침입 탐지 시스템을 위한 효율적인 룰 보호 기법

(A Scheme for Protecting Security Rules in Intrusion Detection System)

손재민*, 김현성*, 부기동*
(Jae-Min Son, Hyun-Sung Kim, Ki-Dong Bu)

요 약 본 논문에서는 기존의 네트워크 기반의 침입탐지 시스템인 Snort에 존재하는 취약성을 해결하기 위한 방법을 제안한다. 현재 룰 기반의 침입탐지 시스템인 Snort에서는 룰 자체를 보호하기 위한 방법을 제공하지 못한다. 이러한 문제를 해결하기 위해서 본 논문에서는 해쉬함수를 이용하여 룰 자체에 대한 보호를 제공할 수 있는 기법을 제안한다. 이러한 기법을 통하여 룰 자체에 대한 무결성과 기밀성을 제공할 수 있을 것이다.

핵심주제어 : 정보보호, 네트워크 보안, 침입탐지 시스템, 룰 보호 기법, 스노트, 해쉬

Abstract This paper proposes a method to solve the weakness in Snort, the network based intrusion detection system. Snort which is the rule-based intrusion detection system dose not supports a protection method for their own rules which are signatures to detect intrusions. Therefore, the purpose of this paper is to provide a scheme for protecting rules. The system with the proposed scheme could support integrity and confidentiality to the rules.

Key Words : Information Security, Network Security, Intrusion Detection System, Rule Protection Scheme, Snort, Hashing.

1. 서 론

침입 탐지 시스템(Intrusion Detection System)의 목적은 호스트 또는 네트웍을 감시하며 자동적으로 침입을 탐지하는데 있다. 공격이 탐지되면 시스템 관리자에게 알려져야 하며 이에 따른 대응 행동이 필요하게 된다. 전통적으로 네트웍 기반의 오용탐지(Misuse Detection) 시스템이 이와 같은 작업을 따랐으며, 이런 방법은 전문가에 의해 네트웍 데이터로부터 룰(미리 선정된 몇 가지 특성들을 추출해 내어 내장된 정책)을 통해 빠르게 침입을 탐지하게 된다[1]. 하지만, 이러 방법에는 미리 선정된 룰들이 공격자에게 노출되어 룰로 정해지지 않은 다른 방법으로 공격이 이루어진다면 그 침입 탐지 시스템은 무용지물이 된다. 이런 문제점을 해결하기 위해서 일부 상용 IDS 제품들은 DB등을 이용하여 룰을 관리하고 보고하고 있다. 그러나 이러한 시스템들은 DB 관리를 위한 추가적인 보안 문제가 대두되고 있다.

본 논문에서는 보안에 대처하기 위해 도입되는 네트웍 침입 탐지 시스템(NIDS)중 현재 널리 사용되고 있는 Snort를 대상으로 룰 보호에 대한 방법을 제안한다. 일반적인 룰의 구성은 Header와 Contents로 되어 있지만 본 논문에서는 룰의 Header에 대한 보호 기법에 대해서만 논의한다. 제안한 방법은 룰에 기밀성과 무 결성을 제공하기 위해서 일 방향 해쉬 함수를 이용한다. 이러한 일 방향 해쉬 함수로써 본 논문에서는 MD5를 이용한다.

* 경일대학교 컴퓨터공학부
(School of Computer Engineering, Kyungil University)

본 논문의 구성은 2장에서 Snort와 MD5에 대하여 간략히 기술하고, 3장에서는 침입 탐지 시스템을 위한 룰 보호방법을 제시하고 생성된 룰을 이용한 탐지 방법을 제시한다. 4장에서는 구현결과를 보여주고, 마지막으로 5장에서 결론을 내린다.

2. 배 경

본 장에서는 침입 탐지 시스템의 두 가지 탐지 모델을 살펴보고, 오용 탐지 모델의 하나인 Snort에 대해서 자세히 살펴본다. 또한 해쉬함수로 잘 알려진 MD5에 대해서 간략히 기술한다

2.1 침입 탐지 시스템

침입 탐지 시스템(Intrusion Detection System)은 시스템의 비정상적인 사용, 오용, 남용 등을 탐지하여 알려주는 시스템을 의미한다. 침입 탐지 시스템은 탐지 모델에 따라 크게 두 가지로 분류할 수 있다[2][3].

- 비정상 탐지 모델

비정상(anomaly) 탐지 모델은 정상적인 행위패턴에서 벗어난 행위를 탐지하는 방법으로, 컴퓨터 자원의 비정상적인 행위에 근거하여 정의된 모델을 이탈하는 경우를 침입으로 간주한다. 그 종류로는 통계적인 방법, 특징 추출 방법, 예측 가능 패턴 생성 방법, 신경망을 이용한 방법 등이 있다.

