通信情報保護學會誌
第5卷 第2號, 1995. 6

# 비정상적인 컴퓨터 행위 방지를 위한 실시간 침입 탐지 병렬 시스템에 관한 연구

## (Real-time Intrusion-Detection Parallel System for the Prevention of Anomalous Computer Behaviours)

유 은 진*, 전 문 석**

## 요 약

Our paper describes an Intrusion Detection Parallel System(IDPS) which detects an anomaly activity corresponding to the actions that interaction between near detection events. IDES uses parallel inductive approaches regarding the problem of real-time anomaly behavior detection on rule-based system. This approach uses sequential rule that describes user's behavior and characteristics dependent on time, and that audits user's activities by using rule base as data base to store user's behavior pattern. When user's activity deviates significantly from expected behavior described in rule base, anomaly behaviors are recorded. Observed behavior is flagged as a potential intrusion if it deviates significantly from the expected behavior or if it triggers a rule in the parallel inductive system.

## 1. 서 론

기존의 보안 장치는 비밀번호와 같은 접근제어 방식을 통해서 인가되지 않은 사용자일 경우 시스템 접근을 막는 방법으로 시스템을 보호하였다. 이러한 방법은 접근제어에 많은 위협적인 요인들이 존재하며, 우회하여 침입하는 경우도 많이 발생하고 있다. 외부적인 침입이나 내부적인 침입으로 부터 보호되기 위하여 모든 환경하에서 접근제어 기법이 신뢰할 수 있는가? 에는 많은 문제점들이 존재한다. 대부분의 보안 시스템은 특권을 갖고 있는 내부 사용자에 의하여 남용되는 경우 시스템에 손상되어지며, 이러한 경우 탐지할 수 있는 방법은 존재하지 않는다.

침입 탐지에 대한 초기 작업중의 하나는 컴퓨터 시스템 추적 데이터를 분석하는 방법을 제시한 Jim Anderson[2]에 의해 연구되었다. 그의 방법

* 숭실대학교 컴퓨터학부
** 숭실대학교 컴퓨터학부

은 다른 이유들에 대해 모아진 데이터를 사용하였으며, 일괄 처리 형식으로 설계되었다. 귀납적 연구[7]는 경험으로부터 습득할 수 있는 컴퓨터를 만드는 것을 목표로 하는 인공지능의 한 분야이며, 본 연구에서 다루어질 방법은 실시간 귀납적 실험 프로그램을 사용하고자 한다. 이것을 인공 지능의 기법을 이용한 추론 방법[7]이라고 말하며, 본 연구는 우선 실시간에 침입을 탐지하기 위하여 K. Chen의 방법을 이용하여 수행하였다. 검색 추적 분석에 의해 설명될 수 있는 침입의 종류는 다음과 같다.

(1) 외부의 침입자

컴퓨터 사용이 허용되지 않은 사람이 비밀번호를 여러번 시도해서 시스템에 침입될 수 있는 경우의 침입자를 말한다.

(2) 내부의 침입자

컴퓨터의 사용은 허용되었지만, 데이터, 프로그램, 모든 자원에 접근할 수 없는 사람이 허락없이 침입하는 경우의 침입자를 말한다.

① 가면을 쓴 형태의 사용자 (다른 사람의 사용자명과 비밀번호를 사용하는 사용자)

② 비밀스러운 사용자 (탐지를 벗어나고 제어 관리에 접근하는 종류의 사용자)

(3) 부적절한 사용자 (컴퓨터의 사용이 허용되고, 모든 장비의 사용이 가능하지만, 이러한 권한을 잘못 사용하는 산업 스파이같은 사용자)

(4) 기타 : 바이러스

본 논문에서는 성공적인 침입 방지 기술에 대해 실시간 귀납적 방법에 의하여 허가없이 불법으로 사용하는 사용자와 정상적인 사용 방법에서 비정상적인 행위를 시도하는 부적절한 사용자를 대상으로 탐지 발견하는데 적용하는 경험적 방법과 추론적 귀납적 방법을 이용하여 연구되었다.

실시간 침입-탐지 병렬 시스템 IDPS는 CPU의 사용, login 시도와 같은 시스템 측정치를 감시하고 그 측정치와 통계적으로 나타난 측정치를 비교한다. IDPS는 병렬 시스템의 한 분야로서, 과거의 침입 행위, 알려진 시스템의 취약점, 각 사용자의 특정 보안정책을 바탕으로한 규칙을 포함한다. 규칙은 사용자가 과거의 행위 패턴에서 벗어났는지와는 별도로 수상한 행위를 기술한다. 감시 시스템으로부터 나타난 데이터는 행위가 의심스러운지 아닌지를 결정하기 위해서 규칙과 일치시켜 본다. 사용자 프로화일을 탐지하는 동안에 IDPS에 의해서 생성된 가정들을 격자 모양의 형태로 저장한다. 본 연구에서는 규칙 데이터베이스에 기록된 가정들 사이에서 일반화된 관계를 기록, 수정하며, 격자 모양의 형태는 Chen[7] 방법으로 대상이되는 사용자 프로화일을 추적 탐지하는데 사용된다.

## 2. 실시간 침입-탐지 병렬 시스템 구조

침입-탐지 병렬 시스템 IDPS의 설계 모델은 그림 2.1처럼 목표되는 시스템, 목표되는 범위 인

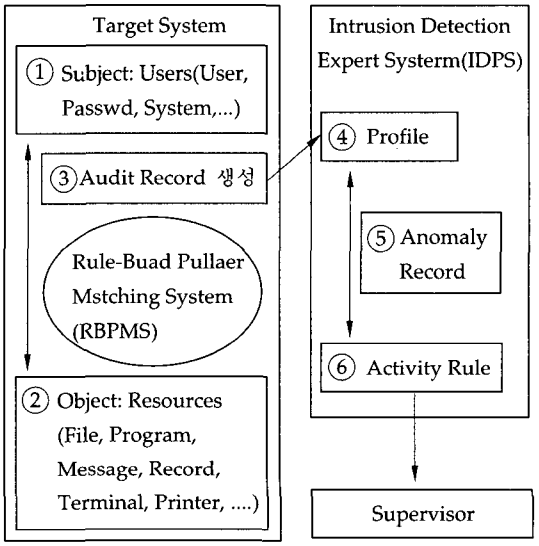


그림 2.1 침입 - 탐지 병렬 시스템 IDPS 설계 모델

터페이스, 침입-탐지 처리 엔진, 사용자 인터페이스 등 네 가지로 분류된다. 침입-탐지 병렬 시스템 환경에서 감시되는 범위는 목표되는 시스템에 연관된 시스템들에 감시 범위가 확산되어서 침입-탐지 기능을 수행할 수 있다. 이와 같은 의미는 목표 시스템에 관련된 모든 시스템에 감시 추적할 수 있는 audit trail의 정보를 이용하여 탐지-감시할 수 있다.

