네트워크 기반 침입탐지시스템 성능향상을 위한 소프트웨어 설계 원리

박종운*, 최홍민*, 은유진*, 김동규**

(주)시큐브* , 아주대학교**

The Software Design Principles to Improve Performance in Network-based Intrusion Detection Systems

Jong-Woon Park*, Hong-Min Choi*, You-Jin Eun*, Dong-Kyu Kim**

Secuve Co, Ltd* , Ajou University**

요 약

정보통신 인프라의 발달과 인터넷을 통한 멀티미디어 서비스 및 대용량 데이터의 처리 증가는 조직의 네트워크 환경의 고속화를 가져왔다. 이러한 네트워크 환경의 변화는 조직으로 유입되는 비정상적인 행위/사건을 감시하는 네트워크 기반 침입탐지시스템(Network-based intrusion detection system, NIDS)의 필요조건의 변화를 동반한다. 즉, 기존 NIDS 연구는 비정상적인 행위/사건의 정확한 판단과 이에 대한 대응 기술에 초점이 맞추어졌으나, 최근에는 이와 더불어 고속 네트워크 환경에서의 NIDS 성능저하를 최소화하기 위한 가용성 확보 기술에 대해 연구가 활발히 진행되고 있다. 따라서 본 논문에서는 고속 네트워크 환경에서 NIDS의 정상적인 운영을 위해 성능에 절대적인 영향을 미치는 요소를 결정하고, 각 요소별 효율적인 설계 원리를 제시한다.

I. 서론

오늘날 조직의 네트워크 환경은 사용자 및 데이터의 폭발적인 증가로 인해 초고속 시대로 변화해가고 있다. 이에 반해 모든 서비스의 네트워크 연결은 외부로부터의 위협 요인을 증가시켜 조직의 자산에 대한 보안을 강화시켜야 하는 당위성을 제공한다. 이러한 환경의 변화 속에서 NIDS는 네트워크 트래픽 정보를 이용해 조직 내/외부로부터의 위협을 판단하는 역할을 수행해왔다.

NIDS는 네트워크 패킷 정보를 기반으로 비정상적인 행위/사건을 판단하기 위해 자신이 보유한 침입유형 목록과의 비교를 수행한다. 이러한 NIDS의 특성은 네트워크 트래픽의 증가 및 새로운 공격유형 확산에 의해 처리 부하의 증가로 가장 중요한 네트워크 패킷 데이터의 손실을 유발할 수 있다. 즉 네트워크 대역폭의 증가 및 새로운 위협원의 증가는 NIDS 성능 문제와 밀접히 연관된 것으로, NIDS는 이러한 변화에 대응해야 한다.

2000년 말까지만해도 NIDS 시장은 100Mbps 이하의 패스트 이더넷(Fast Ethernet) 환경에 적합한 제품이었다. 그러나 2001년 이후 인터넷을 이용한 e비즈니스와 인터넷을 통한 생활업무가 증가함에 따라 네트워크를 통한 데이터는 기하급수적으로 증가하여 한 대의 NIDS로는 조직의 네트워크 트래픽을 처리하지 못해 복수 개의 NIDS를 운영하기에 이르렀다. 또한 로드밸런싱 제품과 같은 부가장비를 동원해 추가 비용을 들여 NIDS의 역할을 분산시키게 됐다. 최근에는 ISP(Internet Service Provider)나 금융권, 증권사, 통신사 등에서 대규모 데이터 전송을 위한 백본망을 구축하여 서비스하고 있고, 대규모 트래픽을 처리할 수 있는 네트워크 보안솔루션에 대한 요구가 더욱 증가함에 따라 NIDS는 고성능의 요구조건을 만족시켜야만 하는 상황에 이르렀다.

본 논문에서는 이와 같은 NIDS의 성능 요구에 부합하기 위해 기존 NIDS의 기능 모델 분석을 통해 성능향상을 위한 소프트웨어 설계 원리를 제시하고자 한다.