- 오용 탐지 모델

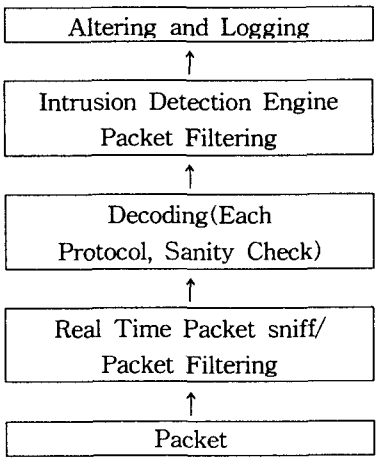
오용(misuse) 탐지 모델은 이미 알려진 공격패턴을 이용한 탐지 방법으로, 시스템이나 응용소프트웨어의 취약점을 통하여 시스템에 침입할 수 있는 잘 정의된 공격을 정의하여, 정의된 모델과 일치하는 경우를 침입으로 간주한다. 종류로는 조건부 확률방법, 전문가 시스템방법, 상태전이분석 방법, 키 입력 관찰 방법, 모델 기반 침입 탐지방법, 패턴 매칭 방법 등이 있다.

2.2 Snort

본 절에서는 패턴 매칭 방법을 사용하는 오용 탐지 모델중의 하나인 Snort 침입 탐지 시스템에 대해서 살펴본다.

Snort는 패킷 수집 라이브러리인 libpcap을 기반으로 한 네트워크 침입 탐지 시스템이다. Snort는 미리 정의된 침입 탐지를 위한 룰들을 이용하여 이와 일치되는 패킷들을 감시하고 기록하고 경고한다.

Snort의 기본구조는 [그림 1]과 같다. 네트워크 상에서 흘러 다니는 모든 패킷을 Filtering하고 패킷에서 필요한 정보를 추출하는 Decoding, 시스템에 존재하는 침입 탐지를 위한 규칙과 입력된 패킷 정보를 기반으로 침입 여부를 탐지하는 Intrusion Detection Engine과 경고 정보와 패킷 정보를 출력하는 Altering and Logging단계로 구성된다.



[그림 1] Snort의 기본구조

본 논문에서는 Snort의 룰을 보호하는 기법을 제안하므로 Snort의 룰에 대해서도 자세히 살펴본다.

Snort의 룰은 다음과 같은 기본 구조를 갖는다.

```
Action Protocol SourceIp SourcePort ->
        DestinationIp DestinationPort (Options...)
```

Action은 침입이 탐지되었을 경우 발생하는 행위이며, 종류에는 alert, log, pass, activate, dynamic이 있으며, 다음으로 Protocol Type이 명시되고, Source와 Destination 각각의 IP와 Port가 명시된다. 마지막으로 여러 종류의 Option들이 명시 될 수 있다.

룰을 설정할 때 유연성(flexibility)을 제공하기 위하여 IP Address, Port 부분은 고정된 값(Fixed Value), 임의의 값(Wildcard), 가변 값이나 범위 값(Variable

or Interval)으로 표현된다. 다음은 실제 Snort룰의 예이다[4].

- Destination Port가 고정된 값인 경우
  ex) log tcp any any -> 192.168.1.1/32 **23**
- Source IP와 Source Port가 각각 임의의 값인 경우
  ex) log tcp **any any** <> 192.168.1.1/32 23
- Source와 Destination의 Port가 범위 값을 가진 경우
  ex) log tcp any **23:100** -> any **1024:**
- Destination IP가 변수 값인 경우
  ex) alert tcp any any -> **$HOME_NET** any

