通信情報保護學會論文誌
第9卷 第1號, 1999. 3

분산환경을 위한 실시간 침입 탐지 모델의 설계

이 문 구*, 전 문 석*

Design and Analysis of Real-time Intrusion Detection Model for Distributed Environment

Moon-Ku Lee*, Moon-Seog Jun*

요 약

기존의 침입 탐지 방법은 침입이 발생했을 때 침입을 바로 탐지하지 못한다. 이러한 문제점을 해결하기 위해 실시간 침입탐지에 대한 연구가 많이 진행되고 있다. 기존의 침입 탐지 시스템들은 대부분 호스트 레벨의 시스템이기 때문에 분산환경과 같은 네트워크 레벨에서 사용하는 경우에는 기존 침입 탐지 시스템을 다른 시스템으로 이식하거나 확장하기가 어렵다. 또한 침입 탐지 시스템들간에 메시지를 주고받을 때 메시지의 비밀성 등을 제공해주어야 한다.

본 논문은 에이전트를 이용한 실시간 침입 탐지 모델을 제안한다. 이 실시간 침입 탐지 모델은 에이전트의 확장성과 에이전트간의 통신 메커니즘을 이용하여 분산 환경에 적용하여 사용할 수 있으며, 기존의 침입탐지시스템의 이식성이나 확장성, 비밀성들을 제공한다.

Abstract

The most of intrusion detection methods do not detect intrusion when it happens. To solve the problem, we are studying a real-time intrusion detection. Because a previous intrusion detection system(IDS) is running on the host level, it difficult to port and to extend to other system on the network level that distributed environment. Also IDS provides the confidentiality of messages when it sends each other.

This paper proposes a model of real-time intrusion detection using agents. It applies to distributed environment using an extensibility and communication mechanism among agents, supports a portability, an extensibility and a confidentiality of IDS.

* 숭실대학교 정보과학대학

1. 서 론

정보시스템이 점차 네트워킹화, 글로벌화 되어가며 확대됨에 따라서 정보의 안전성에 대한 문제가 대두되었다. 이러한 안전성에 문제가 있는 정보의 전송이나 컴퓨터 자원의 공유는 새로운 사회 문제로 인식되어 전산망 내에 있는 사용자에게 보다 안전한 통신을 제공하기 위한 안전성 있는 메커니즘의 개발이 필요하게 되었다. 정보보호에 대한 필요성과 중요성이 날로 높아지고 있으면서 국내외적으로 방화벽을 컴퓨터나 네트워크의 보안에 대한 최선책으로 생각하고 개발에 임하고 있으나 최근 들어서는 방화벽이 보안의 최선책이 아니라 최소한의 대책이라는 말이 나올 정도로 최근의 해킹 기법에 대해서는 무색한 경우가 많이 발생하고 있다. 더욱이 방화벽은 외부의 침입으로부터 컴퓨터의 자원을 보호하기 위한 목적으로 설계되었기 때문에 내부의 권한 있는 사용자에 의한 침입에 대해서는 속수무책인 경우가 많다. 이러한 지능적인 침입을 막기 위해서는 불법적으로 컴퓨터 시스템에 침입하여 중요한 정보들을 손상시키는 행위들을 분산 환경에서 사전에 탐지하고 중지시킬 수 있는 실시간 침입-탐지 시스템 개발이 필요하다.

본 논문에서는 에이전트를 이용한 분산환경에서의 침입탐지 시스템을 제안하였다. 제안한 침입탐지 모듈은 각 호스트 상에서 동작하는 에이전트와 분산 환경에서 동작하는 에이전트 매니저로 구성함으로써, 실시간에 분산 처리할 수 있도록 설계하였다. 본 논문의 구성은 다음과 같다. 2장에서는 침입 탐지 방법에 대해서 기술하고, 3장에서는 지금까지 실시간 침입 탐지 시스템을 구현하기 위해 사용되었던 접근 방법들에 대해 살펴보고, 4장에서는 본 논문에서 제안한 실시간 침입 탐지 모델의 구조와 구성요소를 설명하며, 5장에서는 기존의 실시간 침입 탐지 시스템과 비교 평가한 후, 결론을 제시하였다.

2. 침입 탐지

2.1 침입과 침입 탐지

일반적으로 침입은 "컴퓨터가 사용하는 자원의 무결성, 비밀성, 유용성을 저해하는 일련의 행위들의 집합"이라고 정의된다. 이러한 침입의 형태는 보통 크게 비정상적인 침입과 오용 침입으로 나누어질 수 있다.

비정상적인 침입은 컴퓨터 자원의 비정상적인 행위나 사용에 근거한 침입으로써, 침입의 행위가 예외적인 경우에 해당한다.

반면에, 오용 침입은 시스템과 응용 소프트웨어의 약점을 악용한 잘 정의된 공격 형태를 말한다. 예를 들어 fingerd와 sendmail 버그를 통한 인터넷 Worm의 공격 형태가 오용 침입의 대표적인 예이다.

2.2 비정상적인 침입 탐지 방법

침입 탐지 방법은 행위의 결과에 따른 비정상 침입을 탐지하는 비정상적인 침입 탐지 방법과 오용 침입 탐지 방법의 두 가지로 분류된다.

비정상적인(anomalous) 침입 탐지 방법의 종류는 다음과 같다.

1) 통계적인 방법
2) 특징 추출
3) 비정상적인 행위 측정방법들의 결합
4) 예측 가능한 패턴 생성
5) 신경망

2.2.1 통계적인 방법

통계적인 방법은 비정상적인 침입의 탐지를 주로 통계적으로 처리하는 방법이다. 과거의 경험적인 자료를 토대로 처리하기 때문에 상

당히 정확하게 탐지할 수 있다.