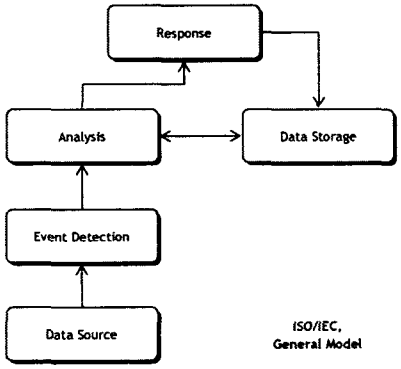
II. NIDS 참조모델(Reference Model)

네트워크 대역폭 및 새로운 공격유형의 증가에 의

한 NIDS의 성능 영향을 분석하기 위해서는 NIDS를 구성하는 기능 블록과 각 기능 블록간의 일련의 데이터 흐름을 살펴볼 필요가 있다. 따라서 본 장에서는 NIDS의 국/내외 표준 모델을 살펴보고 각 기능 블록의 역할 및 관련 데이터를 정의한다.

1. ISO/IEC NIDS 참조 모델 [6]

국제 표준규격을 개발하는 ISO/IEC JTC1의 한 분과인 SC27에서는 TR 15947 표준 문서에 아래 그림과 같이 NIDS의 기능 블록과 데이터 흐름을 정의하고 있다.

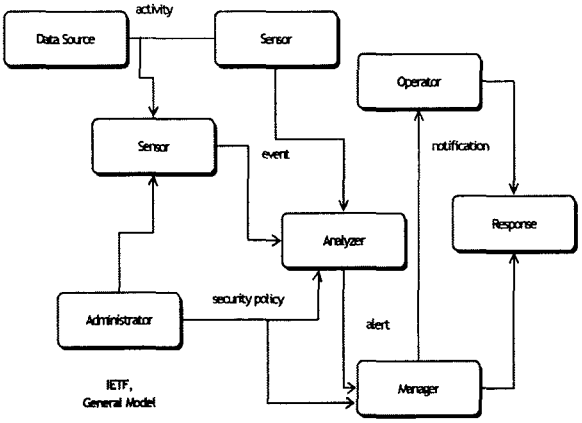


[그림 1] ISO/IEC NIDS 참조 모델

① 데이터 원본(Data Source): 네트워크 관련 정보 및 시스템 자원의 사용 정보
② 행위 탐지(Event Detection): 비정상적인 행위/사건을 판단하는 방안으로 여기서의 행위란 감시하고자 하는 특정 환경, 행동, 데이터의 발생 상황을 의미. 행위의 발생 빈도는 NIDS 성능 결정
③ 분석(Analysis): 행위를 분석하여 실제 침입이 발생할 확률을 결정. 이때 분석을 위해 행위 탐지 결과, 감시대상에서 발생하는 사용자들의 행동 양식, 각 개체 및 시스템 자원 사용 내역 등을 이용하여 비교하는 방법론에 따라 NIDS 성능 결정
④ 대응(Response): 침입 발생 시 이를 관리자에게 알려주는 방안
⑤ 데이터 저장소(Data Storage): 탐지된 행위 결과, 분석에 필요한 데이터, 알려진 침입에 대한 프로파일, 원시 데이터 등을 저장

2. IETF NIDS 참조 모델 [7]

인터넷 표준규격을 개발하는 IETF (Internet Engineering Task Force)의 IDWG (Intrusion Detection Working Group)에서는 아래 그림과 같이 NIDS의 기능 블록과 데이터 흐름을 정의하고 있다.



