패턴 분류를 이용한 침입탐지 시스템 모델
(An Intrusion Detection System Using Pattern Classification)

윤 은 준* 김 현 성** 부 기 동**
(Eun-Jun Yoon, Hyun-Sung Kim, Ki-Dong Bu)

요 약 최근 침입 탐지 시스템에 대한 관심이 증대되고 있다. 침입탐지 시스템에서 침입여부 확인을 위하여 패턴매칭 기법이 주로 사용된다. 기존의 패턴매칭 기법들은 다양한 공격 패턴들에 대한 패턴 비교 시간이 많이 소요되는 문제점이 있었다. 본 논문에서는 기존의 패턴매칭 기법들이 가지고 있는 문제점을 해결하기 위하여 새로운 침입탐지 시스템을 제안한다. 제안한 시스템은 효율적인 패턴 비교를 위하여 룰 패턴을 분류한다. 분류된 패턴은 매칭을 위하여 정형화된 트리로 구현한다. 그러므로, 본 논문에서 제안한 침입탐지 시스템 모델은 효율적으로 네트워크 침입탐지를 수행할 수 있다.

Abstract Recently, lots of researchers work focused on the intrusion detection system. Pattern matching technique is commonly used to detect the intrusion in the system, However, the method requires a lot of time to match between systems rule and inputted packet data. This paper proposes a new intrusion detection system based on the pattern matching technique. Proposed system reduces the required time for pattern matching by using classified system rule. The classified rule is implemented with a general tree for efficient pattern matching. Thereby, proposed system could perform network intrusion detection efficiently.

I. 서 론

네트워크를 통한 침입 유형은 다양해져 가고 침입 기술이 빠르게 발전하고 있지만 침입을 탐지할 수 있는 정보는 네트워크 패킷과 시스템 정보 등으로 한정되어 있다. 따라서, 한정된 정보를 이용하여 침입 탐지의 신뢰성과 탐지율을 높일 수 있는 침입탐지 시스템이 필요하다. 이러한 연구들 중 대부분은, 미리 알려진 침입방법을 규칙 또는 특정패턴으로 정의해 놓고 온라인상의 침입에 대해서 패턴과의 일치정도에 따라 침입을 탐지하고 있다[1][2][3].

네트워크 기반 침입탐지 시스템은 데이터 수집 단계에서 네트워크 패킷으로부터 데이터를 얻어서 가공하여 탐지에 이용하게 된다. 이때 수집된 패킷들은 프로그램상으로 처리하기 쉽도록 가공하여야 한다. 즉, 수집된 패킷을 알맞은 프로토콜에 따라 가공해 주어야 한다. 주로 사용되는 프로토콜이 Ethernet IP, TCP, UDP, ICMP, ARP 등이므로 이에 맞게 적절히 가공해 주어야 할 것이다[3].

