(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。Int. Cl.
G06F 15/00 (2006.01)

(45) 공고일자 2006년09월18일
(11) 등록번호 10-0623552
(24) 등록일자 2006년09월06일

(21) 출원번호 10-2003-0099103
(22) 출원일자 2003년12월29일

(65) 공개번호 10-2005-0068052
(43) 공개일자 2005년07월05일

(73) 특허권자 한국정보보호진흥원
서울특별시 송파구 가락동 78번지 IT벤처타워 서관

(72) 발명자 김영태
서울특별시중랑구중화동316-14

이호재
경기도용인시수지읍죽전2동832-1용인수지벽산타운2단지207-1701

최중섭
서울특별시도봉구창동주공아파트401-1001

이강신
경기도광주시태전동5차성원아파트506-1104

이홍섭
서울특별시송파구문정동150올림픽훼미리타운226-1004

(74) 대리인 김영철
김 순 영

심사관 : 조우연

(54) 자동침입대응시스템에서의 위험수준 분석 방법

요약

본 발명은 대규모 동적 네트워크 환경에서 컴퓨터 관련 보안을 제공하는 자동침입대응시스템에서의 위험수준 분석 방법에 있어서, IDMEF 데이터 모델을 이용하여 침입 탐지 정보를 분류하는 단계(a); 위험평가 지식 베이스를 구축하는 단계(b); 상기 지식 베이스 내의 규칙들을 학습하는 단계(c); 및 상기 지식 베이스로부터 외부 공격의 위험도를 분류하는 단계(d)를 포함하는 것을 특징으로 하는 위험수준 분석 방법에 관한 것이다. 상기 위험수준은 침입탐지 정보, 취약성 정보, 네트워크 대역폭, 시스템의 성능과 중요도, 및 공격의 빈도 등에 의하여 결정된다.

본 발명에 따른 방법은 다양한 침입탐지 정보와 정보시스템의 취약성을 통합 관리함으로써 사이버 공격에 대한 정보시스템의 위험수준을 자동적으로 측정할 수 있다.

대표도

도 5

명세서

도면의 간단한 설명

도 1은 본 발명에 따른 분석 방법이 적용되는 자동침입대응시스템을 도시한 것이고,

도 2는 자동침입대응시스템이 효과적인 보안 및 대응 정책을 설정하기 위한 구성 컴포넌트들의 상호 연동을 도시한 것이며,

도 3은 자동침입대응시스템의 동적 대응(Dynamic Response)의 기본모델을 도시한 것이고,

도 4는 위험분석 메커니즘 과정을 도시한 것이며,

도 5는 정보시스템의 위험수준을 측정하기 위한 동작과정을 도시한 것이고,

도 6 및 도 7은 mstream DDoS 공격이 발생했을 때, 침입탐지시스템이 생성한 침입탐지 정보를 파싱하여 얻은 IDMEF 클래스의 최상위 클래스 및 상세 클래스를 도시한 것이며,

도 8은 침입탐지 환경 및 기술에 따라 다양하게 생성된 탐지정보를 도시한 것이고,

도 9는 IDMEF 데이터 모델의 기본적인 구성을 도시한 것이며,

도 10은 IDMEF 데이터 모델의 상세 구조를 도시한 것이고,

도 11은 침입탐지 정보와 취약성 정보를 표현하는 위험평가 지식베이스의 규칙의 예이며,

도 12는 에이다부스트 알고리즘이고,

도 13 내지 도 16은 본 발명에 따른 위험분석 방법에서 지식베이스의 규칙들을 학습하는 기법으로서, C.4.5, Decision Stump, IB1, PART, 및 나이브 베이즈를 사용한 경우의 분류에러율, 분류속도, 리콜, 및 결정 결과를 도시한 것이다.

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 대규모 동적 네트워크 환경에서 컴퓨터 관련 보안을 제공하는 자동침입대응시스템에서의 위험수준 분석 방법에 있어서, IDMEF 데이터 모델을 이용하여 침입 탐지 정보를 분류하는 단계; 위험평가 지식 베이스를 구축하는 단계; 상기 지식 베이스 내의 규칙들을 학습하는 단계; 및 상기 학습된 지식 베이스로부터 외부 공격의 위험도를 분류하는 단계를 포함하는 것을 특징으로 하는 위험수준 분석 방법에 관한 것이다. 상기 위험수준은 침입탐지 정보, 취약성 정보, 네트워크 대역폭, 시스템의 성능과 중요도, 및 공격의 빈도 등에 의하여 결정된다.

네트워크 상의 공격(attack)에 대응하는 자동침입대응시스템(Automatic Intrusion Response System)에 대한 현재까지의 연구는, i) 방화벽, 라우터, 및 침입방지시스템(IPS, Intrusion Prevention System) 등 보안 컴포넌트들과의 연동, ii) 침입탐지시스템(IDS, Intrusion Detection System) 내 단순대응기능의 내장, 또는 iii) IDIP(Intrusion Detection Isolation Protocol) 또는 CIDF(Common Intrusion Detection Framework) 등의 침입탐지 및 대응 프로토콜 등에 대한 연구로 크게 나누어 볼 수 있다.

그러나, 다양한 보안 컴포넌트들이 제공하는 대응기능은 지역적 탐지에 의한 지역수준에서의 소극적 대응만을 지원하므로, 대규모 분산 네트워크 환경에서 효율적이면서 다양한 대응 메커니즘을 제공하지 못한다는 단점이 있다.

예를 들어, 첫째로, 현존하는 침입탐지시스템은 방대한 양의 오경보를 생성한다는 문제가 있다. 이와 같은 방대한 양의 오경보는 대부분의 분석시스템의 처리 단계에서 많은 시간을 소요하게 하므로, 신속한 대응이 곤란하다. 따라서, 자동침입시스템은 실제 경보중에서 심각한 수준의 공격과 위험한 공격자를 구별해야 할 필요가 있다.

둘째로, 현존하는 침입탐지시스템을 효율적으로 관리하기 위해서는 특별한 노력이 필요하다. 즉, 새로운 공격이 발견될 때마다 침임탐지 패턴을 작성하거나 갱신하여야 하며, 주기적인 로그(log) 분석을 통하여 위협요소가 존재하는지 여부를 파악하여야 한다. 따라서, 대규모 네트워크 영역을 대응영역으로 간주하고, 그에 적합한 보안 및 대응 정책을 결정함으로써 보안관리자의 관리 부담을 감소시키는 것이 바람직하다.

셋째로, 공격 수법이 다양화되고 지능화되면서, 변형공격 및 새로운 공격 등이 계속해서 발견되고 있으나, 새로운 침입탐지 정보에 대해서 다양한 대응을 지원할 수 있는 다양한 효율적인 메커니즘이 제시되어 있지 않다.

넷째로, 대부분의 보안시스템들은 주로 지역적인 영역에 대해서만 보안 및 대응 정책을 지원하고 있다. 따라서, 인터넷의 활성화로 인하여 네트워크 사용범위가 확대되고 있는 시점에서, 대규모 네트워크에서의 적절한 대응 정책을 적용시킬 필요가 있다. 즉, 획일화되고 단순한 대응방법이 아니라, 보안요구 수준 및 위험수준에 따라 적절히 판단하여 서로 다른 수준의 대응 정책을 지원하는 것이 바람직하다.

발명이 이루고자 하는 기술적 과제

본 발명은 상기한 바와 같은 문제점들을 해결하기 위하여 제안된 것으로서, 본 발명에 따른 분석 방법을 사용하는 경우, 사이버 공격에 대한 정보 시스템의 위험 수준을 자동적으로 측정할 수 있기 때문에, 공격에 대하여 적절하게 대응할 수 있다.

따라서, 본 발명의 목적은 자동침입대응시스템에서의 위험분석 방법을 제공하기 위한 것이다.

발명의 구성 및 작용

본 발명은 대규모 동적 네트워크 환경에서 컴퓨터 관련 보안을 제공하는 자동침입대응시스템에서의 위험수준 분석 방법에 있어서, IDMEF 데이터 모델을 이용하여 침입 탐지 정보를 분류하는 단계(a); 위험평가 지식 베이스를 구축하는 단계(b); 상기 지식 베이스 내의 규칙들을 학습하는 단계(c); 및 상기 학습된 지식 베이스로부터 외부 공격의 위험도를 분류하는 단계(d)를 포함하는 것을 특징으로 하는 위험수준 분석 방법에 관한 것이다.

본 발명에서는 위험분석 메커니즘의 효율성과 정확성을 지원하기 위하여, 다양하고, 이질적인 침입탐지 정보에 대한 호환성과 확장성을 지원하는 IDMEF 데이터 모델을 이용하고; 침입탐지 정보와 시스템의 취약성을 위험도에 따라 효율적으로 학습하고, 분류하기 위한 고수준의 위험평가 지식베이스를 구축하며; 상기 지식 베이스 내에 저장되어 있는 규칙들을 학습하기 위하여 C4.5 기계학습 기법을 이용하고; 상기 규칙들을 분류하기 위하여 에이다부스팅(Adaboosting) 메타 학습기법을 이용한다.

이하에서는, 도면을 참조하여 본 발명에 따른 위험분석 방법을 구체적으로 설명한다. 그러나, 본 발명이 하기 실시예에 의하여 제한되는 것은 아니다.

본 발명에 따른 위험수준 분석방법이 적용되는 자동침입대응 시스템은 대응계층(Response Layer) 및 협조계층(Correlation Layer)의 두 계층으로 이루어진다. 도 1은 자동침입대응시스템을 도시한 것이다. 상기 대응계층은 침입탐지시스템 등과 같은 침입탐지 정보생성 부분(D), 대응방법 결정부분(IRA, Intelligent Response Agent), 및 대응실행 부분(미도시)으로 구성된다. 공격에 의한 침입탐지 정보가 발생하는 경우, 일차적으로 대응을 실행하거나 협조계층에서 찾아낸 최적 대응을 실행하는 기능을 수행한다.