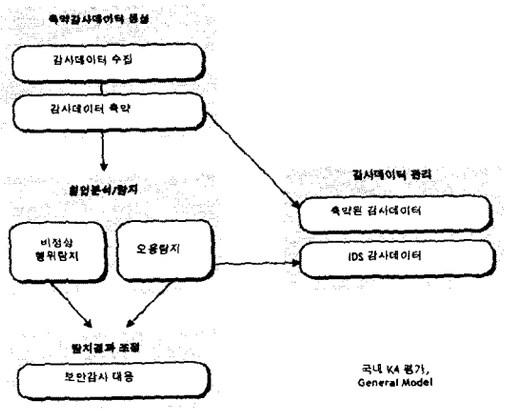
[그림 2] IETF NIDS 참조 모델

① 활동(Activity): 운영자(Operator)가 감시하고 센서(Sensor) 또는 분석기(Analyzer)에 의해 구분 가능한 데이터 수집 단위로, 활동의 빈도에 따라 NIDS 성능 결정
② 관리자(Administrator): NIDS 설치 및 설정, 보안정책의 적용 등 관련 행위에 대해 조직에서의 책임자
③ 경보(Alert): 감시대상이 되는 활동 중에서 분석기에 의해 비정상적인 행위 사실이 탐지되어 매니저(Manager)로 전송되는 메시지
④ 분석기(Analyzer): 센서에 의해 수집된 활동을 분석하여 비정상적인 행위/사건을 판단하는 구성 요소로, 비정상적인 행위/사건 판단 방법에 따라 NIDS 성능 결정
⑤ 데이터 원본(Data Source): NIDS가 비정상적인 행위/사건을 판단하기 위해 수집하는 활동을 생성하는 데이터로 네트워크 대역폭과 관계
⑥ 행위(Event): 센서에 의해 네트워크로부터 수집된 활동을 비정상적인 행위/사건 판단을 위해 분석기로 보내는 가공된 데이터, 센서에서 분석기로 행위를 분산시키는 방법에 따라 NIDS 성능 결정
⑦ 매니저(Manager): 운영자가 NIDS 시스템을 관리할 수 있게 해주는 구성 요소
⑧ 통보(Notification): 매니저가 운영자에게 비정상적인 행위/사건의 발생 사실을 알리기 위한 방법
⑨ 운영자(Operator): 매니저 운영 및 관리
⑩ 대응(Response): 비정상적인 행위/사건 판단 후 취해지는 행동
⑪ 센서(Sensor): 데이터 원본으로부터 감시를 위

한 데이터 수집 및 분석기로 가공된 정보 전달, 데이터 수집 및 데이터 분산 방법에 따라 NIDS 성능 결정

3. 국내 평가기준 NIDS 참조 모델 [8]

국내에는 침입탐지시스템 평가인증을 수행하는 평가기준에 아래 그림과 같이 NIDS의 기능 블록과 데이터 흐름을 정의하고 있다.

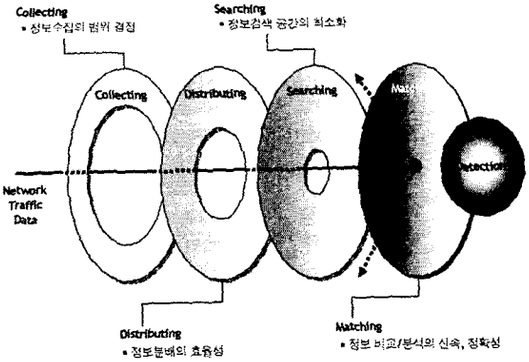


[그림 3] 국내 평가기준 NIDS 참조 모델

① 축약감사데이터 생성: 네트워크 정보를 수집하여 비정상적인 행위/사건 판단을 위한 정보 구성, 감시하고자 하는 범위에 따라 NIDS 성능 결정
② 침입분석/탐지: 축약감사데이터를 이용 비정상적인 행위/사건 결정. 탐지 및 분석 방법에 따라 NIDS 성능 결정
③ 탐지결과 조정: 침입탐지 기록 생성 및 이에 대한 대응
④ 감사데이터 관리: 축약감사데이터 및 침입 기록 저장 관리

III. NIDS 성능결정 요소 분석

2장에서 설명한 NIDS표준 참조모델은 비정상적인 행위/사건 판단까지의 일련의 데이터 흐름을 보여준다. 이때 데이터를 처리하는 각각의 기능 블록은 그 역할 및 입출력 값에 따라 성능에 큰 영향을 미치는지를 결정할 수 있다. 본 논문에서는 기능 블록과 이의 입출력 인자의 데이터 흐름 분석을 통해 네트워크 대역폭의 증가 시, NIDS 성능에 영향을 미치는 소프트웨어 요소로 Collecting, Distributing, Searching, Matching의 4가지 컴포넌트를 선택한다.