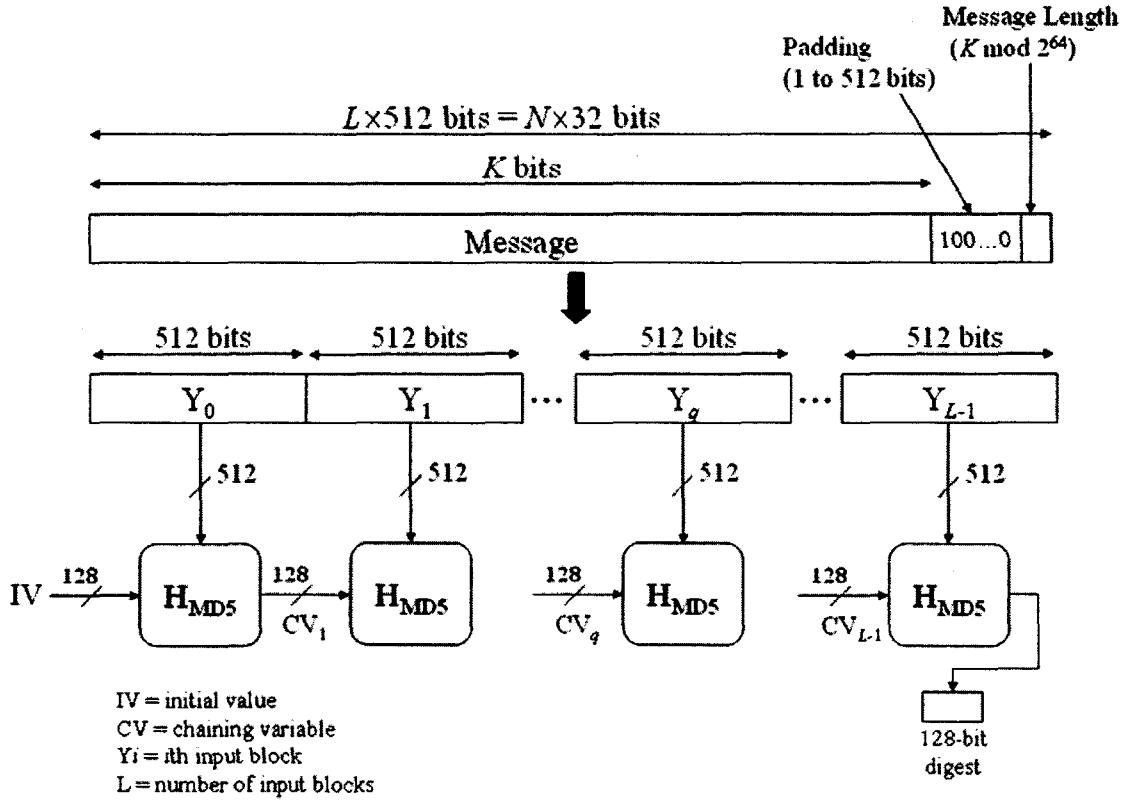
이러한 Snort의 룰 종류를 이해하는 것은 룰을 보호 기법을 제시하기 위해서 꼭 필요한 단계이다.

다음 장에서 이러한 여러 가지 종류의 룰을 해쉬 기법을 이용하여 효율적으로 보호할 수 있는 기법을 제시한다.

2.3 MD5 해쉬 기법

MD5(Message Digest 5) 메시지 다이제스트 알고리즘(RFC1321)은 MIT의 Ron Rivest에 의해 개발되었다. 전사적 공격과 암호 해독에 대한 우려가 고조된 지난 몇 년 동안 MD5는 가장 널리 사용된 안전한 해쉬 알고리즘이다. 이 알고리즘은 임의의 길이의 메시지를 입력으로 취하고 512-비트 블록으로 처리하여 128-비트 메시지 다이제스트를 출력으로 제시한다. 본 논문에서는 이 MD5알고리즘을 이용해 기존의 룰을 입력으로 취하고 128-비트의 출력을 16진수(hex)로 룰을 작성하였다.

[그림 2]는 다이제스트를 만들기 위한 메시지 전체 진행과정을 제시한 것이다. MD5의 알고리즘은 해쉬 코드의 전체 비트가 모든 입력 비트의 함수라는 성질을 갖는다. 기본 함수의 복잡한 반복은 잘 혼합되는 결과를 낸다. 즉, 임의로 선택된 두 개의 메시지가 유사한 규칙성을 가지고 있다 할지라도 같은 해쉬 코드



[그림 2] MD5를 이용한 메시지 다이제스트 생성

를 생성할 수 없다. RFC에서 Rivest는 128-비트 해쉬 코드에 대하여 MD5가 강하다는 것을 추측하였다. 즉, 같은 메시지 다이제스트를 가지는 두 개의 메시지를 추적하는 어려움은 2^{64} 연산의 정도인 반면, 주어진 다이제스트를 가지고 메시지를 찾는 어려움은 2^{128} 연산의 정도이다. 이러한 추측을 반증할 만한 어떠한 분석도 없다[5]. 본 논문에서도 MD5를 이용해 2^{128} 길이의 다이제스트 사용한다.