IDPS 목표 감시 영역은 비정상적 행위에 대해 감시되어지는 감시범위들을 모두 합한 집합으로 구성되어진다. 감시되어지는 여러개의 중앙 시스템이 존재하기 때문에, 감시범위에 대한 것과 목표 시스템의 집합에 대한 것을 모두 적재하여 감시 감독하는 audit 데이터들의 중앙 집결 장치가 하나 이상 여러개가 존재한다.

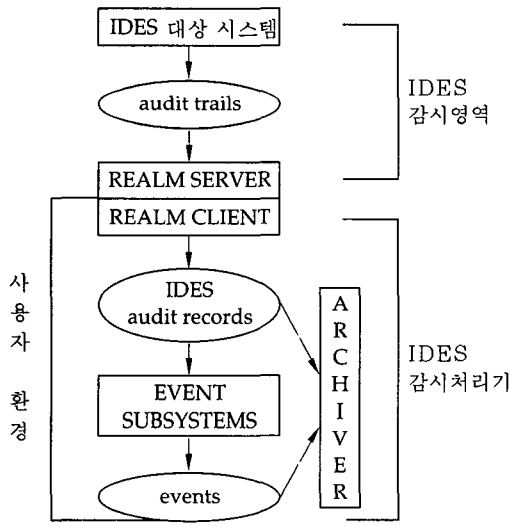


그림 2.2 간단한 IDPS 수행되는 흐름도

그림 2.2에서 실시간 IDPS 기본적인 흐름도는 시스템의 비정산적인 행위를 탐지할 수 있는 핵심 수행함수 흐름도를 행위 탐지 모듈 형태의 기본 골격 흐름를 보여주고 있다.

침입-탐지 병렬 시스템은 감시범위 인터페이스, 통계적이고 비정상적인 행위 탐지기, 비정상적인 행위 탐지 병렬 시스템, 사용자 인터페이스 등으로 구성되고 있다. IDPS 프로세서는 실시간 침입-탐지 병렬 시스템이 구현될 수 있도록 분산 병렬 컴퓨터 구조를 설계하기 위하여 독립된 프로세서 형태로 설계하였다.

## 2.1 실시간 침입-탐지 병렬 시스템에 대한 규칙

침입 탐지에 대한 초기 작업 중의 하나는 컴퓨터 시스템 추적 데이터를 분석하는 방법을 제시한 Jim Anderson에 의해 연구되었다. 그의 방법은 다른 이유들에 대해 모아진 데이터를 사용했으며, 일괄 처리 형식[2]으로 설계되었다. 침입-탐지 병렬 시스템의 규칙 형태에 관한 설명은 다음과 같다.

(1) 추적 레코드 규칙

① 새로운 추적 레코드와 활동 프로화일의 패턴이 일치될때 발생한다.
② 프로화일을 갱신, 예외적 행위를 검사한다.
③ 비정상 활동이 검출되면 비정상 레코드를 생성한다.
④ 변수의 유형에만 의존한다.(통계적 모델)

```
AUDIT-RECORD RULE
    Condition : new Audit.Record
                Audit.Record matches Profile
                Profile.Variable Type = t
    Body :      AuditProcess(Audit-Record, Profile);
END
```

(2) 정기적 활동 갱신 규칙

① 시간적인 길이 p가 수행된 후에 발생한다. ② 프로화일을 갱신, 예외적인 행위를 검사한다.

```
PERIODIC-VARIABLE-UPDATE RULE
    Condition : Clock mod p = 0
                Profile.Period = p
                Profile. Variable-Type = t
    Body :      PeriodProcess(Clock, Profile);
END
```

(3) 비정상 레코드 규칙

① 새로운 비정상 레코드가 규칙에서 주어진 패턴과 일치될 때 발생한다. ② 보안 담당자에게 즉각적인 주의를 출력한다.

```
ANOMALY-RECORD RULE
    Condition : new Anomaly-Record
       Anomaly-Record.Profile matches profile-pattern
       Anomaly-Record.Event matches event-pattern
    Body :    PrintAlert('Suspect intrusion of type..',
                  Anomaly-record);
 END
```

(4) 정기적 비정상 행위 분석 규칙

① 시간간격 마지막인 경우 수행 후에 발생한다.
② 시간대별로 비정상 행위 레코드들을 분석하고, 보안 담당자에게 비정상 행위에 대한 긴급 보고서를 출력한다.

```
PERIODIC-ANOMALY-ANALYSIS RULE
     Condition : Clock mod p = 0
     Body:  Start = Clock - p;
            A = SELECT FROM Anomaly-Records
                WHERE Anomaly-Record.
                    Time-stamp>Start;
            generates summary report of A;
 END
```

## 3. 실시간 귀납적 기계의 탐지 추적

침입-탐지 병렬 시스템은 프로그램의 효율성을 증가시키기 위해 처리중에 여러 다른 모드들을 허용하며, 이 모드들은 시간 간격 정보의 중요성을 구별하는 전략[7]에 의해 구별한다.

침입-탐지 병렬 시스템 전형적인 사건의 개괄적 특징을 기술하면 다음과 같다.

(1) TIME : Log-in 세션이나 사건의 시간 표시에서 주어진 사건의 발생 시간.

(2) DESCRIPTION : 아래와 같이 사건과 관련된 많은 속성

① Event_type : 화일 접근이나 통신망 침입 등과 같은 기록된 사건의 형태
② Image_name : 실행된 적이 있는 실행 가능한 Image의 이름
③ Object_name : 읽혀질 목적물
④ Object_type : 읽혀질 목적물의 형태
⑤ Priviledges_used : 실행 가능한 Image에 의해 사용된 특권
⑥ Status : 실행의 상태
⑦ Process_ID : 처리 식별 코드

침입-탐지 병렬 시스템의 속성들은 시스템의 재조정 없이 추가할 수 있으며, Description의 실제 사용된 예는 다음과 같다.

