



(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) 。 Int. Cl.

H04L 12/26 (2006.01)

H04L 12/56 (2006.01)

G06F 15/00 (2006.01)

G06F 13/14 (2006.01)

(11) 공개번호 10-2007-0078106

(43) 공개일자 2007년07월30일

(21) 출원번호 10-2007-0072587(분할)

(22) 출원일자 2007년07월20일

심사청구일자 2007년07월20일

(62) 원출원 특허10-2005-0134314

원출원일자 : 2005년12월29일

심사청구일자

2005년12월29일

(71) 출원인 충남대학교산학협력단

대전광역시 유성구 궁동 220번지 충남대학교

(72) 발명자 김규호

대전 서구 내동 151번지 코오롱아파트 5동 301호

강석민

대전 유성구 어은동 99번지 한빛아파트 103동 1806호

송일섭

대전 유성구 봉산동 202-11번지 4층 401호

권택근

대전 유성구 신성동 대림두레아파트 106동 902호

(74) 대리인 권오식

박창희

김중관

전체 청구항 수 : 총 4 항

(54) 차입 윈도우와 부분일치상태 기반 패턴 검색 방법

(57) 요약

본 발명은 차입 윈도우에 의한 고속 패킷 검색 시스템 및 그 방법에 관한 것으로, 인터넷으로부터 수신되는 직렬 패킷 데이터를 병렬 데이터 형태인 입력 패킷으로 변환하는 입력망 정합부와, 상기 입력 패킷 인입에 따라 패킷 인입 신호를 발생시키고 상기 입력 패킷을 버퍼링하여 출력하는 입력 패킷 버퍼부와, 적어도 하나의 공격 패턴 정보를 저장하고 있는 메모리와, 상기 입력 패킷 검색시 부분일치상태(Partial Match State) 기반의 바이트 슬라이딩 방식 또는 부분일치상태 기반의 점핑 윈도우 방식을 사용하여 상기 메모리에 저장된 적어도 하나의 공격패턴과 비교하여 검사하고, 공격패턴의 전체 또는 일부가 검출되면 공격패턴 추출신호를 출력하는 패킷 검색부와, 상기 공격패턴 추출신호에 따라 공격패턴이 속한 패킷흐름 제어신호를 출력하는 패킷 출력 제어부 및 상기 패킷흐름 제어신호에 따라 출력 패킷을 네트워크로 출력하지 않고 제거하는 출력망 정합부로 구성된다.

대표도

도 2

특허청구의 범위

청구항 1.

인터넷으로부터 수신되는 직렬 패킷 데이터를 병렬 데이터 형태인 입력 패킷으로 변환하는 단계;

상기 입력 패킷 인입에 따라 패킷 인입 신호를 발생시키고 상기 입력 패킷을 버퍼링하여 출력하는 단계;

상기 입력 패킷 검색시 부분일치상태(Partial Match State) 기반의 바이트 슬라이딩 방식 또는 부분일치상태 기반의 점핑 윈도우 방식을 사용하여 메모리에 저장된 적어도 하나의 공격패턴과 비교하여 검사하고, 공격패턴의 전체 또는 일부가 검출되면 공격패턴 추출신호를 출력하는 단계;

상기 공격패턴 추출신호에 따라 공격패턴이 속한 패킷흐름 제어신호를 출력하는 단계; 및

상기 패킷흐름 제어신호에 따라 출력 패킷을 네트워크로 출력하지 않고 제거하는 단계로 이루어지는 것을 특징으로 하는 차입 윈도우와 부분일치상태 기반 패턴 검색 방법.

청구항 2.

제 1항에 있어서,

상기 해당 입력패킷의 말단 검색시 부분일치상태를 해당 패턴의 해시값(Hash Value) 또는 부분일치 상태 고유번호(Partial Match State Unique ID)를 사용하고 이를 다음번에 입력패킷의 선두 검색시 전체일치상태를 판단하는 데 사용되는 것을 특징으로 하는 차입 윈도우와 부분일치상태 기반 패턴 검색 방법.

청구항 3.

제 2항에 있어서,

상기 부분일치 기반의 점핑 윈도우 방식에서,

해당 입력패킷 후미 말단에서의 검색 데이터가 점핑 윈도우의 크기보다 작을 경우, 필요한 데이터 크기만큼 직전에 검색된 데이터에서 차입해오는 차입 윈도우(Borrow Window) 방식을 사용하는 것을 특징으로 하는 차입 윈도우와 부분일치상태 기반 패턴 검색 방법.

청구항 4.

제 1항 내지 제 3항에서 선택되는 어느 한 항에 있어서,

상기 메모리는 TCAM(Ternary Content Addressable Memory)을 사용하는 것을 특징으로 하는 차입 윈도우와 부분일치상태 기반 패턴 검색 방법.

명세서

발명의 상세한 설명

기술분야

본 발명은 수기가 비트 전송 속도를 가지는 네트워크로부터 인입되는 패킷을 라인 스피드(Line speed)로 의심스러운 데이터 존재 유무를 검사하기 위한 하드웨어 기반의 NIDS(Network Intrusion Detection System)에 대한 것이다.

배경기술

최근 네트워크 대역폭 증가에 따른 인터넷 보급이 확대됨에 따라 인터넷을 통한 침해사태가 자주 보고되고 있으며, 이로 인한 천문학적인 경제 손실이 발생하고 있다. 1990년 초반에는 사용자에게 1Mbps 이상의 대역폭 제공에 따른 많은 어려움이 발생하였으나, 네트워크 기술의 비약적인 발전으로 인해 현재 사용자에게 수십 Mbps 이상의 대역폭을 제공하고 있다. 백본 네트워크의 경우는 포트당 10Gbps를 시스템 차원에서는 수 Tbps의 대역폭을 제공한다. 이러한 네트워크 대역폭 증가와 더불어 네트워크를 통한 공격이 빈번하게 발생하고 있다.

