해쉬함수를 이용한 침입탐지 시스템의 보호기법

손재민°, 김현성, 부기동
경일대학교 컴퓨터공학과
kim@kiu.ac.kr

A Scheme for Protecting Intrusion Detection System using Hash Function

Jae-Min Son°, Hyun-Sung Kim, Ki-Dong Bu
Dept. of Computer Engineering, Kyungil University

요 약

본 논문에서는 기존의 네트워크 기반의 침입탐지 시스템에 존재하는 취약성을 해결하기 위한 방법을 제안한다. 현재 대부분의 룰 기반의 침입탐지 시스템에서는 룰 자체를 보호하기 위한 방법을 제공하지 못한다. 이러한 문제를 해결하기 위해서 본 논문에서는 해쉬함수를 이용하여 룰 자체에 대한 보호를 제공할 수 있는 기법을 Snort를 기반으로 제안한다.

1. 서 론

침입탐지 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 미리 정의된 룰(Rule)을 통해 빠르게 침입을 탐지하게 된다[1]. 하지만, 이런 방법에는 미리 선정된 룰들이 공격자에게 노출되어 룰로 정해지지 않은 다른 방법으로 공격이 이루어진다면 그 침입탐지 시스템은 무용지물이 된다. 이런 문제점을 해결하기 위해서는 침입탐지 시스템의 룰이 보호되어야 한다.

본 논문에서는 네트웍 침입탐지 시스템(NIDS) 중 현재 널리 사용되고 있는 Snort를 대상으로 룰을 보호하기 위한 방법을 제안한다. 이러한 룰의 기밀성을 제공하기 위해서 본 논문에서는 일방향 해쉬함수로 MD5를 이용한다.

2. 배 경

본 장에서는 잘 알려진 공개용 침입탐지 시스템인 Snort에 대해서 살펴보고 해쉬함수로 잘 알려진 MD5에 대해서 간략히 기술한다.

2.1 Snort

침입탐지 시스템은 시스템의 비정상적인 사용, 오용, 남용 등을 탐지하여 알려주는 시스템을 의미한다. 침입탐지 시스템은 탐지 모델에 따라 크게 비정상 탐지 모델과 오용 탐지 모델의 두 가지로 분류할 수 있다[2][3].

오용탐지 모델은 이미 알려진 공격패턴을 이용한 탐지 방법으로, 공개용 침입탐지 시스템인 Snort가 이에 속한다. 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 있다. 다음으로 Protocol Type이 명시되고, Source와 Destination 각각의 IP와 Port가 명시된다. 마지막으로 여러 종류의 Option들이 명시 될 수 있다[4].

룰을 설정할 때 유연성(flexibility)을 제공하기 위하여 IP Address, Port 부분은 고정된 값(Fixed Value), 가변의 값(Variable), 임의의 값(Wildcard) 그리고 범위의 값(Interval)으로 표현된다. 다음은 실제 Snort룰의 예이다.

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

2.2 MD5 해쉬 기법

MD5(Message Digest 5) 메시지 다이제스트 알고리즘(RFC1321)은 MIT의 Ron Rivest에 의해 개발되었다. 이 알고리즘은 임의의 길이의 메시지를 입력으로 취하고 512-비트 블록으로 처리하여 128-비트 메시지 다이제스트를 출력으로 제시한다. 또한 해쉬코드의 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 해쉬코드를 생성할 수 없다[5]. 본 논문에서도 이 MD5알고리즘을 이용해 입력으로는 기존의

룰로 하고, 출력으로는 16진수(hex)의 128-비트 다이제스트로 사용한다.

3. 룰 보호 기법

본 장에서는 잘 알려진 공개용 침입탐지 시스템인 Snort를 기반으로 룰을 보호할 수 있는 기법을 제시한다. 먼저 룰을 보호하기 위한 암호학적 요구사항에 대해서 살펴보고 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 기밀성과 무결성이 제공되어야 한다.

기밀성(Confidentiality)은 정보의 소유자가 원하는 대로 정보의 비밀이 유지되어야 한다는 원칙이다. 정보는 소유자의 인가를 받은 사람만이 접근할 수 있어야 하며, 인가되지 않은 정보의 공개는 반드시 금지되어야 한다는 것이다. 그리고 **무결성**(Integrity)은 정해진 절차에 따라, 그리고 주어진 권한에 의해서만 정보가 변경되어야 한다는 것이다. 정보는 항상 정확성을 일정하게 유지하여야 하며, 인가 받은 방법에 의해서만 변경되어야 한다는 것이다[6].