① Event_type : File 접근 시도
② Event_time : 10_MAR_1993 09:23:03
③ Process_ID : 00000082
④ Image_name : DISK:BACKUP.EXE
⑤ Object_name :
DISK : AUTHORIZATION.DAT
⑥ Object_type : FILE
⑦ Access_Requested :
WRITE, CONTROL
⑧ Status : NORMAL, Normal succesful completion
⑨ Priviledges_used : SYSTEM

침입-탐지 병렬 시스템의 속성 계층들 또한 명백하게 주어진 정보를 추론하는 시스템을 허용하기 위해 기술될 수 있다. 속성 계층은 보안 사건들을 추적 검색에서 명백하게 주어지지 않는 추상적인 패턴들로 일반화하기 위해 침입-탐지 병렬 시스템에 의해 사용된다. 이와 같은 구조가 지역 보안 정책의 추상적인 형태를 특수화하는데 사용되기도 한다.

침입-탐지 병렬 시스템은 Prolog 프로그램 형태로 부록에 추가하였다. 예를 들어, 침입-탐지 병렬 시스템 속성 계층이 주어진다.

9 A.M. to 5 P.M. are business hours ; and
EMACS is an editor; and
MH CHOI is a member of the X project.

"the command EMACS is invoked by MH CHOI at 2 P.M"인 사건은 "editor가 X 프로젝트의 멤버인 MH CHOI에 의해 사용되었음" 과 같이 좀더 구체적인 기술로 실험 프로그램에 의해 실행된다. 한편, "editor가 업무시간 중에 X 프로젝트의 멤버에 의하여 사용되면 증명 프로시저를 실행하라"와 같은 지역 보안 정책의 일반화된 형태가 또한 기술된다.

실시간 침입-탐지 병렬 시스템에 의해 생성된 규칙의 시간 부분은 다른 사건과의 시간 관계를 포함하고 있다. Audit 레코드를 이용하여 임의의 사용자가 침입자인지, 정당한 사용자인지를 빠르게 판단하기 위해서는 여러가지 병렬처리 형태의 알고리즘이 연구되어야 하고, 이것들을 수행하기 위해서는 다음과 같은 세 가지 형태를 우리는 고려하게 된다.

### 3.2 사용자 프로화일 생성

각 사용자나 사용자들 그룹의 프로화일의 기초를 형성하는 보안 활동 규칙들은 침입-탐지 병렬 시스템을 거쳐 생성된다. 이 규칙들은 이전의 보안 검색 기록에 기초한 사용자나 사용자 그룹의 행동 패턴을 기술한다. 더구나 각 규칙은 만족할 만한 정확도에 의해 다음의 가능한 사건들을 예측하는 순차적인 패턴을 기술한다. 프로화일의 구성요소는 다음과 같다.

(1) 주체(Subject)와 객체(Object)에 독립적인 요소들
- ① 변수명
- ② 행동 패턴
- ③ 예외 패턴
- ④ 자원 사용 패턴
- ⑤ 기간 : 측정을 위한 시간 간격
- ⑥ 변수 유형 : 측정과 통계 모델의 특별한 형태를 정의하는 추상적인 데이터 형태의 이름
  예 평균과 표준 편차 모델을 갖는 계수기
- ⑦ 임계값
  - 통계적 테스트에 사용되는 정의 한계
  - 연산 모델 : 상한
  - 표준 편차의 값

(2) 주체와 객체에 관련된 요소들
- ① 주체 패턴
- ② 객체 패턴
- ③ 값 :
  - 현재 추정된 값
  - 이전의 값들의 분포를 표현하기 위해서 통계적 모델에 의해서 사용
  - 표준편차의 값
  - 연산 수치

침입-탐지 병렬 시스템의 규칙 데이터 베이스 시스템은 Prolog에 의해서 작성되었으며, 부록에 추가하였다. 여기서 생성된 간단한 규칙의 예는 다음과 같다.

```
E1 − E2 − E3
→ (E4 = 95% : E5 = 5%)
```

E1, E2, E3, E4, E5는 앞 절에서 기술한 형태의 보안 활동들이다. 규칙은 E1이 E2에 따르고 ("−" 표기에 의해 표현된) E2는 E3에 따른다면 E4가 따를 경우는 95%이며, E5가 따를 경우는 5%라는 것을 나타낸다. 침입-탐지 병렬 시스템은 실제적으로 위에서 주어진 예보다 좀더 일반화된 규칙들(사건들의 시간적인 관계나 사건 묘사들에 관해)을 생성할 수 있다.

침입-탐지 병렬 시스템은 다음의 시간적 관계가 있다고 가정하자.

```
E1 − * → (E2 = 100%)
```

( * 는 어떤 하나의 사건이 지원됨을 표시한다. 하나의 규칙에 * 의 어떠한 수도 허용된다.)

두 사건들 사이의 시간적 관계는 또한 값들의 범위로서 기술될 수 있다. 예를 들어,

```
"E1과 E2는 0, 1, 2개의 사건들에
의해 분리된다"
```

다음은 규칙들이 생성된 과정을 보여주기 위해 설계된 간단한 예이다. 순차적인 규칙들의 각 문자가 사건을 표시한다면,

```
A-B-C-S-T-S-T-A-B-C-A-B-C
```

다음의 규칙들이 생성되어질 것이다.

R1 : A − B → (C ,100%)
R2 : C → (S, 50% : A 50%)
R3 : S → (T, 100%)
R4 : T → (A, 50% : S, 50%)

규칙들은 처리 과정에서 발생하는 것들을 처리하지 않는 대안적인 설명들로서 보여질 수 있다. R2와 R4는 넓게 발산하는 예측을 이끌고 "좋은" 가정이 아니기 때문에 결과적으로 규칙 베이스로부터 제거되어질 수 있다. R1과 R3는 더많은 사건들을 더 정확하게 설명하거나 처리하는 것처럼 보여지고 미래에 편차 감지에 사용될 수 있기 때문에 사용자 프로화일에 남겨질 수 있다.