님다(Nimda), 코드레드(Code-Red)와 같은 웜 바이러스 및 트로이 목마와 같은 백도어(Backdoor) 해킹 등이 그 대표적인 예가 될 것이다. 이러한 네트워크를 통한 침해를 감지하고 이를 운용자에게 알림으로써 네트워크를 보호하기 위한 NIDS(Network Intrusion Detection System) 시스템이 개발되었다.

이러한 종래의 NIDS는 소프트웨어 기반으로 동작함에 따라 포트당 수 Gbps 전송속도로 인입되는 패킷에 대해 대역폭 저하 및 실시간 검색이 불가능하다. 즉, 종래의 NIDS는 주로 소프트웨어를 이용한 방법으로서, 이러한 기술을 적용한 시스템은 고속으로 인입되는 패킷을 네트워크의 전송속도의 저하 없이 네트워크를 보호하기 어려우며, 이는 네트워크의 성능을 저하시키는 병목 요소로 작용하게 된다.

또한, 이러한 장치는 소프트웨어로 침입 탐지 과정을 수행함에 따라 순간적인 과도한 공격에 의해 관련 소프트웨어 처리를 위한 프로세서에 과부하가 발생하여 분산 서비스 거부 공격인 DDoS(Distributed Denial Of Service)에 취약한 문제점을 내포하고 있다.

최근 들어 이러한 문제를 해결하기 위해 하드웨어 기반의 NIDS가 고안되고 있으며, NP(Network Processor)와 TCAM(Ternary Content Addressable Memory)을 이용하는 방법이 대표적인 사례이며, 이러한 NP(Network Processor)와 TCAM(Ternary Content Addressable Memory)을 이용한 종래의 기술은 크게 두 가지로 구분될 수 있다.

첫번째는 TCAM(Ternary Content Addressable Memory)을 이용하여 패킷 내에 의심스러운 패턴을 찾을 때까지 1 바이트씩 옮기는 방법이다. 이러한 방법을 바이트 슬라이딩 윈도우 방법이라 한다. 이러한 경우, 현재 가장 빠른 TCAM의 액세스 속도가 4ns (nano-second: 10억분의 1초)임을 감안한다면 이 방법을 적용한 시스템은 최대 2Gbps 내외의 네트워크 정합 속도를 제공하게 된다. 이는 현재 포트당 10Gbps 네트워크의 대역폭을 지원할 수 있으나 현재 고속 대용량 네트워크에서는 공격 패턴 검색 추출이 불가능하다.

또한, 이 방법은 한개의 데이터 흐름(flow)에 속하는 다중 패킷 및 한 개의 패킷이 여러개의 패킷으로 분할된 분할(Fragment) 패킷의 경우 패킷간의 데이터 종속성 문제로 인해 패킷을 재조립(Reassembly)하여야 한다. 패킷 재조립을 위해서는 패킷을 임시로 저장하기 위한 메모리가 필요하며 대용량 NIDS에서는 백만개 이상의 흐름이 존재한다.

즉, 종래의 대용량 NIDS에서는 패킷 재조립을 위해 대용량의 메모리를 소요하게 되는 문제점을 내포하고 있으며, 이러한 방법은 가격에 민감한 상업용 NIDS에서는 부적합하다.

두 번째는 상기 설명한 방법에서의 속도 문제를 해결하기 위해 한 바이트씩 옮기는 방법이 아니라, 데이터 검색을 위한 고정된 크기의 점핑 윈도우를 설정하여 해당 윈도우 크기만큼 패킷내의 데이터를 동시에 검색하는 방법이다. 이를 m-바이트 점핑 윈도우(이하, 점핑 윈도우라 칭함)라 한다. 이 방법에 있어서 만일 윈도우의 크기가 4 이상일 경우 10Gbps 이더넷 지원이 가능하다.

즉, 점핑 윈도우의 경우 패킷 내에 포함된 데이터의 크기가 점핑 윈도우의 크기보다 적거나 마지막 검색할 데이터의 크기와 점핑 윈도우의 크기가 상이할 경우 임의의 데이터로 점핑 윈도우의 부족한 양만큼 채우도록 되어 있는데 이러한 임의의 데이터로 인해 문제를 야기하는 패킷이 아닌 경우도 공격 패킷으로 검출하여 데이터 전송을 방해하는 요소로 작용할 수 있다는 문제점이 있었다.

발명의 내용

해결 하고자하는 과제

따라서, 본 발명의 목적은 상기와 같은 문제점을 해결하기 위한 것으로서, 패킷 재조립 방법이 필요하지 않으며, 고속 대용량 NIDS에 적용이 가능한 부분 일치 상태(Partial Match State) 기반의 바이트 슬라이딩 윈도우, 차입 윈도우 및 부분 일치 상태(Partial Match State) 기반의 차입 윈도우(Borrow Window) 및 이를 적용한 시스템을 제공함에 있다.