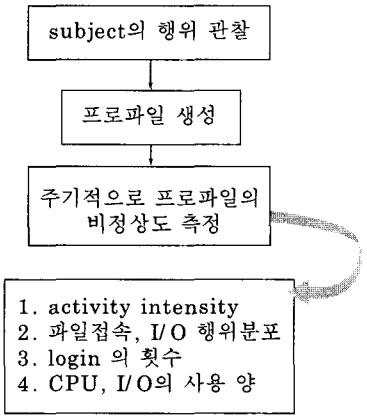


그림 1 통계적인 방법

탐지 방법은 먼저 사용자나 사용자가 실행시킨 프로세스의 행위를 관찰하고, 각각의 행위에 대한 profile을 생성한다. 생성된 profile들을 주기적으로 관찰하여 profile의 비정상적인 행위를 측정하는 abnormality를 측정한다.

profile의 abnormality를 측정하는 방법들은 다음과 같이 여러 가지 방법들이 있다.

1) activity의 intensity를 측정하는 방법
2) 파일 접속이나 I/O 행위의 분포의 측정 방법
3) login의 횟수를 측정하는 방법
4) CPU나 I/O의 사용 양을 측정하는 방법

위의 방법들을 사용하여 사용자의 행위가 정상적인 행위인지 비정상적인 행위인지를 측정 할 수 있다. 통계적인 방법의 장점은 통계적으로 잘 연구된 방법들을 사용할 수 있다는 것이다. 단점은 다음의 네 가지를 들 수 있다:

1) 통계적인 방법은 어떤 행위의 발생 순서에 민감하지 못하다는 것이다.
2) 순수한 통계적인 침입 탐지 시스템은 어떤 행위의 변경사항이 작은 경우는 정상이라고 학습시킬 수 있기 때문에 점차적으로 비정상적인 행위도 정상이라고 잘못 판단할 수 있다.
3) 어떤 행위가 정상인지 비정상인지를 판가름할 수 있는 임계치(threshold)를 설정하기가 힘들다.
4) 순수한 통계적인 방법을 사용하여 모델링 할 수 있는 행위의 종류가 제한적이다.

2.2.2 특징 추출

특징 추출은 특정 침입 패턴을 추출하는 방법으로, 탐지방법은 경험적인 침입 탐지 측정 도구의 집합을 설정하고, 침입의 예측, 분류 가능한 침입 탐지 도구의 부분집합을 결정하여 침입을 예측, 분류한다.

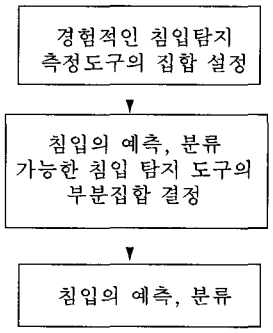


그림 2 특징 추출

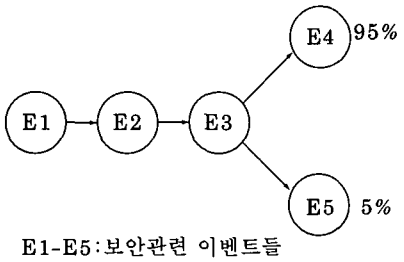
2.2.3 비정상적인 행위 측정방법들의 결합

이 방법은 여러 비정상적인 행위 측정방법들을 사용하여 각각의 결과를 통합하여 특정 행위가 정상인지 비정상인지를 측정하는 방법이다. 사용하는 측정 방법들은 Bayesian Statistics, Covariance Matrices, Belief Network 등을 사용한다.

2.2.4 예측 가능한 패턴 생성

이 방법은 특정 행위를 이루는 이벤트의 순

서가 랜덤하지 않고 인식 할 수 있는 패턴이라는 가설에 근거하며, 이벤트간의 상호관계와 순서를 설명할 수 있다. time-based rule을 사용하여 각각의 이벤트에 시간을 부여할 수 있으며, 이벤트의 순서가 올바른 경우에도 시간의 간격에 따라 주어진 이벤트들이 정상인지 비정상인지 탐지할 수 있다.



E1-E5:보안관련 이벤트들

그림 3 예측 가능한 패턴 생성

예를 들어, 그림 3과 같이 E1이 발생한 후 E2와 E3가 발생하고 E4가 발생할 확률이 95% 이고, E5가 발생할 확률이 5%라면 E4가 발생하는 것이 정상이고, E5가 발생하는 것은 비정상일 것이다. 이와 같이 발생할 확률이 적은 이벤트가 발생한 경우는 침입이라고 간주하여 탐지할 수 있게 된다. 이 방법의 단점은 rule에서 표현되지 않은 행위의 패턴은 비정상이라고 인식할 수 없다는 것이다. 장점은 특정 행위가 일련의 순서를 가진 패턴이라면 이 행위의 다양한 변형 형태를 다룰 수 있으며, 의심이 가는 전체 로그인 세션보다 더 자세하게 관련된 이벤트들을 처리할 수 있기 때문에 이벤트 레벨에서의 비정상적인 행위를 탐지할 수 있다.

2.2.5 신경망

이 방법은 명령어의 순서를 신경망으로 학습시켜서 다음에 수행될 명령어를 미리 예측할 수 있다. 다음에 수행되는 명령어를 예측할 수 있기 때문에 정상적인지 비정상적인지를 탐지할 수 있다.