# 4. 침입-탐지 시스템 IDPS의 경고 관리 구조

## 4.1 대상 시스템의 측정 기준

경고 보고를 위한 병렬 시스템를 구현하는데 필요한 측정 기준과 대상 시스템의 주요 파라미터와 원격 호스트 기준에 의하여 여러가지 측정 대상들은 다음 도표의 기준치들을 이용하였다. 침입-탐지 수행은 규칙 지식 베이스에서 측정된 모든 지식들을 기반으로 기록하고 통계적 데이터에 의하여 정상적인 기준치에 벗어나는 경우는 비정상적인 행위로 간주해서 침입-탐지를 감시하는 시스템을 구현할 수 있다.

표 4.1 원격 호스트 측정 기준

| 측 정 기 준 | 설 명 |
|---|---|
| 사용자 account (선형적) | 각 account가 읽혀진 횟수 |
| 활동 유형 (선형적) | 각 유형별 활동 네트워크량 |
| 시간별 사용(선형적) | 매 시간별 활동 네트워크량 |
| 유형에 따른 시간별 사용(선형적) | 각 유형별/시간별 활동 네트워크량 |
| 그릇된 login 시도(선형적) | 그릇된 시도 횟수 |

표 4.2 사용자 측정기준(a)

| 측 정 기 준 | 설 명 |
|---|---|
| CPU 사용빈도(통상적) | CPU 사용빈도 |
| I/O 사용빈도(통상적) | I/O 사용빈도 |
| 사용처 (선형구분에 의한) | 각 위치에서의 연결 횟수 |
| Mailer 사용빈드 | Mailer가 사용된 횟수 |
| Editor 사용빈도 | Editor가 사용된 횟수 |
| 컴파일러 사용빈도 | 컴파일러가 사용된 횟수 |
| 쉘 사용빈도(선형구분에 의한) | 쉘이 불리운 횟수 |
| 윈도우 명령어 사용빈도 | 윈도우 명령어 사용된 횟수 |
| 프로그램 사용빈도 | 프로그램이 사용된 횟수 |
| 시스템 호출빈도 | 시스템 호출이 사용된 횟수 |
| 디렉토리 사용빈도 | 디렉토리가 읽혀진 횟수 |
| 명령어 사용빈도(통상적) | 명령어가 불리워진 횟수 |
| 디렉토리 생성(통상적) | 생성된 디렉토리 수 |

| | |
|---|---|
| 디렉토리 지움(통상적) | 지워진 디렉토리 수 |
| 디렉토리 읽음(통상적) | 디렉토리가 읽혀진 횟수 |
| 디렉토리 수정(통상적) | 디렉토리 수정 횟수 |
| 화일 사용빈도(선형적) | 각각의 읽혀진 디렉토리 횟수 |
| 화일 사용빈도(2진적) | 화일이 Access되었는지 |
| 임시화일의 사용빈도(통상적) | 임시화일의 Access 수 |

표 4.3 사용자 측정기준(b)

| 측 정 기 준 | 설 명 |
|---|---|
| 화일생성(통상적) | 생성된 화일의 수 |
| 화일지움(통상적) | 지워진 화일의 수 |
| 화일읽음(통상적) | 읽거나/읽혀진 화일의 수 |
| 화일수정(통상적) | 수정된 화일의 수 |
| 읽혀진 사용자ID(선형적) | 사용자 ID가 바뀐 횟수 |
| 읽혀진 사용자ID(2진적) | 사용자 ID가 읽혀졌는지의 여부 |
| 시스템 에러(통상적) | 시스템관련 에러 횟수 |
| 유형별 시스템 에러(선형적) | 각 유형별 발생된 에러 횟수 |
| 추적 레코드 활동성(선형적) | 매 시간당 추적 레코드 수 |
| 시간별 활동성(2진적) | 매 시간별 추적 레코드의 기록 여부 |
| 사용일수 (선형적) | 매일 발생된 추적 레코드 수 |
| 사용일수(2진적) | 추적 레코드가 매일 받아졌는지 |
| 원거리 네트워크 활동성(통상적) | 원거리 네트워크 활동량 |
| 유형별 네트워크 활동성(선형적) | 각 유형별 네트워크 활동량 |
| 호스트별 네트워크 활동성(선형적) | 원격의 호스트별 네트워크량 |
| 근거리 네트워크 활동성(통상적) | 근거리 내의 네트워크량 |
| 유형별 근거리 네트워크량(선형적) | 각 유형별 네트워크량 |
| 호스트별 네트워크량(선형적) | 호스트별 네트워크량 |

표 4.4 대상 시스템 측정기준

| 측 정 기 준 | 설 명 |
|---|---|
| 대상에 대한 활동성(통상적) | 레코드 생성 수 |
| CPU 사용빈도 | 사용된 CPU 시간 |
| I/O 사용빈도 | 사용된 I/O |
| 그릇된 login 시도(통상적) | 그릇된 login 시도 횟수 |
| 시간별 그릇된 login 시도(선형적) | 매 시간별 그릇된 login 시도 횟수 |
| 에러(통상적) | 시스템 에러 수 |
| 유형별 에러(선형적) | 유형별 에러 횟수 |
| 시간별 에러(선형적) | 매 시간별 에러 횟수 |
| 네트워크 활동량(통상적) | 네트워크 활동량 |
| 유형별 네트워크 활동량(선형적) | 각 유형별 네트워크량 |
| 호스트별 네트워크량(선형적) | 원격지 호스트별 네트워크량 |

이러한 기준치를 중심으로 침입-탐지 시스템은 경고 보고 시스템을 구현할 수 있다. 다음 알고리즘은 감시를 하는 사이에 경고 보고 메세지를 교환하는 과정에 따라 서술되었으며, 감시를 하는 시스템과 감시를 받는 시스템으로 나누어 서술하였다. 현재 컴퓨터 통신 네트워크 경고 기능의 경고 전송을 구현하였으며, 구현 환경에 적당한 관리 대상을 선정하여 경고 전송을 묘사하고 있다. 일반적인 네트웨크 침입-탐지와 경고 발생 기능을 컴퓨터 통신 네트워크 관리에 맞도록 하는 많은 연구 필요하다.