3. 룰 보호 기법

본 장에서는 잘 알려진 공개용 침입탐지 시스템인 Snort를 기반으로 룰을 보호할 수 있는 기법을 제시한다. Snort와 같은 룰 기반의 침입 탐지 시스템은 침입 탐지를 위해 시스템에 저장하고 있는 룰과 비교하여 침입 여부를 판별한다. 여기서 시스템에 저장된 룰을 Plain Text를 저장된다. 그러나 침입 탐지 시스템의 룰이 침입자에게 노출된다면 침입자는 이 룰을 통하여 시스템의 취약성을 알게 되므로 침입 입 탐지 시스템 자체가 무력화 될 것이다.

이러한 문제를 해결하기 위하여 본 장에서는 침입 탐지 시스템의 중요한 요소 중 하나인 룰을 해쉬함수를 이용하여 침입자로부터 보호할 수 있는 방법을 제안한다. 시스템의 전체적인 구성은 [그림 3]과 같다. 침입탐지 시스템의 룰의 보호를 위한 부분과 네트웍상의 패킷으로 침입탐지를 위한 부분으로 구성되어 있다.

본 장에서는 먼저 룰을 보호하기 위한 암호학적 요구사항에 대해서 살펴보고 2장에서 살펴본 여러 가지 룰 예제에 따른 룰 보호 방법을 제안한다. 그리고 제안한 시스템에서 침입 여부를 탐지하는 방법에 대해서 살펴본다.

3.1 보안 요구사항

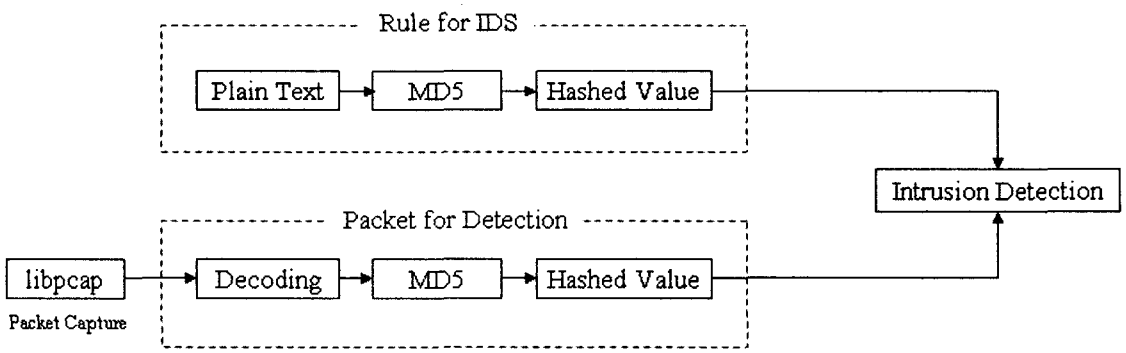
룰 보호를 위해서는 기밀성(Confidentiality)과 무결성(Integrity)이 제공되어야 한다. 먼저 본 절에서는 기밀성과 무결성에 대해서 살펴본다.

- 기밀성은 정보의 소유자가 원하는 대로 정보의 비밀이 유지되어야 한다는 원칙이다. 정보는 소유자의 인가를 받은 사람만이 접근할 수 있어야 하며, 인가되지 않은 정보의 공개는 반드시 금지되어야 한다. 기밀 자료는 비밀성이 노출되지 않도록 반드시 인가된 자에 의해서만 접근이 가능해야한다.
- 무결성은 정보는 정해진 절차에 따라, 그리고 주어진 권한에 의해서만 변경되어야 한다는 것이다. 정보는 항상 정확성을 일정하게 유지하여야 하며, 인가 받은 방법에 의해서만 변경되어야 한다. 무결성을 보장하기 위한 정책에는 정보 변경에 대한 통제뿐만 아니라 오류나 태만 등으로부터의 예방도 포함되어야 한다. 즉, 정보는 의도적이던, 우발적이던 간에 허가 없이 변경되어서는 안 된다[6].

본 논문에서는 이러한 기밀성과 무결성을 제공하기 위하여 룰에 해쉬함수를 적용한 기법을 제시한다.

3.2 보호된 룰 생성

본 절에서는 Snort의 룰을 보호하기 위해서 룰에



[그림 3] 시스템 전체 구성

해쉬함수를 적용하여 보호된 룰을 생성하기 위한 방법을 제안한다. [그림 4]는 보호된 룰의 기본 형식을 보여준다. 보호된 룰의 기본 형식은 각각의 필드가 고정된 값, 가변적인 값, 임의의 값 그리고 범위의 값의 네 가지 경우가 존재한다. 이들을 구별하기 위해서 본 논문에서는 추가적인 Flag 필드를 둔다.