IRA는 침입탐지시스템에서 탐지한 외부의 공격에 대하여 어떻게 대응할 것인지를 결정한다. 상기 대응의 결정은 기존의 침입탐지 및 대응 정보에 대한 학습, 침입탐지 정보의 위험도(공격의 강도 및 의도), 정보시스템의 위험 수준, 및 현재 시스템의 방어 수준 등에 의하여 결정된다. 결정된 대응은 어떠한 대상에 대하여 어떠한 종류의 대응을 수행할 것인가에 대한 메타 정보로 표현된다.

협조계층은 지역대응조율부(LDC, Local Domain Coordinator) 및 전역대응조율부(GDC, Global Domain Coordinator)로 구성된다. LDC는 대응계층의 침입탐지 정보, 대응 정보, 및 주변의 상황 정보를 참조하여 이미 내려진 대응을 해제하거나

강화하는 방법으로 대응을 최적화하는 기능을 수행한다. LDC의 관리영역은 LDC에 설정된 관리영역(통상적으로, 물리적 네트워크 세그먼트로서 지역적인 보안 도메인을 표시한다)으로 제한된다. 또한, LDC에 의하여 행해진 대응과 관련된 정보는 GDC로 전달된다. GDC 및 LDC는 대규모 분산 네트워크 환경에서의 전역 상황을 고려하여 최적화하는 기능을 수행한다.

대응계층과 협조계층으로 구성되어 있는 자동침입대응시스템은 사이버 공격에 대하여 효율적인 보안·대응 정책을 설정할 수 있다. IRA는 지역보안 도메인과 자신에 대한 공격에 대하여 빠른 대응을 수행하며, 수행된 대응이 적절한 대응인지 아닌지는 LDC와 GDC를 통해 결정된다. 또한, 네트워크에 새로운 정보시스템 또는 네트워크가 설치되는 경우, 구성정보를 LDC와 GDC에 등록함으로써 전역 보안 도메인을 효율적으로 관리할 수 있다. 즉, IRA, LDC, GDC가 정보 시스템, 지역 보안 도메인, 전역 보안 도메인을 서로 분리하여 각각 관리하기 때문에 새로운 정보시스템 또는 네트워크가 추가되더라도 전체 보안 네트워크에 미치는 영향은 매우 적다.

한편, 상기 자동침입탐지시스템은 침입탐지시스템(Host/Network IDS Generator), 방화벽(BC, Boundary Controller), 및 관리도구(Manager) 등을 구성 컴포넌트로서 더 포함한다. 도 2는 자동침입대응시스템이 효과적인 보안 및 대응 정책을 설정하기 위한 구성 컴포넌트들의 상호 연동을 도시한 것이다.

이하에서는, 상기 자동침입탐지시스템의 동적 대응 과정을 설명한다.

도 2에 도시되어 있는 바와 같이, IRA의 주요 기능인 지식기반의 동적 대응 메커니즘(Dynamic Response Mechanism)은 대규모 네트워크환경에서 사이버 공격에 대한 동적보안 및 대응정책을 지원한다.

이러한 동적 대응(Dynamic Response)의 기본모델은 도 3에 도시되어 있는 바와 같이, IDMEF 모델 및 위험분석 모델을 통하여 다양한 침입탐지 환경에서 보고된 침입탐지 정보의 위험도와 시스템의 취약성을 분류하고, 적절한 보안 및 대응 정책을 결정하며, 실시간으로 지역적인 대응을 실행한 후, 손상된 중요 데이터에 대하여 손실 측정 및 복구를 진행하는 과정으로 이루어진다. 상기 동적 대응 모델은 IDMEF 데이터 모델, 위험분석 모델, 보안·대응정책, 동적대응 선택, 대응·평가, 및 손실측정·복구로 구성되어 있다.

상기 IDMEF 데이터 모델은 침입탐지시스템, 대응시스템 및 관리시스템 사이의 정보를 공유하기 위한 데이터 형식과 교환 절차들을 정의한다. IDMEF 모델은 모든 탐지정보에 대하여 표준화된 표현을 제공하며, 침입탐지시스템의 탐지환경 및 능력에 따라 간단하고 복잡한 침입탐지 정보를 함께 표현할 수 있도록 설계되었다.

상기 위험분석 모델은 침입탐지 정보를 IDMEF 클래스들로 분류한 후, 이 클래스들을 바탕으로 구축된 위험평가 지식베이스로부터 공격의 위험도(공격 강도와 공격 의도)를 분류한다. 그리고 분류된 공격의 위험도를 바탕으로 공격의 빈도, 시스템의 중요도, 주위의 위험 상황 등을 적용함으로써 정보시스템의 위험수준를 측정한다. 상기 모델은 위험평가 지식베이스 내에 저장되어 있는 침입정보와 취약성 정보에 대한 규칙을 학습하고 분류하기 위하여 C4.5 기계학습(Machine Learning) 기법을 이용하며, 학습 데이터에 대한 분류의 정확성을 높이기 위하여 에이다부스트(AdaBoost) 메타-학습기법을 이용한다.

상기 보안 및 대응정책은 대규모 네트워크 환경에서 중요 시스템과 네트워크를 보호하기 위하여 보안 관리자에 의하여 관리되며, 동적대응 선택 메커니즘에 의해 자동적으로 변경될 수 있다.

동적대응 선택 알고리즘은 상기 보안 및 대응정책을 기반으로 위험분석 모델에서 분류된 정보시스템의 위험수준과 IDMEF 클래스들을 분석함으로써 적절한 보안수준과 대응수준(대응모듈, 대응방법)을 선택한다.

대응 및 평가는 보안 및 대응 정책의 실행을 담당하며, 이에 대한 정책보안 수준과 대응 수준의 적절성, 침입탐지시스템의 정확성, 위험분석 모델의 정확성 등을 평가함으로써 지능적이고 고성능의 자동침입대응시스템을 관리 및 유지하는데 이용된다.

손실 측정 및 복구는 악의적인 파일 혹은 프로세스의 갱신 및 삭제가 발생했을 경우, 정보시스템의 손실을 측정하고 손상된 파일이나 프로세스 등을 복구한다. 이 기능은 침입탐지시스템으로부터 이벤트가 발생하지 않더라도, 독자적이면서 주기적으로 정보 시스템에 대한 손실을 측정한다.

이하에서는, 상기 자동침입탐지시스템의 동적 대응 과정 중 위험분석 메커니즘을 설명한다.

본 발명에 따른 위험분석 메커니즘은 침입탐지, 네트워크관리, 시스템 성능, 취약성 평가 등의 시스템들에 의해 생성된 다양한 정보를 이용하여 사이버 공격에 대한 공격의 위험도를 분류하고 정보시스템의 위험수준을 측정하는 기능을 담당한다. 도 4는 이러한 기능을 도시한 것이다.

본 발명에 따른 위험분석 방법은 공격의 위험도를 정확하게 분류하기 위하여 2단계 검색 기능을 지원한다. 정보시스템의 위험수준을 측정하기 위한 동작과정은 도 5에 도시되어 있는 바와 같다.

먼저 전처리기에서는, 다양한 침입탐지시스템에서 XML 형식으로 생성된 침입탐지 정보(IDMEF 메시지)를 수신받아, IDMEF 클래스별로 파싱을 진행한다. 이때, 수신받은 메시지의 파싱은 XML 라이브러리 내에 있는 DOMParser() 함수를 이용한다. 도 6 및 도 7은 mstream DDoS 공격이 발생했을 때, 침입탐지시스템이 생성한 침입탐지 정보를 파싱하여 얻은 IDMEF 클래스를 Internet Explorer 6.0 프로그램에서 오픈한 것이다.

이후 IDMEF 클래스 내에서 취약점 식별자의 존재여부를 확인한다. 이 때, 상기 파싱된 IDMEF 클래스 중에서 분류(Classification) 클래스 내에 "CAN-2000-0138"이 존재하는 지를 도 7에 도시되어 있는 바와 같이 검사한다. 이 과정은 현재의 공격이 알려지지 않은(unknown) 공격인지를 판단하는 과정이다. 알려지지 않은 공격인 경우(즉, 취약점 식별자가 존재하지 않는 경우), 위험평가 모듈이 수행된다. 반면에, 이미 알려진(known) 공격인 경우(즉, 취약점 식별자가 존재하는 경우), 공격 DB 검색모듈이 수행된다. 위험평가 모듈과 공격 DB 검색모듈은 공격강도와 공격의도를 나타내는 공격의 위험도를 분류 또는 검색한다.

위험평가 모듈은 파싱된 IDMEF 클래스들과 취약성 데이터베이스 정보를 이용하여, 이미 구축된 위험평가 지식베이스로부터 공격의 위험도를 분류하고, IDMEF 클래스들과 공격의 위험도를 이용하여 학습을 진행한다. 동시에 위험수준 결정모듈로 분류결과를 전달한다.

상기 학습과정에서는 C4.5 알고리즘을 이용하는 것이 바람직하다. 상기 분류과정에서는 분류의 정확성을 높이기 위하여 C4.5를 여러번 수행시킬 수 있는 에이다부스트(AdaBoost) 알고리즘을 이용하는 것이 바람직하다.

이후, 보안 관리자에게 알려지지 않은 탐지정보에 대한 분류 결과를 제공한다. 보안 관리자는 위험평가 모듈에서 보고된 정보, 공격 DB 분석, 손실 측정 결과 등을 바탕으로 취약성 식별자를 등록한 후, 공격 DB에 등록한다.

공격 DB 검색모듈은 IDMEF의 분류 클래스에 존재하는 취약성 식별자를 이용하여 공격 DB를 검색한다. 검색결과가 없는 경우, 위험평가 모듈이 수행하게 된다. 검색결과가 존재하는 경우, 위험수준 결정 모듈로 검색 결과를 전달한다.

위험수준 결정 모듈은 공격의 위험도, 네트워크 트래픽 양, 시스템 성능, 시스템 중요도, 동일 공격의 빈도 등에 관한 정보를 이용하여 정보시스템의 위험수준을 결정한다.

이와 같이, 본 발명에 따른 위험분석 메커니즘이 적용되는 시스템은, 공격자의 공격 강도와 정보시스템의 취약성 및 위험도를 자동적으로 분석할 수 있어 위험도 기반의 보안 정책과 대응 정책을 지원하는데 도움을 제공할 수 있다.

이하에서는, 침입탐지 정보에 대한 공격의 위험도를 분류하고 학습하는 기능을 담당하는 위험평가 모듈에 대하여 상세하게 설명한다.

