主題

정책 기반 네트워크에서 침입탐지에 대한 연구

대구가톨릭대학교 컴퓨터정보통신공학부 교수 전 용 희
대구가톨릭대학교 컴퓨터정보통신공학부 박사과정 장 정 숙

차 례

1. 서론

인터넷의 폭발적인 성장으로 인하여 여러 가지 문제점들이 발생하고 있으며, 그 중에서 개인 정보에 대한 불법적인 접근과 네트워크를 통한 공격을 들 수 있다. 이에 따라 보안 기술의 중요성이 높아지고 있다. 기존의 인증 체계 및 호스트 내부의 보안 체계는 시스템에 의존적인 측면이 강하고 빠른 속도로 변화하는 네트워크 공격 기술에 대응하기에는 어려움이 있다. 정책 기반의 네트워크 관리 기술은 분산 네트워크 환경에서 보안관리를 가능하게 하고 자신의 네트워크에 있는 특정 트래픽에 영향을 미치게 된다. 정책 기반 네트워킹(PBN: policy-based networking)은 최근 네트워킹 분야에서 관심이 증가되고 있는 기술 중의 하나이다. 현재, 장비 위주의 네트워크 인프라는 관리의 분산, 통합의 어려움, 트래픽 보장의 어려움, 보안과 인증의 분산 등 여러 가지 문제를 가지고 있다. 또한 네트워크가 점차 거대하여지고 복잡하여 짐에 따라, 이들 장비를 관리하기 위한 비용의 상승이 커다란 문제가 되고 있다. 정책 기반 네트워크 관리는 네트워크에서 제공하는 QoS(Quality of Service), 보안(security) 및 자원을 공통된 형태로 제공함으로써 효율적인 네트워크 관리를 목적으로 하고 있다.

정책(policy)이란 다른 조건에서 네트워크의 행위를 제어하는 일련의 규칙이라고 할 수 있다[1]. 모든 사용자들에게 균일(uniform) 혹은 최선(best-effort) 서비스를 제공하기보다는 정책-실행 네트워크는 우선순위(priority) 혹은 차등 서비스(DS: Differentiated Services)와 같은 다른 사용자-수준의 특성을 고려할 수 있으며 또한 각 패킷에 대하여 동적으로 그러한 조치를 결정할 수 있다. 인터넷 사용의 급격한 증가로 네트워크 관리 측면에서의 효율적인 보안 관리방안으로 보안 정책을 적용하여 동적으로 변화되는 네트워크 상태를 관리할 수 있다. 네트워크 운용상

의 정책이란 현재 가지고 있는 자원에 대한 모든 정보를 가지고 어떻게 활용할 것인가에 대한 원칙과 계획을 말한다. 보안 도메인(security domain)은 하나 이상의 실행점(enforcement point)에서 실행되는 공통 보안 집합을 공유하는 통신 개체와 자원들의 집합을 의미한다.

효율적인 네트워크의 보호를 위하여 네트워크를 경유한 공격에 대한 빠른 탐지와 적절한 대응을 할 수 있는 침입 탐지 시스템에 대하여, 특히 분산 침입 탐지 시스템에 대한 연구가 세계적으로 많이 진행되고 있다. 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. 그러나 국내에서는 아직 분산 침입 탐지 시스템 컴포넌트 사이의 통신 메커니즘과 일반적인 통신 모델에 대하여 발표된 연구 결과는 거의 없는 실정이다.

분산 침입 탐지 시스템의 한 구현으로써, 개별 시스템 단위의 과다한 트래픽 분석과 다양한 침입유형에 보다 능동적으로 대응하기 위해서 지역적 보안환경에서 광역적인 보안환경으로 적용하기 위한 글로벌 네트워크 보안 제어 프레임워크 기술이 대두되고 있다. 글로벌 네트워크 보안 제어 프레임워크에서는 각 지역 망의 출력 트래픽들의 종합 분석과 망의 구성과 상태정보 그리고 관리정보 및 통계정보를 통한 다단계 분석으로 침입 예측 및 환경에 적합한 대응 정책의 결정이 가능하게 된다. 이를 위해서 고속화 침입탐지 엔진, 이들의 정보를 축약하기 위한 기법의 개발, 이들의 정보를 전달하기 위한 프로토콜의 표준화, 계층적인 침입 분석의 개발과 그들 정보를 공유하기 위한 협력 메커니즘의 수립 그리고 종합적인 침입 대응 시나리오 등이 필요하다. 이를 효과적으로 관리하기 위한 보안 관리 프레임워크 구조로는 IETF 정책 프레임워크가 있다. IETF 정책 프레임워크는 각 보안 정책 도메인 관리를 위한 보안 정책서버를 두고 각 보안 정책 도메인은 사이버 공격 분석 및 이벤트 정보 수집을 위하여 다중 에이전트들을 분산 구성한다[2].

2. 정책 기반 네트워크

보안 정책 시스템은 중요한 정보와 다른 자원들이 특정한 시스템에서 관리되어 분산되는 방법을 규제하는 법 혹은 규칙을 설정한다. 보안 정책 시스템(SPS: Security Policy System)은 보안 정책 데이터베이스(SPD: Security Policy Database), 보안 정책 서버(SPS: Security Policy Server) 그리고 정책 클라어언트(PC: Policy Client)로 구성되며 보안 정책 프로토콜(SPP: Security Policy Protocol)을 사용하여 정보를 교환한다. 침입 탐지를 위한 보안 정책은 보호되어야 할 정보 자산, 필요한 침입 탐지 시스템(IDS: Intrusion Detection System)의 형태, IDS의 위치, IDS가 탐지할 공격의 유형, 특정 공격이 식별되었을 때 제공될 대응 혹은 경보의 형태를 정의한다. 정책의 한 예로, 규칙-기반 정책은 IP 주소, 시간, 프로토콜, 그리고 차단, 로긴, 경고 혹은 통과 허용 같은 조치를 명시하기 위한 지시와 같은 qualifier를 사용하여 보안 정책을 자동으로 시행하도록 해준다.

네트워크 운용상의 정책이란 현재 가지고 있는 자원에 대한 모든 정보를 가지고 어떻게 활용할 것인가에 대한 원칙과 계획을 말한다. 분산 네트워크 환경에서 보안 관리를 가능하게 하고 자신의 네트워크에 있는 특정 트래픽에 영향을

미치게 하는 정책기반의 네트워크 관리(PBNM: policy-based network management) 기술 개발이 필요하다. 정책 기반 관리 구조는 IETF에서 정책 관리를 위한 정책관리도구(PMT: Policy Management Tool), 정책 저장소(Policy Repository), 정책 결정을 위한 Policy Consumer (Policy Decision Point), 정책 적용을 위한 Policy Target(Policy Enforcement Point) 등의 기능적 컴포넌트들을 포함한다(그림 1 참조).