과제 해결수단

본 발명의 차입 윈도우와 부분일치상태 기반 패턴 검색 방법은 인터넷으로부터 수신되는 직렬 패킷 데이터를 병렬 데이터 형태인 입력 패킷으로 변환하는 단계; 상기 입력 패킷 인입에 따라 패킷 인입 신호를 발생시키고 상기 입력 패킷을 버퍼링하여 출력하는 단계; 상기 입력 패킷 검색시 부분일치상태(Partial Match State) 기반의 바이트 슬라이딩 방식 또는 부분 일치상태 기반의 점핑 윈도우 방식을 사용하여 메모리에 저장된 적어도 하나의 공격패턴과 비교하여 검사하고, 공격패턴의 전체 또는 일부가 검출되면 공격패턴 추출신호를 출력하는 단계; 상기 공격패턴 추출신호에 따라 공격패턴이 속한 패킷 흐름 제어신호를 출력하는 단계; 및 상기 패킷흐름 제어신호에 따라 출력 패킷을 네트워크로 출력하지 않고 제거하는 단계로 이루어지는 특징이 있으며, 상기 해당 입력패킷의 말단 검색시 부분일치상태를 해당 패턴의 해시값(Hash Value) 또는 부분일치 상태 고유번호(Partial Match State Unique ID)를 사용하고 이를 다음번째 입력패킷의 선두 검색시 전체일치상태를 판단하는 데 사용된다. 상기 부분일치 기반의 점핑 윈도우 방식에서, 해당 입력패킷 후미 말단에서의 검색 데이터가 점핑 윈도우의 크기보다 작을 경우, 필요한 데이터 크기만큼 직전에 검색된 데이터에서 차입해오는 차입 윈도우(Borrow Window) 방식을 사용하는 특징이 있으며, 상기 메모리는 TCAM(Ternary Content Addressable Memory)을 사용하는 특징이 있다.

효 과

본 발명에 따르면, 부분일치 상태 기반의 바이트 슬라이딩 윈도우 방법과 차입 윈도우 방법을 제공함으로써, 수 기가비트로 인입되는 패킷에 대해 네트워크 대역폭 저하 없이 실시간으로 공격 패턴을 검색 및 추출할 수 있는 효과가 있다.

또한, 패킷 재조립을 위한 부가적인 메모리가 필요 없으므로 대용량 고속 NIDS 시스템에 적합한 구조를 가지는 효과가 있다.

발명의 실시를 위한 구체적인 내용

상기한 목적을 달성하기 위한 본 발명에 따른 차입 윈도우와 부분일치상태 기반 패턴 검색 시스템의 일 측면에 따르면, 인터넷으로부터 수신되는 직렬 패킷 데이터를 병렬 데이터 형태인 입력 패킷으로 변환하는 입력망 정합부와, 상기 입력 패킷 인입에 따라 패킷 인입 신호를 발생시키고 상기 입력 패킷을 버퍼링하여 출력하는 입력 패킷 버퍼부와, 적어도 하나의 공격 패턴 정보를 저장하고 있는 메모리와, 상기 입력 패킷 검색시 부분일치상태(Partial Match State) 기반의 바이트 슬라이딩 방식 또는 부분일치상태 기반의 점핑 윈도우 방식을 사용하여 상기 메모리에 저장된 적어도 하나의 공격패턴과 비교하여 검사하고, 공격패턴의 전체 또는 일부가 검출되면 공격패턴 추출신호를 출력하는 패킷 검색부와, 상기 공격패턴 추출신호에 따라 공격패턴이 속한 패킷흐름 제어신호를 출력하는 패킷 출력 제어부 및 상기 패킷흐름 제어신호에 따라 출력 패킷을 네트워크로 출력하지 않고 제거하는 출력망 정합부로 구성된다.

상기 패킷 검색부는 해당 입력패킷의 말단 검색시 부분일치상태를 해당 패턴의 해시값(Hash Value) 또는 부분일치 상태 고유번호(Partial Match State Unique ID)를 사용하고 이를 다음번째 입력패킷의 선두 검색시 전체일치상태를 판단하는 데 사용된다.

상기 패킷 검색부가 부분일치 기반의 점핑 윈도우 방식에서, 해당 입력패킷 후미 말단에서의 검색 데이터가 점핑 윈도우의 크기보다 작을 경우, 필요한 데이터 크기만큼 직전에 검색된 데이터에서 차입해오는 차입 윈도우(Borrow Window) 방식을 사용한다.

상기 메모리는 TCAM(Ternary Content Addressable Memory)을 사용한다.

한편, 상기한 목적을 달성하기 위한 본 발명에 따른 차입 윈도우와 부분일치상태 기반 패턴 검색 방법의 일 측면에 따르면, 인터넷으로부터 수신되는 직렬 패킷 데이터를 병렬 데이터 형태인 입력 패킷으로 변환하는 단계와, 상기 입력 패킷 인입에 따라 패킷 인입 신호를 발생시키고 상기 입력 패킷을 버퍼링하여 출력하는 단계와, 상기 입력 패킷 검색시 부분일치상태(Partial Match State) 기반의 바이트 슬라이딩 방식 또는 부분일치상태 기반의 점핑 윈도우 방식을 사용하여 메모리에 저장된 적어도 하나의 공격패턴과 비교하여 검사하고, 공격패턴의 전체 또는 일부가 검출되면 공격패턴 추출신호를 출력하는 단계와, 상기 공격패턴 추출신호에 따라 공격패턴이 속한 패킷흐름 제어신호를 출력하는 단계 및 상기 패킷흐름 제어신호에 따라 출력 패킷을 네트워크로 출력하지 않고 제거하는 단계로 이루어진다.

상기 해당 입력패킷의 말단 검색시 부분일치상태를 해당 패턴의 해시값(Hash Value) 또는 부분일치 상태 고유번호(Partial Match State Unique ID)를 사용하고 이를 다음번에 입력패킷의 선두 검색시 전체일치상태를 판단하는 데 사용된다.

상기 부분일치 기반의 점핑 윈도우 방식에서, 해당 입력패킷 후미 말단에서의 검색 데이터가 점핑 윈도우의 크기보다 작을 경우, 필요한 데이터 크기만큼 직전에 검색된 데이터에서 차입해오는 차입 윈도우(Borrow Window) 방식을 사용한다.

상기 메모리는 TCAM(Ternary Content Addressable Memory)을 사용한다.