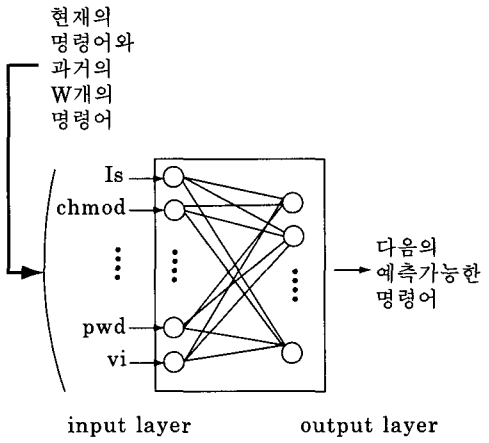


그림 4 신경망을 이용한 침입 탐지

그림 4에서 보면 현재의 명령어와 과거에 수행시켰던 명령어들을 input layer에 입력시키면 신경망은 스스로 학습하여 다음에 수행될 명령어를 output layer에서 나타내게 된다. 신경망의 장점은 통계적인 방법과 같이 자료의 특성에 의존할 필요가 없으며, 노이즈가 많은 데이터에 잘 탐지할 수 있고, 출력에 영향을 줄 수 있는 다양한 방법들 사이의 상관관계를 잘 나타낼 수 있다.

단점은 신경망의 토폴로지와 각각의 구성요소 사이에 주어지는 가중치(weight)를 여러 번 학습시킨 후에야 결정할 수 있다는 것이다. 또한, 현재의 명령어와 과거의 w개의 명령어를 설정할 때 w의 크기는 신경망 설계에 중요한 요소로 설정하기 어렵다. w의 크기가 큰 경우에는 보다 정확한 결과를 얻을 수 있지만 학습 시간이 많이 걸리고, w의 크기가 적은 경우에는 학습 시간은 빠르지만 결과가 정확하지 않을 수 도 있기 때문이다.

2.3 오용 침입 탐지 방법

오용(misuse) 침입 탐지 방법의 종류는 다음과 같다.

1) 조건부 확률 (Conditional Probability)
2) 전문가 시스템 (Production/Expert System)
3) 상태 전이 분석 (State Transition Analysis)
4) 키-스트로크 관찰 (Keystroke Monitoring)
5) 모델에 근거한 침입 탐지 (Model-based Intrusion Detection)

2.3.1 조건부 확률

오용(misuse) 침입을 탐지하기 위해 다음과 같은 조건부 확률을 사용한다.

$$P(Intrusion|Event\ Pattern) = P(Event\ Pattern|Intrusion)\frac{P(Intrusion)}{P(Event\ Pattern)}$$

이벤트 패턴중에서 특정 이벤트가 침입일 확률은 위의 조건부 확률의 공식과 같이 계산된다. 예를 들어, 전산망에서 전문가는 침입 발생에 따른 선행 확률과 $P(Intrusion)$을 한정하고, 망에서 시스템의 침입 보고표가 만들어지면 이벤트 순서의 각 유형별 $P(EventSequence\ vert\ Intrusion)$을 결정하고, 전체 침입집단에서 이벤트 순서가 발생하는 관계빈도는 이 확률로 제공된다. 이와 비슷하게 제시된 intrusion-free 감사 추적l 집단은 검사와 테이블화 함으로서 확률 $P(EventSequence\ vert \neg Intrusion)$를 결정한다. 주어진 두 조건부 확률과 이벤트 순서의 선행 확률

$P(Event\ Sequence) = P(ES|I) \cdot P(I) + P(ES|\neg I)$로부터 Bayesian 식을 이용하여 상기식의 좌측을 결정한다.

2.3.2 전문가 시스템

전문가 시스템은 if-then-rule에서 공격 패턴들에 대한 지식을 표현하고, 감사 추적 이벤트와 일치하는 fact들을 명시한다. 일치하는 공격 패턴을 발견하는 경우에는 if-then-rule에 따라 처리한다. 그림 5와 같이 if-then-rule에 일치하는 경우에는 rule의 action을 수행하게 된다.

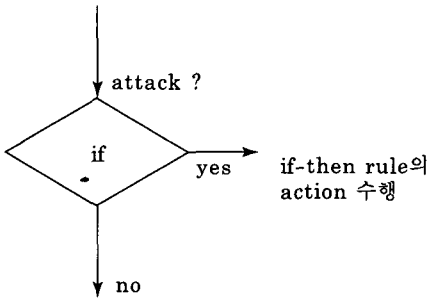


그림5 전문가시스템을 이용한 침입탐지

2.3.3 상태 전이 분석

상태 전이 분석(State Transition Analysis)은 공격 패턴을 특정 시스템의 상태 전이(state transition)의 순서로 표현한다.

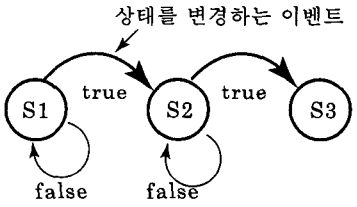


그림6 상태전이분석을 이용한 침입탐지

그림 6과 같이 특정 행위가 수행된 이후의 시스템 상태를 state라고 표현한다. 하나의 state에서 다음 state로 이동하기 위해서는 이벤트들이 필요로 하게 되는데, 하나의 state가 주어진 조건을 만족하면 다음의 state는 어떤 것인지를 알 수 있게 된다.

2.3.4 키-스트로크 관찰

이 방법은 공격 패턴의 발생을 결정하기 위해 사용자의 키-스트로크를 감시한다. 공격 패턴을 나타내는 특정 키-스트로크 순서를 패턴화한다.