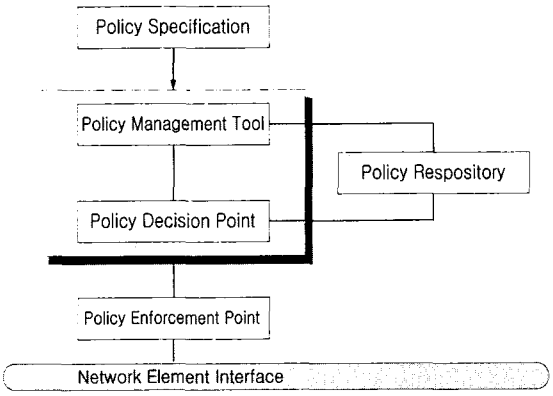


그림 1. 정책 기반 보안 시스템의 컴포넌트

정책 관리부는 망 운용자 서버의 목적 및 사업자의 목표에 따라 결정된 망 운용규칙을 일관성 있는 정책 데이터로 변환하기 위해서 PFDL (Policy Framework Definition Language)을 이용한다. 망 관리 정책은 정책 저장소에 저장되며, 망 내의 분산되어 있는 정책 결정부(PDP)에 의해 실시간으로 검색되고, 정책 저장소에 수용되는 데이터는 정책을 결정하기 위한 정책 결정조건과 결정된 정책에 따라 적용되어야하는 정책 동작으로 구성된다. 저장된 정책을 조회하거나 생성된 신규 정책을 저장하기 위한 프로토콜로 디렉토리 서비스에 널리 이용되고 있는 Lightweight Directory Access Protocol (LDAP)가 이용된다.

3. 침입 탐지 시스템

3.1 개요

침입(intrusion)은 컴퓨터가 사용하는 자원의 무결성(integrity), 기밀성(confidentiality), 가용성(availability)을 저해하는 일련의 행위들의 집합 또는 컴퓨터 시스템의 보안정책(SP: Security Policy)을 파괴하는 행위로 규정한다. 침입 탐지 시스템은 대부분 침입 차단 시스템과 연계하여 네트워크 단계 혹은 호스트 단계에서 비정상적인 사용, 오용 등의 침입을 관리자가 실시간으로 탐지를 할 수 있는 시스템이며 침입 탐지 유형에 따라 비정상 탐지(Anomaly Detection), 오용 탐지(Misuse Detection) 등으로 구분한다. 일반적으로 접근 시 정해진 모델을 벗어나는 경우를 탐지하는 것을 비정상 탐지라 하며, 침입이라고 정해진 모델과 일치하는 경우를 오용 탐지라 한다. 또한, 웹 서비스와 같은 호스트에 설치되어 설치된 호스트만을 대상으로 침입 탐지를 하는 것을 호스트-기반 IDS라고 하며 일정 부분의 네트워크 전체를 대상으로 침입 탐지를 하는 것을 네트워크-기반 IDS라고 한다.

기존의 침입 탐지 시스템들은 다양한 침입에 능동적으로 대처하는데 어려움이 많으며 대규모 네트워크 환경에서의 효율적인 탐지에 적합하지 않는 구조를 가지고 있다. 침입 탐지 시스템은 시스템 스스로가 침입 여부에 대한 판정을 내리고 적절한 대응을 할 수 있어야 한다. IDS가 요구하는 일반적인 특성은 다음과 같다[3]:

- 최소한의 감독으로 연속적인 실행
- 결점 허용성
- 전복(subversion)에 대한 저항
- 시스템에 최소 오버헤드 부과

- 구성 가능성
- 배치의 용이성
- 시스템 변화에 대한 적응성
- 공격 탐지 가능성

3.2 구성 요소

현대적인 IDS 모델을 처음 제시한 Dorothy E. Denning은 IDS를 "대상시스템에 대한 비인가된, 비정상적인 행동을 탐지, 구별하고 이에 대응하는 기능을 가진 시스템"으로 정의하였다[4]. IDS는 공격 방법을 기반으로 하여 공격자의 침입을 탐지하므로 신기술 적용이 빠른 편이며, 침입 차단 시스템에서 막지 못하는 내부자의 공격도 차단 가능한 특징이 있다. 또 접속 IP에 상관없이 침입을 차단하는데 이는 시스템의 상태 정보를 보고 판단하기 때문에 가능하다. 그리고 IDS는 시스템 침입에 즉시 대응할 수도 있고, 탐지에 그치지 않고 침투경로를 따라 공격자를 적발할 수 있는 능동적인 대응도 가능한 시스템이다.

그림 2에서와 같이 네트워크로부터 감사 자료를 탐지부로 보내게 되고, 감사 자료와 데이터베이스 내에 공격 패턴들과 비교를 하게 된다. 만약 패턴이 일치하면 대응부는 경고를 발생시켜 네트워크 내에 모든 호스트로 전송하여 침입에 대한 빠른 대응을 하게 된다.

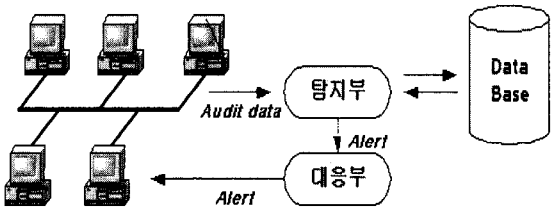


그림 2. 침입 탐지 시스템의 개념적 동작

침입 탐지 시스템은 데이터 소스로부터 데이터를 수집하는 단계, 데이터를 필터링하고 축약하는 단계, 분석 및 침입 탐지 단계, 그리고 보고 및 대응단계로 이루어져있다(그림 3 참조).

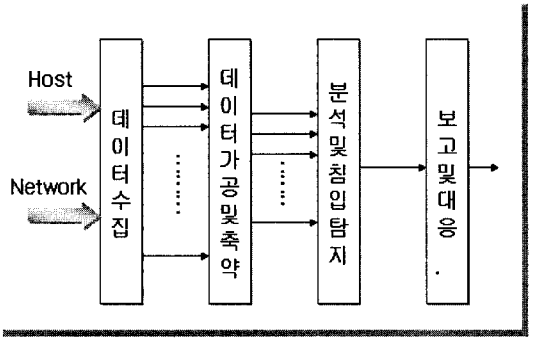


그림 3. 침입탐지시스템의 구성요소

가. 데이터 수집 단계

침입 탐지에 필요한 감사 자료에 대한 수집은 침입탐지의 데이터 수집 구조 방법을 통하여 수집 가능하며 일반적인 방법으로는 호스트 기반과 네트워크 기반 두 가지로 나눌 수 있다. 먼저 호스트 기반에는 System Source, Accounting, Syslog로 탐지를 하는데, 실시간이 어렵다는 단점이 있다. 네트워크 기반에서는 SNMP 정보와 네트워크 패킷을 이용하여 탐지에 필요한 데이터를 수집한다.

나. 데이터 가공과 축약(Data Filtering & Reduction) 단계

수집 단계에서 모아진 자료는 의미 있는 정보로 가공되고, 실시간 침입 판정을 위한 최소한의 정보와 자체의 감사 레코드(Audit Record)로서의 의미를 가지는 축약을 한다.