가공된 패킷을 가지고 침입여부를 판단

* 경일대학교 대학원 컴퓨터공학과
** 경일대학교 컴퓨터공학과 교수

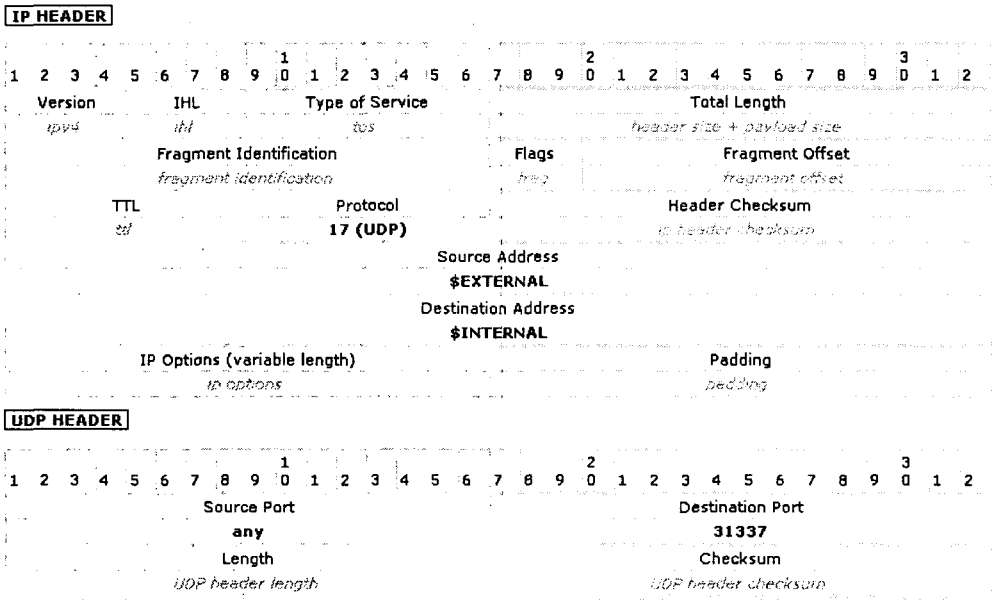

그림 1 Back Orifice 공격 패킷의 프로토콜 구조

시 침입검사 방법은 패턴매칭 방법을 사용하게 된다. 패턴매칭 방법은 알려진 공격방법을 일정 패턴 형식으로 가지고 있으면서 현재 행위와 공격으로 설정된 패턴을 비교하여 일치 여부로 침입을 탐지하는 방식이다. 패턴매칭을 통한 기존의 시스템에서는 시간이 많이 걸리는 문제점이 존재한다[3].

이러한 문제점을 보완하기 위한 방법으로는 공격 유형 중 탐지시 침입의 조건이 비슷하거나 중복되는 것이 있다는 점을 이용하여 중복되거나 비슷하게 탐지되는 공격을 유형별로 패턴화하는 방법 등이 있다[5].

본 논문에서는 기존의 패턴매칭 기법들이 가지고 있는 문제점을 해결하기 위하여 새로운 침입탐지 시스템을 제안한다. 시스템을 구성하기 위하여 먼저 효율적인 패턴 비교를 위하여 룰 패턴을 분류한다. 분류된 패턴은 매칭을 위하여 정형화된 트리로 구현되어 공격 패킷을 탐지를 할 때 사용한다. 이렇게 함으로서, 기존의 침입탐지 방법보다 효율적인 처리 시간을 제공할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 침입 유형 분류에 관하여 기술하고, 3장에서는 패턴 분류 기법을 이용한 침입탐지시스템 모델에 관하여 설계하고 구현을 한다. 마지막으로 4장에서는 결론을 맺는다.

II. 침입유형 분류

침입의 유형은 다양하다. 침입탐지 시스템을 구성하기 위하여 다양한 공격 유형에 대한 분석의 이론적 배경이 필요하다. 본장에서는 대표적인 6가지 침입 유형에 대하여 분석하고 룰을 구축하는 방법을 소개한다[2].

2.1 Back Orifice

트로이 목마와 트로이 목마에 대한 스캐닝은 1997년 중반부터 현재까지의 공격 중 많은 수를 차지한다. Back Orifice는 그림 1과 같이 디폴트 포트는 31337 UDP이다[1]. 그림 1에서 $EXTERNAL은 외부 네트워크 범위의 값으로 정의된 출발지 IP 주소를 가진다. $INTERNAL은 내부 네트워크의 범위의 값으로 정의된 목적지 IP 주소를 가진다.

Back Orifice 공격 패킷의 탐지 방법은 그림 1과 같이 패킷의 IP 헤더 프로토콜이 UDP이며 출발지 주소가 EXTERNAL, 출발지 포트가 any, 목적지 주소가 INTERNAL, 목적지 포트가 31337을 가지는 패킷의 Content Data 부분에서 Content 내용 중 "|ce63 d1d2 16e7 13cf 39a5 a586|" 패턴을 가지는 패킷에 관한 룰을 구축하면 된다[7].

2.2 Trinoo DDOS

분산 서비스 거부 공격은 한 호스트에 있는 악의적인 사용자는 하나 또는 많은 ICMP echo request를 조작하게 되는데, 이러한 패킷은 목적지 호스트의 위장된 출발지 IP와 확장 사이트의 브로드캐스트 주소를 갖으며, 이러한 많은 확장 호스트는 공격의 강도를 확대시켜 나가는 공격법이다[4].

최근까지 여섯 개의 다른 DDOS 프로그램이 알려져 있다. Trinoo, TFN(Tribre Flood Network), TFN2K, Stacheldraht(German for barbed wire), Mstream(mass stream), Shaft이다.

Trinoo 소프트웨어는 마스터라는 호스트를 통제하고, 데몬이라고 알려진 호스트를 공격하는데 사용한다. 클라이언트와 마스터간, 마스터와 데몬간의 통신은 TCP와 UDP를 사용하여 이루어진다. 표준 포트(27665, 27444, 31335)는 있지만, 이러한 포트는 변경될 수 있다. Trinoo는 희생 호스트에 임의의 목적지 포트에 단지 UDP Flooding을 보낼 수 있다. 그림 2.2는 Trinoo 공격유형 중 HELLO 공격 패킷의 프로토콜 구조이다.