대부분의 침입탐지시스템들은 동일 공격에 대해서도 탐지환경 및 탐지기술에 따라 서로 다른 탐지정보를 보고한다. 즉, 알려진 또는 알려지지 않은 모든 공격에 대하여 호스트, 네트워크, 응용 기반의 탐지 환경과 시그너처(signatures), 명세(specification), 비정상행위(anomalies), 및 정책(policy) 기반의 탐지기술 등에 따라 다양하고 이질적인 탐지 정보를 생성할 수 있다. 도 8은 침입탐지 환경 및 기술에 따라 다양하게 생성된 탐지정보를 도시한 것이다.

따라서, 본 발명에서는 다양하고 이질적인 침입탐지시스템 간의 호환성 및 확장성을 향상시키기 위하여 IETF(Internet Engineering Task Force)에서 현재 표준화가 진행중인 XML 형식을 지원하는 IDMEF(Intrusion Detection Message Exchange Format)를 적용하였다. IDMEF는 의심스러운 이벤트에 대하여 자동화된 침입탐지시스템들에 의해 침입탐지 정보를 표현하는 표준데이터 형식이다. IDMEF 데이터모델은 침입탐지시스템에서 관리시스템으로 보내지는 탐지 정보의 객체 지향적 표현이다.

IDMEF 데이터 모델은 다음과 같은 문제점들을 고려하고 있다.

즉, 탐지 정보는 본래 이질적(heterogeneous)이다는 점(어떤 탐지 정보는 근원지, 목적지, 이름, 및 사건발생 시간 등의 적은 정보만을 표현하지만, 어떤 탐지 정보는 포트 또는 서비스, 프로세스, 및 사용자 정보 등과 같이 보다 많은 정보를 제공한다);

침입탐지 환경이 서로 다르다는 점(어떤 침입탐지 환경에서는 네트워크 트래픽을 분석하여 공격을 탐지하고, 어떤 침입탐지 환경에서는 운영체제 로그 혹은 응용감사(audit) 정보를 이용하기 때문에, 동일한 공격에 대하여 서로 다른 침입탐지 환경에서 보고된 탐지 정보가 반드시 동일한 정보를 포함하는 것은 아니다);

침입탐지시스템의 능력은 서로 다르다는 점(보안 도메인에 따라 다소 적은 탐지 정보를 제공하는 침입탐지시스템을 설치하거나, 보다 많은 탐지 정보를 제공하는 복잡한 침입탐지시스템을 설치할 수 있다);

운영체제 환경이 서로 다르다는 점(공격은 설치된 네트워크 또는 운영체제의 종류에 따라 서로 다르게 관찰되고 보고된다); 및

공급자의 목적이 서로 다르다는 점(다양한 이유로 인하여 공급자들은 자신들이 분류한 공격 유형에 따라 유용하고 적절한 정보를 제공하는 침입탐지시스템을 공급한다)을 고려한다.

따라서, IDMEF 데이터 모델은 모든 탐지 정보에 대하여 표준화된 표현을 제공하며, 침입탐지시스템의 탐지 환경 및 능력에 따라 간단하고 복잡한 탐지 정보를 함께 기술할 수 있도록 설계되었다. 도 9는 IDMEF 데이터 모델의 기본적인 구성을 도시한 것이다.

모든 IDMEF 메시지들의 최상위 클래스는 IDMEF-메시지(Message) 클래스이다. 상기 IDMEF-메시지 클래스의 하위클래스로서 경보(Alert)와 하트비트(Heartbeat)의 두 가지 메시지 유형이 존재한다. 도 10에 도시되어 있는 바와 같이, 각각의 메시지 내에 보다 상세한 정보를 표현하기 위하여, 각 메시지 유형에 대한 하위클래스가 사용된다.

본 발명에서는 사이버 공격에 대한 공격의 강도와 의도를 모두 포함하고 있는 공격의 위험도를 분류하기 위하여 침입탐지 정보와 취약성 정보를 통합 관리할 수 있는 위험평가 지식베이스를 구축한다. 상기 지식베이스에 사용된 속성들은 몇 개의 IDMEF 클래스와 취약성 데이터베이스의 정보로 구성된다. IDMEF 클래스는 Snort NIDS, Arach NIDS 등과 같은 침입탐지시스템의 침입패턴을 참조하며, 취약성 정보는 ICAT 취약성 데이터베이스를 참조한다. 또한, 침입탐지 정보, 취약성 정보, 네트워크 대역폭, 시스템의 성능과 중요도, 및 공격의 빈도 등을 고려한다.

정보시스템의 취약성 정보는 IDMEF의 참조필드 내에서 취약성 식별자인 CVE의 존재 유무에 의해 결정된다. CVE는 침입 유형이 admin, dos, user, file인 경우에만 존재한다. 이는 침입자가 정보시스템의 잠재적인 취약성을 이용하여 정보시스템을 손상시킬 수 있음을 의미한다. 그러나 침입유형이 recon인 경우에는 CVE 번호가 침입탐지 정보의 참조필드에 포함되어 있지 않다. 이는 공격자가 다양한 정보를 수집하기 위한 목적으로 오직 침입의 시도만을 행하며 정보시스템을 손상시키지 않음을 의미한다. 취약성 데이터베이스에서 정보시스템의 손실 유형(Loss_Type), 취약한 시스템의 유형(Exposed_System_Type), 취약한 컴포넌트 (Exposed_Component) 등과 같은 속성들을 추출함으로써 침입탐지시스템이 침입탐지 정보를 발생시킨 원인, 즉 정보시스템 내의 어떤 취약점을 이용하여 공격을 했는지에 대한 침입자에 대한 공격의 의도를 파악할 수 있다.

하기 표 1은 IDMEF의 기본 클래스들과 취약성 데이터베이스의 속성들을 반영한 위험평가 지식베이스를 구성하는 기본적인 속성 리스트들을 나타낸 것이다.

[표 1] 위험평가 지식베이스를 구성하는 기본적인 속성 리스트

속성 이름	필드	설명	자료형
취약성 식별자	CVE_ID	CVE, CAN 번호	실수형
공격 패턴	Attack_Pattern	침입탐지 정보의 패턴	문자열
공격 유형	Attack_Type	공격 강도의 유형 (admin, user, dos, file, recon, other)	문자열
손실 유형	Loss_Type	가용성, 기밀성, 무결성의 위반 여부	문자열
시스템의 취약성 유형	Exposed_System_Type	취약점이 있는 시스템 유형(os, server, application, protocol, encryption, other)	문자열
컴포넌트의 취약성 유형	Exposed_Component	취약점이 있는 시스템 컴포넌트	문자열
공격 위치	Attack_Location	공격이 시작된 위치(local, remote)	문자열
근원지 주소의 위조 여부	Source_Spoofed	근원지 주소의 위조여부 판단 (unknown, yes, no)	문자열
근원지 위치	Source_Location	근원지 IP 주소의 위치(internal, external)	문자열
근원지 프로세스	Source_Process	근원지 시스템에서 실행중인 프로세스	문자열
근원지 프로토콜	Source_Protocol	근원지 시스템에서 사용된 프로토콜	문자열
근원지 포트 번호	Source_Port_Num	근원지 시스템에서 사용된 포트 번호	실수형
목적지 위조 여부	Target_Decoy	목적지 IP 주소의 위조 여부 (unknown, yes, no)	문자열
목적지 위치	Target_Location	목적지 IP 주소의 위치(internal, external)	문자열
목적지 프로세스	Target_Process	목적지 시스템에서 실행중인 프로세스	문자열
목적지 프로토콜	Target_Protocol	목적지 시스템에서 사용된 프로토콜	문자열
목적지 포트 번호	Target_Port_Num	목적지 시스템에서 사용된 포트 번호	실수형
목적지 파일 상태	Target_File_Status	비 인가된 파일의 접근, 생성 및 갱신 등을 결정	문자열
목적지 손상 파일	Target_File	목적지 시스템에서 손상된 파일	문자열
공격의 위험도	Severity	공격의 강도과 취약성을 빨리 판단하는데 사용됨	문자열

상기 표 1에서는 Snort NIDS 및 Arach NIDS의 두 종류의 네트워크 기반의 침입탐지시스템만을 이용하고 있지만, 다른 네트워크 또는 호스트 기반의 침입탐지시스템을 쉽게 추가할 수 있다. 이 때, Source_Process, Target_Process, Exposed_System _Type, 및 Exposed_Component, Target_File 속성은 값을 포함하지 않을 수 있다.

도 11은 침입탐지 정보와 취약성 정보가 위험평가 지식베이스의 규칙들로 어떻게 표현되고 있는지를 도시한 것이다.

전술한 바와 같이, 침입탐지 정보와 취약성 정보를 이용하여 위험평가 지식베이스를 구축하고, 상기 지식베이스는 공격의 위험도를 분류하는데 사용된다.

이하에서는, 알려지지 않은 공격에 대한 침입탐지 정보에 대하여 공격의 강도를 분류하고 학습할 수 있는 C4.5 기계학습 기법과, 분류의 정확성을 높이기 위한 부스팅(boosting) 알고리즘으로 에이다부스트 메타-학습 기법에 대하여 설명한다.

본 발명에 따른 위험평가 방법에서는 기계학습 및 분류에 WEKA 라이브러리의 J48 알고리즘을 사용하였다. J48은 ID3 이후에 나온 C4.5 결정 트리 알고리즘을 자바(JAVA) 언어로 동일하게 구현한 것이다. WEKA에서 지원 가능한 알고리즘으로는 결정 트리, k-nearest neighbor, 나이브 베이즈(naive bayes), 및 assocision rules 등이 있다.

상기 C4.5 기법은 결정 트리를 구축하여 학습과 분류를 진행하므로, 결정 트리 알고리즘에 속한다. 결정 트리 알고리즘은 결과를 분류할 수 있는 최적의 트리를 생성하는 것이 목적이며, 최적의 트리를 생성하기 위해서는 속성을 선정하는 순서가 중요하다. 속성의 선정 순서에 따라 트리의 구성도가 바뀔 수 있고, 트리의 구성도에 따라서 트리의 복잡도가 복잡해질 수도 있고 간단해질 수도 있기 때문이다.