F	SrcIP	F	SrcPort	F	DstIP	F	DstPort

F : Flag, Src : Source, Dst : Destination

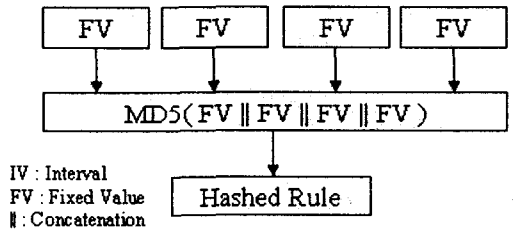
[그림 4] 보호된 룰의 형식

2.2절에서 제시되었던 룰은 크게 세 가지 형태로 다음과 같이 나뉠 수 있다.

- 고정된 값의 필드를 갖는 룰
- 범위의 값의 필드를 갖는 룰
- 가변값과 임의의 값의 필드를 갖는 룰

각각의 처리 방법은 다음과 같다.

1) 고정된 값을 갖는 필드



[그림 5] 고정된 형태의 룰 생성과정

룰을 생성하는데 있어서 고정된 값을 갖는 필드는 [그림 5]와 같이 바로 해쉬함수를 적용한다. 고정된 값은 다음 [예제 1]과 같이 수행된다.

[예제 1] 고정된 IP(203.230.91.200)를 가지는 경우

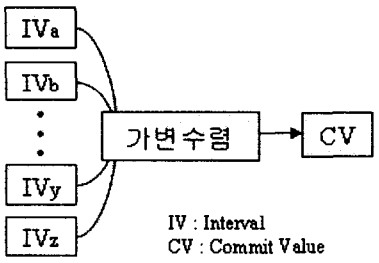
- 기존의 룰 : 203.230.91.200
- 해쉬함수 적용 : MD5("203.230.91.200");
- 보호된 룰 : e8e67f06b8d6fec9a6bc6062b0c4f41c

2) 범위 값을 갖는 필드

고정된 값의 필드 값과는 다르게 범위의 값을 갖는 필드는 그 각각의 값들을 모두 해쉬 취해서 각각의 값을 룰에 저장하면 된다. 그러나 그 값의 범위에 따라서 룰의 개수가 증가하는 문제가 발생하게 된다. 이러한 문제를 해결하기 위해서 범위 값을 갖는 필드에 대해서는 하나의 대표값으로 대응시키기 위한 [그림 6]과 같이 구성된 가변수렴 알고리즘과 어떤 값이 수렴된 결과 값과 일치하는지 확인하기 위한 [그림 7]과 같이 구성된 가변발산 알고리즘을 이용한다[7].

가변수렴 알고리즘은 다음과 같이 범위의 값에서 최소값과 최대값을 인자로 가지며 범위의 차인 Interval과 Interval의 배수이며 임의의 값인 CommitVal 그리고 최소값과 CommitVal의 차인 DecommitVal를 결과값으로 생성한다.

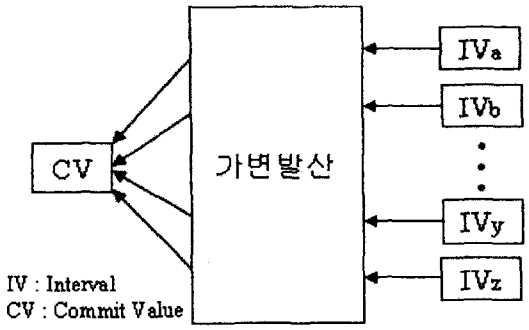
```
가변수렴(MinValue, MaxValue) {
  Interval = MaxValue - MinValue + 1;
  CommitVal = Random() * Interval;
  DecommitVal = MinValue - CommitVal;
  return Interval, CommitVal, DecommitVal;
}
```



[그림 6] 가변수렴 알고리즘의 동작방법

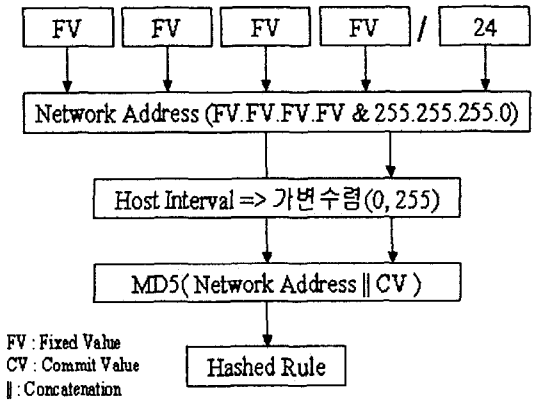
가변발산 알고리즘은 범위에 값에 속하는지 여부를 알아볼 값과 가변수렴 알고리즘에 포함된 Interval과 DecommitVal를 인자로 가지며 그 인자들에 의해 CommitVal2를 결과값으로 생성한다.

```
가변발산(CaptureNum, Interval, DecommitVal) {
  CommitVal2 = CaptureNum - DecommitVal;
  CommitVal2 -= (CommitVal2 % Interval);
  return CommitVal2;
}
```



[그림 7] 가변발산 알고리즘의 동작방법

위의 알고리즘에서 결과값으로 생성된 CommitVal와 CommitVal2의 일치여부에 따라 침입이 판단된다.