이하, 본 발명의 바람직한 실시예의 상세한 설명이 첨부된 도면들을 참조하여 설명될 것이다. 도면들 중 참조번호들 및 동일한 구성요소들에 대해서는 비록 다른 도면상에 표시되더라도 가능한 한 동일한 참조번호들 및 부호들로 나타내고 있음에 유의해야 한다. 하기에서 본 발명을 설명함에 있어, 관련된 공지 기능 또는 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략한다.

도 1은 방화벽을 이용한 네트워크를 보호하기 위한 네트워크 상호 접속도이다.

도 1에 도시된 바와 같이, 공격자 PC(10)가 IP 망을 통해 연결된 원격장치 1, 2 또는 3(20a, 20b, 20c)을 공격할 경우 NIDS와 같은 방화벽(30)을 통하여 보호하도록 한다. 종래의 NIDS는 소프트웨어를 기반으로 하여 동작을 한다. 이러한 소프트웨어 기반의 NIDS는 IP 패킷내의 데이터를 검사하여 공격 패턴을 추출하는데 프로세서의 성능을 소모할 뿐 아니라, 수 기가비트 속도로 입력되는 IP 패킷을 전송 속도에 맞추어 공격 패턴 검색 및 추출하는데 어려움이 있어 네트워크의 전송 속도를 저하시키는 원인이 된다.

도 2는 본 발명에 따른 하드웨어 기반의 NIDS 내부 블록도이다.

도 2에 도시된 바와 같이, 본 발명은 크게 입력망 정합부(100)와, 입력 패킷 버퍼부(200)와, 패킷 검색부(300)와, 고속 검색 메모리(400)와, 패킷 출력 제어부(500) 및 출력망 정합부(600)로 구성된다.

즉, IP 망으로부터 수신되는 직렬 패킷 데이터는 입력망 정합부(100)에 의해 병렬 데이터 형태인 입력 패킷으로 변경되어 입력 패킷 버퍼부(200)로 인입된다.

입력 패킷 버퍼부(200)는 순간적으로 과도하게 입력되는 패킷을 버퍼링하는 기능을 수행하며, 한 개의 패킷이라도 입력되면, 패킷 인입이라는 신호를 통하여 검색할 패킷이 있다는 것을 패킷 검색부(300)에 통보한다.

패킷 검색부(300)는 입력 패킷 버퍼부(200)를 통하여 해당 패킷과 고속 검색 메모리(400)로부터 읽어온 공격 패턴간 상호일치 과정을 수행하여 일치할 경우에는 공격 패킷으로 분류하여 패킷 출력 제어부(500)로 공격 패킷 추출 신호를 전송하게 된다.

특히, 상기 패킷 검색부(300)는 해당 입력 패킷의 말단 검색시 부분일치상태를 해당 패턴의 해시값(Hash Value) 또는 부분일치 상태 고유번호(Partial Match State Unique ID)를 사용하고 이를 다음번에 입력 패킷의 선두 검색시 전체일치상태를 판단하는 데 사용한다. 여기서, 상기 해시값(Hash Value)은 해당 입력 패킷의 말단 검색시 부분일치상태를 체크하기 위해 이용되는 값을 의미한다.

공격 패킷 추출 신호를 수신한 패킷 출력 제어부(500)는 공격 패킷이 속한 흐름을 통하여 원격 공격자가 지속적으로 공격할 것으로 간주하여 상위 관리자에게 통보하게 된다.

출력망 정합부(600)는 상기 패킷 출력 제어부(500)로부터 패킷 흐름 제어 신호를 수신하는 경우, 해당 출력 패킷을 외부망으로 출력하지 않고 해당 패킷을 제거하게 되며, 패킷 흐름 제어 신호를 수신하지 않는 경우는 병렬로 수신한 해당 패킷을 직렬 형태로 변환한 출력망 패킷을 외부망으로 출력하게 된다. 상기 언급한 모든 과정은 파이프라인 형태로 병렬 처리된다.

도 3은 일반적인 바이트 슬라이딩 윈도우 동작 과정을 나타내는 동작 흐름도이다.

도 3에 도시된 바와 같이, 하드웨어 기반의 NIDS에서 공격 패턴 검색을 위해 적용된 바이트 슬라이딩 윈도우 방법을 예를 들어 도식화 한 것으로, 입력되는 패킷 내에서 찾고자 하는 공격 패턴은 "GATT"로 패킷의 가운데에 위치하고 있다.

이러한 바이트 슬라이딩 윈도우 방법은 찾고자 하는 공격 패턴을 패킷의 처음부터 패턴이 일치할 때까지 한 바이트씩 이동하는 방법으로서, 첫번째 공격 패턴 검색 시도에서 한 바이트 오른쪽으로 이동하여 두번째 시도를 하게 된다. 이러한 형태로 패턴이 발견될 때까지 계속 한 바이트씩 이동하여 6번째 시도에서 공격 패턴 "GATT"를 발견하게 된다.

이러한 방법은 공격 패턴 검색에 있어서 한 바이트씩 이동하므로 패턴 길이에 따라 검색 시간이 길어질 뿐 아니라, 패킷이 네트워크의 여러가지 이유로 인해 다중 패킷(Multi-Packet)으로 구성될 경우 패킷간의 데이터 종속성 문제로 인해 패킷 재조립 과정이 필요하게 되며, 이는 대용량 NIDS는 흐름의 수가 증가함에 따라 부가적인 대용량 메모리가 소요되는 시스템 부담을 가지게 된다.

도 4는 바이트 슬라이딩 윈도우에서 패킷 재조립에 의한 다중 패킷 검색 동작 흐름도이다.