결정 트리 알고리즘은 속성을 선정하는 순서를 결정하기 위하여 정보 이론(Information Theory)을 이용하며, 이는 불확실성(Entropy)과 정보획득 (Information Gain)을 이용한다. 불확실성(Entropy)이란 현재 상태에서 각 종류별 클래스들이 섞여 있는 정도이다. 즉, 여러 종류의 클래스들이 섞여 있을수록 불확실하다고 말할 수 있으며, 섞여있는 각 클래스들의 데이터 수가 비슷할 때 더욱 불확실하다고 할 수 있다. 따라서, 한 종류의 클래스들로 이루어졌을 경우에는 불확실성이 0이며, 두 종류의 클래스로 이루어져 있고 각 클래스의 수가 같을 경우에는 불확실성이 1이다.

하기 수학식 1은 불확실성을 측정하는 수식이다:

[수학식 1]

$$Entropy(S) \equiv \sum_{i=0}^{c}(-p_i \log_2 p_i)$$

상기식에서,

S는 전체 데이터 집합이고,

c는 클래스를 나타내며,

Pi는 전체 데이터 집합 S에 대한 i번째 클래스(c) 집합의 확률이다.

Gain은 현재 상태에서 임의의 속성을 선정하여 데이터를 분류하는 경우, 예측할 수 있는 불확실성이 줄어든 정도이다. 불확실성이 줄어든 정도가 높다는 것은 그 속성을 사용하였을 경우 그만큼 데이터를 명확하게 분류할 수 있다는 것을 의미한다. 따라서, 속성을 선정하기 위해서는 현재 상태에서 각 속성에 대하여 Gain 값을 구하고 그중 Gain 값이 가장 높았던 속성을 먼저 선정하여 데이터를 분리하여야 한다.

하기 수학식 2는 Gain 값을 구하는 수식이다.

[수학식 2]

$$Gain(S,A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

상기 식에서,

S는 전체 데이터 집합을 의미하고,

A는 하나의 속성 이름을 의미하며,

Gain(S,A)는 전체 데이터 집합 S에서 속성 A를 선택하여 분류하였을 경우 얻을 수 있는 불확실성이 낮아지는 정도이고,

v는 속성 A에 속하는 각각의 속성 값들을 의미하며,

Sv는 속성 A의 특정 속성 값 v를 지니는 데이터들의 집합이고,

Entropy(Sv)는 Sv에 대한 불확실성이다.

부스팅 알고리즘은 임의로 주어진 학습 알고리즘의 정확성을 극대화시킬 수 있다. 특히, 이 알고리즘은 에러율(error rate)이 50%보다 약간 낮은 임의의 약성 학습 알고리즘(weak learning algorithm)을 강성 학습 알고리즘(strong learning algorithm)으로 강화시킴으로써 에러율을 최소화시킨다. 또한, 부스팅 알고리즘은 M번의 반복 시도에서 C4.5, Decision Stump, IB1, 나이브 베이즈, 및 PART 등과 같은 다수의 약성 학습 알고리즘을 개별적으로 적용함으로써 분류에 대한 에러율을 최소화시킬 수 있다.

에이다부스트(AdaBoost)의 기본적인 아이디어는 학습데이터 집합에 대하여 분포도(distribution) 또는 가중치 집합을 유지하는 것이다. 즉, 이전에 학습된 약성 분류자(weak classifiers)의 가중치 합을 이용하여 강성 분류자(strong classifier)를 구축하는 것이다. 가중치를 이용하여 새로운 분류자를 학습하는 방법에는 샘플링을 이용한 부스팅(boosting by sampling) 및 가중치를 이용한 부스팅(boosting by weighting)의 두 가지 방법이 있다. 샘플링을 이용한 부스팅에서 학습 인스턴스들은 가중치에 비례하는 가능성(probability)을 지닌 학습데이터 집합으로부터 대체용으로 선출된다. 이 방법은 모든 반복에서 변경되는 과정을 제외하고는 가중치가 배깅(bagging)과 동일한 재샘플링(resampling) 방법이다. 가중치를 이용한 부스팅에서 동일한 학습 데이터 집합은 각각의 반복에서 학습 알고리즘에 주어지며, 가중치는 에러 함수를 최소화하도록 구성하기 위해 직접 사용된다. 본 발명에서는 동일한 데이터 집합을 학습하는 가중치를 이용한 부스팅을 사용하였다.

에이다부스트 알고리즘의 동작과정은 다음과 같다. 먼저 모든 학습 데이터에 대하여 동일한 가중치를 설정한다. 이 알고리즘의 M번의 반복과정은 아래의 단계들에 의해 수행된다:

① 학습데이터와 가중치 분포도에 대하여 약성 또는 기본 학습기(base learner)를 이용하여 기본 분류자(base classifier)를 구축한다. 예를 들어, C4.5, Decision Stump, IB1, PART, Nave Bayes 등을 사용할 수 있다.

② 학습데이터의 집합에서 잘못 분류된 학습 인스턴스들을 결정하고 보다 큰 가중치를 할당한다.

③ N번의 수행 후에 반복을 멈추고 기본 분류자들의 가중치의 합을 출력한다.

도 12는 에이다부스트 알고리즘의 단계별 절차와 가중치 갱신 방법을 요약한 것이다.

이후, 상기 에이다부스트 방법에 따라 외부 공격의 위험도를 분류할 수 있다. 표 2는 위험수준은(Risk Level)을 DOD와 SANS에서 분류한 것이다.

[표 2] 위험 수준의 예

위험수준	설명
Green (정규 활동)	· 뚜렷한 활동이 없음
Blue (증가되는 공격의 위험)	· 일반적인 위협을 가리키는 지시 및 경고 · 의심스럽거나 알려진 CNA(Computer Network Attack) 능력을 가진 잠재적인 적들을 포함하는 지역적인 이벤트 · 정보시스템 조사(probe), 검사(scan) 혹은 감시(survelliance)에 의해 탐지된 활동
Yellow (특정 공격의 위험)	· 지시 및 경고는 특정 시스템, 위치, 단위, 운영을 목표로 한 공격을 지시 · 네트워크 조사(probe), 검사(scan) 혹은 집중적인 조사(reconnaissance)에 의해 탐지된 활동 · 관리 네트워크의 운영에 영향을 미치지 않고 시도된 네트워크의 부당한 접근(penetration) 혹은 DOS
Orange (제한된 공격)	· 제한된 공격을 지시하는 지능공격 평가 · 관리 도메인의 운영에 제한된 영향을 미치는 정보시스템 공격 - 최소한의 성공, 성공적인 방해 공작 - 손상된 데이터 혹은 시스템은 거의 혹은 전혀 없음 - Mission을 성취할 수 있는 단위
Red (일반적인 공격)	· 관리네트워크의 운영에 영향을 미치는 성공적인 정보시스템 공격 · 제기능을 약화시키는 널리 알려진 사건 · Mission failure를 야기시키는 뚜렷한 위험

본 발명에 따른 방법에서 지식베이스의 규칙들을 학습하는 기법으로서, C.4.5, Decision Stump, IB1, PART, 및 나이브 베이즈(Naive Bayes)를 각각 사용하여 실험하여 분류에러율(Error rate), 분류속도(Speed), 리콜(Recall, 총 적합 건수에 대하여 검색된 적합 건의 비), 및 결정(Precision, 검색결과의 전체건수 중에서 검색목적에 적합한 건수의 비)을 비교하였다.

상기 실험에서는 SNORT와 ArachNIDS의 다양한 침입 규칙들과 ICAT 취약성 데이터베이스의 취약성 정보를 결합시켜 각각 50개, 100개, 150개, 200개 및 250개의 학습 데이터를 사용하였다.

분류에러율, 분류속도, 리콜, 및 결정에 대한 실험 결과를 도 13 내지 도 16에 도시하였다. 상기 실험으로부터 알 수 있는 바와 같이, C4.5를 분류 학습기로서 적용하는 경우, 성능이 가장 우수함을 알 수 있다.

발명의 효과

전술한 바와 같이, 본 발명에 따른 위험분석 방법을 사용하는 경우, 다양한 침입탐지 정보와 정보시스템의 취약성을 통합 관리함으로써 사이버 공격에 대한 정보시스템의 위험수준을 자동적으로 측정할 수 있다. 또한, 본 발명에 따른 자동침입대응시스템을 사용하는 경우, 대규모 네트워크 영역을 대응 영역으로 간주하고, 그에 맞는 보안 및 대응 정책을 결정하기 때문에, 보안 관리자의 관리부담을 감소시킬 수 있다.

(57) 청구의 범위

청구항 1.

동적 네트워크 환경에서 컴퓨터 관련 보안을 제공하는 자동침입대응시스템에서의 위험수준 분석 방법에 있어서,

침입 주체의 위험도 및 공격 대상의 중요도를 분석할 수 있도록 IDMEF 데이터 모델을 이용하여 침입 탐지 정보를 그 속성에 따라 공격 정보, 취약성 정보, 근원지 정보 및 목적지 정보로 분류하는 단계(a);

상기 공격 정보, 취약성 정보, 근원지 정보 및 목적지 정보를 저장하기 위한 위험평가 지식 베이스를 구축하는 단계(b);

상기 지식 베이스 내의 규칙들을 학습하는 단계로서, 상기 학습 단계는, 외부 공격의 위험도를 분석하기 위한 기준이 되는 속성을 선정하기 위해 상기 위험평가 지식 베이스에 저장된 공격 정보, 취약성 정보, 근원지 정보 및 목적지 정보에 대한 불확실성(Entropy) 값과 정보획득(Information Gain) 값을 구하는 단계를 포함하는 단계(c); 및

상기 학습된 지식 베이스로부터 외부 공격의 위험도를 분류하되, 상기 공격 정보, 취약성 정보, 근원지 정보 및 목적지 정보 중에서 정보획득 값이 가장 높은 정보를 먼저 선정하여 분류하는 단계(d)를 포함하는 것을 특징으로 하는 위험수준 분석 방법.

청구항 2.