[그림 4] NIDS 성능 결정 요소

각 성능 결정 요소를 구체적으로 설명하면 다음과 같다.

3.1. 감사데이터 수집 범위 (Collection Range of Audit Data)

모든 NIDS 제품들은 트래픽 정보 수집을 기반으로 비정상적인 행위/사건을 결정한다. 이러한 특성은 초고속 네트워크 환경에서 NIDS 정보수집 기능의 병목현상을 유발하여 성능저하를 초래하는 가장 큰 이유가 된다. 따라서 감사데이터의 수집 범위를 효율적으로 줄이는 것은 NIDS 성능 결정에 큰 영향을 미친다.

3.2. 감사데이터 분배 (Distributing of Audit Data)

일반적으로 NIDS가 수집한 데이터는 NIDS를 구성하는 프로세스/쓰레드 단위의 비정상적인 행위/사건 판단 컴포넌트로 전송된다. 일반적으로 NIDS 정보 수집 모듈과 비정상적인 행위/사건 판단 모듈간에는 성능 향상을 위해 병렬 프로세스로 구현되기 때문에 모듈간의 효율적인 데이터 분배는 NIDS 성능 향상에 큰 영향을 미친다.

3.3. 검색 공간 최소화 (Minimization of Search Space)

NIDS에서 비정상적인 행위/사건을 결정하는데 사용되는 탐지패턴 목록은 프로세스/쓰레드 단위로 관리된다. 초고속 네트워크 환경에서 수집된 감사데이터는 탐지패턴 목록을 검색하여 비정상적인 행위/사건을 결정하기 위해 가능한 모든 경우의 수를 비교하게 된다. 이러한 특성은 탐지패턴 목록의 효

율적인 분할, 즉 로드밸런싱을 통해 수집된 감사데이터와의 연관성을 최소화하여 성능향상을 최대화할 수 있다.

3.4. 고속 패턴매칭 (High-speed Pattern Matching)

NIDS에서 가장 부하가 많이 걸리는 부분으로, 수집된 감사데이터는 연관된 탐지패턴 목록과의 비교를 통해 비정상적인 행위/사건을 판단한다. 대부분의 오용(Misuse) 기반의 탐지기법을 적용하는 NIDS는 수집된 감사데이터의 수가 증가할수록, 자신이 보유하고 있는 침입유형 목록이 증가할수록 급격한 성능저하를 보인다. 따라서 효율적인 고속 패턴매칭을 위한 설계는 성능 향상에 큰 영향을 미친다.

Ⅳ. NIDS 성능 향상을 위한 설계 원리

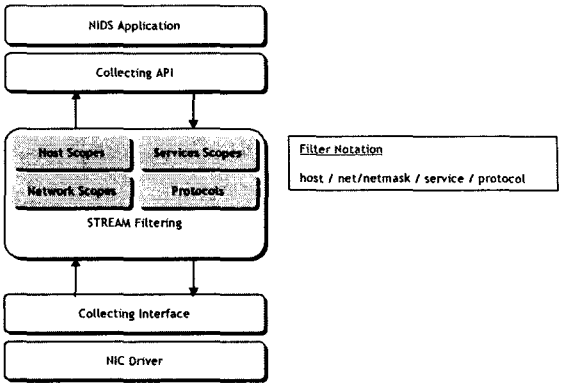
본 장에서는 3장에서 구분한 NIDS 성능 향상 요소별로 실제 개발 시 적용할 수 있는 설계 원리를 제시한다.