Trinoo HELLO 공격의 탐지 방법은 UDP 헤더의 목적지 포트가 31335를 가지는 패킷의 Content Data 부분에서 Content가 "*HELLO*"인 패턴을 가지는 패킷에 관한 룰을 구축하면 된다.

III. 침입탐지 시스템

본 논문에서 제안한 침입탐지 시스템은 표 1과 같이 4가지 단계로 구성된다. 먼저 원시데이터를 수집하고 이렇게 수집된 데이터는 침입탐지에 이용하기 위해 축약되고 가공된다. 이 가공된 데이터를 바탕으로 분류기법이 적용된 룰패턴과 비교하여 침입 여부를 판별하며 그 결과에 따른 후속 조치를 취한다.

표 1 제안한 침입탐지 시스템 처리단계

단계	처리 내용
데이터 수집/가공	· 네트워크 패킷 수집 · 데이터 축약/가공
룰 패턴 분류	· 룰트리 생성 · 룰트리에 룰입력
침입 탐지	· 패턴 매칭
대응	· log 남김 · 관리자 경고

기존의 시스템에서 사용된 패턴매칭 방법에서는 다양한 공격 패턴들에 대한 패턴 매칭 시간에 많이 소요되어 효율적인 침입 판정이 어려웠다. 그러한 단점을 보완하기 위한 하나의 해결책은 다양한 패턴들의 규칙들을 분류하는 것이다. 예를 들면, 공개 침입탐지 시스템인 Snort의 룰 파일들은 공격 유형별로 분류가 이루어져 있다[2][6]. 하지만, 공격별 파일 분류가 되었다고 하더라도 많은 데이터의 패턴 매칭이 필요하다.

이러한 문제를 해결하기 위한 하나의 방편으로 본 논문에서는 공격 패킷과 설정된 룰패턴들과의 비교횟수를 간결화 할 수 있는 패턴 분류 방법을 제안한다. 또한 분류된 패턴은 매칭을 위하여 정형화된 트리로 구현하여 메모리상에서 효율적인 패턴 매칭을 제공한다. 그림 2는 본 논문에서 제안한 침입탐지 시스템의 구성을 보여준다.

3.1 데이터 수집 가공 모듈

이 모듈에서는 시스템에 전송되는 모든 네트워크 패킷을 수집하여 전처리 한다.

3.1.1 패킷 수집(Packet Capturing)

네트워크 패킷 수집 모듈은 libpcap-0.4 라이브러리를 이용하였다. Libpcap 라이브러리는 UC Berkeley에서 개발한 패킷 수

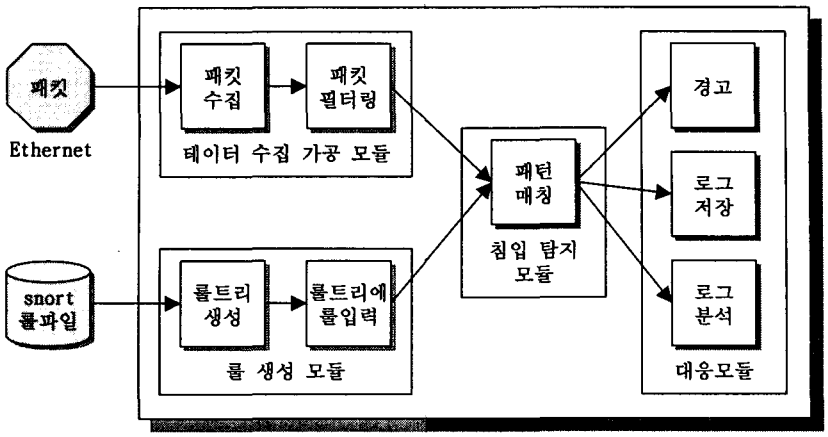

그림 2 침입탐지 시스템

집을 효과적으로 할 수 있게 만든 공용 라이브러리이다. Libpcap 라이브러리는 패킷 Capturing Library로써 시스템에 독립적으로 사용자 레벨에서 패킷을 잡을 수 있도록 해주는 장점이 있다. Libpcap 라이브러리는 기본적으로 BPF(BSD 패킷 Filter)를 지원하여 Ethernet device driver를 Promiscuous 모드로 열어서 모든 패킷들을 가져오기 때문에 사용자가 패킷 수집을 위한 드라이버를 별도로 만들 필요가 없다. Libpcap 라이브러리는 패킷을 네트워크 인터페이스로부터 획득하여 사용자가 접근할 수 있는 사용자 메모리에 복사한 다음 사용자가 지정한 함수를 메모리에 복사된 패킷에 수행하도록 한다[2][5].

