 느슨	중간
클라이언트 수	중간(1000)	아주 많음(1,000,000)	많음(100,000)	많음(100,000)
서버 수	적음(10)	많음(100,000)	많음(100,000)	적음(100)

표 3. 푸시 시스템과 이벤트 기반 시스템 비교

	푸시 시스템	이벤트 기반 시스템
목적	시기적절한 데이터 분배	이벤트 통지
참가자 역할	비동기	동기
광고 정책	단순한 광고(채널)	표현적 광고 언어
구독 정책	단순한 구독(채널)	표현적 구독 언어
이벤트의 빈도	낮음에서 중간	높음
이벤트의 수	낮음에서 중간	높음
페이로드의 크기	대규모	소규모
생산자/소비자 상호연결	정적 채널과 정적 생산자	생산자와 동적 바인딩
이벤트 그룹핑	채널	이벤트 패턴
필터링	데이터 전송 요구 감소	이벤트 수 감소

Ⅵ. 요약과 결론

본 논문에서는 침입 탐지를 위한 효과적인 분산 통신 모델을 도출하고자, 먼저 기존의 침입 탐지 시스템인 DIDS, AAFID, EMERALD 그리고 GrIDS에 대한 통신 메커니즘에 대한 분석을 하였다. 기존의 침입 탐지 시스템은 TCP/IP 기반에서 특정한 메시지 전송으로 컴포넌트 사이에서 통신한다. DIDS는 OSI 통신 모델기반 네트워크를 관리하는 프로토콜로서 컴포넌트 사이 메시지 교환에 CMIP 통신 프로토콜을 사용한다. AAFID는 계층적인 구조에서 컴포넌트들이 표준메시지를 이용하여 정보를 분배한다. EMERALD도 계층적인 구조에서 이벤트 스트림을 대상으로 하는 인터페이스 명세와 전송계층을 통하여 처리되는 메시지 시스템을 이용한다. 그리고 GrIDS는 컴포넌트들이 대규모 분산 활동으로부터 행위 그래프를 만드는 것으로 메시지를 교환하는 메시지 기반 통신 메커니즘을 사용한다.

현재 인터넷의 분산 플랫폼처럼 이용되고 있는 클라이언트 서버 통신모델과는 대비되는 통신 모델로서 피어 투 피어 통신 모델이 있다. 피어 투 피어 통신 모델은 데이터 중심 그리고 정보의 효율적인 보급을 요점으로 한다. 피어 투 피어 통신에서는 피어들 사이 관심의 정보를 구독과 광고를 통하여 정보를 교환한다. 본 논문에서는 피어 투 피어 통신 모델에서 이벤트 기반과 푸시 기반에 대한 통신 모델을 분석하였다. 이벤트 기반 통신에서는 이벤트 통지가 요점이며 효과적인 이벤트 통지 서비스는 이벤트 통지 프로토콜과 이벤트 통지 하부구조로서 가능하다. 이벤트 기반은 느슨한 결합으로 높은 확장이 가능하고, 푸시 기반은 채널을 사용하여 더 높은 확장이 가능하다. 따라서 피어 투 피어 통신 모델이 대규모 망으로의 사용에 적절하다고 사료된다. 피어 투 피어 통신 모델을 분산 침입 탐지 시스템에 적용하여 관심사를 등록하고 효율적인 관심 정보 분배를 통하여 차세대 분산 침입 탐지 시스템으로 이용할 수 있을 것이다.

향후 연구계획으로는 피어 투 피어 통신 모델을 기반으로 분산 침입에 대한 탐지 정보들을 피어들에게

보급하는 과정을 시뮬레이션 하여 그 성능을 검증해 보는 것이다.

참고문헌

[1] Eugene H. Spafford and Diego Zamboni, "Intrusion detection using autonomous agents", Computer Networks, 34(4):547-570, October 2000.

[2] S. Snapp, J. Brentano, and G. Dias et al. DIDS (Distributed Intrusion Detection System) motivation, architecture, and an early prototype. In Proceedings of the 14th National Computer Security Conference, October 1991.

[3] Jai Sundar Balasubramaniyan, Jose Omar Garcia-Fernandez, David Isacoff, Eugene Spafford, and Diego Zamboni. An architecture for intrusion detection using autonomous agents. In Proceedings of the Fourteenth Annual Computer Security Applications Conference, pages 13 24. IEEE Computer Society, December 1998

[4] Phillip A. Porras and Peter G. Neumann. EMERALD: event monitoring enabling responses to anomalous live disturbances. In 1997 National Information Systems Security Conference, Oct 1997.

[5] S. Staniford-Chen, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, C. Wee, R. Yip, and D. Zerkle. GrIDS-a graph based intrusion detection system for large networks. In Proceedings of the 19th National Information Systems Security Conference, September 1996.

[6] Rebeca Bace and Peter Mell. NIST Special Publication on Intrusion Detection System, National Institute of Standards and Technology

[7] http://www.vertel.com/products/tmn.asp

[8] C. Kahn, P. Porras, S. Stamofprd-Chen, and B. Tung. A common intrusion detection framework. In Journal of Computer Security, 1998.