4.1. 감사데이터 수집을 위한 설계 원리

본 절에서는 NIDS의 정보수집 기능 블록이 데이터 수집 시, 비정상적인 행위/사건 판단을 위해 필요한 데이터만을 수집하도록 설계하여 정보수집 기능 블록의 정체 및 연계된 다른 기능 블록의 성능저하를 최소화할 수 있는 설계 원리를 제시한다.

설계 원리 1. NIDS는 필요한 정보만을 선별적으로 수집해야 한다. 이는 침입탐지 기능 블록의 입력 인자를 최소화하는 효과를 준다. 이를 위해 NIDS는 감시범위를 지정하여 필터링을 수행한다. 일반적으로 IP기반의 NIDS는 감시대상 영역을 지정하기 위해 Network Address, Subnet Mask 정보를, 감시대상 호스트를 지정하기 위해 IP Address 정보를, 감시영역에서 제공되는 서비스를 지정하기 위해 Port 정보를, 그리고 프로토콜 정보를 지정할 수 있다.

아래 그림은 일반적인 TCP/IP 스택(Stack)의 스트림(Stream) 처리 로직에 NIDS 정보수집 및 필터링 모듈을 추가한 것이다.



[그림 5] 감시범위 필터 구조

NIDS는 네트워크에서 발생하는 단 하나의 이벤트 처리를 위해 네트워크 디바이스로부터 NIDS 응용 프로그램까지의 일련의 데이터 처리 과정을 거친다. 이는 이벤트 중 가장 큰 부하가 발생하는 커널 공간 vs. 사용자 공간 사이의 전이(Switching)를 유발하는 것으로, 네트워크 이벤트 발생의 증가는 NIDS 성능과 반비례 관계를 가진다. 따라서 NIDS는 커널 공간과 사용자 공간 사이의 전이를 최소화하기 위해 발생하는 이벤트를 최소화한다.

설계 원리 2. NIDS는 감시대상 프로파일을 유지해야 한다. 이는 감시대상에 대한 운영체제(OS), 응용서비스(Application)에 대한 정보를 유지하여 수집된 데이터의 공격 유효성 판단에 이용한다. 즉 현재의 NIDS 감시범위 내에서는 수집된 데이터가 공격으로 의미가 없을 경우 필터링하여 침입 판단 기능 블록의 입력 인자를 최소화한다.

NIDS는 네트워크 이벤트의 선별적인 수집 과정을 거친 후 아래와 같은 감시대상 프로파일(Target Profile) 정보에 의해 NIDS 응용에서 가장 큰 부하를 차지하는 침입 판단 기능 블록의 입력 인자를 최소화할 수 있다.

앞서 설명한 디자인 패턴 1이 감시대상 범주 지정에 의한 하위 계층에서의 1단계 필터링이라면, 디자인 패턴 2는 감시대상 프로파일 정보 구성에 의한 상위 계층에서의 2단계 필터링으로 볼 수 있다.

```
# Target Profile for NIDS
target :- OS | SW # OS definition or
                                    # Software Product Lists
OS :- name, version                 # OS name and versions
SW :- name, version                 # Software Product name and versions
```

[그림 6] 감시대상 프로파일 구조

또한 디자인 패턴 2는 NIDS의 False Positive 확률을 줄이는 부가 효과를 얻을 수 있다. 실제로 감시대상 네트워크에 존재하는 운영체제가 유닉스 계열일 경우, 윈도우즈 기반의 취약성을 이용한 침입 시도는 공격 유효성이 없는 NIDS의 부하를 증가시키는 원인에 지나지 않는다. 따라서 감시대상 프로파일에 의한 공격 유효성 검증은 NIDS 성능저하를 최소화하고 이와 더불어 False Positive 확률을 줄이는 역할을 동시에 수행한다.