제 1 항에 있어서, 상기 위험수준 분석은 상기 위험수준은 침입탐지 정보, 취약성 정보, 네트워크 대역폭, 시스템의 성능과 중요도, 및 공격의 빈도 등에 의하여 결정되는 것을 특징으로 하는 방법.

청구항 3.

제 1 항에 있어서, 상기 동적 네트워크 환경은 대규모 분산 네트워크 환경인 것을 특징으로 하는 방법.

청구항 4.

제 1 항에 있어서, 상기 IDMEF 데이터 모델에는 상기 자동침입대응시스템에 구비되어 있는 침입탐지시스템, 대응시스템 및 관리시스템 사이의 정보를 공유하기 위한 데이터 형식 및 교환 절차가 정의되어 있는 것을 특징으로 하는 방법.

청구항 5.

삭제

청구항 6.

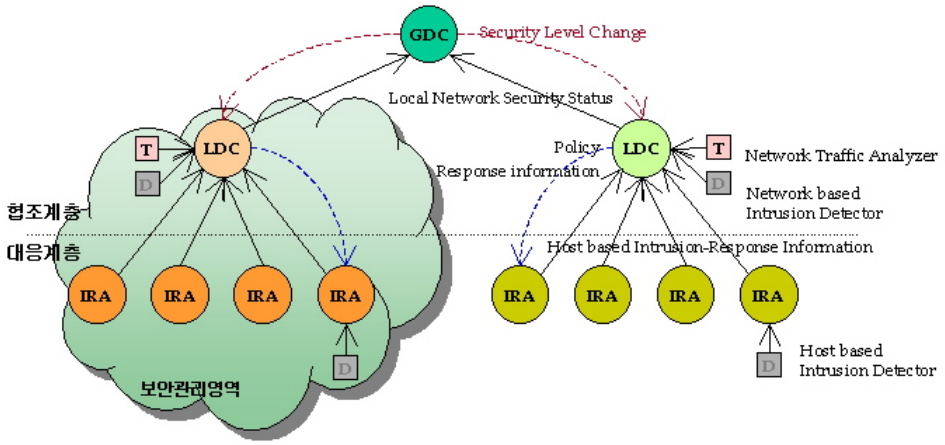
제 1 항에 있어서, 상기 지식 베이스 내의 규칙들을 학습하는 단계(c)는 C4.5 기계학습 기법을 이용하는 것을 특징으로 하는 방법.

청구항 7.

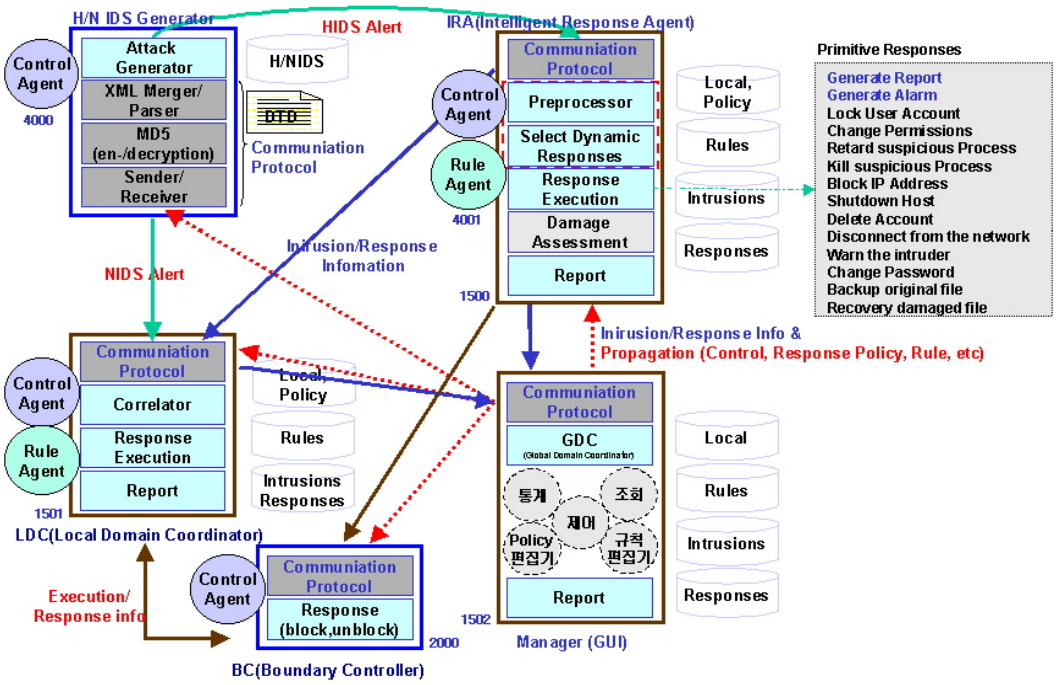
제 1 항에 있어서, 상기 학습된 지식 베이스로부터 외부 공격의 위험도를 분류하는 단계(d)는 에이다부스트 메타 학습 기법을 이용하는 것을 특징으로 하는 방법.

도면

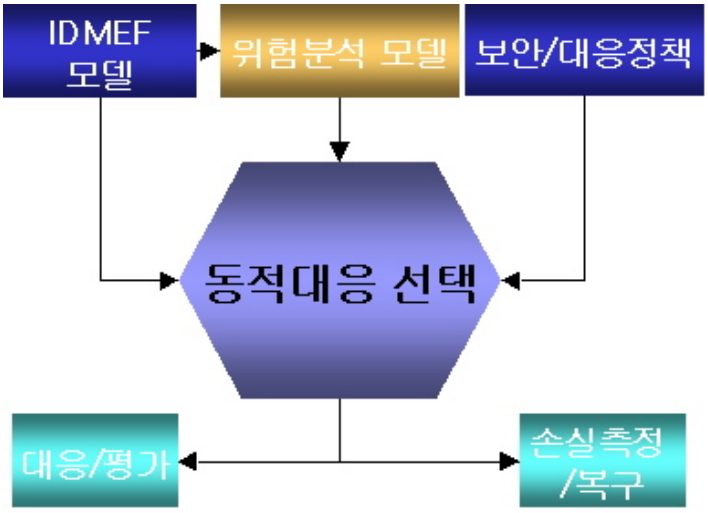
도면1



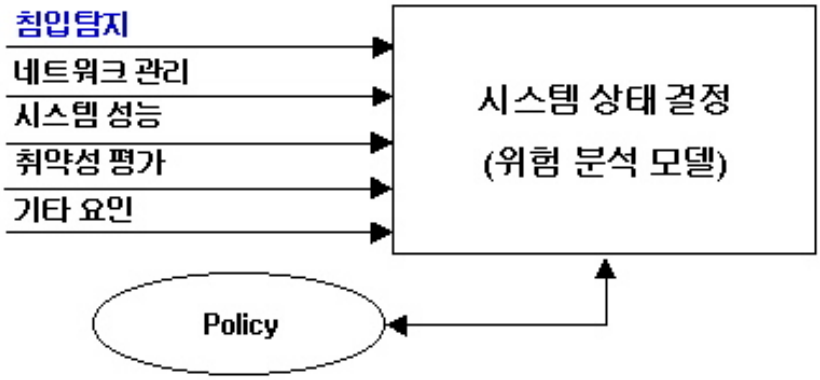
도면2


H/N IDS Generator
HIDS Alert
IRA(Intelligent Response Agent)
Control Agent
Attack Generator
XML Merger/ Parser
MD5 (en-/decryption)
Sender/ Receiver
4000
H/NIDS
DTD
Communiation Protocol
Control Agent
Rule Agent
4001
Communiation Protocol
Preprocessor
Select Dynamic Responses
Response Execution
Damage Assessment
Report
1500
Local, Policy
Rules
Intrusions
Responses
Primitive Responses
Generate Report
Generate Alarm
Lock User Account
Change Permissions
Retard suspicious Process
Kill suspicious Process
Block IP Address
Shutdown Host
Delete Account
Disconnect from the network
Warn the intruder
Change Password
Backup original file
Recovery damaged file
Inirusion/Response Infomation
NIDS Alert
Inirusion/Response Info & Propagation (Control, Response Policy, Rule, etc)
Communiation Protocol
Control Agent
Correlator
Response Execution
Rule Agent
Report
1501
LDC(Local Domain Coordinator)
Local Policy
Rules
Intrusions Responses
Communiation Protocol
GDC
(Global Domain Coordinator)
통계
조회
제어
Policy 편집기
규칙 편집기
Report
1502
Local
Rules
Intrusions
Responses
Manager (GUI)
Execution/ Response info
Control Agent
Communiation Protocol
Response (block,unblock)
2000
BC(Boundary Controller)