다. 분석 및 침입 탐지(Analysis & Intrusion Detection)

분석 및 침입 탐지 단계에서는 비정상적인 침입 탐지와 오용 침입 탐지로 침입을 탐지하

게 된다.

라. 보고 및 대응(Report & Response) 단계

침입 발견 시 즉각적인 보고와 해당 조치 사항을 수행하고 대응하여 침입 진행 상황을 보고하는 단계이다. 침입에 대한 대응으로서 IDS가 몇 가지 행동들(예: closing holes, shutting services down, logging an intruder)을 취할 때 이런 IDS를 능동적이라 하고, 만약 IDS가 약간의 경고나 통지만 한다면 그러한 IDS는 수동적이라고 한다.

3.3 분류 및 특징

가. 호스트 IDS(HIDS)

호스트 IDS는 모니터 될 각 호스트에 상주하는 에이전트를 채택하고 있다. 에이전트는 사건, 시스템 로그, 커널(kernel) 로그, 중요한 시스템 파일과 비인가된 변경을 조사하는 감사가 가능한 자원이나 의심스러운 활동 패턴 등을 조사한다. 정상적인 것이 아닌 것이 인지될 때, 경보(alert)나 SNMP 트랩이 자동으로 발생된다. 전통적인 HIDS는 내부자 위협을 탐지하는데 매우 좋고 통상적으로 광범위한 손해 평가와 데이터 수사학(data forensics)을 제공한다.

HIDS 접근의 단점은 중요 시스템 상에 에이전트를 배치해야할 필요가 있고 감사 정책에 면밀한 주의를 기울여야 한다는 필요성이다. 이것으로 인하여 전통적인 HIDS는 IDS 중에서 가장 배치하기 힘이 든다. 또한 호스트가 침해되면 경보도 없고 무슨 일이 발생하였는지, 손실된 것을 결정하기 위한 수사 자료가 없게 된다. HIDS는 정책 준수 결정을 지역적으로 하며 관리자에게는 단지 경보만 전송한다. 그러므로 네트워크 대역폭을 작게 사용하는 특징이 있다.

HIDS의 변형된 형태로 집중(centralized) 호스트 기반 침입 탐지가 있다. 이 방법은 분석을 위하여 감시된 파일, 로그 등을 관리자에게 보내어 집중적으로 분석을 한다. 호스트가 만일 침해되더라도 모든 필요한 정보를 관리자에게 보냄으로써 더욱 안전한 면이 있다. 그러나 정보를 전송하는데 많은 대역폭이 필요하다.

나. 네트워크 IDS(NIDS)

네트워크 IDS는 네트워크 상의 패킷들을 실시간으로 감시하여 오용 패턴을 탐지한다. 패킷들을 이미 알려진 "침입 시그너처"의 데이터베이스와 대조함으로써 혹은 비정상을 탐지하기 위하여 프로토콜 복호(decode)를 수행하거나, 혹은 둘 다를 수행함으로써 이루어진다. 이 흔적(signature) 데이터베이스는 새로운 공격이 발견될 때마다 정기적으로 갱신된다. 의심스러운 행위가 인지되면, 네트워크 기반 IDS는 경보를 발생하거나 공격하는 연결을 즉시 종료할 수 있다. 또한 방화벽과 연동하여 공격자를 차단하도록 하기위한 새로운 규칙을 자동으로 정의할 수 있다.

기존의 대부분의 네트워크-기반 IDS는 "무차별 모드"(promiscuous mode)로 동작한다. 그러므로 패킷들이 IDS 장치로 향하든지 안하든지에 관계없이 지역 구간 상에 있는 모든 패킷들을 조사한다. 이것의 예로는 스니퍼(Sniffer)와 같은 네트워크 모니터가 있다. 시스템 자원을 많이 사용하기 때문에 별도의 호스트가 통상적으로 필요하다. 잘 알려진 네트워크-기반 침입 탐지 시스템으로 Cisco, Cybersafe, ISS, Shadow 등이 있다[5].

네트워크 IDS의 장점은 다음과 같다:

- 보호 범위가 넓다.
- 설치가 용이하다.
- 네트워크 트래픽의 성질에 대한 상세한 정보를 제공한다.
- HIDS에 비하여 성숙한 기술이다.
- 침입에 대하여 실시간으로 대응이 가능하다.

다. 네트워크 노드 IDS(NNIDS)

NNIDS는 네트워크 기반 IDS의 몇 가지 제한점을 극복한 혼합형 IDS의 한 형태이다. NNIDS 에이전트는 네트워크 패킷을 잡아서 프로토콜 분석을 수행하고 시그너처 데이터베이스 목록과 비교한다는 점에서 네트워크 기반 IDS와 유사한 방법으로 동작한다. 그러나 이 에이전트는 위치하고 있는 네트워크 노드로 향하는 패킷만 단지 관계한다. 이 에이전트는 호스트의 프로토콜 스택 내부에 설치되기 때문에 때때로 스택-기반 IDS라고도 한다.

NNIDS는 회선상의 모든 패킷을 조사할 필요가 없기 때문에 더욱 고속으로 되고 시스템 자원도 작게 필요로 한다. 그래서 과도한 오버 헤드를 부과하지 않고 기존 서버에 설치될 수 있다. 전통적인 네트워크-기반 IDS에서는 문제가 되는, 과도한 부하를 가진 구간이나, 교환 네트워크 환경 혹은 회선 상에 암호화된 트래픽을 가진 가상 사설망(VPN: Virtual Private Networks) 구현 등에 적합하다.

위와 같이 IDS를 세 가지 형태로 구분하였지만, 보다 효과적인 IDS를 위하여 네트워크와 호스트 기반 침입 탐지를 결합하여 사용하는 것이 바람직하다. 이 경우, 어디에 각 형태를 사용하고 데이터를 어떻게 통합하는 가가 실제적이고 중요한 관심사이다. 본 논문에서는 호스트 기반 IDS와 네트워크 기반 IDS를 결합한 혼합형 IDS를 기반으로 각 IDS 시스템들에서 들어오는 정보를 종합적으로 상호 관련하여 다중 도메인 상의 네트워크에서 침입 탐지를 통합적으로 수행할 수 있는 구조를 제시한다.

3.4 침입 유형과 탐지도구

가. 침입 유형

침입탐지시스템에서 네트워크 공격을 탐지하는 일반적인 공격 유형은 다음과 같다[6].