[그림 8] 범위를 가진 형태의 룰 생성과정

범위의 값을 갖는 필드는 [그림 8]과 같이 가변수렴 알고리즘을 사용하여 [예제 2]와 같이 단일한 값으로 매칭 시킨 후 해쉬를 수행한다.

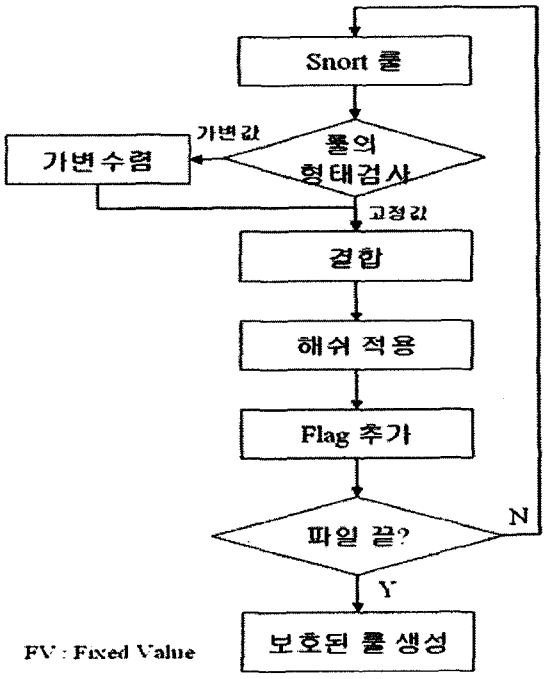
[예제 2] 범위가 있는 IP(203.230.91.200/24)의 경우

기존의 룰에서 Mask 연산을 통해 네트웍주소를 찾고 그에 속하는 호스트의 주소가 이 룰의 범위를 나타낸다. 호스트의 주소는 "203.230.91.0~255"과 같고, 범위의 값이므로 다음과 같은 가변수렴 알고리즘을 통해 대표값을 생성하고, 마지막으로 해쉬 알고리즘인 MD5를 적용하면,
"MD5("203.230.91.**가변수렴(0, 255)**")" 다음과 같이 보호된 룰이 생성된다.
- 기존의 룰 : 203.230.91.200/24
- 보호된 룰 : 437ac7b8311fdd92d0797d2cfc60d48e

3) 가변값과 임의의 값을 갖는 필드

가변값의 필드는 각 가변값이 가질 수 있는 값들을 하나의 룰로 변환하여 고정된 값의 필드일 경우와 같은 형태로 해쉬를 적용한다. 또한, 임의의 값을 갖는 필드의 경우에는 모든 값이 허용되어야 하므로 그 필드를 공백으로 두어서 처리한다.

이러한 세 가지 형태의 필드처리 방법을 통하여 룰에 기밀성을 제공할 수 있을 것이다. 룰 생성 시스템의 전체적인 구성도는 [그림 9]와 같고 이를 통해 룰 형식의 다양함에서 오는 문제를 효율적으로 해결할 수 있다.



[그림 9] 룰 생성 시스템 구성도

3.3 침입 탐지

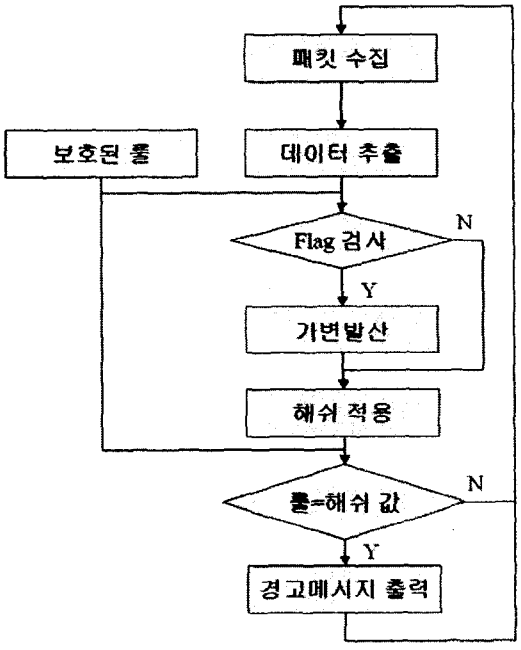
본 절에서는 3.2절에서 제안한 룰 생성기법에 의해 생성된 룰을 이용한 침입 탐지 수행 방법을 살펴본다. 전체적인 수행 과정은 [그림 10]과 같다. 이는 크게 고정된 해쉬 값을 가지는 경우와 가변된 해쉬값을 가지는 경우의 두 가지 형태로 나뉠 수 있다.