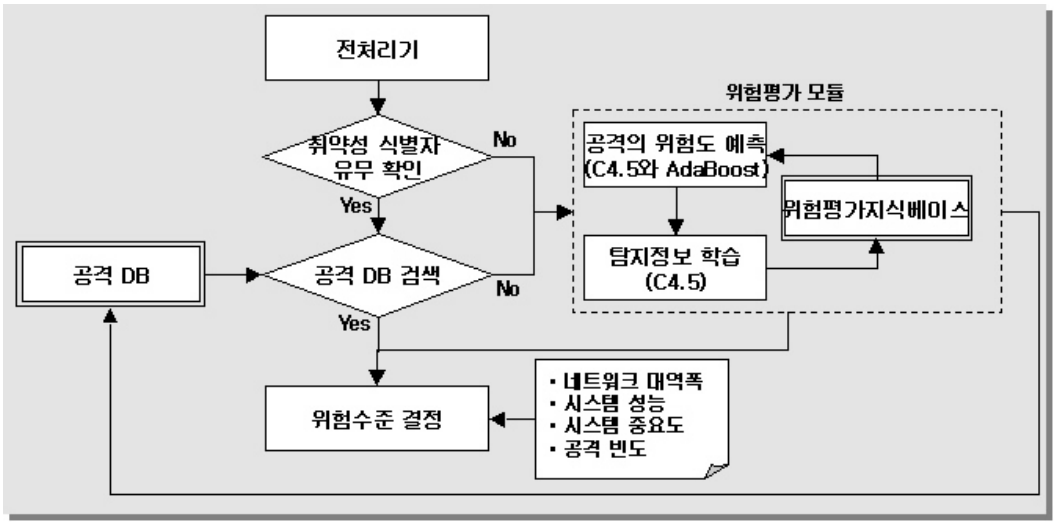
도면3


IDMEF 모델
위험분석 모델
보안/대응정책
동적대응 선택
대응/평가
손실측정 /복구

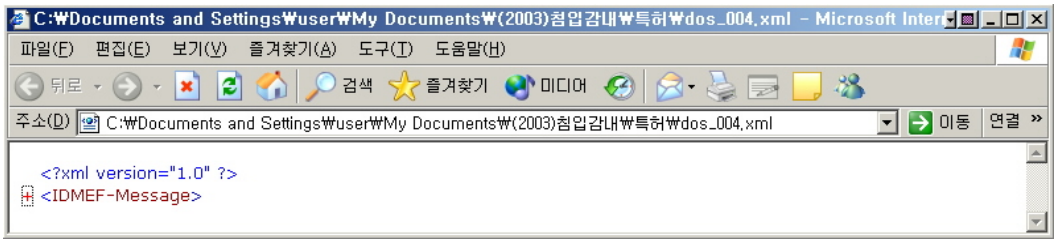
도면4


침입탐지
네트워크 관리
시스템 성능
취약성 평가
기타 요인
시스템 상태 결정
(위험 분석 모델)
Policy

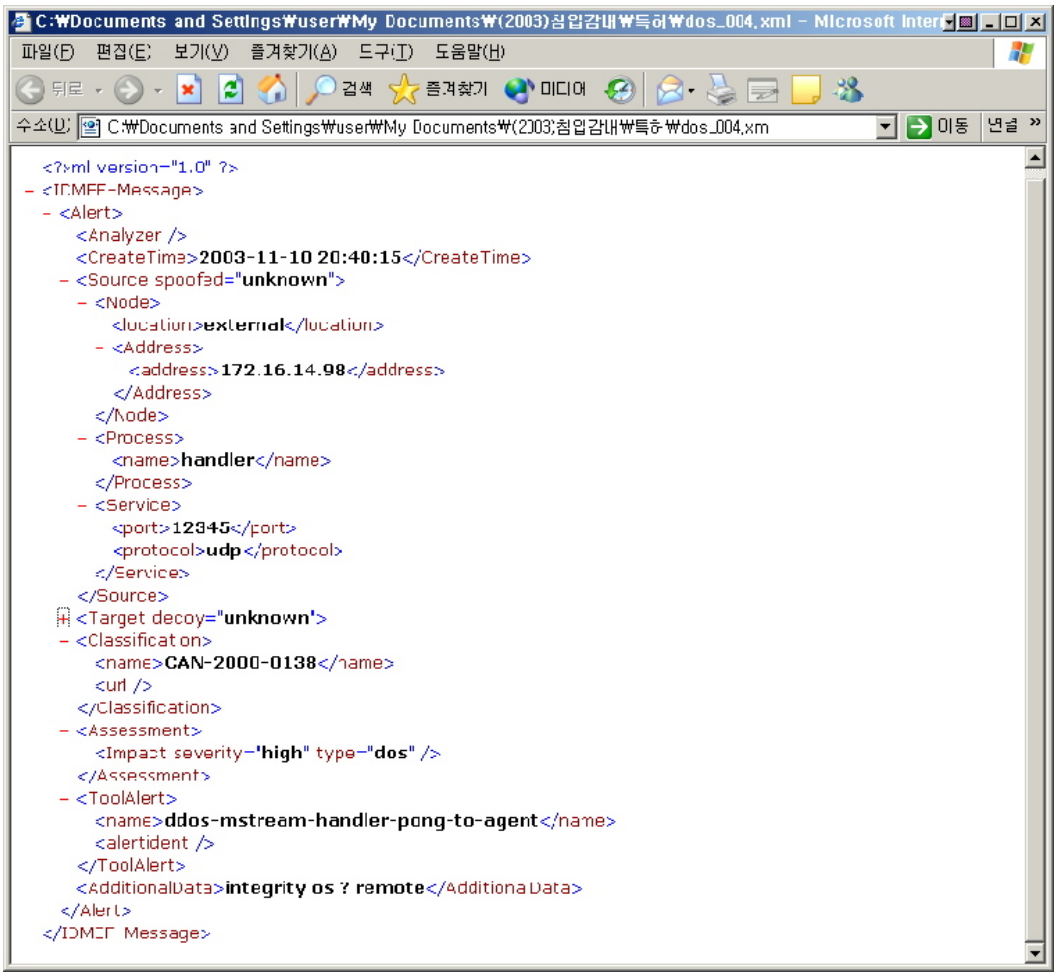
도면5


전처리기
취약성 식별자 유무 확인
No
Yes
공격 DB
공격 DB 검색
No
Yes
위험평가 모듈
공격의 위험도 예측 (C4.5와 AdaBoost)
위험평가지식베이스
탐지정보 학습 (C4.5)
위험수준 결정
• 네트워크 대역폭
• 시스템 성능
• 시스템 중요도
• 공격 빈도

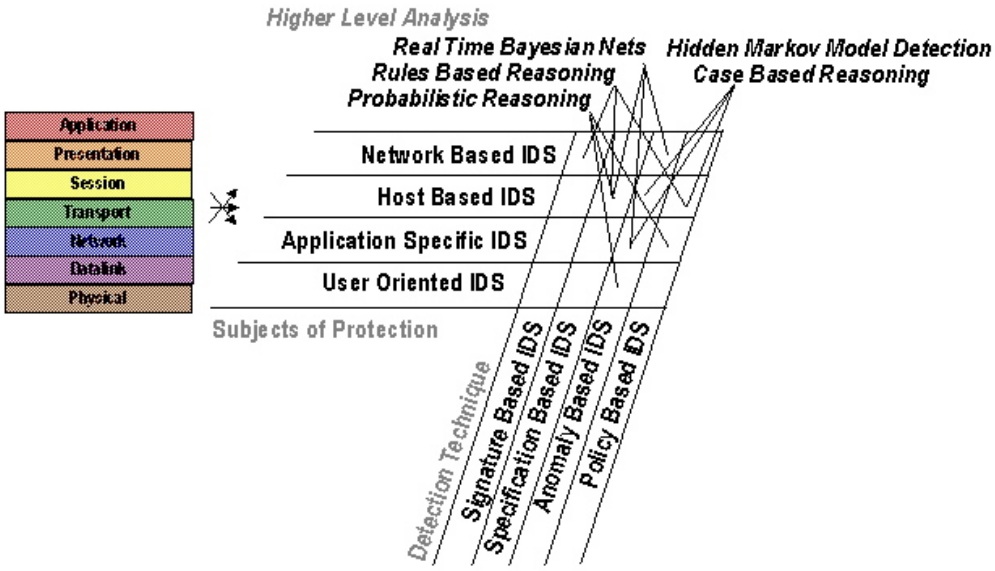
도면6


C:₩Documents and Settings₩user₩My Documents₩(2003)침입감내₩특허₩dos_004.xml - Microsoft Inter
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
뒤로 검색 즐겨찾기 미디어
주소(D) C:₩Documents and Settings₩user₩My Documents₩(2003)침입감내₩특허₩dos_004.xml
이동 연결
<?xml version="1.0" ?>
<IDMEF-Message>

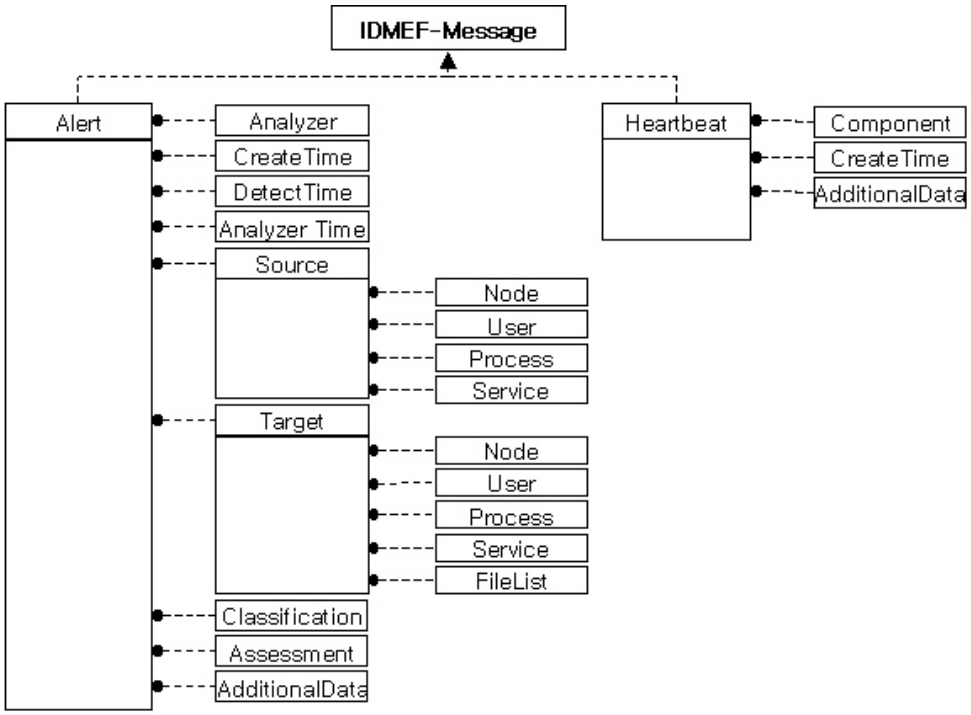
도면7


C:₩Documents and Settings₩user₩My Documents₩(2003)침입감내₩특허₩dos_004.xml - Microsoft Inter
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
뒤로 검색 즐겨찾기 미디어
주소(D) C:₩Documents and Settings₩user₩My Documents₩(2003)침입감내₩특허₩dos_004.xm
이동 연결
<?xml version="1.0" ?>
- <IDMEF-Message>
- <Alert>
<Analyzer />
<CreateTime>2003-11-10 20:40:15</CreateTime>
- <Source spoofed="unknown">
- <Node>
<location>external</location>
- <Address>
<address>172.16.14.98</address>
</Address>
</Node>
- <Process>
<name>handler</name>
</Process>
- <Service>
<port>12345</port>
<protocol>udp</protocol>
</Service>
</Source>
+ <Target decoy="unknown">
- <Classification>
<name>CAN-2000-0138</name>
<url />
</Classification>
- <Assessment>
<Impact severity="high" type="dos" />
</Assessment>
- <ToolAlert>
<name>ddos-mstream-handler-pong-to-agent</name>
<alertident />
</ToolAlert>
<AdditionalData>integrity os ? remote</AdditionalData>
</Alert>
</IDMEF-Message>