[9] Steven Cheung, Rick Crawford, Mark Dilger, Jeremy Frank, Jim Hoagland, Karl Levitt, Jeff Rowe, Stuart Staniford-Chen, Raymond Yip, Dan Zerkle. The Design of GrIDS- A GraphBased Intrusion Detection System. Department of Computer Science University of California at Davis, Davis CA 95616, January 26, 1999.

[10] E.Koutsofios and S.North et al. Drawing graphs with dot. Technical report, AT&T Bell Laboratories, November 1996.

[11] http://www.infosys.tuwien.ac.at/Staff/pooh/papers/BIBOS

[12] M. Hauswirth and M. Jazayeri. A component and communication model for push systems. In Proceedings of ESEC/FSE 99 - Joint 7th European Software Engineering Conference (ESEC) and 7th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE-7), Toulouse, France, September 1999

[13] A. Carzaniga, D.S. Rosenblum, and A.L. Wolf "Design and Evaluation of a Wide-Area Event Notification Service". ACM Transactions on Computer Systems, 19(3):332-383, Aug 2001.

[14] G. Cugola, E. Di Nitto, and A. Fuggetta. The JEDI event-based infrastructure and its application to the development of the OPSS WFMS. Technical report. CEFRIEL, Politecnico di Milano, Via Fucini, 2, 20133 Milano, Italy, August 1998.

[15] M. Hauswirth and M. Jazayeri. A component and communication model for push systems. In Proceedings of ESEC/FSE 99 - Joint 7th European Software Engineering Conference (ESEC) and 7th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE-7), Toulouse, France, September 1999

[16] Dipl.-Ing. Manfred Hauswirth. Internet-Scale Push Systems for Information Distribution Architecture, Components, and Communication. Institut fur Informations systeme, August 1999.

[17] Rajeev Gopalakrishna, Eugene H. Spafford. A Framework for Distributed Intrusion Detection using Interest-Driven Cooperating Agents. Center for Education and Research in Information Assurance and Security, Purdue University.

[18] http://www.infosys.tuwien.ac.at/Staff/pooh/papers/PushIssues/

[19] M. Hauswirth and M. Jazayeri. A component and communication model for push systems. In Proceedings of ESEC/FSE 99 - Joint 7th European Software Engineering Conference (ESEC) and 7th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE-7), Toulouse, France, September 1999

[20] Antonio Carzaniga, David S. Rosenblum, Alexander L. Wolf. SIENA: Wide-Area Event Notification Service. WISEN, 1998. http://www.ics.uci.edu/~irus/wisen/,http://www.isr.uci.edu/events/twist/twist2000/program.html

장 정 숙

1991년 경일대학교 공과대학 컴퓨터공학과 졸업(학사), 1992년~1995년 대구가톨릭대학교 교육대학원 전자계산교육전공(석사), 1998년~현재 대구가톨릭대학교 대학원 전자계산학 전공 박사과정, 1995년~현재 유진컴퓨터 대표, 관심분야 차세대인터넷, 통신망 보안, 이동 에이전트, QoS 보장 기술, 고속 통신망 응용 서비스

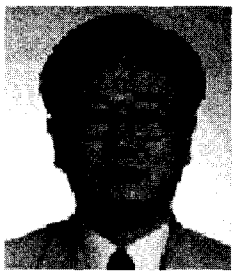
전 용 희

1978년 고려대학교 전기공학과 졸업(BS), 1987년 미국 플로리다공대 대학원 컴퓨터공학과 수료, 1989년 미국 노스캐롤라이나주립대 대학원 Elec. and Comp. Eng. 졸업(MS), 1992년 미국 노

스캐롤라이나주립대 대학원 Elec. and Comp. Eng. 졸업(Ph. D.), 1978~1978년 삼성중공업(주) 근무, 1978~1985년 한국전력기술(주) 근무, 1989~1989년 노스캐롤라이나주립대 Dept of Elec. and Comp. Eng. TA, 1989~1992년 노스캐롤라이나주립대 부설 CCSP(Center For Comm. & Signal Processing) RA, 1992~1994년 한국전자통신연구원 교환전송기술 연구소 선임 연구원, 1994~현재 대구가톨릭대학교 컴퓨터·정보통신공학부 학부장, 공과대학장, 관심분야 초고속 통신망 프로토콜, 통신망 성능분석, QoS 보장 기술, 고속통신망 응용 서비스, 통신망 보안

장 종 수

1984년 경북대학교 전자공학과 학사, 1986년 경북대학교 대학원 전자공학과 석사, 2000년 충북대학교 대학원 컴퓨터공학과 박사, 1989년 7월~현재 한국전자통신연구원 정보보호연구본부 네트워크보안연구부 네트워크보안구조연구팀 팀장/선임연구원, 관심분야 Network Security, Active Network, Biometry



손 승 원

1984년 경북대학교 전자공학과 졸업 (공학사), 1994년 연세대학교 전자공학과 졸업 (공학석사), 1999년 충북대학교 전자공학과 졸업 (공학박사), 1991년~현재 한국전자통신연구원 정보보호연구본부 네트워크보안연구부 부장/책임연구원, 관심분야 Network Security, Optical Security, Active Network, Biometry