- 고정된 형태의 필드 : 입력된 패킷의 필드값을 해쉬 취해서 룰의 해당 필드의 해쉬값과 비교한다.

- 범위를 가진 형태의 필드 : 입력된 패킷의 필드값을 바로 해쉬하지 않고 가변발산 알고리즘을 통하여 계산된 결과값을 해쉬 취해서 룰의 해당 필드의 해쉬값과 비교한다.

제안된 룰 보호기법을 이용한 침입탐지 과정은 다음과 같다.

(과정1) : 수집된 원시 데이터에서 필요한 데이터를 추출한다[8].
(과정2) : 룰의 Flag를 보고 고정된 값을 가지는 룰인지를 판단한다. 범위를 가지는 값을 갖는 필드일 경우엔 추출된 데이터를 가변발산 알고리즘을 적용한다.
(과정3) : (과정2)의 결과값을 해쉬한다.
(과정4) : 룰의 해당 필드의 값과 (과정3)의 결과값을 비교하여 일치할 경우 경고 메시지를 출력한다.

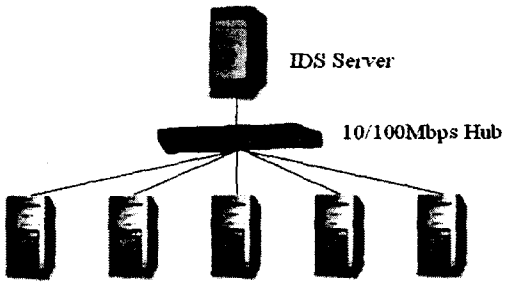


[그림 10] 탐지 시스템 구성도

4. 구현 결과

실험한 환경은 다음과 같다. [그림 11]은 전체 시스템 구성을 보여준다.

- **IDS Server**
  Pentium III-733MHz, Linux kernel-2.4.x, Snort 2.0
  Realtek RTL8139 Family PCI Fast Ethernet NIC
- **하위 Network**
  Pentium III-700MHz, Windows 2000 professional
  Realtek RTL8029(AS) PCI Ethernet Adapter



[그림 11] 네트웍 환경

본 논문에서는 MD5해쉬 알고리즘을 통하여 Snort의 룰을 보호하는 시스템을 구축하였다. 즉, 입력된 패킷의 값과 보호된 룰의 비교를 통해서 침입여부를 탐지하였다.

[그림 12]는 Snort 룰의 형식을 보여주며, [그림 13]은 Snort의 룰을 보호하기 위하여 본 논문에서 제안한 스키마를 적용한 결과를 보여준다. 여기서 Flag의 값이 'n'이면 고정된 형태의 룰을 의미하며, 그 값이 'y'이면 범위를 가진 형태의 룰을 의미한다.

고정된 형태의 룰인 경우는 바로 해쉬를 수행하였고, 범위를 가진 형태의 룰인 경우는 가변수렴 한 후 룰에 해쉬를 수행하였다.

[그림 12]에서 보는 바와 같이 기존의 룰에서는 룰의 내용을 알 수 있지만, [그림 13]와 같이 MD5 해쉬 함수를 통해 보호된 룰의 내용은 기밀성과 무결성이 보장된다.

[그림 14]는 본 논문에서 구성한 시스템의 탐지 결과를 보여준다.