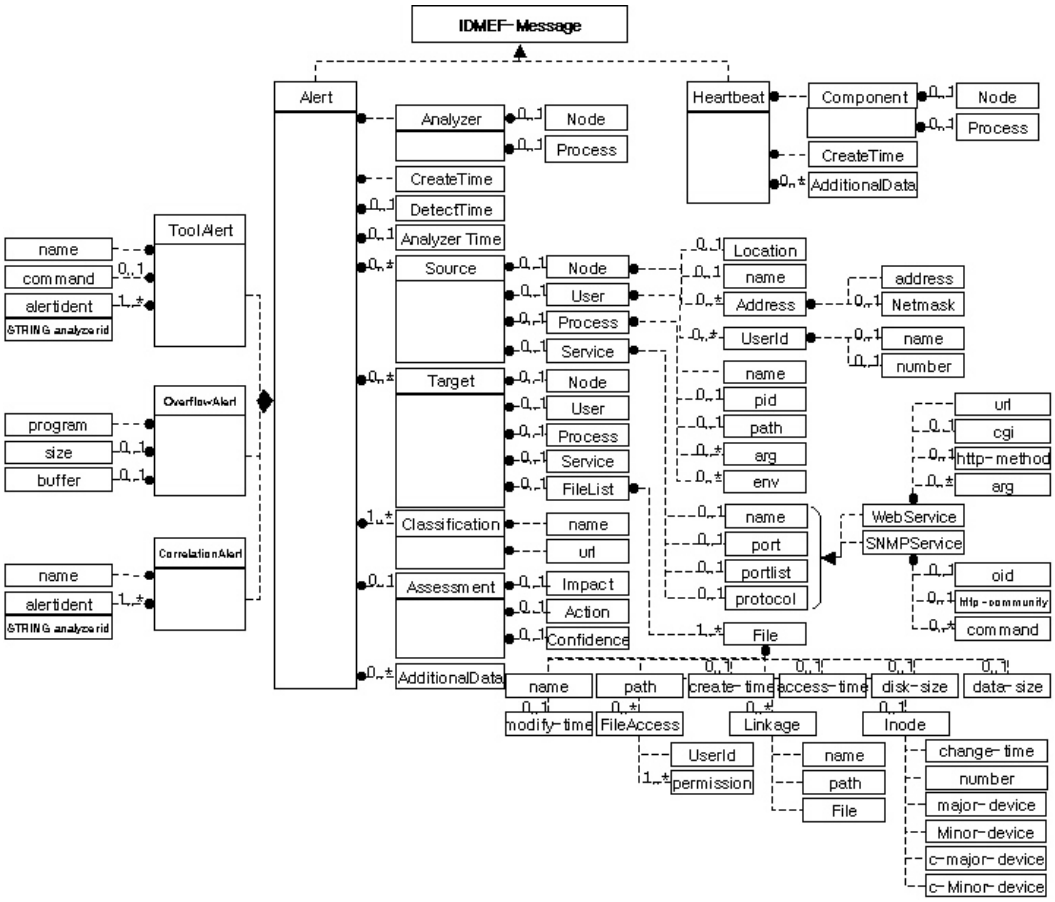
도면8


Higher Level Analysis
Real Time Bayesian Nets
Rules Based Reasoning
Probabilistic Reasoning
Hidden Markov Model Detection
Case Based Reasoning
Application
Presentation
Session
Transport
Network
Datalink
Physical
Network Based IDS
Host Based IDS
Application Specific IDS
User Oriented IDS
Subjects of Protection
Detection Technique
Signature Based IDS
Specification Based IDS
Anomaly Based IDS
Policy Based IDS

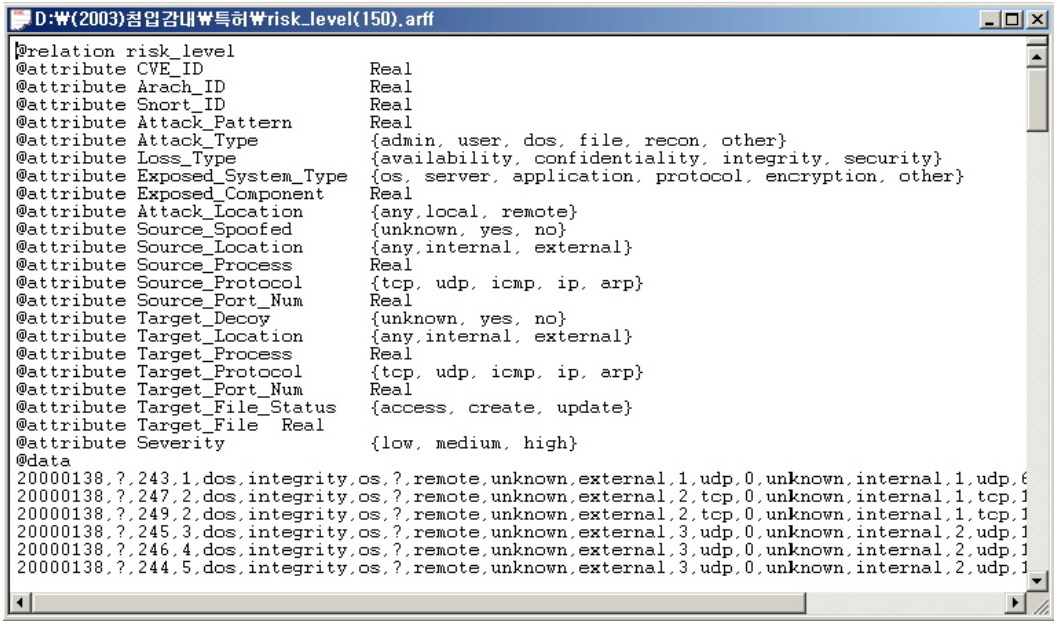
도면9


IDMEF-Message
Alert
Analyzer
CreateTime
DetectTime
Analyzer Time
Source
Node
User
Process
Service
Target
Node
User
Process
Service
FileList
Classification
Assessment
AdditionalData
Heartbeat
Component
CreateTime
AdditionalData

도면10


IDMEF-Message
Alert
Analyzer
Node
Process
CreateTime
DetectTime
Analyzer Time
Source
Node
User
Process
Service
Target
Node
User
Process
Service
FileList
Classification
name
url
Assessment
Impact
Action
Confidence
AdditionalData
Location
name
Address
address
Netmask
UserId
name
number
name
pid
path
arg
env
name
port
portlist
protocol
File
name
path
create-time
access-time
disk-size
data-size
modify-time
FileAccess
UserId
permission
Linkage
name
path
File
Inode
change-time
number
major-device
Minor-device
c-major-device
c-Minor-device
Heartbeat
Component
Node
Process
CreateTime
AdditionalData
ToolAlert
name
command
alertident
OverflowAlert
program
size
buffer
CorrelationAlert
name
alertident
url
cgi
http-method
arg
WebService
SNMPService
oid
http-community
command

도면11


D:₩(2003)첨입감내₩특허₩risk_level(150).arff
@relation risk_level
@attribute CVE_ID Real
@attribute Arach_ID Real
@attribute Snort_ID Real
@attribute Attack_Pattern Real
@attribute Attack_Type {admin, user, dos, file, recon, other}
@attribute Loss_Type {availability, confidentiality, integrity, security}
@attribute Exposed_System_Type {os, server, application, protocol, encryption, other}
@attribute Exposed_Component Real
@attribute Attack_Location {any,local, remote}
@attribute Source_Spoofed {unknown, yes, no}
@attribute Source_Location {any,internal, external}
@attribute Source_Process Real
@attribute Source_Protocol {tcp, udp, icmp, ip, arp}
@attribute Source_Port_Num Real
@attribute Target_Decoy {unknown, yes, no}
@attribute Target_Location {any,internal, external}
@attribute Target_Process Real
@attribute Target_Protocol {tcp, udp, icmp, ip, arp}
@attribute Target_Port_Num Real
@attribute Target_File_Status {access, create, update}
@attribute Target_File Real
@attribute Severity {low, medium, high}
@data
20000138,?,243,1,dos,integrity,os,?,remote,unknown,external,1,udp,0,unknown,internal,1,udp,
20000138,?,247,2,dos,integrity,os,?,remote,unknown,external,2,tcp,0,unknown,internal,1,tcp,
20000138,?,249,2,dos,integrity,os,?,remote,unknown,external,2,tcp,0,unknown,internal,1,tcp,
20000138,?,245,3,dos,integrity,os,?,remote,unknown,external,3,udp,0,unknown,internal,2,udp,
20000138,?,246,4,dos,integrity,os,?,remote,unknown,external,3,udp,0,unknown,internal,2,udp,
20000138,?,244,5,dos,integrity,os,?,remote,unknown,external,3,udp,0,unknown,internal,2,udp,

도면12

입력 (T: 학습데이터 집합, M: 최대 반복횟수)

초기화 $w_1(x_i) = 1/N$

Do for k = 1 to M

1) 학습데이터와 L를 이용하여 t_k를 구축한다.

2) 오분류 에러율을 계산한다.

$$\varepsilon_k = \sum_{i=1:\, t_k(x_i) \neq y_i}^{N} w_k(x_i)$$

만일 $\varepsilon_k \geq 0.5$이면 $\varepsilon_k = 0$, Do 문의 수행을 멈춘다.