도 4에 도시된 바와 같이, 바이트 슬라이딩 윈도우 방법을 다중 패킷에 적용하였을 경우 패킷 재조립 과정이 필요한 예를 도식화 한 것이다. 입력 패킷들로부터 찾고자 하는 공격 패턴이 "GATT"인 경우로서, 공격 패턴의 "GAT"는 i번째 패킷의 끝부분에 존재하고 나머지 공격 패턴 "T"는 i+1번째 패킷의 첫 부분에 발생한다.

즉, 상기 도 3에서 설명한 일반적인 바이트 슬라이딩 윈도우 방법에서는 i번째 패킷에서는 공격 패턴이 패킷의 마지막 부분에 바이트 슬라이딩 윈도우보다 작은 크기로 존재하기 때문에 공격 패턴의 일부인 "GA"를 찾지 못한다. 또한, i+1번째 패킷의 첫부분에 존재하는 "TT"의 패턴도 앞서 언급한 연유로 인해 발견할 수 없게 된다. 이러한 데이터 종속성을 가지는 다중 패킷에서 공격 패턴을 찾기 위해서는 패킷 재조립 과정을 거쳐야 한다.

이에 따라, 공격 패턴을 찾기 위한 i번째 바이트 슬라이딩 윈도우보다 1 바이트 적은 데이터를 메모리에 저장한 후 i+1번째 패킷과 저장된 데이터를 덧붙여 검색하는 패킷 재조립 방법을 적용하게 된다. 예를 들어, 도 4에 나타난 바와 같이, 1번째 패킷의 마지막 부분에서 바이트 윈도우보다 1 바이트 적은 데이터인 "CGA"를 메모리에 저장한 후 i+1번째 패킷을 수신할 때 "CGA" 값을 i+1번째 패킷의 앞부분에 덧붙이는 패킷 재조립 과정을 수행하여 "CGATT"로 시작하는 패킷을 생성한 후 새로이 생성된 패킷을 이용하여 바이트 슬라이딩 윈도우 방법을 적용한다.

이러한 패킷 재조립 과정은 앞서 언급한 바와 같이 바이트 슬라이딩 윈도우보다 1 바이트 적은 데이터 즉, "CGA" 값을 메모리에 저장하기 위한 메모리가 필요하다. 일반적인 소호(SOHO: Small Office Home Office) 및 기업급 라우터에서는 흐름 수가 많지 않기 때문에 시스템 차원에서 큰 부담은 없다. 그러나, 백본 망에 설치되는 캐리어급 라우터에서는 흐름의 수가 수백만에 이르므로 추가적으로 필요한 메모리의 크기가 기가비트 네트워크당 수 백만 바이트 메모리가 필요하게 시스템 차원에서 많은 부담을 가지게 된다.

도 5는 본 발명 따른 고속 검색 메모리로 적용한 TCAM의 동작 흐름도이다.

도 5에 도시된 바와 같이, 하드웨어 기반의 NIDS에 적용되는 일반적인 CAM(Content Address Memory)은 디지털 로직의 "H" 및 "L" 상태에 대해 검색 과정을 수행한다. 그러나 TCAM은 "H" 및 "L" 상태 외에 don't care 상태에 대한 검색 기능을 제공한다. 이러한 don't care 상태 검색 기능은 패턴 검색을 위해 필요한 엔트리의 수를 획기적으로 줄일 수 있다. 예를 들면, 도 5에서와 같이 "CGT*T" 엔트리의 경우 연속된 패턴인 "CGT" 이후의 한 바이트는 어떠한 패턴이 입력되어도 공격 패턴을 검출할 수 있게 된다.

도 6은 본 발명에 따른 패킷 재조립 과정이 불필요한 부분일치 상태 기반의 바이트 슬라이딩 윈도우의 동작 흐름도이다.

도 6에 도시된 바와 같이, 부분일치 상태 기반의 바이트 슬라이딩 윈도우의 동작 방법을 간략하게 도식화 한 것으로, 바이트 슬라이딩 윈도우 방식을 패킷 재조립 없이 공격 패턴 검색 및 검출하는 방법이다.

상술한 바와 같이 종래의 바이트 슬라이딩 윈도우 방식은 데이터 종속성 문제에 의해 패킷 재조립을 해야만 다중 패킷 검색이 가능했다. 즉, 종래의 바이트 슬라이딩 윈도우 방법에서는 윈도우 크기보다 1 바이트가 적은 데이터를 무조건 저장하여 이를 다음 패킷의 선두에 덧붙여 공격 패턴을 검색하여야만 한다. 따라서, 앞서 언급한 바와 같이 패킷 재조립은 대용량 메모리를 필요로 하는 시스템 부담을 가지게 된다.

반면, 본 발명에 따른 부분일치 상태 기반의 바이트 슬라이딩 윈도우 방법은 패킷내에 공격 패턴 유무를 검색하여 공격 패턴의 일부만 있을 경우 이에 해당하는 1 바이트로 구성된 부분일치 상태 값만을 저장하게 된다. 즉, 종래의 방법에 비해 본 발명에서는 부가적으로 필요한 메모리를 바이트 슬라이딩 윈도우의 크기를 m 으로 하였을 경우, 모든 패킷이 공격 패킷일 경우 최소 $1/(m-1)$ 이하로 줄일 수 있게 된다.

다시 말해, 본 발명은 선수 부분일치 상태 및 후미 부분일치 상태 두 가지로 구성된다. 후미 부분일치 상태값은 다음 바이트 슬라이딩 윈도우의 선수 부분일치 상태와 연결된다.

도 6에 나타난 바와 같이 i 번째 패킷에서 검색된 공격 패턴 "GA"는 후미 부분일치 상태에 설정된 2를 저장한 후, $i+1$ 번째 패킷에서 "TT" 패킷을 검출하면 2번지에 저장된 데이터와 동일한지 확인 후 일치하면 후미 부분일치 상태값을 확인하게 된다.