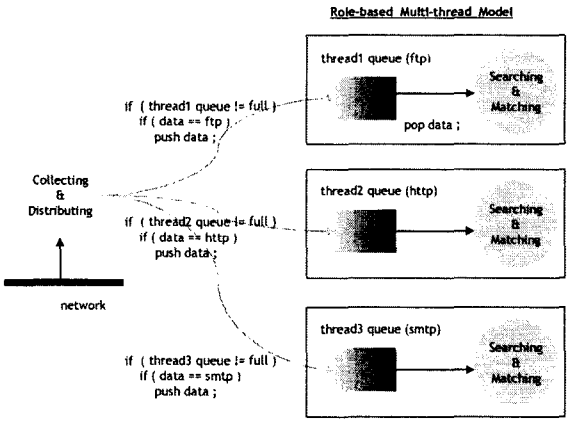
4.2. 감사데이터 분배를 위한 설계 원리

본 절에서는 NIDS의 정보수집 기능 블록과 침입탐지 기능 블록간의 이벤트 전달 비용을 최소화하여 중간에서의 정체 및 데이터 손실에 의한 성능저하를 줄일 수 있는 설계 원리를 제시한다.

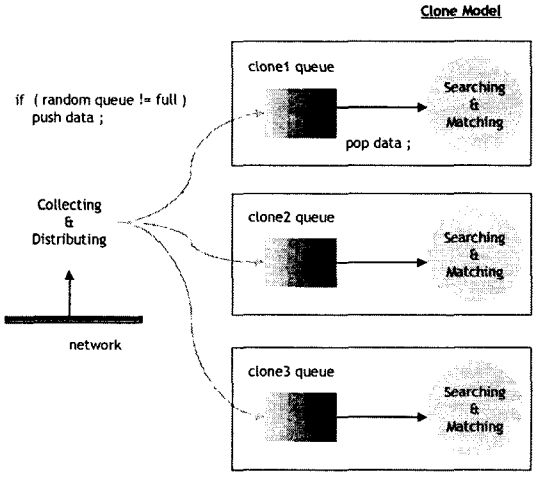
설계 원리 3. NIDS는 수집된 데이터를 침입 판단 기능 블록으로 전달하기 위해 공정한 로드밸런싱을 수행해야 한다. 일반적으로 NIDS가 수집한 데이터는 감시영역의 업무 특성에 따라 고유한 데이터 분포를 가진다. 이는 수집 기능 블록의 로드밸런싱이 트래픽 분석에 의한 특정 기준으로 이루어질 경우, 특정 환경에서 정체가 일어날 수 있음을 의미한다. 따라서 로드밸런싱이 효율적으로 수행되기 위해서는 첫째로 수집 기능 블록은 특정 기준에 의한 분배를 적용하지 않아야 하며, 둘째로 판단 기능 블록은 이를 위해 클론(Clone) 속성을 가져야 한다.

아래 [그림 7]은 일반적인 NIDS 침입 판단 기능 블록의 구현 모델이다. 각 판단 기능 블록마다 담당하는 역할이 정해져 있어 수집 기능 블록은 이를 기준으로 데이터를 로드밸런싱 하게 된다. 즉 그림에서 확인할 수 있듯이 각 판단 기능 블록의 역할이 특정 서비스를 기준으로 정해졌을 경우, 수집 기능 블록은 수집된 데이터의 서비스 필드 비교를 부가적으로 수행해야 한다.

이에 반해 [그림 8]은 침입 판단을 위해 클론 기반의 로드밸런싱을 구현한 모델이다. 그림에서 볼 수 있듯이 별도의 부가적인 데이터 분석이 필요 없이 수집 기능 블록에서 판단 기능 블록으로 데이터 전송이 이루어진다.



[그림 7] 역할 기반 로드밸런싱 모델



[그림 8] 클론 기반 로드밸런싱 모델

설계 원리 4. NIDS는 수집된 데이터를 침입 판단 기능 블록으로 전달할 때, 독립된, 직접적인 (Separated, Directed) 통신 메커니즘을 적용해야 한다. 일반적으로 데이터 수집 속도와 침입 판단의 속도 차이는 편중된 부하 발생 시 데이터 손실을 유발하여 데이터 손실을 초래한다. 따라서 각 기능 블록 사이의 통신 메커니즘 및 큐(버퍼)의 역할은 중요하다.