2.3.5 모델에 근거한 침입 탐지

이 방법은 오용(misuse)의 발생을 탐지하기 위해 evidential reasoning을 사용한다. evidential reasoning은 특정 모델을 만들어서 관련된 공격 패턴들을 데이터베이스로 구축하여 특정 공격패턴이 발생하는 경우 이 데이터베이스를 참조하여 탐지한다.

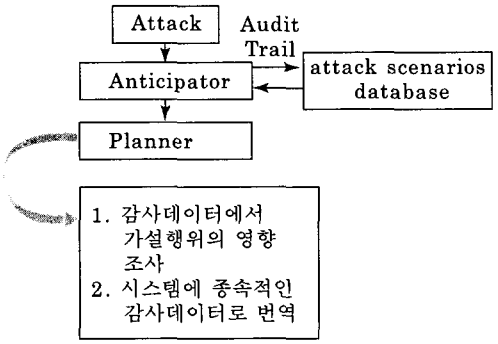


그림 7 모델에 근거한 침입탐지

그림 7을 보면 특정 공격패턴이 발생한 경우, Anticipator는 attack scenario database에서 관련된 패턴을 검색하고, 검색결과를 Planner에게 알려준다. Planner는 hypothesized behaviour의 영향을 조사하고, 시스템이 읽을 수 있는 감사 추적 데이터로 번역한다.

3. 실시간 침입 탐지

3.1 실시간 침입 탐지의 개요

인터넷의 확장에 따라서 네트워크를 통한 침입의 가능성이 증가되었고, 이에 따라 시스템이나 네트워크 침입을 즉각적으로 탐지하고 대처할 능력이 있는 기술이 필요하게 되었으며, 또한, 이러한 기술을 이용하여 자동으로 침입을 탐지, 보고, 조치하는 자동화된 시스템이 필요하다.

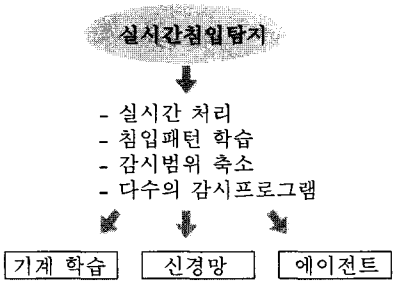


그림8. 실시간 침입탐지의 발전동향

기계학습, 신경망, 에이전트 등은 이러한 목적을 만족하여 실시간 침입탐지를 구현하는 토대가 될 수 있다.

3.2 기계학습을 이용한 실시간 침입탐지

기계학습을 이용한 침입탐지는 주어진 시스템을 관찰하여 얻어지는 감사기록을 이용하고, 정상적인 행위를 특성화하여 학습시킨 후 비정상적인 조건을 탐지하는 방법이다. 대표적인 시스템으로는 Wisdom & Sense 시스템이 있다. 기계학습의 분야는 크게 개념학습(Concept learning), 클러스터링(Clustering), 예측학습(Predictive learning) 그리고 특징추출(Feature extraction) 등으로 분류할 수 있는데, 이러한 특성들을 침입탐지 시스템에 적용하는 것이다.

개념 학습(Concept learning)이란 구성요소들을 몇 개의 범위로 분류하도록 시스템을 학습한다. 그리고 구성요소의 속성들을 조정한 후, 임의의 작업 세션이 침입인지 정상인지를 판별하기 위해 이용 가능한 다양한 속성을 사

용하며, 부적절한 속성 중에서 관련 있는 속성을 수집하도록 한다.

클러스터링(Clustering)은 구성요소들을 유사성(similarity) 범위를 사용하여 관련 있는 구성 요소들의 그룹으로 분할하는데, 범위와 분류 규격이 필요하며, 사용자, 세션, 자원 억세스 요구 등 관련 있는 구성요소들의 집합으로 분류한다.

예측 학습(Predictive learning)은 시간적인 데이터와 이산적인 이벤트의 순서에서 침입 이벤트를 학습 할 수 있는 기능으로 주로 Markov와 time-series model을 사용한다.

특징 추출(Feature extraction)은 관련이 없는 특징들로부터 관련이 있는 특징들을 구별하여, 관련이 있는 특징들을 이벤트를 나타내는 함수로 결합한다.

3.3 신경망을 이용한 실시간 침입 탐지

신경망을 이용한 침입탐지는 융통성 있는 인식 기능을 침입탐지에 사용한 방법으로써, 사용자나 시스템 행위의 적용 모델링(Adaptive modeling)을 할 수 있다는 것이 큰 장점이다. 신경망을 이용한 침입탐지는 알려지지 않은 공격 패턴을 취급하는데 유용하다. 활용범위로써는 특정 바이러스 패턴을 학습하거나 바이러스 발생 시 수행할 대책을 설정, 사용자와 시스템의 정상 상태의 모델화, 비정상적인 행위 발견 시 수행할 대책 설정 등에 사용되고 있다.

3.4 에이전트

에이전트란 한 호스트의 특별한 측면 또는 비정상적인 보고 그리고 흥미 있는 행동을 감시한다. 에이전트는 시스템의 행위가 비정상적이라고 생각되면 에이전트들끼리 협력하여 시스템의 행위를 감시한다. 즉, 하나의 에이전트는 시스템에서 작은 부분을 감시하고, 또한 다수의 에이전트를 이용하여 복잡한 침입 탐지 시스템 구축이 가능하다. 에이전트들은 서로 독립적으로 활동하며, 동적으로 시스템에 추가되거나 삭제 또한 가능하다.