후미 부분일치 상태값이 0일 경우 공격 패턴을 모두 검색한 것으로 간주하고 해당 패킷을 전송하지 않도록 한다. 만일 $i+1$ 번째 패킷에서 검색된 데이터가 "ATT"일 경우도 공격 패턴으로 검색이 가능한데 이는 i 번째 패킷에서 저장된 후미 부분일치 상태값이 2이므로 부분일치 상태 연결이 정상적으로 이루어지지 않으므로 공격 패턴으로 간주하지 않는다. 이는 부분일치 상태 충돌에 의한 오동작을 방지할 수 있다. 이 때, 선수 부분일치 상태값이 0일 경우는 이전에 검색된 공격 패턴이 없다는 것을 의미하며, 후미 부분일치 상태값이 0일 경우는 잔존하는 공격 패턴이 없다는 것을 의미한다. 예를 들어, "GATT"의 경우 선수 및 후미 부분일치 상태값이 0이므로 찾고자 하는 공격 패턴이라는 것을 의미한다.

도 7a는 일반적인 점핑 윈도우의 기본 동작 흐름도이다.

도 7a에 도시된 바와 같이, 상술한 바이트 슬라이딩 윈도우 방법에 비해 공격 패턴 검색 속도를 m 배 증가시키는 m 바이트 점핑 윈도우의 기본적인 동작 방법을 도식화 한 것이다. 이러한 점핑 윈도우 방식은 바이트 슬라이딩 윈도우 방법이 한번에 한바이트씩 이동하면서 검색하는데 반해 점핑 윈도우는 한번에 m 바이트씩 검색하여 이동하는 방법이다.

점핑 윈도우는 m 바이트 단위로 공격 패턴을 검색하므로, 공격 패턴이 나타날 수 있는 모든 형태에 대한 정보를 가지고 있어야 하지만 일반적인 CAM으로 이러한 모든 형태의 정보를 기술하기에는 메모리 낭비외에도 정보기술 방법에서도 어려움이 따르게 된다. 따라서, 점핑 윈도우 방법은 이러한 문제점을 해결하기 위해 상기 도 5에서 설명한 TCAM을 이용하여 공격 패턴이 나타날 수 있는 모든 정보를 기술하며 이를 TCAM 엔트리라 한다.

도 7a에서는 m 의 크기를 4로 설명하였으며, 엔트리의 수는 $2m-1$ 인 7개가 된다. 7개 엔트리 묶음을 점핑 윈도우라하며, 첫 번째 점핑 윈도우에서는 패킷내에 선두에 위치한 4바이트에 대해 동시에 검색한다. 공격 패턴이 없을 경우 4 바이트 오른쪽으로 이동하며, 동일한 방법으로 4 바이트에 대해 검색하는 형태로 진행된다.

이러한 점핑 윈도우는 바이트 슬라이딩 윈도우에 비해 m 배 패턴 검색 속도를 향상시키는 장점을 가지는 반면 검색하고자 하는 데이터의 양이 점핑 윈도우의 크기와 서로 상이할 경우 임의의 데이터를 이용하여 점핑 윈도우 크기와 동일한 데이터를 생성한 후 검색하게 되는데 이러한 경우 오동작할 가능성이 있다.

도 7b는 다중 패킷상에서의 점핑 윈도우의 동작 흐름도이다.

도 7b에 도시된 바와 같이, 검색하고자 하는 데이터의 양과 점핑 윈도우의 크기가 서로 상이할 경우 발생할 수 있는 가능성에 대해 간략하게 도식화하였다.

검색하고자 하는 패킷의 크기가 점핑 윈도우의 크기보다 적거나, 패킷의 마지막 부분에서 검색하고자 하는 데이터의 양이 점핑 윈도우의 크기보다 적을 경우 임의의 데이터를 채워서 점핑 윈도우를 생성하여 검색하게 된다. 이러한 경우 임의의 데이터가 다른 공격 패턴의 일부인 경우 공격 패턴으로 간주될 가능성이 있으며, 이는 임의의 데이터로 채운 가공된 패킷에 의해 네트워크의 데이터 흐름을 방해할 가능성이 높다.

예를 들면, 도 7b에서 i 번째 패킷의 두 번째 점핑 윈도우에서 검색하고자 하는 데이터는 "GA"이며, 이는 점핑 윈도우 크기인 4와 서로 맞지 않다. 만일 임의의 데이터를 "00"으로 채울 경우 "00"을 "GA"의 뒤에 붙일 경우 새로이 생성되는 데이터는 "GA00"이 되며, 이는 점핑 윈도우에서 검색할 수 없게 된다. 따라서, "00"은 항상 "GA"의 앞에 추가되어야 한다. 이러한 경우 단일 공격 패턴에 대해 검색할 경우에는 문제점을 야기하지 않으나, 검색 데이터에 대해 한개 이상의 공격 패턴 규칙(이하, 다중검색패턴이라 칭함)을 적용할 경우 오동작 할 가능성을 내포하고 있다.

즉, 새로이 생성된 데이터 "GA00"에 대해 "GATT"와 "00"의 두 가지 공격 패턴 검색을 할 경우 공격 패턴 "GATT"의 일부인 "GA"에 대해서는 정상적인 검출을 할 뿐 아니라 "00"에 대해서도 공격 패턴으로 검출하게 된다. 여기서 "00"은 입력된 패킷의 데이터가 아니라 점핑 윈도우의 크기를 맞추기 위해서 삽입된 임의의 데이터에 의해 검출된 것이므로 오동작을 하게 된다. 예를 들어, "GA00"은 다중 패턴 검색 중 공격 패턴이 존재하여 오동작을 발생하지 않는다. 그러나 만일 공격 패턴이 없는 경우 점핑 윈도우와 크기를 맞추기 위해 삽입된 "00"에 의해 공격 패턴으로 인지하게 되어 정상적인 패킷이 전송되지 못하는 결과를 초래하게 된다.