[그림 7/8]과 같이 일반적으로 프로세스 간 혹은 두 통신 객체 사이의 큐(버퍼) 역할을 하는 메커니즘은 1:1로 동작해야 한다. 큐가 복수 개인 경우 통신 객체에 의해 공유되면 부가적인 임계영역(Critical Region) 제어에 의한 부하가 발생한다. 또한 두 통신 객체는 제 3의 다른 객체를 통하지 않는 직접적인 통신 채널을 가져야 한다. 예를 들어 운영체제의 메모리 큐(Memory Queue)의 경우 커널 공간과 사용자

공간의 전이가 발생하여 스케줄러에 의한 부가적인 성능저하가 발생하지만, 공유 메모리(Shared Memory)의 경우 전이가 발생하지 않아 성능저하가 최소화된다.

4.3. 검색 공간 최소화를 위한 설계 원리

본 절에서는 NIDS의 침입탐지 기능 블록에서 비정상적인 행위/사건을 판단하기 위해, 수집한 데이터와 자신이 보유하고 있는 침입유형 (탐지패턴) 목록 비교 시 검색 시간을 최소화하여 성능저하를 줄일 수 있는 설계 원리를 제시한다.

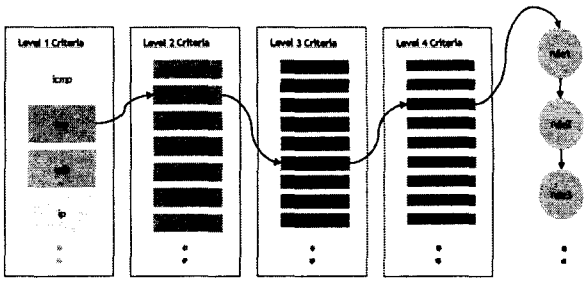
설계 원리 5. NIDS는 침입 판단 기능 블록에서 유지하는 침입유형 목록과 수집된 데이터의 빠른 비교를 위해 침입유형 목록을 다단계 그룹화 한다. 이 때 그룹화를 결정짓는 기준은 수집된 데이터를 통해 구분할 수 있고, 해쉬와 같은 형태의 데이터를 생성하여 한번에 접근 가능한 고유 값으로 선택해야 한다.

다음은 Misuse 기반 NIDS에서 침입유형 목록의 그룹화를 위해 사용하는 침입유형의 구조이다. RuleInfo는 침입유형에 대한 일반적인 정보를, RuleHeader와 RuleOption은 실제 침입 판정을 위해 사용되는 정보를 나타낸다.

```
# Pattern Structure
RuleInfo     := {id, name, ...}              # General information
RuleHeader := {protocols, ports, ip, ...} # criteria of grouping
RuleOption  := {flags, ttl, signatures, ...} # additional information
```

[그림 9] 침입유형 구조

이 중에서 침입유형 구조에 정의된 RuleHeader의 각 요소들은 침입유형 목록을 다단계로 그룹화 할 수 있는 기준으로 사용된다. 일반적으로 프로토콜, 포트 번호, IP 주소 값이 그룹화의 기준으로 이용된다. 이는 아래 그림과 같이 침입유형 목록의 멀티, 단일비교 검색 구조를 만들어 수집된 데이터와 침입유형 목록의 비교해야 할 수를 급격히 감소시킨다.



[그림 10] 침입유형 목록의 다단계, 단일비교 구조

4.4. 고속 패턴매칭을 위한 설계 원리

본 절에서는 NIDS의 침입탐지 기능 블록에서 비정상적인 행위/사건을 판단하기 위해 수집한 데이터와 자신이 보유한 침입유형 목록을 비교할 때 비교 시간을 최소화함으로 성능저하를 줄일 수 있는 설계 원리를 제시한다.