- IP spoofing attack(IP 속임 공격)
- Packet sniffing(패킷 훔쳐보기)
- Passwords attacks(패스워드 공격)
- Sequence number prediction attack (순서번호 예측공격)
- Session hi-jacking attacks. (세션 가로채기 공격)
- Shared library attack (공유 라이브러리 공격)
- Social engineering attack(사회 공학 공격)
- Technological vulnerability attack (기술적 취약성 공격)
- Trust-access attacks(신뢰-접근 공격)

대부분 IDS로 보고 되는 컴퓨터 침입은 세 가지 형태로 분류한다[7]:

- 스캐닝 침입(Scanning Intrusion)
- 서비스 거부 침입 (Denial of Service Intrusion)
- 침투 침입(Penetration Intrusion)

1) 스캐닝 침입

스캐닝 침입은 침입자들이 다양한 패킷들을 대상 네트워크로 보내어 대상 네트워크 혹은 시스템을 조사할 때 사용한다. 스캐닝 침입은 시스템 자체를 붕괴시키지는 않으며, 이들 활동에 사용되는 다양한 도구들은 다음과 같다: Network Mappers, Port Mappers, Network Scanners, Port Scanners, Vulnerability Scanners.

스캐닝 침입으로 다음과 같은 결과를 얻을 수 있다.

- 대상 네트워크의 위상
- 방화벽에 허가된 네트워크 트래픽의 형태
- 네트워크에 활동 중인 호스트
- 호스트에 수행중인 운영체제
- 호스트에 수행하는 소프트웨어

• 탐지된 모든 소프트웨어에 대한 버전

스캐닝 침입은 이용 가능한 자원을 발견하고, 시스템의 취약점을 분석하지만 법에 위배되지는 않는다. 그러나 IDS의 흔적으로 합법적인 사용과 침입의 의도로 스캔 하는 것 사이의 차이를 식별할 수 있다.

2) 서비스 거부 침입

서비스 거부(DOS: Denial Of Service) 침입은 대상 네트워크 시스템 혹은 서비스들을 느리게 하거나 멈춤을 시도한다. 서비스 거부 공격은 인터넷 사회에서는 일반적인 침입이며, 전자 상거래에서 구매하려는 접근을 불가능하게 하여 주요한 손실을 가져온다. 서비스 거부 침입에는 다음과 같은 두 가지 주요한 침입의 형태가 있다:

• 결점 이용(Flaw Exploitation) DOS 침입

대상 시스템의 소프트웨어 결점을 이용하는 침입이다. 그 예로서 윈도우 대상 시스템에 예기치 못한 대규모의 ping 패킷을 보내는 "Ping of Death"가 있다. 대상 시스템은 이 비정상적 패킷을 운용하지 못하므로 대상 시스템은 붕괴된다. 자원을 소모하는 DOS 침입은 대상 시스템의 CPU 시간, 메모리, 디스크 공간, 버퍼 공간 그리고 대역폭 자원을 모두 소모하므로 결과적으로 시스템은 붕괴되는 것이다.

• 플러딩(flooding) DOS 침입

대상 시스템이 운영 가능한 것 보다 더 많은 정보를 단순히 시스템 혹은 시스템 컴포넌트로 보낸다. 이 침입으로 대상 시스템의 처리능력을 압도하는 충분한 정보를 보내지는 못하지만 대상 네트워크 연결을 독점 할 수 있으므로 그 밖의 자원 사용을 부인하게 하여 자원의 가용성을 위반한다. 분산(Distributed) DOS란 DOS 공격의 한 형태이다. DDOS는 침입자가 다중 컴퓨터를 이용하여 침입을 하는 단순한 플러딩 DOS 침입이다. 침입하는 대규모 다중 컴퓨터들은 침입자의 컴퓨터에 의해 집중적으로 제어되어 대상 시스템은 붕괴된다.

일반적인 플러딩 DOS 공격에는 대응책이 가능하지 않지만 이들 공격을 진정시키거나 혹은 방지하는 다른 방법으로, 다음과 같은 IDS에 대한 몇 가지 해결책이 존재한다[7]: 동적인 IDS 대응, 통신 채널의 분리, 분산화된 비-계층 IDS 이용, 이동 에이전트로 컴포넌트 회복 가능.

3) 침투 침입

침투 침입(Penetration Intrusion)은 비권한 획득 그리고 시스템의 특권, 자원 혹은 데이터의 대체를 포함한다. 이 침입은 무결성과 제어권을 위반한다. 이 침입은 소프트웨어 결점의 다양한 개발로 시스템의 제어를 획득 가능하고, 침입 방법이 세분화되어 있어 영향이 다양하다. 그 일반적인 형태는 다음과 같다: User to Root, Remote to User, Remote to Root, Remote Disk Read, Remote Disk Write.

나. 침입 탐지 도구

상업적인, 연구적인 그리고 공개적인 침입 탐지 도구들이 빠르게 개발되고 있으며 다양하게 이용 가능하다. 일반적으로 알려져 있는 상업적인 침입 탐지 도구로는 Real Secure가 있으며, Tripwire는 상업적 그리고 공개적으로 둘 다 이용이 가능하며, 그리고 공개적인 침입 탐지 도구들로는 Shadow와 Snort가 있다[5]. 각각에 대한 특성은 다음과 같다.

• Realsecure : 인터넷 보안 시스템이 개발하였으며, 실시간 IDS로서 네트워크 기반 인식 엔진, 호스트 기반 인식 엔진 그리고 관리자의 모듈과 같이 세 가지 부분으로 구성

된다.

- Tripwire : 침입 탐지에 대한 파일 무결성 평가 도구이다. 시스템 파일 정보를 데이터베이스로 생성하고 파일 내용을 기반으로 전에 생성한 내용과 현재 정보를 비교하여 파일 길이와 암호화 체크섬을 한다.
- Shadow : CIDER 프로젝트에서 개발되었으며, 센서와 분석 스테이션 둘 다를 사용하며, 센서는 방화벽 바깥쪽 같은 보통 네트워크에서 주요한 모니터링 위치에 주거하고 분석 스테이션은 방화벽 안쪽에 주거한다.
- Snort : 초경량 네트워크 침입탐지 시스템이며 소스가 공개된 공개 침입 탐지 도구이다. 실시간 트래픽 분석과 패킷 로긴 능력이 있다. 프로토콜 분석과 문장 매칭 그리고 다양한 공격 탐지에 사용되며 트래픽 기술에 대하여 유연한 규칙(rule) 언어와 탐지엔진을 사용한다[8].

4. 수집 메커니즘 및 장치

4.1 직접 모니터링과 간접 모니터링

침입 탐지 시스템이 물리적 현상으로부터 직접 데이터를 획득하여 모니터 되는 컴포넌트에서 조건 혹은 행위를 측정 할 때 이것을 직접 모니터링(direct monitoring)이라 하고, 침입 탐지 시스템이 정보를 획득하기 위해서 분리 메커니즘 혹은 도구에 의존 할 때 이것을 간접 모니터링(indirect monitoring)이라 한다. 즉, 직접 모니터링은 객체의 특성을 측정 혹은 관찰하는 것이고, 간접 모니터링은 그 특성을 가지는 객체의 영향을 측정하거나 관찰하는 것이다. 예를 들면, 유닉스 호스트에서 CPU 부하를 관찰하는 ps 명령어의 사용은 직접 모니터링으로 간주된다. 그것은 ps가 커널에서의 해당 데이터 구조로부터 부하 데이터를 추출하기 때문이다. 또한 만약 CPU 부하가 로그 파일에서 기록되어 후에 읽혀진다면, 이것은 간접 모니터링으로 간주한다. 그것은 관찰을 위하여 분리 메커니즘(로그 파일, 네트워크 패킷)에 의존하기 때문이다. 이와 같이, 데이터 수집 방법에 따라 직접 혹은 간접 모니터링으로 분류된다.