**[알고리즘 4.1]** 컴퓨터 네트워크 경고 보고

```
While (delete (any_MO))
 { notify( );
   If (node = confirmed)
      then { Al_request( );
             Wait_Al_request( );
             Al_confirmed( );
             report_to_user( ); }
      else  If (node = non_confirmed)
              then { Al_request( ); }
              else If (error( ))
                   then {
                     retransmit( ); }
 system( );
   exit( );  }
While (receive_Al_request( ))
 { information_collection( );
   diagnose( );
   recovery_procedure( );
   Al_report_indication( );
   If (node = confirmed)
      then { Al_request( ); }
   system( );
   exit( );  }
```

알고리즘 수행은 감시를 받는 시스템에 있는 자원에 의하여 통지된 경고를 사용하여 확인형과 비확인형 서비스에 따라서 수행을 달리한다. 그러나 양쪽 서비스 기능은 감시를 하는 시스템으로 수행하며, 확인형 서비스의 경우에만 감시를 하는 시스템의 경고 데이터 응답 확인을 기다린 뒤, 이를 감시를 받는 시스템의 사용자에게 제공한다. 또한 경고 침입-탐지 시스템 구성 혹은 시스템 처리 에러가 발생할 시에는 재전송 과정을 수행한다. 감시를 하는 시스템은 감시를 받는 시스템으로 제공된 경고 데이터를 수령하고, 탐지 프로세스와 경고 발생 프로세스를 사용하여 보다 정련된 결과를 사용자에게 제공한다.

## 4.2 침입-탐지 병렬 시스템 구현

실시간 침입-탐지 병렬 시스템에서 사용 가능한 정보의 특성과 침입-탐지 관리 문맥 기반 성질은 규칙을 기반으로 하는 접근 방법에 적당하다. 문제 결정을 수행하기 위한 절차적 지식은 규칙에 근거한 시스템으로 구현될 수 있다. 규칙에 근거한 지식 베이스 시스템을 이용하여 병렬 탐지 과정에 대한 문제를 결정하기 위하여 추론 엔진을 사용한다. 일반적인 추론 시스템은 각 사용자 명령에서 일어나는 데이터를 사용하여 비정상적인 행위를 탐지하는 근원적인 증상을 제공하고 있다.

탐지 과정은 문제를 결정하기 위하여 추론 기관(Inference Engine)을 이용하고 있다. 일반적인 추론 시스템은 각 자원에서 발생된 데이터를 사용하여 잘못된 자원의 근원인 증상을 결정하는 것이다. 그러나, 네트워크 시스템에서 발생할 수 있는 침입-탐지의 요인은 감시를 하는 시스템에 의한 경고 보고 데이터의 처리 기능과 관련된 관리 기능의 협조를 통한 지식 베이스 그리고 이를 결정하는 규칙 베이스 시스템인 탐지기반 시스템의 탐지 프로세스와 경고 발생 프로세스의 결정만으로 가능하다. 경고 발생 과정은 규칙 베이스에 근거하여 경고 발생 제안 구조를 사용하고 있다.

경고보고 기능 서비스가 경고 데이터를 발생할 경우 어떻게 수행을 하게 되며, 관리 대상인 트랜스포트층 연결관리 대상과 로그관리 대상으로 침입-탐지를 연구하였다. 탐지 프로세스의 효율성을 보면 트랜스포트층 연결관리 대상을 적용한다. 침입-탐지 병렬 시스템의 네트워크 구성은 구성관리 성격으로 네트워크의 연결성을 나타낸다. 즉, 어느 감시를 하는 시스템과 세 개의 감시를 받는 시스템이 존재하고, 각 감시를 받는 시스템은 감시를 하는 시스템과 연결이 이루어질 수 있으며, 각 감시를 받는 시스템 사이에 연결 설정이 가능한 경우이다. 또한, 각 연결상태 사이에는 중간 노드 시스템이 존재할 수 있다. B 시스템은 여러 노드가 연결될 수 있는 A와 C사이의 중간 시스템이라고 가정된다. 다음 예제는 침입-탐지 결과를 추출하는 데 연결성의 결함은 복잡한 편이며, 구성관리와 관련되므로 선택되었다. 따라서, 전송 결함과 관련된 경고의 원인 추출은 다른 대상 시스템과의 연동이 필요하다.

감시를 하는 시스템이 세 개의 감시를 받는 시스템 A, B, C와 연결되어 있다고 가정하자. 만일 시스템 A가 B와의 전송 경고의 하나인 호출 설정 에러를 감시하는 시스템에게 제공하고, 시스템 C도 시스템 B와의 호출 설정 에러를 감시하는 시스템에게 제공할 경우, 지식 베이스를 사용하여 탐지한다. 두 경고에 따른 지식과 감시를 하는 행위로부터 "감시를 받는 시스템 B가 호출설정 기능에 관한 침입-탐지가 있다."는 것을 발견할 수 있다. 발견된 탐지에 따라 경고 발생에 관한 사항들을 감시하는 시스템의 사용자에게 제공하거나, 경고 발생 시스템에게 요구할 수 있다. 따라서 경고 발생 프로세스는 침입-탐지 부분의 격리를 위한 통신 활동을 먼저 수행할 것이며, 다른 관리 기능 서비스, 혹은 사용자를 통하여 침입-탐지 문제를 해결한다.

호출 설정에 따른 사고 보고를 추론 규칙으로 나타내면 다음과 같다. 여기서, 네트워크의 연결은 →을 기호로 사용하였으며, 중간 노드의 표시는 제외하였다. A → B는 A와 B 사이의 연결을 의미한다.

fail fail success
A → B, C → B, A → C
---
결론: 당연히 탐지되지 못한다.

위의 추론 시스템과 지식 베이스를 통한 규칙 베이스의 침입-탐지가 성공적으로 감시되었는지에 대한 정보 구성은 다음과 같다.

1. IF (C1: The call-setup between A and B is failed)
   THEN A or B may be failed.
2. IF (C2: The call-setup between C and B is failed)
   THEN C or B may be failed.
3. IF (C3: The call-setup between A and C is successed)
   THEN (P1: B must be failed.)

1과 2의 추론으로 감시하는 시스템은 C3의 관리 활동을 실행하고 3의 추론 결과를 얻는다.