도 8은 본 발명에 따른 부분일치 상태 기반의 차입 윈도우의 동작 흐름도이다.

도 8에 도시된 바와 같이, 상기 도 7a 및 도 7b에서 언급한 점핑 윈도우의 문제점을 해결하기 위해 본 발명에서 제안한 차입 윈도우의 동작에 대해 설명하기로 한다.

차입 윈도우는 검색 데이터 크기와 점핑 윈도우 크기가 서로 상이할 경우, 임의의 데이터를 삽입하는 점핑 윈도우와 달리 윈도우의 이동 방향을 필요한 데이터 크기만큼 반대로 이동하는 방법이다. 즉, 앞선 데이터를 이용하여 새로운 검색 데이터를 생성하는 방법이다. 도 8에서 i 번째 패킷에서 첫 번째 점핑 윈도우는 종래의 방법과 동일한 검색 방법을 수행한다.

두 번째 점핑 윈도우에서는 검색 데이터 크기와 점핑 윈도우 크기가 서로 상이한 경우로, 본 발명에서는 필요한 데이터의 크기만큼 앞선 데이터로부터 차입한다. 즉, 두 번째 점핑 윈도우에서는 첫 번째 점핑 윈도우에서 검색한 데이터 중 일부인 "TC"를 차입하여 두 번째 검색하고자 하는 데이터 "GA"와 상호 연결하여 "TCGA"라는 새로운 검색 데이터를 생성하게 된다.

이와 같이 새롭게 생성된 검색 데이터는 점핑 윈도우를 통해 공격 패턴 검색 과정을 거치게 된다. 생성된 검색 데이터는 점핑 윈도우를 통해 공격 패턴 검색 과정을 거치게 된다. 공격 패턴 검색 과정을 통해 "GA"라는 데이터가 존재함을 발견하게 되며, 해당 패킷을 네트워크를 통해 전송하지 못하게 된다. 즉, 공격 패턴으로 검색된 "GA"는 상기 도 6에서 언급한 부분일치 상태 방법을 기반으로 검색하게 된다.

또한, 본 발명에서 제안한 부분일치 상태 기반의 바이트 슬라이딩 윈도우 방법과 차입 윈도우는 부분일치 상태 기반의 차입 윈도우로 구성할 수도 있다. 즉, 점핑 윈도우의 크기를 1로 구성할 경우 바이트 슬라이딩 윈도우와 동일한 형태로 동작할 수 있다.

이상에서는 본 발명에서 특정의 바람직한 실시예에 대하여 도시하고 또한 설명하였다. 그러나, 본 발명은 상술한 실시예에 한정되지 아니하며, 특허 청구의 범위에서 첨부하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 기술분야에서 통상의 지식을 가진 자라면 누구든지 다양한 변형 실시가 가능할 것이다.

도면의 간단한 설명

도 1은 방화벽을 이용한 네트워크를 보호하기 위한 네트워크 상호 접속도.

도 2는 본 발명에 따른 하드웨어 기반의 NIDS 내부 블록도.

도 3은 일반적인 바이트 슬라이딩 윈도우 동작 과정을 나타내는 동작 흐름도.

도 4는 바이트 슬라이딩 윈도우에서 패킷 재조립에 의한 다중 패킷 검색 동작 흐름도.

도 5는 본 발명 따른 고속 검색 메모리로 적용한 TCAM의 동작 흐름도.

도 6은 본 발명에 따른 패킷 재조립 과정이 불필요한 부분일치 상태 기반의 바이트 슬라이딩 윈도우의 동작 흐름도.

도 7a는 일반적인 점핑 윈도우의 기본 동작 흐름도.

도 7b는 다중 패킷상에서의 점핑 윈도우의 동작 흐름도.

도 8은 본 발명에 따른 부분일치 상태 기반의 차입 윈도우의 동작 흐름도.