직접 모니터링은 신뢰성(reliability), 완전성(completeness), 부피(volume), 확장성(scalability), 그리고 적시성(timeliness)에서 간접 모니터링보다 더 나은 탐지 능력을 보인다. 그러나 직접 모니터링은 생성 정보의 형태와 모니터 되는 컴포넌트에 대하여 보다 특정한 방법에서 설계되어야 하는 메커니즘으로 구현이 복잡하다는 단점을 가진다.

4.2 호스트 기반과 네트워크 기반

실제적으로, 데이터 수집 방법은 다음 정의에 의하여, 일반적으로 호스트 기반 혹은 네트워크 기반으로 분류된다.

- 호스트 기반(host-based) 데이터 수집
  시스템의 상태, 메모리의 내용, 혹은 로그파일을 호스트에 상주하는 소스로부터 데이터를 획득하는 것.
- 네트워크 기반(network-based) 데이터 수집
  패킷이 네트워크를 통하여 지나갈 때 패킷을 포획함으로써 데이터를 획득하는 것.

호스트 기반 데이터 수집이 타당한 이유로는 다음과 같은 것이 있다[9]: 정확한 데이터 수집, 발생하는 모든 사건에 대한 보고가능, 네트워크 기반에서 발생하는 삽입 및 속임수(evasion) 공격 문제가 없음, 데이터 수집과 통일에 대한 문제가 없음, 그리고 호스트 내부의 활동(action)을 관찰 가능.

네트워크 기반 데이터 수집은 기존의 네트워

크에 IDS가 전개되므로 호스트에 아무런 변경을 주지 않는다. 이런 이유로 많은 상용 침입 탐지 시스템은 네트워크 기반 데이터 수집을 사용한다. 또한 다른 호스트에서 완전히 보이지 않음으로써, 네트워크 상에서의 행동을 관찰하기 위한 편리한 장점을 제공한다.

"네트워크 기반"으로 통상 간주되는 침입 탐지 시스템은 간접/네트워크 기반 모니터링 메커니즘에 해당하며, 간접/호스트 기반과 모든 직접 모니터링 메커니즘은 "호스트 기반" IDS에 해당한다. 최근에는 완전한 모니터링을 위해서 호스트 기반과 네트워크 기반 컴포넌트 둘 다를 사용하는 추세이다.

4.3 수집 장치

가. 센서(sensor)

센서는 엔진(engine)이라고도 하며, 고속 네트워크 상의 대량의 트래픽을 감시하기 위한 설치 가능한 소프트웨어 혹은 어플라이언스(appliance)-기반 기술이다. 센서는 네트워크의 특정 위치에 놓여진다. 센서는 처리기-집중 장치이며 일반적으로 정확하게 동작하기 위하여 자체적인 PC나 어플라이언스를 요구한다. 어플라이언스는 입출력(I/O) 선택 사항, 처리 속도 및 메모리가 목적에 따라서 틀리는 특정 컴퓨터이다. 센서는 침입 증거를 찾기 위하여 모든 네트워크 트래픽을 분석하여 네트워크 IDS 정책의 매개변수에 따라서 중앙에 위치한 관리자에게 정보를 보고한다.

센서는 완전한 독립 감시 도구이며 모든 경보와 다른 기록 정보를 지역 하드 드라이브에 저장하며, 실시간으로 경보를 발생한다. 센서와 매니저 사이의 통신은 DES(Data Encryption Standard)와 같은 암호 기술을 사용하여 안전한 터널을 사용한다. 각 매니저는 한 개의 프로세서 시스템에서 수십 개의 센서들을 수용할 수 있다. 모든 센서들이 특정 위치에 있는 것을 보증하기 위하여 혹은 특정 형태가 동일한 정책을 적용받기 위하여 논리적으로 함께 그룹 될 수 있다. 이것을 트리 안에서 센서 그룹이라 한다.

나. 에이전트(agent)

에이전트는 호스트 기반 침입탐지 시스템에서 특정 PC에 설치된 소프트웨어를 말한다. 에이전트 소프트웨어는 일반적으로 작은 분량이며 처리력(processing power)을 거의 소모하지 않는다. 에이전트의 기능은 호스트 상의 특정 파일이나 로그를 감시하여 특정 파일이 접근, 변경, 삭제 혹은 복사될 때 호스트 기반 보안 정책에 따라서 중앙 관리자에게 보고한다. 에이전트는 호스트 상에서 정책 준수를 결정하고 보안 정책에서 위반을 단지 보고하기 때문에 지능적인 소프트웨어라고 여길 수 있다. 호스트 기반 에이전트의 예로는 Tripwire, CyberSafe, AXENT, ISS 등이 있다.

모든 직접 모니터링 방법은 호스트 기반이다. 호스트의 직접 모니터링은 다음 정의에 따라 외부 혹은 내부 에이전트를 사용하여 수행된다[9].

• 외부 에이전트(external agent)
호스트에서, 한 조각의 소프트웨어가 컴포넌트(하드웨어, 소프트웨어)를 관찰하여 IDS에게 유용한 데이터를 보고한다. 컴포넌트와 코드는 분리하여 구현된다.

• 내부 에이전트(internal agent)
호스트에서, 한 조각의 소프트웨어가 컴포넌트(하드웨어, 소프트웨어)를 관찰하여 IDS에게 유용한 데이터를 보고한다. 외부 에이전트와 다른 점은 컴포넌트와 코드는 통합하여 구현된다.

내부 에이전트는 소프트웨어 혹은 하드웨어

컴포넌트 안에 구축될 수 있다. 예를 들면, Unix 커널 안에 내장된 프로세스-정보를 수집하는 컴포넌트일 수도 있고 혹은 네트워크 인터페이스 카드의 펌웨어에서처럼 하드웨어 컴포넌트 안에도 구축이 가능하다. 내부 에이전트는 모니터 되는 컴포넌트의 소스 코드의 부분이다. 내부 에이전트는 이미 존재하는 프로그램에 추가될 수 있고, 이 경우 소스 코드 계측의 경우로 고려된다. 이상적으로, 내부 에이전트는 변경을 하고 에러를 고치는 비용과 노력이 적게 드는 프로그램의 개발동안 추가되어야 한다. 프로그램의 어떤 부분도 IDS에 의하여 사용되는 데이터를 제공하는 한 내부 에이전트로 간주된다. 직접 데이터 수집을 위한 외부 에이전트와 내부 에이전트는 다른 강점과 약점을 가진다. 그러므로 침입 탐지 시스템에서 함께 사용될 수 있다. 다음은 내부 에이전트와 외부 에이전트의 장단점을 기술한 것이다[10].