규칙 1, 2, 3으로 부터,

IF (C1 and C2 and C3) THEN P1

을 추론하여 새로운 규칙은 앞으로의 연결 설정 침입-탐지에 대한 추론 생성 규칙(production rule)으로 사용될 수 있으며, A와 C의 침입-탐지도 마찬가지로 추론될 수 있다. 또한 다른 침입-탐지에 관한 생성 규칙이 제공될 것이며, 탐지와 경고 발생 기능의 협조에 바람직한 지식들을 구성하고, 추론을 어느 정도 자동화시킬 수 있다. 따라서 연결과 같은 침입-탐지의 처리는 침입-탐지를 가지고 있는 시스템을 격리시키는 구성 관리의 기능을 수행하여 네트워크 연결을 다시 구성함으로 경고 발생한다.

위의 예제는 통신에 관한 침입-탐지의 추론 예제이나, 관리 대상 자체의 침입-탐지로 인한 환경에 관한 경고와 처리 경고 일부분의 발견은 간단한 혹은 경고 보고 데이터 자체로도 적당한 탐지를 얻을 수 있다. 예를 들면, 용량 임계 경고인 처리 경고의 타입은 시스템의 로그가 가득차여 있는지를 나타내는 경고를 제공함으로 감시하는 시스템은 탐지를 더욱 단순히 수행할 수 있다. 경고 발생 과정은 심도를 고려하여 로그를 빈 상태로 수행하던지, 혹은 다른 방법으로 비교적 쉽게 처리할 수 있다. 그러나, 관련된 경고의 보고가 계속하여 수령되는 경우에는 실제로 이전에 수령된 경고에 의하여 탐지와 경고 발생의 수행을 반복함으로 감시하는 시스템의 부하를 증진시킬 수 있다. 지식 베이스 상태에서 경고를 수령하여 탐지 프로세스가 처리하는 경우, 관련된 경고를 다시 처리하는 탐지 프로세스 수행의 낭비이며, 이를 사전에 방지하는 것은 감시하는 시스템으로 하여금 수령된 경고나 사건 보고를 처리할 수 있는 능력을 제공한다.

구현된 모델 시스템의 특징으로 기존 병렬 시스템의 경고 데이터 관련성을 지식 베이스를 사용하여 탐지한 상황을 살펴보고 탐지한다. 구현된 실시간 침입-탐지 병렬 시스템에서 노드의 실패에 따른 침입-탐지 문제의 추적을 위하여 구성된 규칙 베이스의 부분적인 표현은 다음과 같다.

```
IF (reception of session problem from
    other networks)  &
   (problem is already outstanding)
THEN (problem on any node)
```

위의 규칙 표현처럼 경고 보고의 관련성을 지식 베이스에 저장하여 탐지 시스템이 파악하고 처리할 수 있도록 하였다. 탐지 프로세스로 하여금 항상 관련된 경고가 존재하는지를 조사하도록 구성하였으므로, 관련된 경고가 감시하는 시스템에게 전달되면 탐지 프로세스를 가동시킴으로 불필요한 탐지 과정을 수행할 수 있다.

비정상적인 행위를 감지하기 위한 방법으로 편차에 의한 실시간 침입-탐지 병렬 시스템에 대한 측정 기준과 보안 기준에 대하여 알아 보자. 실시간 침입-탐지 병렬 시스템에 의해 생성되거나 보안관리에 의해 입력된 규칙들의 집합으로 비정상적인 행위는 일련의 사건들이 규칙 R의 왼편을 가동한다면 후속의 사건들이 R에 의해 예측된 대로 인정된 장기간의 패턴(예를 들어 R의 오른편)으로부터 크게 이탈하면 그 행위가 감지되어진다. 예를들면, 다음과 같은 규칙 R이 있다.

$$A - B \rightarrow (C = 100\%)$$

(A 뒤에 B가 따라 나오면 C는 항상 따르는 것이 기대된다는 것을 표시한다.)

그러면, 일련의 사건들은 다음과 같다.

$$A - B - D$$

처음 두 사건들은 A와 B는 연속해서 정확하게 수행되므로 오류가 없고, 세번째 사건 D는 규칙의 오른편에 맞추어지지 않기 때문에 패턴의 침해로써 고려될 것이다. 각 사건은 다수의 속성이 연관되어 복잡한 묘사가 될 것이다.

## 5. 실시간 침입-탐지 시스템의 분석과 비교

지금까지 다루어진 방법은 네 개의 주요 모듈로 구성된 원형으로 구현되었다. 자료 수집과 변환 모듈, 실시간 침입-탐지 병렬 시스템에 기초한 사용자 프로화일 생성 모듈, 예외 사항 감지 모듈과 사용자 연결 모듈, 이 시스템의 구조는 다음 [그림 5.1]에서 보여진다.

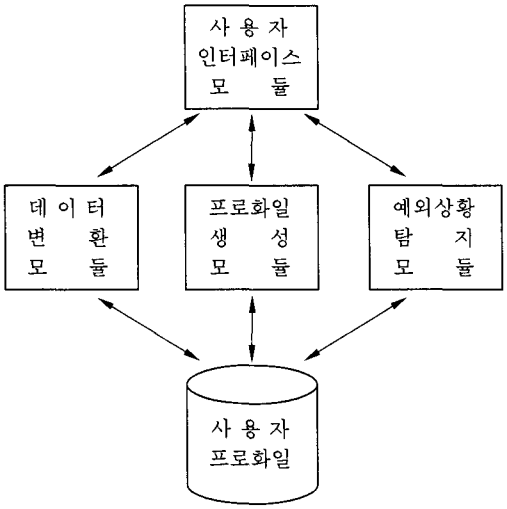


그림 5.1 시스템 구조

이 프로그램은 CONVAX 시스템에서 32 메가 바이트의 메모리를 가지고 실행된다. 추적 사건들은 사용자들의 내용들과 함께 모아진다. 적절한 경고 메시지는 사용자가 시스템에 들어올 때 표시된다. 프로화일의 생성은 많은 계산을 요구하기 때문에 자동적으로 적절한 계산 자원이 가능해질 때까지 처리를 뒤로 연기한다. 다음은 사용자 프로화일로부터 비정상적인 행위가 탐지된 프로그램으로부터의 보고이다.