표 2 필터된 패킷의 구조

구성요소	설명
protocol	프로토콜 (ip=1028, tcp=6, udp=17, icmp=1)
src IP	출발지 IP
src port	출발지 포트번호
dest IP	목적지 IP
dest port	목적지 포트번호
etc ...	etc ...

3.1.2 패킷 필터링(Packet Filtering)

Libpcap 라이브러리를 이용하여 패킷 수집 모듈을 작성한 후, 수집된 패킷을 패킷필터링 모듈에서 각 프로토콜(IP/TCP/UDP/ICMP)에 맞게 포인터를 옮겨가며, 내용중 일정한 규칙을 가지고 분석하여, 헤더 파일에 정의된 각 프로토콜의 구조체에 넣고, 다시 축약된 감사자료를 위한 네트워크 구조체에 넣어, 링크드리스트로 만들어 패턴매칭에 사용한다. 표 2는 필터된 패킷(Filtered Packet)의 구조체이다. 필터된 패킷의 구조체와 다음에서 제안한 룰트리의 패턴과 패턴매칭(Pattern Matching)을 한다.

3.2 룰 패턴 분류 모듈

많은 양의 필터링 패킷을 실시간으로 패턴매칭 하기 위해서 정형화된 룰트리를 이용한다. 이러한 공격 패킷의 빠른 탐지를 위해 룰 패턴 분류 모듈에서는 룰의 패턴을 분류하고 분류된 패턴에 맞는 룰트리를 생성한다.

3.2.1 룰트리 생성

네트워크를 통한 공격유형은 다양하여, 이러한 다양한 공격 유형을 패턴 매칭을 통해 하나 하나씩 비교하려면 상당히 많은 시간이 소요되며, 시스템의 과부하를 가져온다. 본 논문에서는 이러한 문제점을 보완하기 위해서 공개 침입탐지 시스템인

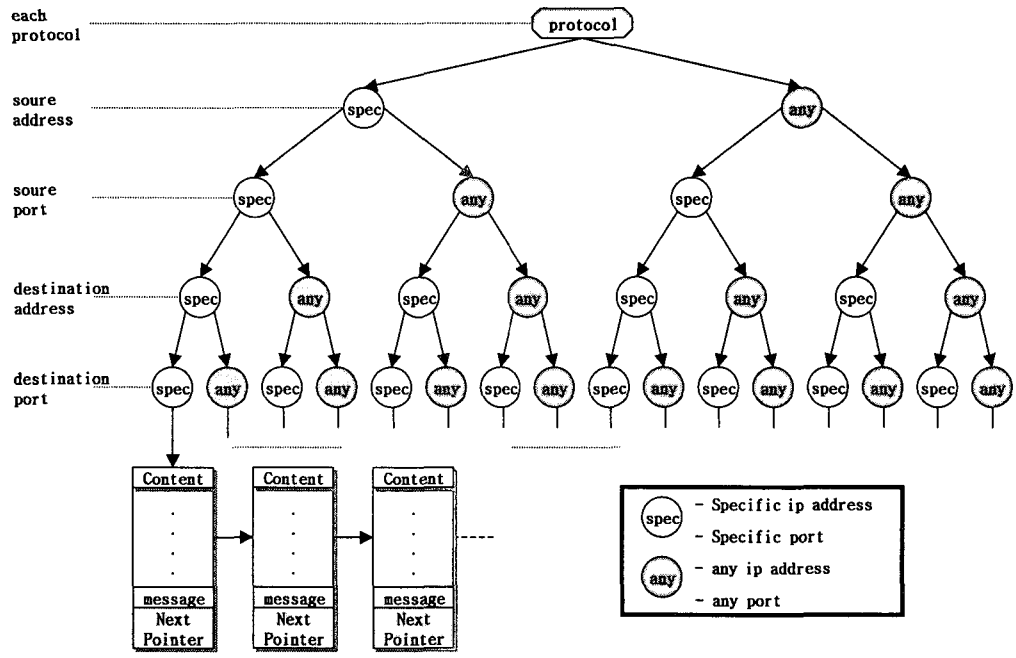

그림 3 프로토콜 단위의 패턴 분류 트리(PCT)