설계 원리 6. NIDS는 침입 판단 기능 블록에서 수집된 데이터의 Payload 부분과 침입유형에 정의된 Signature 부분을 비교하기 위해 고속의 멀티 패턴매칭 알고리즘을 사용한다. NIDS의 기능 요소 중 가장 부하가 큰 부분으로 데이터의 특성에 따라 가장 빠른 성능을 보이는 복수 개의 알고리즘을 적용할 수 있다.

오용(Misuse) 기반 NIDS의 경우 현재까지 알려진 침입유형 목록을 유지하며 네트워크 이벤트 발생 시 이와 비교를 수행하게 된다. 이는 새로운 침입유형이 증가할수록 단일 비교해야 할 일의 증가를 의미한다.

또한 일반적으로 Brute-force, Regular Expression, KMP(Knuth-Morris-Pratt), BM(Boyer-Moore) 등의 패턴매칭 알고리즘은 비교하고자 하는 데이터의 특성에 따라 성능에 차이를 나타낸다. 즉, NIDS가 보유한 침입유형을 표현하는 데이터의 특성에 따라 패턴매칭 알고리즘을 선택한다면 최적의 성능향상을 이룰 수 있다.

아래 그림은 이를 위한 침입유형 정의 시 패턴매칭 알고리즘을 정의하는 구조를 보여준다.

```
# Multi-Pattern Matching Definition Language
# algorithm 0 - Regular Expression
# algorithm 1 - KMP
# algorithm 2 - BM
Pattern 1 : Algorithm 1
Pattern 2 : Algorithm 2
Pattern 3 : Algorithm 1
Pattern 4 : Algorithm 0
Pattern 5 : Algorithm 1
.....
```

[그림 11] 멀티 패턴매칭 사용 구조

V. 결론

오늘날 많은 조직의 네트워크 환경이 초고속을 수용할 수 있도록 바뀌면서 NIDS는 고속 트래픽에 의한 성능저하 문제를 해결하고자 다양한 연구를 시도하고 있다. NIDS는 네트워크 트래픽 정보를 기반으로 비정상적인 행위/사건을 판단하는 특성을 가지므로, 트래픽의 증가는 NIDS 부하의 증가를, 부하의 증가는 NIDS가 비정상적인 행위/사건 판단의 근거로 이용하는 데이터의 손실을 초래한다.

따라서 본 논문에서 제시한 NIDS의 성능 결정 요소 4가지는 NIDS가 초고속 네트워크 환경에서 적응하기 위한 필수 고려사항이라 할 수 있다. 또한 각 성능 결정 요소마다 제시된 설계 원리는 NIDS 개발 시 지침이 될 수 있을 것이다. 물론 본 논문에서 제안한 것들이 NIDS 표준 모델의 분석을 통해 나온 것이므로, 새로운 NIDS 모델은 더욱 다양한 설계 원리를 만들어 낼 수 있을 것이다.

참 고 문 헌

[1] Peter Jackson. "Introduction to Expert Systems". International Computer Science Series. Addision Wesley, 1986.

[2] Sandeep Kumar, Eugene H, Spafford. "A Pattern Matching Model for Misuse Intrusion Detection System". Proceedings of the 17th National Information Security Conference, 1994.

[3] Sandeep Kumar, Eugene H, Spafford. "A Software Architecture to support Misuse Intrusion Detection". Proceedings of the 18th National Information Security Conference, 1995.

[4] R. Sekar, Y. Guagn, S. Verma, T. Shanbhag. "A High-Performance Network Intrusion Detection System". 1999.

[5] Wallace Wadge. "Achieving Gigabit Performance on Programmable Ethernet Network Interface Card". May 2001.

[6] IT intrusion Detection Framework, ISO/IEC TR 15947, Octorber 2002.

[7] Intrusion Detection Message Exchange Requirements. IETF IDWG, October 2002.

[8] 정보통신망 침입탐지시스템 평가기준. 정보통신부, 2000년 7월.