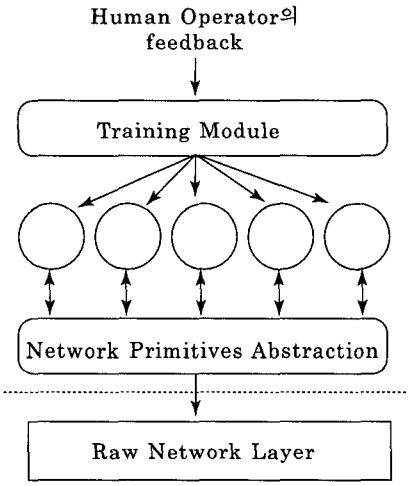


그림 9. 에이전트를 이용한 침입탐지

고유의 에이전트를 이용한 침입탐지 방법은, 하나의 커다란 단일 침입탐지 모듈대신 소규모의 자율적인 에이전트 집단으로 침입탐지 모듈을 구성하는 방법이다. 이러한 방법은 단일 침입탐지 모듈에 비해 많은 장점을 지니는데, 다수의 에이전트를 이용하여 복잡한 침입탐지 시스템 구축이 가능하며, 주요위협을 확인하여 에이전트가 인식할 수 있도록 학습이 가능하다. 한번 학습되면 시스템의 오버헤드가 감소하게되고, 시스템의 감시가 실패하는 경우 에이전트는 그 상태를 기억하여 재 수행 시 성능 저하를 방지한다. 만약 문제가 되는 에이전트가 발생하는 경우 그 에이전트의 기능을 취소하고 네트워크 환경에서 쉽게 수정이 가능하다. 에이전트의 추가 및 삭제가 용이하여 큰 시스템으로의 확장과 구축이 쉽다.

표 1 Perl로 작성된 에이전트의 예제

```
Package YourAgent;

# Keep $VERSION in sync with RCS revision
$VERSION = do { my @r = (q$Revision: 1.1 $ =~ /\d+/g);
sprintf "%d."."%02d" x $#r, @r }; $VERSION = $VERSION;

%PARAMETERS=(Description => "Agent description",
                    CheckPeriod => 10);   # Seconds

use AAFID::Agent;
use AAFID::Common;
@ISA=qw(AAFID::Agent);

sub Init {
  my $self=checkref(shift);
  # Optional initialization code here.
}

sub Check {

  my $self=checkref(shift);
  # Mandatory check code here.
 return ($status, $message);
}

sub Cleanup {
  my $self=checkref(shift);
  # Optional cleanup code here.
}

# This is the template for implementing a new command called MYCMD.
# Any number of new command may be de fined by an agent.
#
#  sub command_MYCMD {
#    my ($self, @message, %params)=@_;
#    # Code to implement the command MYCMD here.
#    return undef;
#    # If a return value is needed, use something like:
#    # return { code => $code, Message => $msg };
#  }

 # Very important, to make the agent both loadable and standalone.
 _EndOfEntity;
```

3.4.1 AAFID2

AAFID(Autonomous Agents for Intrusion Detection)2 시스템은 퍼듀 대학에서 에이전트를 이용하여 개발한 침입 탐지 시스템이다.

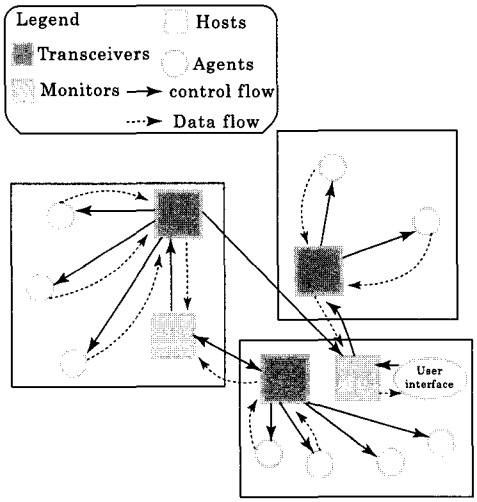


그림10. AAFID2의 구조

AAFID2 시스템은 그림 10에서 보는 것과 같이 에이전트, 트랜시버, 모니터의 세 가지 구성요소들로 구성된다. 이 구성요소들은 AAFID 개체(entities)나 단순히 개체라고 불리고, 이 개체들로 구성된 전체 침입 탐지 시스템은 AAFID 시스템이라고 불린다. AAFID 시스템은 네트워크상의 많은 호스트들에 널리 분포된다. 각각의 호스트는 많은 에이전트들을 가지고 있으며, 에이전트는 호스트에서 발생하는 이벤트들을 감시한다. 하나의 호스트에서 모든 에이전트들은 트랜시버에게 그들의 발견사항을 보고한다. 트랜시버는 호스트 당 하나씩 존재하며 호스트에서 동작중인 모든 에이전트들의 행위를 감독한다. 트랜시버는 호스트에서 동작하는 에이전트들을 제어하며 에이전트에게 구성(configuration) 명령어들을 시작하거나 중단하고 전송할 수 있다. 트랜시버는 또한 에이전트들에게서 받은 데이터들로부터 데이터들을 정리하여 모니터에게 그 결과를 보고하고 각각의 모니터는 트랜시버들의 행위를 감독한다. 모니터는 네트워크상의 데이터에 접속해서 높은 레벨의 상호관계(correlation)를 분석하고 여러 호스트들 사이의 연관된 침입을 탐지한다. 모니터는 계층적 방식으로 구성될 수 있기 때문에 하위 레벨의 모니터가 높은 레벨의 모니터에게 보고할 수 있으며, 궁극적으로 정보를 제공하고 사용자 인터페이스로부터 제어 명령어를 얻을 수 있다.