Snort의 룰 파일을 탐지룰로 사용하였다. Snort의 룰 파일들을 이용하여 패턴들을 각 프로토콜(ip, tcp, udp, icmp) 단위로 분류를 한 후, 정의된 출발지 주소와 포트번호 그리고 정의된 목적지 주소와 포트번호 인지를 분류하여 메모리상에서 룰 트리로 생성하면 빠른 패턴매칭이 이루어진다. 그림 3은 위의 사항을 기반으로 작성한 정형화된 패턴 분류 트리(PCT)이다.

3.2.2 패턴 분류 트리(PCT)

패턴 분류 트리(PCT)는 트리 부분인 룰헤드 부분과 트리 하위에 링크드리스트로 연결된 룰옵션 부분으로 나누어진다. 패턴 분류 트리(PCT)의 any 부분은 정의되지 않은 IP 주소와 Port 번호이며, spec 부분은 정의된 IP 주소와 Port 번호이다. 예를 들면, destination IP와 Port 번호가 192.168.0.1 80인 경우는 specific ip와 specific port 번호에 해당된다.

그림 3과 같이 정형화된 패턴 분류 트리(PCT)를 이용한 패턴매칭을 통하여 정상적인 패킷과 공격 패킷을 탐지한다. 따라서, 패턴을 분류하지 않은 방법보다 패턴비교횟수를 줄일 수 있으며 처리속도를 향상시킬 수 있다.

3.3 패턴 매칭(Pattern Matching)

패턴 매칭을 위해서 필터링한 패킷(Filtered Packet)의 프로토콜을 패턴 분류 트리(PCT)의 루트에서 비교하며, 계속해서 source ip address와 port 번호, destination ip address와 port 번호를 비교하여가며 진행한다. 패턴 분류 트리(PCT)의 단말노드는 링크드리스트로 구성된 rule content 부분과 rule option 부분을 패턴 매칭하여 침입여부를 빠르게 판정하게 된다.

그림 4는 각 프로토콜에 대한 6종류의 공격 패킷에 대한 패턴매칭과정을 도식화하였다. 표 3은 패턴매칭시 사용할 필터링한 6종류의 공격 패킷의 구조체 내용을 표로 나타내었다.

정형화된 패턴 분류 트리는 모든 공격 유형의 특징을 충분히 반영하였으며, 다양한 공격 유형에 대하여 처리 할 수 있도록 설계하고 구현되어 효율적으로 네트워크 침입탐지를 수행한다.