본 논문에서는 룰에 기밀성과 무결성을 제공하기 위하여 해쉬 함수를 이용한다.

3.2 보호된 룰 생성

본 절에서는 Snort의 룰을 보호하기 위해서 룰에 해쉬 함수를 적용하는 방법을 제안한다. 그림 1은 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 1. 보호된 룰의 형식

2.1절에서 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

룰을 생성하는데 있어서 고정된 값을 갖는 필드는 바로 해쉬 함수를 적용한다.

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 가변수렴 알고리즘과 어떤값이 수렴된 결과 값과 일치하는지 확인하기 위한 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;
}
```

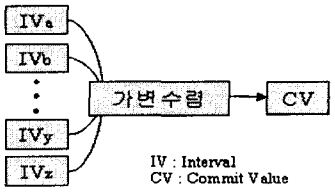

그림 2. 가변수렴 알고리즘의 동작방식

가변발산 알고리즘은 다음과 같이 범위의 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;
}
```

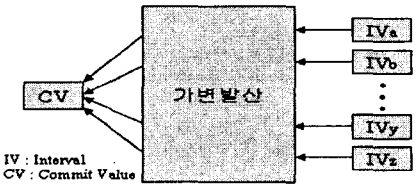

그림 3. 가변발산 알고리즘의 동작방식

위의 알고리즘에서 결과값으로 생성된 CommitVal와 CommitVal2의 일치여부에 따라 침입이 판단된다.

범위의 값을 가변수렴 알고리즘을 사용하여 예제 1과 같이 단일한 값으로 매칭 시킨 후 해쉬를 수행한다.

예제 1. 범위가 있는 IP(203.230.91.200/24)의 경우

기존의 룰	203.230.91.200/24
룰 생성의 중간단계	203.230.91.0~255(호스트 주소 구함) MD5("203.230.91.**가변수렴(0, 255)**");
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.

3.3 침입탐지

본 절에서는 3.2절에서 제안한 룰 생성기법에 의해 생성된 룰을 이용한 침입탐지 수행 방법을 살펴본다. 전체적인 수행 과정은 크게 고정된 해쉬값을 가지는 경우와 범위의 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.

(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.

4. 구현 결과

실험한 환경은 다음과 같이 그림 4에서 보여준다. 본 논문에서는 MD5 해쉬 알고리즘을 통하여 Snort의 룰을 보호하는 시스템을 구축하였다. 즉, 입력된 패킷의 값과 보호된 룰의 비교를 통해서 침입여부를 탐지하였다. 그림 5는 본 논문에서 구성한 시스템의 탐지 결과를 보여준다.

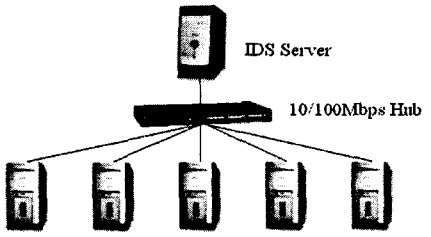

그림 4. 네트워 환경

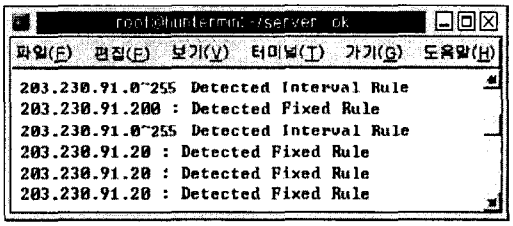

그림 5. 탐지 결과

5. 결 론

본 논문에서는 침입탐지 시스템에서 Snort를 기반으로 룰을 보호하기 위한 기법을 제안하고 구현하였다. 룰에 기밀성과 무결성을 제공하기 위하여 해쉬함수로써 MD5를 이용하였다. 제안된 기법을 통하여 기존의 침입탐지 시스템의 룰을 보호함으로써 보다 효율적인 보안을 제공할 수 있을 것으로 기대된다.

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 공격기법의 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 conferens 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.