표 1은 Perl로 작성된 에이전트의 예제를 나타낸다. 에이전트는 초기화, 검사할 조건 설정, 코드의 삭제, 사용자 정의 명령어 설정 부분들로 구성된다.

4. 분산환경의 실시간 침입탐지

4.1 분산환경의 침입탐지시 고려사항

분산환경에서 실시간 침입 탐지 기능을 수행하기 위해서는 먼저 다중 호스트들을 대상으로 하는 침입탐지에 대해 분석해야 한다. 각각의 호스트에서 발생하는 보안 관련 이벤트들 분석의 경우 여러 사용자들에 의해 많은 세션들이 발생하기 때문에 여러 호스트에서 발생하는 보안 관련 이벤트들을 분석하여야만 한다. 그리고 다중 호스트들의 이벤트 데이터 간의 상호관계를 분석하려면, 여러 호스트들에서 발생한 이벤트 데이터들은 서로 연관성을 가지고 하나의 이벤트가 다른 이벤트에 영향을 줄 수 있기 때문에 상호 관계에 대해 분석하여 하나의 이벤트가 다른 이벤트에 어떤 영향을 미치는지를 조사해야 한다.

이처럼 분산환경에서는 시스템을 계층적으로 구성하여 다양한 레벨의 이벤트에 대한 추상화가 이루어져야 한다. 사용자, 호스트, 서브넷, 도메인 등과 같이 다양한 레벨을 지원함으로써 침입을 쉽게 규정하고 효과적으로 침

입에 대응할 수 있다.

분산환경에서 다중 호스트들의 침입을 탐지할 때는 이기종 호스트들의 환경, 분산 침입 패턴, 다중 호스트들간의 이벤트의 순서를 나타내기 위한 전체 시간의 설정과 같은 문제점들이 발생한다.

다중 호스트들의 침입을 탐지할 때의 문제점 중 첫째, 이기종 호스트들의 환경에서 고려해야 할 사항으로 다중 호스트들은 서로 다른 환경에 존재하기 때문에 각각의 호스트의 구조나 운영체제, 감사 서브 시스템의 특징들을 고려해야 한다. 감사 서브 시스템의 특징들은 이벤트 형태의 레이아웃이나 이벤트 데이터의 의미, 감사 데이터의 수집, 저장, 구성 등이 고려된다.

두 번째 문제점은 분산 침입 패턴이다. 침입 패턴 탐지는 각 호스트의 취약점 집합에 의해 결정되며, 각각의 호스트에서 개별적으로 처리되기 때문에 호스트 레벨에서는 정상이지만 네트워크 레벨에서 볼 때 보안상 취약한 사용자 행위들 즉, 분산 침입 패턴은 탐지하기 어렵게 된다. 사용자 행위의 임계치(threshold)가 호스트들마다 서로 다르기 때문에 호스트 기반의 오용 프로파일(anomaly profile)의 구축이 어렵다. 전체 사용자들의 행위 감시는 각 호스트에서 발생한 방대한 양의 감사 데이터의 분석을 필요로 한다.

세 번째 문제점은 다중 호스트들간의 이벤트의 순서를 나타내기 위한 전체 시간의 설정이다. 각 호스트간의 시간과 통신 지연은 감사 데이터의 타임 스탬프에서 불 일치성을 나타내고, 이벤트 횟수, 시간들과 같은 오용행위를 탐지하는데 사용하는 측정 방법들에 영향을 준다.

이처럼 분산 환경에서 실시간 침입탐지를 행하고자 할 때의 문제점을 고려하여 실시간에 침입 탐지를 행하고자 한다면, 실시간 처리와 침입 패턴의 학습 그리고 감시 범위를 축소하고 다수의 감시 프로그램 등을 이용하여야 한다.

4.2 제안한 실시간 침입탐지 모델

현재까지의 침입 탐지 시스템은 중앙에서 분석하므로 실패의 위치가 한곳에 집중되어있으며, 단일 호스트에서 모든 정보를 처리하므로 확장성이 제한되고, 침입 탐지 시스템에 재구성 또는 능력을 부과하기가 어렵다. 그리고 네트워크 데이터의 분석은 손상될 수 있는 등의 많은 문제점을 갖고 있어서, 분산환경의 실시간 침입탐지는 어렵다. 그러나 제안하는 시스템은 서로 독립적으로 활동하고, 동적으로 시스템에 추가되거나 삭제가 가능하기 때문에 다수의 에이전트들을 이용하여 복잡한 시스템의 구축이 가능한 에이전트를 이용한다. 또한 지역 분석기와 여러 호스트상의 지역 분석기를 전역 타이머와 프로파일 데이터 베이스를 이용한 전역 분석기를 사용한다. 전역 분석기는 지역 분석기가 미처 탐지하지 못한 침입도 탐지할 수 있다.

4.2.1 제안한 모델의 구조

기존의 침입탐지 시스템들은 대부분 단일 호스트 상에서 침입탐지를 기반으로 구현되어 왔다. 하지만 최근에 보고되고 있는 바에 의하면, 단일 호스트 상에서 이루어지는 침입의 형태보다, 분산 네트워크 상에서 좀 더 복잡하고 다양한 방법으로 이루어지는 침입이 점차로 증가하고 있는 추세이다. 단일 호스트 상에서 뿐 아니라 분산환경에서 효율적으로 침입탐지를 하기 위해서, 본 연구에서는 Perl로 프로그램을 구현한 다수의 독립적인 에이전트들로 구성된 지역 분석기(Local Evaluator)와 전역 분석기(Global Evaluator)로 구성된 침입탐지 모델을 제안하였다. 지역 분석기는 각 호스트

에서 동작하는 침입탐지 모듈이며, 전역 분석기는 분산환경에서의 취약점을 보완하기 위해서 구성된 침입탐지 모듈이다.