Account Name : F00
Time of Analysis :
10-MAR-93 13:31:00 14:31:00
Security Events Collected : 302
Rules Based on Site Specific Policy : 9
Unusual Sequence of Events : 6
Average Deviation for Unusual
Sequences : 17%
Events Not Covered by Profile :
0 (0%)
Violations Based on Site Specific
Policy : 0 (0%)

비정상적인 행위가 있는 사용자 프로화일의 규칙들 중에서 하나는 다음과 같다.

RULE : 155
The occurrence of the event:

위의 규칙에서, 합계는 규칙이 발동되어진 총 시간이며, 엔트로피 값은 과거의 규칙에 의해 만들어진 예측들을 일관성 있게 나타낸다. 보안 담당자에게는 비정상적인 보안 사건에 대해 촛점을 맞출 수 있는 능력은 매우 중요하다. 보안 관리는 잠재적으로 나쁜 의도를 가진 침입에 관한 보안 사건들의 관계를 잘 파악하는 것이다. 왜냐하면, 어떤 의미와 보안 사건들의 순서는 확실한 순서를 제공하기 때문이다.

이 실험에 대한 분석을 통해, 한 그룹의 사용자들은 CONVAX/VMS 상에서 주어진 시간 내에 평균 46 개의 서로 다른 실행가능한 이미지를 작동시키며, 512 개의 다른 화일을 읽는 것을 알 수 있다. 두번째 사용자들의 그룹에 대한 비슷한 분석을 통해, 비슷한 시간 내에 다른 실행가능한 정보를 작동시키며, 276 개의 다른 화일을 읽는다는 것을 알 수 있다. 그러나, 두번째 사용자 그룹보다 광범위한 활동을 하는 첫번째 사용자 그룹에 대해 실시간 침입-탐지 병렬 시스템이 생성한 규칙을 분석함으로써, 첫번째 사용자 그룹은 활동의 연속이라는 측면에서 더욱 명확한 행동 패턴을 지니고 있다는 것을 알 수 있다.

0.25보다 작은 엔트로피 값을 가진 생성된 약 9.5%의 규칙들은 주어진 시간을 초과한 63.5%의 보안 사건에 대하여 설명할 수 있다.

이것을 통해 첫번째 사용자 그룹은 연속적 규칙에 대해 설명되어질 수 있는 일관된 행위를 했음을 알 수 있으며, 결과적으로 비정상적 시스템의 탐색 감도는 증가한다.

## 6. 결 론

기존의 감시 시스템은 단지 비밀번호 방법에 의해서 시스템을 보호하고 있지만, 본 논문에서는 비밀번호 방법에서 탐지될 수 없는 여러가지 내부적인 침입들을 방지할 수 있는 방법들에 중점을 두고 연구되었다. 대부분의 시스템 침입의 경우는 합법적으로 작업하는 사용자와는 현저하게 다른 작업을 많이 수행함으로써 시스템 상에서 탐지될 수 있으며, 프로토타입 IDPS는 독립적으로 시스템 상에서 수행될 수 있었으며, 네트워크를 통해서 대상 시스템으로부터 감시 레코드(Audit record)를 얻은 것으로 정보 보호에 위배되는 행위들을 실시간 내에 탐지하기 위해 시스템에서 독립된 방법을 제공하게 된다.

본 연구에서는 시뮬레이션을 통해 대부분 사용자의 활동에서 연속 패턴의 의미있는 수를 발견할 수 있으며, 사용자에 대해서 설정된 패턴을 통해 즉각적인 대응이 가능하다. 차세대 실시간 침입-탐지 병렬 시스템은 더욱 강력하게 탐지될 수 있는 기능을 갖도록 신뢰성 있고 견고한 설계가 될 것이다. 통신 네트워크에서 위험도가 높은 환경하에서 하나 이상의 프로세서가 침입-탐지를 발생할 경우를 대비해서 목표가 되는 시스템만 집중적으로 침입-탐지 시스템을 수행하는 것이 아니라, 분산 정책에 의해서 부분적으로 공동 감시 영역을 중복함으로서 침입-탐지가 많이 일어나는 상황에도 반드시 탐지되고 보안 관리자에게 즉시 보고될 수 있는 실시간 IDPS 체제가 구현되어야 할 것이다. 이러한 분산처리 환경하에서 일어날 수 있는 통신 문제는 아직도 많이 존재하므로 강력한 시스템 구현이 장차 연구되어야 할 것이다.

## 참 고 문 헌


[1] H.S. Teng, An Expert System Approach to Security Inspection of a VAX/VMS System in a Network Environment, Proceedings of the 10th National Computer Security Conference Baltimore, Sept, 1987.

[2] J.P. Anderson, Computer Security Threat Monitoring and Surveillance, James P. Anderson Co., Fort Washington, PA, April 1980.

[3] T.F. Lunt, Automated Audit Trail Analysis and Intrusion Detection : A Survey, Proceedings of the 11th National Computer Security Conference, Baltimore, MD, October, 1988.

[4] D.E, Denning and P.G. Neumann, Requirements and Model for IDES a Real-Time Intrusion Detection System, Computer Science Laboratory, SRI International, 1985.

[5] M.M. Sebring, Eric W. Shellhouse, R. Alan Whitehurst, Expert Systems in Intrusion Detection: A Case Study, Proceedings of the 11th National Computer Security Conference Baltimore, MD, October 1988.

[6] H.S. Vaccaro and G.E. Liepins, Detection of Anomalous Computer Session Activity, Proceedings of the 1989 IEEE Symposium on Security and Privacy, May 1989.

[7] K. Chen, An Inductive Engine for the Acquistion of Temporal Knowledge, Ph.D. Thesis, Department of Computer Science, Univer-sity of Illinois at Urban-Champaign, 1988.

[8] G. DeJong, An Approach to Learning from Observation, a chapter in Michalski, R.S., Carbonell, J.G. and Mitchell, T.M. (eds), Morgan Kaufmann Inc., 1986.

[9] D. McDermott, A Temporal Logic for Reasoning About Processes and Plans, Cogntive Science, Volume 6, 1982, pp101-156.

[10] L.A. Rendell, A New Basis for State-space Learning Systems and a Successful Implementation, Artificial Intelligence, pp369-392.

[11] L.A. Rendell, A General Framework for Induction and a Study of Selective Induction, reprot No. UIUCDCS-R-86-1270, Department of Computer Science, University of Illinois, Urbana, Illinois, April, 1986.