1) 내부 에이전트

- 장점:
  - 정보의 생성과 사용 사이에서 최소의 지연
  - 침입자의 추적을 은폐하기 위한 데이터 수정이 불가능
  - 분리 처리되지 않기 때문에 쉽게 불능화 혹은 수정 불가능
  - 네트워크 트래픽과 처리 부하의 감소
  - 단일 호스트에 많은 에이전트를 반영 가능
  - 모니터링하는 프로그램의 부분으로 구현되기 때문에, 필요한 어떤 정보에도 접근 가능

- 단점:
  - 구현은 모니터 되는 프로그램 소스 코드에 접근이 요구됨
  - 모니터 되는 프로그램에 대하여 수정을 요구하기 때문에, 구현이 어려움
  - 모니터 할 프로그램과 같은 언어에서 구현될 필요
  - 만약 부정확하게 구현 혹은 설계된다면, 속하는 프로그램의 기능 혹은 성능을 심하게 손상 가능
  - 다른 운영체제에서 갱신, 수정 그리고 이식이 어렵고, 혹은 같은 프로그램의 다른 버전에서조차 어렵다
  - 감소된 이식성

2) 외부 에이전트

- 장점:
  - 호스트로부터 쉬운 변경, 추가 혹은 삭제
  - 어떤 프로그래밍 언어에서도 구현 가능

- 단점:
  - 데이터의 생성과 사용 사이에 지연이 존재
  - 에이전트가 정보를 획득하기 전에 침입자에 의해 변경 가능
  - 침입자에 의해 불능화 혹은 수정 가능.
  - 연속적으로 수행되기 때문에 성능에 영향
  - 정보에 대한 제한된 접근(유닉스 명령어, 시스템 호출)

5. 정보 보고 및 분석

5.1 정보 보고의 형태

현재 IDMEF의 메시지로는 두 가지가 정의되어 있다; Alert과 Heartbeat.

1) 경보(alert) 클래스

일반적으로 분석기(analyzer)가 조사하도록 배치되어 있는 어떤 이벤트를 탐지할 때마다, 자신의 매니저에게 경보 메시지를 보낸다. 경보 메시지는 단일 탐지 이벤트 혹은 복수의 탐지 이벤트

일 수 있다. 경보는 외부 이벤트에 대응하여 비동기적으로 발생한다. 현재 경보는 다음과 같이 세 가지로 분류된다[11].

- ToolAlert 클래스: 공격 도구 혹은 트로이 목마 같은 악성 프로그램의 사용에 관련되는 추가적인 정보를 가지며, 이러한 도구들을 식별할 수 있을 때 분석기에 의하여 사용될 수 있다.
- CorrelationAlert 클래스: 경보 정보의 상호 관련(correlation)에 관련되는 추가적인 정보를 가진다. 한 개 이상의 이미 전송된 경보를 함께 그룹하기 위함이다.
- OverflowAlert 클래스: 버퍼 오버플로 공격에 관련되는 추가적인 정보를 가진다. 분석기로 하여금 오버플로 공격 자체에 대한 상세한 내용을 제공하도록 하기 위함이다.

2) Heartbeat 클래스

매니저에게 분석기의 현재 상태를 나타내기 위하여 사용된다. Heartbeat는 정기적인 기간에 전송되도록 되어있다. 분석기로부터의 Heartbeat 메시지의 정기적인 수신은 분석기가 현재 운영중임을 매니저에게 나타내며, 메시지가 없을 경우 분석기 혹은 네트워크 연결이 실패되었다는 것을 지시한다.

3) 이벤트 정보

이벤트 스트림의 종류로는 운영 체제 감사 레코드, 네트워크 트래픽, 응용 로그, 시스템 콜 등과 같은 여러 가지 형태가 있다.

5.2 상호 관련 및 분석

현재의 보안 시스템은 시스템 간의 상호 운용성이 부족하여 대규모 망에서 효과적인 침입 탐지를 수행하는데 어려움이 있다. 이에 따라 대규모 분산 시스템에서의 침입 탐지 시스템 사이의 정보 교환 등에 대한 기술 개발이 절실히 요구된다. 다음은 침입 탐지 정보를 상호 관련함으로써 침입 탐지를 효율적으로 수행할 수 있는 몇 가지 예를 보여준다[12]:

침입 탐지를 위한 대표적인 정보는 소스, 목적지, 공격의 유형 등을 포함한다. 도메인 간에서 전송되는 이런 정보의 상호 연관의 대표적인 경우는 다음과 같다.

- 경우 1: 동일한 소스, 동일한 목적지 및 동일한 경보 등급에 속하는 경보. 이 경우는 단일 웹 서버에 대하여 연속적인 웹 서버에 대한 공격을 시도하는 공격자의 경우이다.
- 경우 2: 동일한 소스 및 목적지를 가진 경보. 이 경우는 목적지에서 이용 가능한 여러 가지 서비스에 대한 연속적인 공격을 탐지할 수 있다.
- 경우 3: 동일한 목적지와 동일 경보 등급에 속하는 경보. 이 경우는 단일 목적지에 대하여 분산 공격을 탐지하기 위하여 사용된다.
- 경우 4: 동일한 소스와 동일 경보 등급에 속하는 경보. 이 경우는 DNS(domain name server) 집합에 대하여 연속적인 공격을 시도하는 공격자를 발견하기 위하여 사용된다.
- 경우 5: 동일한 목적지를 가진 경보. 이 경우는 분산 공격을 탐지하기 위하여 사용된다.
- 경우 6: 동일한 소스를 가진 경보. 이 경우는 다양한 목적지에 대하여 여러 가지 공격을 수행하는 한 공격자를 탐지하기 위하여 사용된다.
- 경우 7: 동일한 공격 등급을 가진 경보. 이 경우는 최근 해커 메일링 리스트에 게시된 신규 공격 등이 개시되는 상황이다.

6. 정책기반 IDS 모델링 및 성능평가

6.1 시뮬레이터 설계와 모델 구현

정책기반 IDS 모델 구현과 성능평가를 위해서 시뮬레이터 설계와 구현에는 OPNET Modeler를 사용하였다. 그림 4는 정책기반 프레임워크를 이용하여 설계한 정책기반 IDS 평가 모델이다.