4.2.2 에이전트

에이전트는 플랫폼에 독립적이므로 이 기종간에 에이전트 구성이 가능하고, 주어진 에이전트의 성격에 따라 필요한 클래스만으로 구성이 가능하다. 이들 에이전트는 동적으로 시스템에 추가되거나 삭제가 가능하며, 한 에이전트가 감시에 실패하는 경우, 에이전트는 그 상태를 기억하여 재 수행 시에 성능이 저하되는 것을 방지한다. 또한 에이전트에 문제가 발생하는 경우 해당 에이전트 기능은 취소할 수 있다.

4.2.3 전역 분석기(Global Evaluator)

전역 분석기는 지역 분석기에서 얻어지는정보를 바탕으로, 지역 분석기에서 미처 탐지하지 못한 침입을 탐지할 수 있도록 하였다.

전역 타이머는 다중 호스트들간의 이벤트의 순서를 나타내기 위한 전체시간을 설정하기 위해 사용한다.

전역 분석기의 프로파일 데이터베이스는 각각의 호스트의 구조나 운영체제, 이벤트 형태의 레이아웃이나 이벤트 데이터의 의미 등을 고려하여 관련 정보를 관리, 저장한다.

지역 분석기는 일반적으로 하나의 호스트에서 발생하는 침입의 행위를 탐지하는데 사용하며 지역 오용 데이터베이스와 지역 통계 데이터베이스를 가진다.

4.2.4 지역 분석기(Local Evaluator)

지역 분석기는 각 호스트에 동작하는 에이전트들로부터 정보를 받아 침입 여부를 판별하는 침입 모듈이다. 지역 분석기의 구성은 그림 12와 같다.

지역 분석기는 에이전트 라우터와 지역 규칙 데이터베이스, 지역 통계 데이터베이스로 구성된다.

에이전트 라우터는 에이전트에 관한 정보 관리하며, 에이전트의 연결 및 해제, 메시지 전송 및 수신에 관여하게 된다.

지역 규칙 데이터베이스는 하나의 호스트에서 발생하는 오용 침입을 탐지하는데 사용된다. 지역 통계 데이터베이스는 사용자 행위의

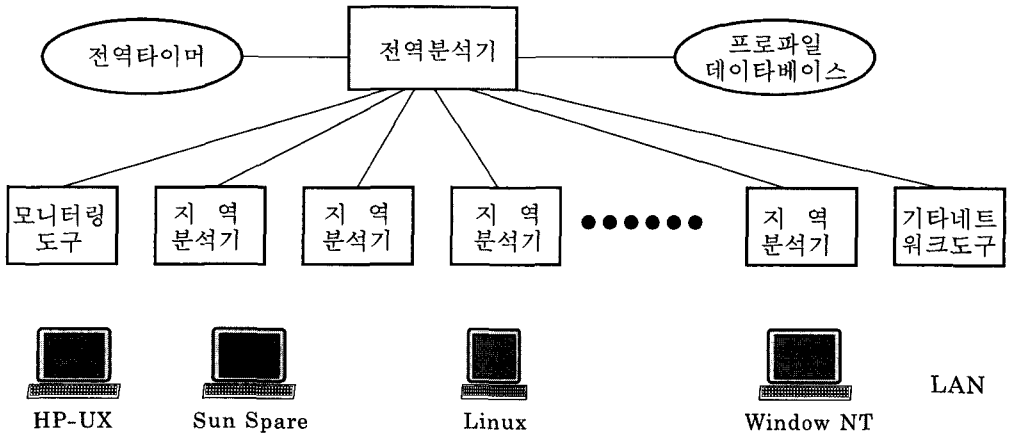


그림 11. 제안한 모델의 전체 구성도

임계치를 설정하여 사용자의 현재 행위와 프로파일의 값을 비교하여 침입여부를 판별한다.

5. 비교 평가

퍼듀 대학에서 개발한 AAFID 시스템과 제안한 모델을 다음과 같이 항목별로 비교하고자 한다.

표 3과 같이 제안한 모델과 AAFID 시스템은 에이전트를 이용하여 실시간으로 침입을 탐지할 수 있으며, 분산환경에 적합하여 분산 침입 패턴을 탐지할 수 있다. 제안한 모델은 전역 분석기의 전역 타이머와 전체 프로파일 데이터베이스를 이용하여 다중 호스트들간의 감사데이터의 타임스탬프를 제공하고, 이 기종 호스트별 프로파일을 제공하지만, AAFID 시스템은 이러한 기능들을 제공하지 않는다.

표 3. 제안한 모델과 AAFID의 비교

비교항목	제안한 시스템	AAFID2
에이전트의 기능	있음	있음
분산 침입 패턴	탐지 가능	탐지 가능
다중 호스트들의 감사데이터의 타임스탬프	제공함	제공하지않음
이기종 호스트별 프로파일	제공함	제공하지않음

6. 결 론

본 논문에서 제안한 실시간 침입 탐지 모델은 자율적인 에이전트 시스템을 이용해서 분산된 환경에서 서로 분리되어 있는 각각의 에이전트 상호간에 서로 유용한 정보를 교환할 수 있으며, 분산 환경하에서는 이러한 에이전트 시스템을 이용하므로 해서 좀더 경쟁력 있는 실시간 침입탐지를 설계할 수 있다. 전역 타이머와 전역 분석기와 전역분석기의 프로파일 데이터베이스, 지역분석기와 지역 오용 데이터베이스와 지역 통계 데이터베이스를 통하여 다중 호스트들이 서로 다른 환경에 존재하여 발생하는 이벤트 형태의 레이 아웃이나 이벤트 데이터의 의미의 불일치 문제와 사용자 행위의 임계치가 호스트들마다 서로 다르기 때문에 발생하는 호스트 기반의 오용 프로파일의 구축이 어려운 문제, 각 호스트간의 시간과 통신 지연에 의한 감사 데이터의 타임 스탬프에 대한 불일치 문제들을 해결할 수 있다.