[12] N. Rescher, A. Urquhart, Temporal Logic, Springer-Verlag New York-Wien, 1971.

[13] B. Russell, The Principles of Mathematics, W. W. Norton & Company Inc., New York, 1974.

[14] P.E. Utgoff, Shift of Bias for Inductive Concept Learning, In Machine Learning, An Artificial Intelligence Approach, Volume II, R.S.Michalski, eds. Morgan Kaufmann Publishers, Inc., 1986.

[15] P.H. Winston, Learning Structural Descriptions from Examples, chapter in the book, The Psychology of Computer Vision, P.H. Winston(Ed.), McGraw Hill, 1975.

[16] 데이터 보호 기반 기술 연구(Ⅰ, Ⅱ), 한국 전자 통신 연구소.

[17] 암호학 입문, 한국 전자 통신 연구소.

## 부 록

;;; Prolog 프로그램: 시간 침입-탐지 병렬 시스템

Time-based Inductive Machine Output

```
;;; Following are some of the more
    interesting rules discovered by TIM.
;;;
rule7291: #S(RULE DESCRIPTION
   ((NIL(AND(TREATMENT(X)=A))(((80
      87))((65 67))(=((15 20)))NIL T T))
    (NIL(AND(TREATMENT(X)=C))(((140
      157))((32 50))(=((90 125)))NIL T T)))
   CHILDREN NIL
   PARENTS(rule7044 rule7002)
   COVERS-EPISODES(episode6885
      episode6884 episode6886 episode6887)
   ENTROPY 0.56233513
   EMER 0.5004024
   P-COVERED 3
   N-COVERED 1
   TOTAL-COVERED 4)

rule7266: #S(RULE DESCRIPTION
   ((NIL(AND(TREATMENT(X)=C))(((140
      157))((32 32))(=((90 125)))NIL T T)))
   CHILDREN NIL
   PARENTS(*ROOT*)
   COVERS-EPISODES(episode6886
      episode6884 episode6885)
```

```
   ENTROPY 0
   EMER 0
   P-COVERED 2
   N-COVERED 0
   TOTAL-COVERED 2)

rule7024: #S(RULE DESCRIPTION
   ((NIL(AND(TREATMENT(X)=C))
   (((80 87))((67 67))(=((15 20)))NIL T T)))
   CHILDREN NIL
   PARENTS(*ROOT*)
   COVERS-EPISODES(episode6886
      episode6884 episode6885)
   ENTROPY 0
   EMER 0
   P-COVERED 2
   N-COVERED 0)

rule7044: #S(RULE DESCRIPTION
   ((NIL(AND(TREATMENT(X)=C))
  (((140 157))((32 50))(=((90 125)))
   NIL T T)))
   CHILDREN (rule 7291)
   PARENTS(*ROOT*)
   COVERS-EPISODES(episode6889
   episode6887 episode6886 episode6884
   episode6885)
   ENTROPY 0.67301166
   EMER 0.6365142
   P-COVERED 3
   N-COVERED 2
   TOTAL-COVERED 5)

rule7002: #S(RULE DESCRIPTION
  ((NIL(AND(TREATMENT(X)=C))
  (((80 87))((65 67))(=((15 20)))
   NIL T T)))
   CHILDREN (rule7291)
   PARENTS(*ROOT*)
   COVERS-EPISODES(episode6888
   episode6887 episode6886 episode6884
   episdoe6885)
   ENTROPY 0.67301166
   EMER 0.6365142
   P-COVERED 3
   N-COVERED 2
   TOTAL-COVERED 5)
;;; learning from multiple sequence

;;; The fule to be found is
;;;  (exist()=1)x(forall y(linked(x y))) →
        k in two time units
;;;
;; Definition of "linked", which is a
        commutative binary predicate
;; (for undirected graphs).
     The classifying event is defined as
;; the propostion "k"
;;
(define-functor 'linked:commutativity
     t:arity 2:type 'boolean)
(define-functor 'k:commutativity t:arity
    p:type 'boolean)
(define-k-exp'(k()=t))
;;
;; Definitions of assumptions
;;
(define-asusmptions
   :closed-world t
   :sequence t
   :induction-type 'characteristic
   :strategy 'conservative
   :labeling 'event-labeling
   :single-named-object nil)
;;
;; Definition of backgound knowledge: a
```

```
   node is assumed to be linked
;; to itself.
;;
(define-bk-rules'((forall x(implies
  t(linked(x x) = t)))))
;;
;; Allows no embiguity so that only
  perfect rules will be found.
;;
(setq uniformness-threshold 0)

; Definition of episode #1
;
(define-event e11
  (make-event
      :objects '(a b c d)
      :description'(and(linked(a b)))
      :time 1))
(define-event e12
  (make-event
      :objects'(a b c d)
      :description'(and(linked(a b))
          (linked(b c))(linked(c d)))
      :time 3))

(define-event e14
  (make-event
      :objects'(a b c d)
      :description'(and(linked(a b))
         (linked(c d)))
      :time 4))
(define-event e15
  (make-event
      :objects'(a b c d)
      :description '(and(linked(a b))
         (linked(b c))(linked(b d)))
      :time 5))
(define-event e16
  (make-event
      :objects'(a b c d e)
      :description'(and(linked(a b))
         (linked(b c))(linked(c d))
         (linked(a d))(linked b d)))
      :time 6))
(define-event k1
  (make-event
      :object()
      :description'((k))
      :time 7))
(define-episode'(e11 e12 e13 e14 e15
      e16 k1))
```

## □ 著者紹介


### 유 은 진(Yu, Eun-Jin)

1977년 2월 : 숭실대학교 전산학(학사)
1980년 9월 : 숭실대학교 전산학(석사)
1993년 3월 : 숭실대학교 전산학(박사과정)
1984년 3월 : 한국교육개발원 연구원
1987년 3월-현재 : 대유공업전문대학 조교수



### 전 문 석(Jun, Moon-Seog) 종신회원

1980년 2월 : 숭실대학교 전산학(학사)
1986년12월 : University of Maryland 전산학(석사)
1988년12월 : University of Maryland 전산학(박사)
1989년 8월 : Morgan State University 전산수학과 조교수
1991년 2월 : New Mexico State University 부설
Physical Science Lab. 책임연구원
1991년 3월-현재 : 숭실대학교 컴퓨터학부 부교수