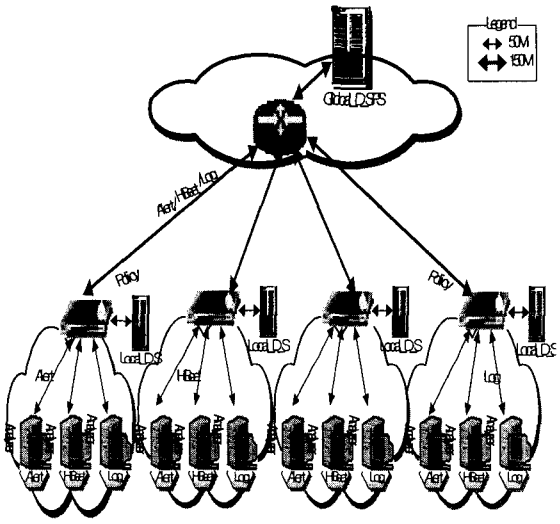


그림 4. 정책기반 IDS 평가 모델

정책기반 IDS 평가 모델은 전역적인 침입 탐지를 수행하기 위해서 지역적인 도메인(Local_Domain)과 상위의 전역적인 도메인(Global_Domain)으로 계층적인 구성을 하였다. 지역적인 도메인에는 각 분석기(Analyzer)와 지역적인 도메인 서버(Server)로 구성되며 전역적인 도메인은 보안정책서버(Security Policy Server)로 구성된다. 지역적인 도메인의 분석기에는 에이전트들이 있다. 각 에이전트들은 보안정책이 관장하는 침입을 분석하여 경보(Alert)와 현 분석기의 상태를 정기적으로 알리는 정보(Heartbeat) 그리고 로그에 관한 이벤트 정보(Log)를 상위의 매니저인 지역적인 도메인 서버로 보고한다. 지역적인 도메인 서버는 분석기로부터 보고받은 탐지 정보를 처리 후, 전역적인 분석을 위하여 최상위의 전역적인 도메인에게 보고한다. 전역적인 도메인은 하위의 매니저인 지역적인 도메인 서버들로부터의 보고 받은 탐지 정보를 보안정책서버(SPS)의 최종 분석을 통하여 정책을 다시 하위의 지역적인 도메인으로 또한 각 분석기에게 정책으로 하달한다.

본 논문에서는 정책기반 IDS 성능평가를 위해서 시뮬레이터 모델을 구현한다(그림 4참조). 각 분석기는 경보 메시지와 로그에 관한 스트림을 비동기적으로, 그리고 분석기의 현재의 상태를 알려주는 HeartBeat 메시지를 동기적으로 상위의 매니저에게 보고하도록 하였다. 표 1은 분석기에서 매니저로 전달되는 이벤트 메시지의 크기이다.

<표 1> 이벤트 크기(단위: 바이트)

이벤트	Alert	HeartBeat	Log
크기(BYTE)	512	512	440

시뮬레이터 구현에서 각 노드 이벤트 전송 율은 양방향으로 설정하였으며 지역적인 도메인 내 연결은 50Mbps 그리고 전역적인 도메인과의 연결은 150Mbps를 통하여 연결하였다.

6.2 모의실험 및 성능분석

모의실험과 성능분석에서는 정책기반 IDS 평가모델을 대상으로 개발된 시뮬레이터를 이용한 성능 모의실험 결과를 제시하고 분석한다. 경보 전달과 정책 전달 과정은 독립적으로 모의실험하였다. 먼저 보안정책에 따라 각 분석기가 침입을 탐지한 후 전역적인 처리를 위하여 전역적인 도메인의 보안정책서버에게 정보를 전달하는 네트워크 수준의 보고를 성능 분석하였으며, 다음으로 최종적인 보안정책서버의 정책을 지역적인 도메인 내 분석기까지 정책을 하달하는 성능을 분석하였다. 성능분석 파라미터로는 지연(delay)만을 사용하였다.

그림 5와 6은 침입 탐지 정보들을 전역적인 도메인 내 보안 정책 서버에게 보고할 때의 평균 지연 성능을 나타낸다. 그림 5에서 전역적인 도메인과 지역적인 도메인에서 입력 부하 크기(즉, 네트워크 이용률)의 변화에 따른 패킷별 평균 지연을 보여준다.

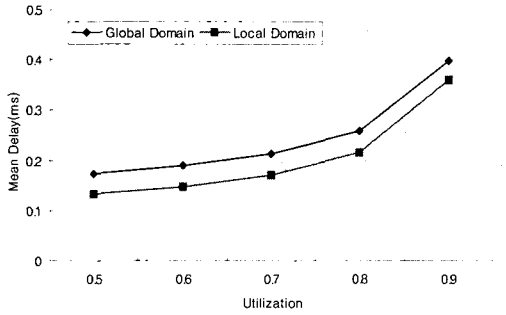


그림 5. 네트워크 이용률에 따른 지연

그림 6은 각 이벤트 종류에 따른 지연 성능을 나타낸다. 경보와 로그 이벤트는 지연이 이용률에 따라 증가하는 추이를 보이며 분석기 상태 정보(HeartBeat)는 네트워크의 상태 변화에서도 정기적인 보고가 이루어지므로 가시적인 지연 성능은 이용률에 따라 경보와 로그의 지연에 비하여 상대적으로 변화가 작은 것으로 분석된다. 그림 7은 상태 정보(HeartBeat)를 좀 더 상세하게 분석한 것이다. 지연이 미세하게 이용률에 따라 증가함을 나타낸다.

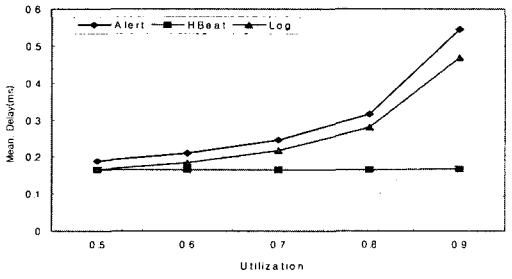


그림 6. 이벤트 종류에 따른 지연

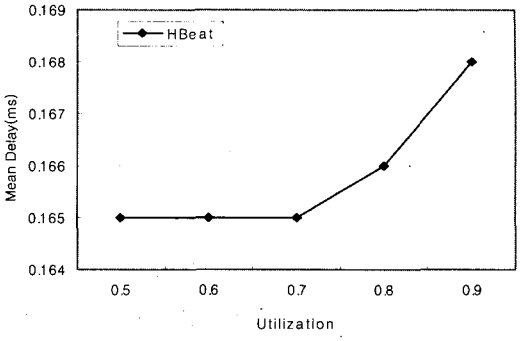


그림 7. HeartBeat 클래스 지연

그림 8은 전역적인 도메인 내 보안 정책 서버의 정책을 이용률에 따라 지역적인 도메인 내 분석기까지 하달하는 지연 성능을 보여준다. 지역적인 도메인과 분석기까지는 네트워크 수준의 지연 성능 차이를 나타낸다.

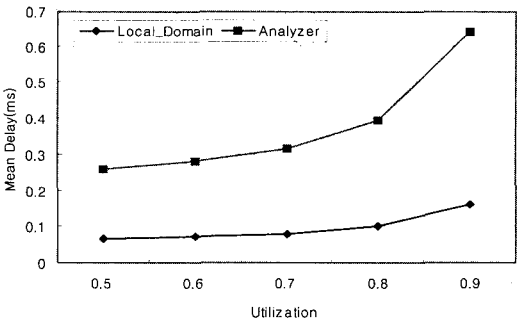


그림 8. 정책 전달 지연

7. 맺음말

본 논문에서는 정책 기반 망에서의 침입 탐지 기술에 대하여 핵심 기술을 분석하고, 다양한 형태의 데이터를 이용하여 에이전트로 구성된 분석기와 매니저인 지역적인 도메인 그리고 전역적인 도메인 내 정책 서버로 구성되는 정책기반 분산 침입 탐지 시스템의 통신 모델을 시뮬레이션을

통하여 성능분석 하였다. 이를 위하여 정책기반 분산 침입 탐지 시스템을 모델링하고 시뮬레이터를 설계하여 시스템의 네트워크 레벨에서 성능평가를 수행하였다. 이벤트 생성 모델과 상위계층의 서비스를 위한 모델을 설계하고 시뮬레이션에서 이용하였다. 다양한 형태의 시나리오를 이용하여 통신 메커니즘의 결정 요인 중의 하나인 지연 성능 인수를 사용하였다.