향후에 연구할 방향으로는 이 모델에 근거하여 각 구성요소와 통합된 침입 탐지 시스템을 개발하는 것이다.

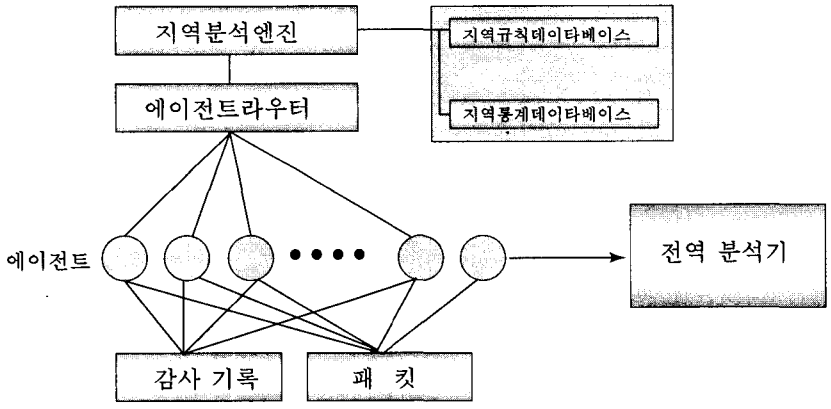


그림 11. 지역 분석기의 구성

참 고 문 헌

[1] Dorothy E. Denning, "An Intrusion Detection Model," In IEEE Transaction on software engineering , Number 2, Feburary 1987

[2] Mark Crosbie, Bryn Dole, Todd Ellis, Ivan Krsul, and Eugene Spaddord. IDIOT-User Guide. COAST Laboratory, Purdue University, 1398 Computer Scince Building, West Lafayette, IN 47907-1398, September 1996. Available at http://www.cs.purdue.edu/coast/coast-library.html

[3] Mark Crosbie, Gene Spafford, "Defending a Computer System using Autonomous Agents," Technical Report CSD-TR-95-022, Purdue University, March 11, 1994.

[4] Joseph Barrus " A Distributed Autonomous-Agent Network-Intrusion Detection and Reponse" Proc., 1998 Commnad and Control Research and Technology Symposium, Monterey CA, June-July 1998.

[5] Yannis Labrou, Tim Finin, "A Proposal for a new KQML Specification," Technical Report TR-CS-97-03 University of Maryland Baltimore County, Feb 3, 1997

[6] Mark Crosbie and Eugene Spafford. Defending a computer system using autonomous agent. In Proceedings of the 18th National Information System Security Conference, Oct 1995.

[7] Mark Crosbie and Gene Spafford. Active defense of a computer system using autonomous agent. Technical Report 95-008, COAST Group, Department of Computer Science, Purdue University, West Lafayette, IN 47907-1398, fEB 1995.

[8] Dorothy E. Denning. An Intrusion-Detection Model. IEEE Transactions on Software Engineering, 13(2):222-232, February 1987.

[9] William M. Farmer, Joshua D. Guttman, and Vipin Swarup. Security for mobile agent: Issues and requirements. In Proceedings of the 19th National Information Systems Security Conference, volume 2, pages 591-597. National Institute of Standards and Technology, October 1996.

[10] Stephanie Forrest, Steven Hofmeyer, Anil Somayaji, and Thomas Longstaff. A sense of self for Unix processes. In Proceedings of the 1996 IEEE Symposium on Security and Privacy. IEEE, IEEE Computer Press, 1996.

[11] Stephanie Forrest, Steven A. Hofmeyr, and Anil Somayaji. Computer Immunology. Communications of the ACM, 40(10):88-96, October 1997.

[12] R. Heady, G. Luger, A. Maccabe, and M. Servilla. The Architecture of a Network Level Intrusion Detection System. Technical report, University of New Mexico, Department of Computer Science, August 1990.

[13] L. Heberlein, G. Dias, K. Levitt, B. Mukherjee, J. Wood, and D. Wolber. A Network Security Monitor. In Proceedings of the IEEE Symposium on Research in Security and Privacy, May 1990.

[14] Eurene Spafford and Diego Zamboni. Aframework and prototype for a distributed intrusion detection system. Technical Report 98-06, COAST Laboratory, Purdue University, Wwst Lafayette, IN 47907-1398, May 1998.

□ 著者紹介

이 문 구

1984년 숭실대학교 전산과(학사)
1993년 이화여자대학교 교육대학원 전산학과(석사)
1996년 ~현재 숭실대학교 대학원 전산과 박사수료
1997년 ~현재 명지 전문대학 전산과 겸임교수

※ 주관심분야 :

전 문 석

1980년 숭실대학교 전자계산학과(학사)
1986년 University of Maryland, Computer Science(석사)
1998년 University of Maryland, Computer Science(박사)
1989년 Morgan State Univ. 부설 Physical Science Lab. 책임 연구원
1991년 ~현재 숭실대학교 컴퓨터학부 부교수

※ 주관심분야 :