5. 결 론

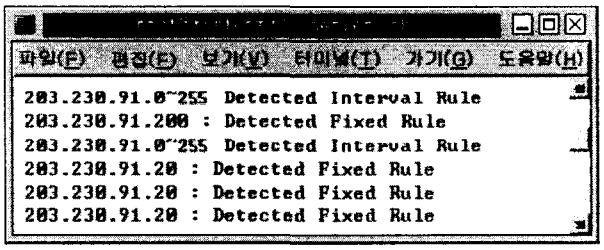
기존의 룰 기반 침입탐지 시스템인 Snort에서는 룰이 중요한 역할을 하지만 정작 그 룰에 대한 보호는 전혀 고려되지 못했다. 그래서 본 논문에서는 룰 기반 침입탐지 시스템의 하나인 Snort를 기반으로 룰을 보

```
log tcp 203.230.91.20 any -> 219.249.69.202 any (option)
log tcp 203.230.91.200 any -> 219.249.69.202 any (option)
log tcp 203.230.91.241 any -> 219.249.69.202 any (option)
log tcp 203.230.91.200/24 any -> 219.249.69.202 any (option)
```

[그림 12] Snort Rule

```
log tcp n 10951400d4950b10c0797ca891b0468e n 61c0b43a925df1c7e8a2827102ae77ae
log tcp n e8e67f06b8d6fec9a6bc6062b0c4f41c n 61c0b43a925df1c7e8a2827102ae77ae
log tcp n dfe0879144a47a9da5ab091b2e9159b0 n 61c0b43a925df1c7e8a2827102ae77ae
log tcp y 437ac7b8311fdd92d0797d2cfc60d48e n 61c0b43a925df1c7e8a2827102ae77ae
```

[그림 13] Hashed Rule



[그림 14] Detected Result

호하기 위한 기법을 제안하고 구현하였다. 그 기법은 룰에 기밀성과 무결성을 제공하기 위하여 해쉬함수인 MD5를 이용한다. 하지만 실제 Snort의 룰에서는 다양한 형태의 룰이 존재한다. 그 각각의 형태에 맞게 해쉬하기 위하여 가변수렴 알고리즘과 가변발산 알고리즘을 제안하였다. 제안된 기법을 통하여 기존의 침입탐지 시스템의 룰을 보호함으로써 보다 효율적인 보안을 제공할 수 있을 것으로 기대된다.

본 논문에서는 룰의 Header에 대한 보호기법을 제안하였지만, 앞으로는 룰의 Contents의 보호에 대해서도 연구가 필요하다. 또한 해쉬를 적용하여 룰을 보호하는 시스템, 암호화하여 룰을 보호하는 시스템, 및 기타 여러 경우에 대한 성능의 비교 분석에 관한 연구가 필요하며 현재 진행중에 있다.

참 고 문 헌

[1] Paul E.Proctor, *Practical Intrusion Detecton Handbook*, Prentice Hall, 2001.

[2] 한국 정보보호 센터, 침입탐지 모델 분석 및 설계, 1996.

[3] 한국 정보보호 진흥원 기술문서 - 네트워크 공격 기법의 패러다임 변화와 대응방안, 2000.

[4] Snort, http://www.snort.org

[5] William Stallings, *CRYPTOGRAPHY AND NETWORK SECURITY PRINCIPLES AND PRACTICE*, Prentice Hall, 2003.

[6] 조완수, *정보 시스템 보안*, 홍릉과학출판사, 2003.

[7] A Juels and M Wattenberg, "A Fuzzy Commitment Scheme", *In Proceedings of the second ACM conference on computer and communication security CCS'99*. Singapore, 1999, pp. 28-36

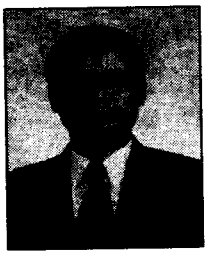
[8] 서승호 외 5, *TCP/IP 프로토콜 분석 및 네트워크 프로그래밍*, 정익사, 2002.

손 재 민 (Jae-Min Son)

2000년 3월 ~ 현재 경일대학교 컴퓨터공학과
(관심분야 : 정보보안, 네트워크 보안, IDS)

김 현 성 (Hyun-Sung Kim)

1996년 2월 경일대학교 컴퓨터공학과 공학사
1998년 2월 경북대학교 컴퓨터공학과 공학석사
2002년 2월 경북대학교 컴퓨터공학과 공학박사
2002년 3월 ~ 현재 경일대학교 컴퓨터공학과 교수
(관심분야 : 정보보안, 암호 알고리즘, 암호 프로세서 설계, IDS, PKI)



부 기 동 (Ki-Dong Bu)

1984년 경북대학교 전자공학과 전자계산기 전공
1988년 경북대학교 대학원 전산공학전공 공학석사
1996년 경북대학교 대학원 전산공학전공 공학박사
1983년 ~ 1985년 포항종합제철 시스템개발실
2001년 9월 ~ 2002년 8월 게이오대학 교환교수
1998년 ~ 현재 경일대학교 컴퓨터공학과 교수
(관심분야 : 데이터베이스, GIS, 시멘틱 웹)