3) 학습 인스턴스에 대하여 가중치를 재부여한다.

$$w_{k+1}(x_i) = \frac{w_k(x_i)}{z_k} \times \begin{cases} \exp(-\alpha_k), & \text{if } t_k(x_i) = y_i \\ \exp(\alpha_k), & \text{if } t_k(x_i) \neq y_i \end{cases}$$

where, $\alpha_k = \frac{1}{2}\log\left(\frac{1-\varepsilon_k}{\varepsilon_k}\right), z_k = 2\sqrt{(1-\varepsilon_k)\cdot\varepsilon_k}$

출력 $t^*(x) = sign\left\{\sum_{k=1}^{M} \alpha_k \cdot t_k(x)\right\}$

$T = \{x_i, y_i\}_{i=1}^{N}$: 학습 인스턴스의 집합

x_i: i 번째 학습 인스턴스

$y_i \in$ {low, medium, high}: x_i의 분류 클래스

N: 모든 학습 인스턴스의 개수

L: 약성 또는 기본 학습기 (C4.5)

M: 최대 시도 횟수(maximum trials)

$w_k(x_i)$: k 번째 시도에서 i 번째 인스턴스의 가중치

ε_k: k 번째 시도에서 모든 인스턴스의 에러율

t_k: w_k일 경우 T로부터 L에 의해 생성된 분류자

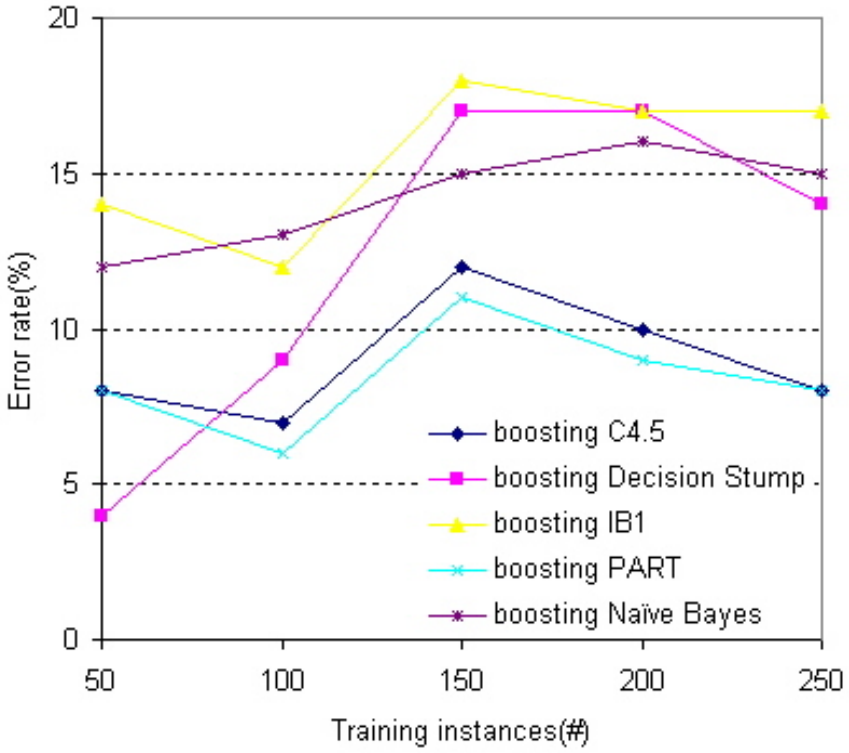
t^*: 복합 분류자(the composite classifier)

$t_k(x_i)$: i 번째에 예측된 클래스

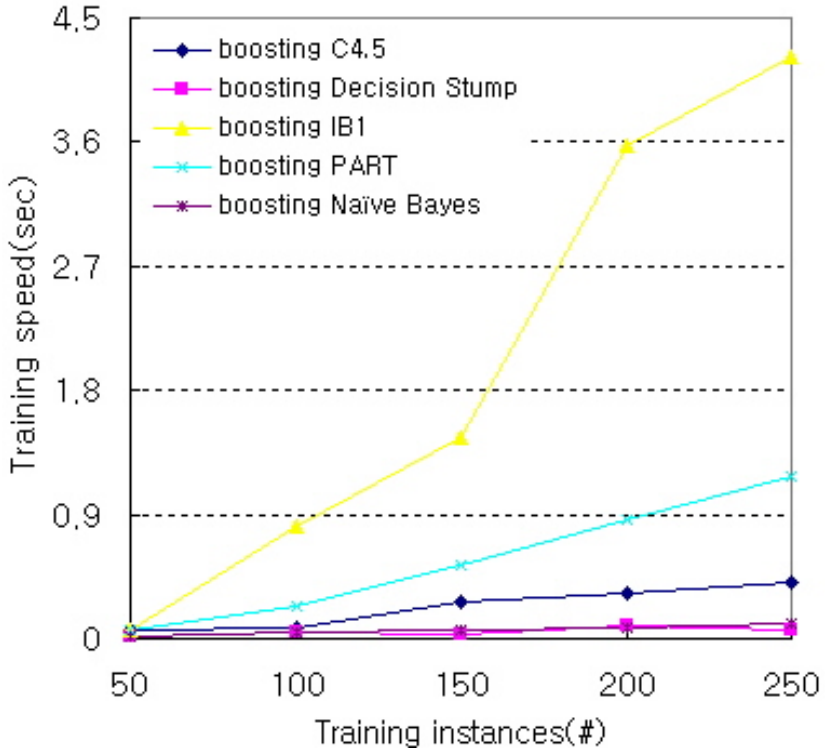
α_k: 분류자의 가중치(the weight of a classifier)

z_k: 정규화 인자(the normalization factor)

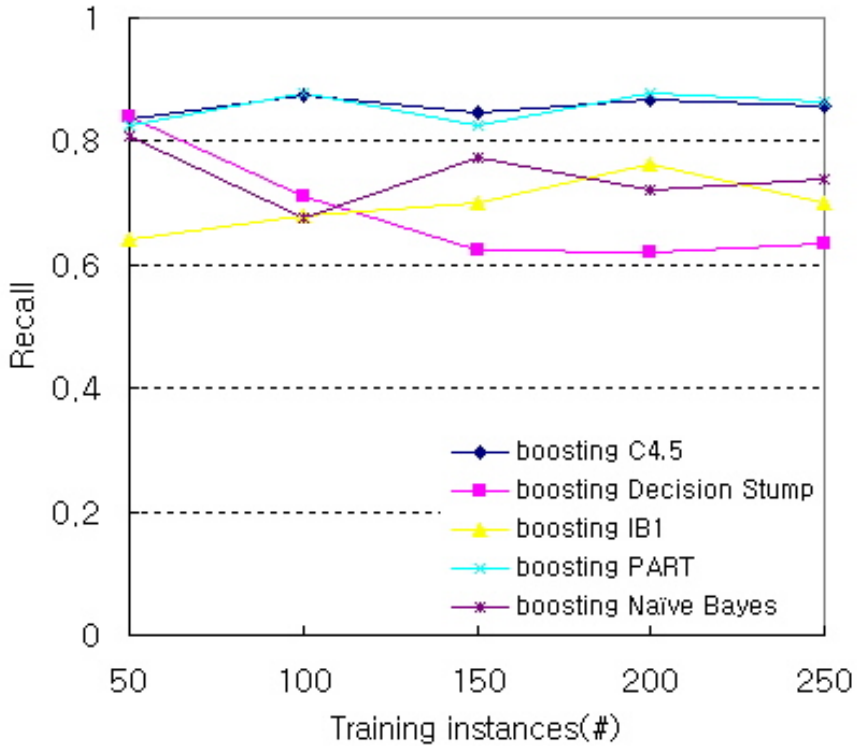
도면13


20
15
10
5
0
Error rate(%)
50
100
150
200
250
Training instances(#)
boosting C4.5
boosting Decision Stump
boosting IB1
boosting PART
boosting Naïve Bayes

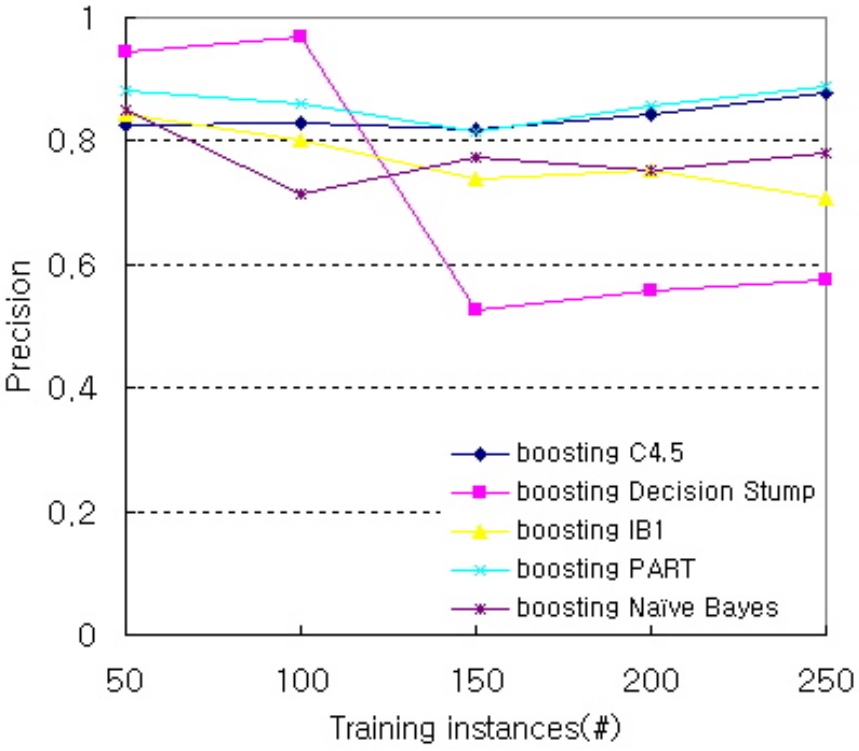
도면14


4.5
3.6
2.7
1.8
0.9
0
Training speed(sec)
50
100
150
200
250
Training instances(#)
boosting C4.5
boosting Decision Stump
boosting IB1
boosting PART
boosting Naïve Bayes

도면15


1
0.8
0.6
0.4
0.2
0
Recall
50
100
150
200
250
Training instances(#)
boosting C4.5
boosting Decision Stump
boosting IB1
boosting PART
boosting Naïve Bayes

도면16


1
0.8
0.6
0.4
0.2
0
Precision
50
100
150
200
250
Training instances(#)
boosting C4.5
boosting Decision Stump
boosting IB1
boosting PART
boosting Naïve Bayes