3.4 대응 모듈

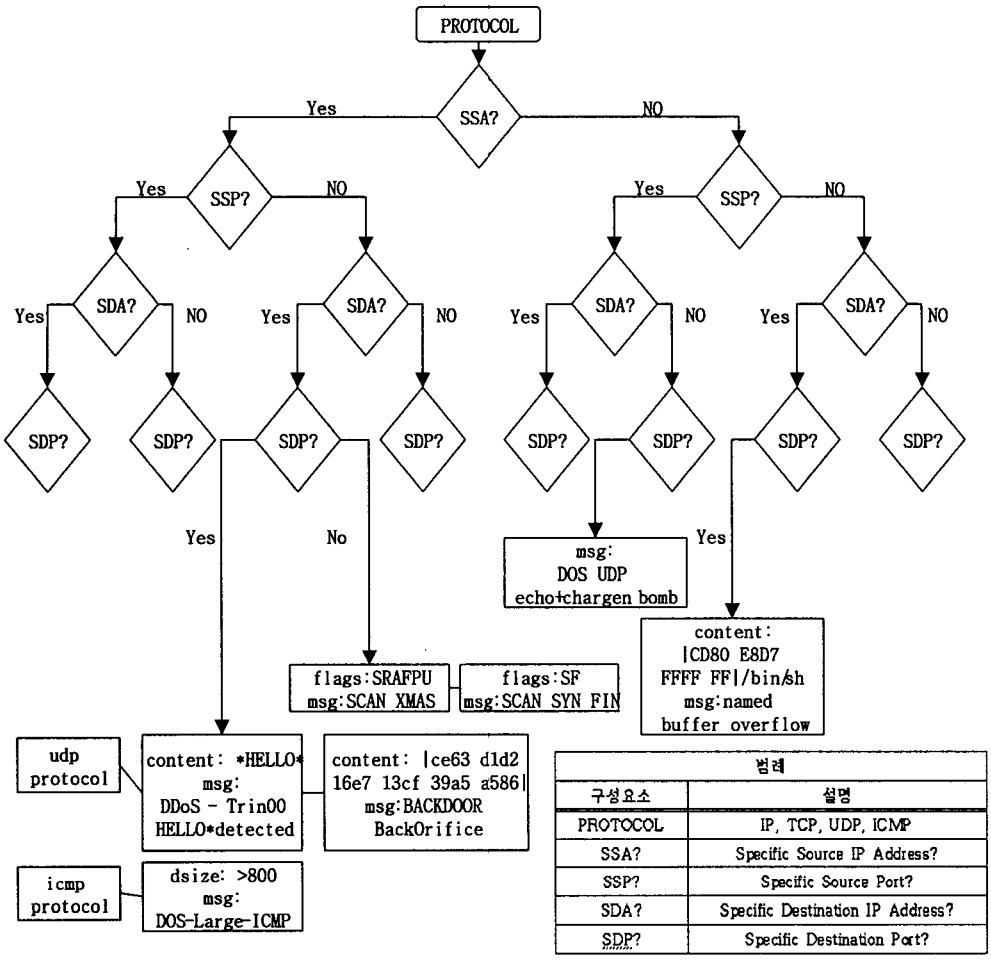

그림 4 각 프로토콜에 대한 패턴매칭과정

표 3 필터링된 패킷의 구조체

PROTOCOL	SSA	SSP	SDA	SDP	OPTIONS	MESSAGE	ACTION
udp	EXTERNAL_NET	any	INTERNAL_NET	31337	content: "|ce63 d1d2 16e7 13cf 39a5 a586|"	BACKDOOR BackOrifice access	alert
tcp	EXTERNAL_NET	any	INTERNAL_NET	any	flags:SRAFPU	SCAN XMAS	alert
tcp	EXTERNAL_NET	any	INTERNAL_NET	any	flags:SF	SCAN SYN FIN	alert
udp	any	19	any	7	없슴	DOS UDP echo+chargen bomb	alert
icmp	EXTERNAL_NET	any	INTERNAL_NET	any	dsize: >800	DOS-Large-ICMP	alert
udp	EXTERNAL_NET	any	INTERNAL_NET	31335	content:"*HELLO*"	DDOS Trin00 (*HELLO*detected)	alert

침입으로 판정받은 공격 패킷은 침입 시간, 침입자 주소, 공격 정보 등을 관리자에게 경고(alert)를 하고 데이터베이스에 공격 패킷의 대한 정보를 로그(log)로 남긴다. 관리자는 공격 패킷에 대한 분석을 통하여 효율적인 사후 대책을 수립한다.

IV. 결론

본 논문에서는 패턴 분류 기법을 이용한 침입 탐지 시스템을 구현하였다. 먼저, 공격 패킷에 대한 효율적인 패턴 비교를 위하여 룰 패턴을 분류하였다. 분류된 패턴은 매칭을 위하여 정형화된 트리로 구현되어 효율적인 패턴 매칭을 할 수 있었다. 구현한 시스템은 효과적인 패턴 분류와 분류된 패턴에 맞는 정형화된 트리를 생성함으로써 패턴 비교 횟수를 감소시킴으로써 탐지시간을 줄일 수 있었다.

향후 연구 과제로는 지능화된 공격 유형에 대한 패턴 분석과 신종해킹기법에 의한 공격 패턴을 패턴 매칭 이전 단계에서 정확하게 분석하여 정형화된 룰을 구축하여 오탐율을 줄이는 방법 연구가 필요하다.

참고문헌

[1] Stephen Northcutt, Judy Novak, "Network Intrusion Detection An Analyst's Handbook Second Edition". Infobook. 2001. 10.

[2] Kejji Takeda, Hiroshi Isozaki "네트워크 침입탐지 부정침입의 검출과 대책" 2000. 5.

[3] 한국전자통신연구원. "인터넷 침입차단 시스템". 1999. 12.

[4] 이근수, 박지현외, DDos 공격에 대한 탐지 및 추적시스템 제안, 한국정보보호학회 종합학술발표회 논문집, Vol.11, No.1. 2001.

[5] 김주영, 강창구외, 네트워크 패킷 분석을 통한 침입탐지 기법 개발, 한남대학교 컴퓨터공학과 논문지

[6] http://www.snort.org

[7] http://www.whitehats.com