본 논문에서 제시한 정책기반 분산 침입 탐지 시스템은 보안관리 프레임워크 구조에 의해 정책 도메인 내에서 발생하는 모든 보안 이벤트 정보를 수집하고 이를 체계적으로 관리하여 정책 도메인에 따른 보안상황을 분석하므로 종합적인 네트워크 보안관리가 가능하다[13-19]. 제안된 구조에서 주요한 점은 지역적인 도메인 내 분석기에 의해 탐지되는 각 지역적인 도메인 사이의 관심 전달은 제안된 계층 모델에서 지능적인 방법으로 에이전트들의 협동을 통하여 가능하다는 것이며, 상위 레벨에서는 도메인 간 정보의 수집을 통하여 전역적인 침입 분석이 가능하다는 것이다.

본 논문에서의 정책 기반 분산 침입에 대한 성능 분석은 종합적인 네트워크 보안관리 시스템을 설계하는 데 이용 될 수 있을 것으로 사료되며, 향후 연구 과제로는 본 시뮬레이터 모델에서의 보완점인 결점에 적응 할 수 있는 방안과 관심 전달에 대하여 연구하고 모의 실험하여 성능을 평가하는 것이다.

참고문헌

[1] Madalina Baltatu, Antonio Lioy and Daniele Mazzocchi, "Security Policy System: status and perspective", pp. 278-284.

[2] http://www.ietf.org/html.charters/ipsp- charter .html

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

[4] Dorothy E. Denning, "An Intrusion-Detection Model", IEEE Software pp. 222-232, February 1987.

[5] John McHugh, Alan Christie, and Julia Allen, "Defending Yourself: The Role Of Intrusion Detection Systems", IEEE Software, October 2000.

[6] W. J. Buchannam, Handbook of Data Communications and Networks, Kluwer, 1998.

[7] Peter Mell, Donald Marks, and Mark McLarnon, "A denial-of-resistant intrusion detection architecture", Computer Networks 34, pp. 641-658, 2000.

[8] http://www.snort.org

[9] Thomas E. Daniels and Eugene H. Spafford, "Identification of host audit data to detect attacks on low-level IP vulnerabilities", Journal of Computer Security, 7(1), pp.3-35, 1999.

[10] Diego Martin Zamboni, Using Internal Sensors for Computer Intrusion Detection, Ph.D. dissertation, Purdue University, CERIAS TR 2001-42, August 2001.

[11] IETF Intrusion Detection Working Group, draft-ietf-idwg-idmef-xml-06.txt, Intrusion Detection Message Exchange Format Data Model and

Extensible Markup Language(XML) Document Type Definition, Dec. 2001.

〔12〕 Herve Debar and Andreas Wespi, "Aggregation and Correlation of Intrusion-Detection Alerts", RAID 2001, Springer-Verlag LNCS 2212, pp. 85-103, 2001.

〔13〕 김기영, 장종수, 신영석, "분산 시스템 기반의 보안정책 정보 공유 기술", 한국통신학회지, 제 19 권 제 8호, pp.1184-1192, 2002년 8월.

〔14〕 최종연구보고서, 정책기반의 보안 제어용 분산 프락시 에이전트 기술 개발에 관한 연구, 한국전자통신연구원, 2001 12월.

〔15〕 전용희, 장정숙, 장종수, 손승원, "네트워크 보안 시스템을 위한 에이전트 기술", 한국통신학회지 제 18권 9호, pp.1184-1199, 2001 9월.

〔16〕 장정숙, 전용희, 장종수, 손승원, "침입 탐지를 위한 보안 에이전트에 관한 연구", 제 6회 통신 소프트웨어 학술대회 논문집, pp. 45-49, 2001 7월, 강원도 속초.

〔17〕 장정숙, 전용희, "내부 센서를 이용한 침입탐지 시스템에 관한 연구", 한국정보보호학회 종합학술발표회 논문집, 제 11권 1호, pp.161-165, 2001 11월, 서울.

〔18〕 Jung-Sook Jang, Yong-Hee Jeon, Jong-Soo Jang and Seung-Won Sohn, "A Study on the Application of Mobile Agents for Intrusion Detection System", Proc. of the 4th International Conf. on Advanced Comm. Tech.(ICACT 2002), pp.684-688, Feb. 2002.

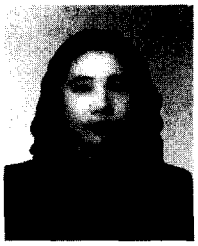
〔19〕 장정숙, 전용희, 장종수, 손승원, "분산 침입 탐지 시스템을 위한 통신 모델", 한국통신학회지, 제 19권 8호, pp. 1168-1183, 2002년 8월.

※ 본 연구는 2001학년도 대구가톨릭대학교 연구비 지원에 의한 것임.

전 용 희

1978년 고려대학교 전기공학과 졸업(공학사)
1985~1987년 미국 플로리다공대 대학원 컴퓨터공학과
1989년 미국 노스캐롤라이나주립대 대학원 Elec. and Comp. Eng. 졸업(공학석사)
1992년 미국 노스캐롤라이나주립대 대학원 Elec. and Comp. Eng. 졸업(공학박사)
1978~1978년 삼성중공업(주) 근무
1978~1985년 한국전력기술(주) 근무
1989~1989년 미국 노스캐롤라이나주립대 Dept of Elec. and Comp. Eng. TA
1989~1992년 미국 노스캐롤라이나주립대 부설 CCSP(Center For Comm. & Signal Processing) RA
1992~1994년 한국전자통신연구원 광대역통신망연구부 선임 연구원
1994~현재 대구가톨릭대학교 컴퓨터·정보통신공학부 교수
2001.3~2003.2 동 공과대학장 역임

관심분야 : 차세대인터넷, 통신망 성능분석, QoS 보장기술, 네트워크 보안



장 정 숙

1991년 경일대학교 공과대학 컴퓨터공학과 졸업(학사)
1992년~1995년 대구가톨릭대학교 교육대학원 전자계산교육전공(석사)
1998년~현재 대구가톨릭대학교 대학원 컴퓨터·정보통신공학 전공 박사 수료

관심분야 : 차세대인터넷, 네트워크 보안, 통신망 성능분석, 고속 통신망 응용 서비스