* 도면의 주요 부분에 대한 부호의 설명 *

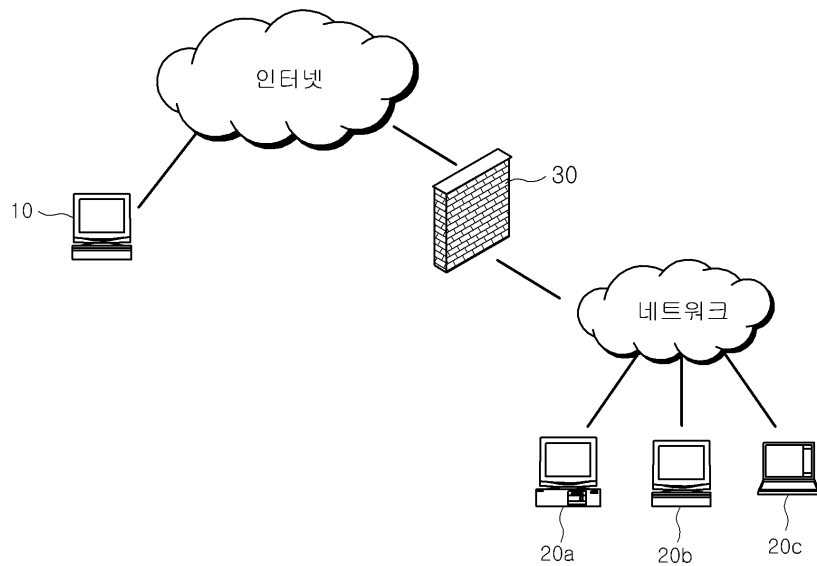
100 : 입력망 정합부 200 : 입력 패킷 버퍼부

300 : 패킷 검색부 400 : 고속 검색 메모리

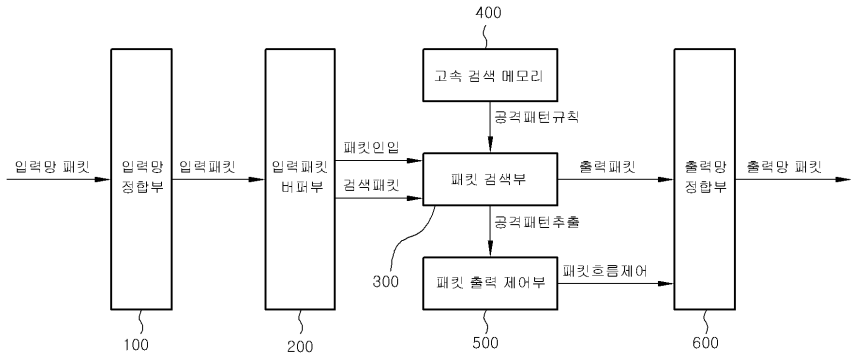
500 : 패킷 출력 제어부 600 : 출력망 정합부

도면

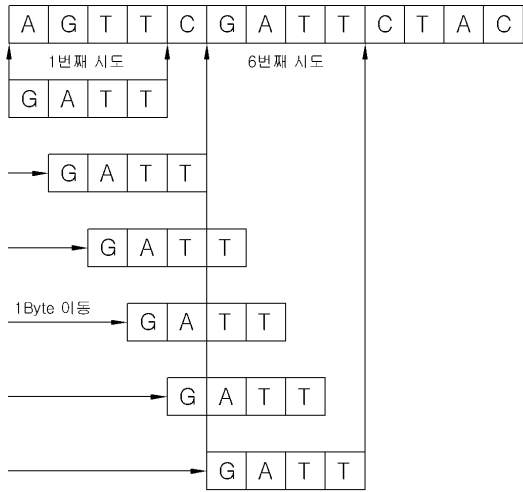
도면1



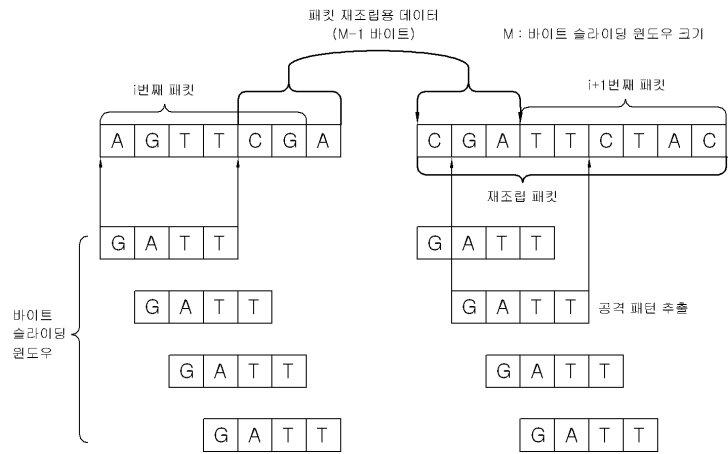
도면2



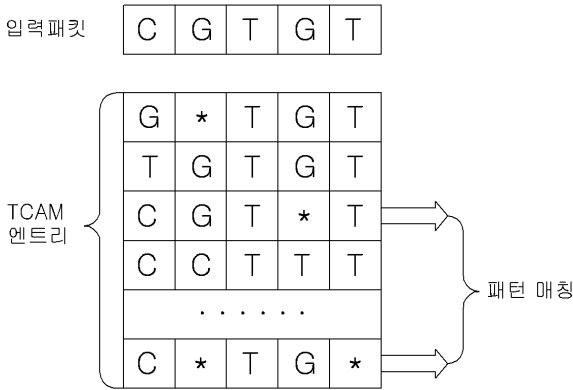
도면3



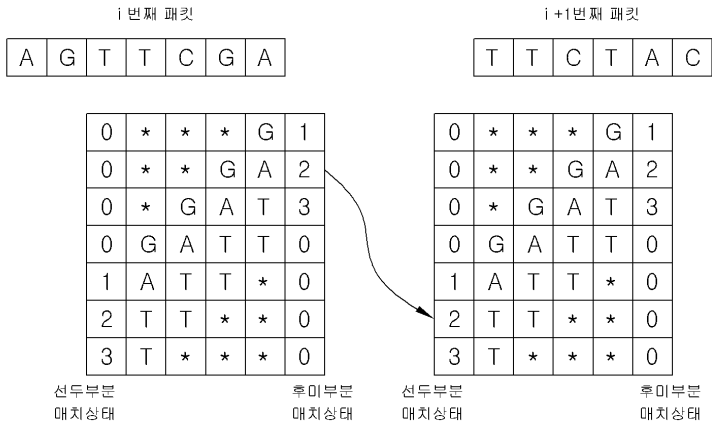
도면4



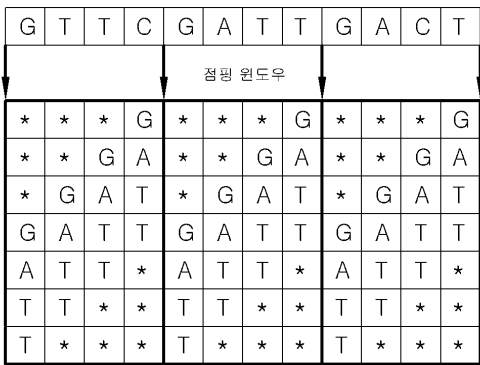
도면5



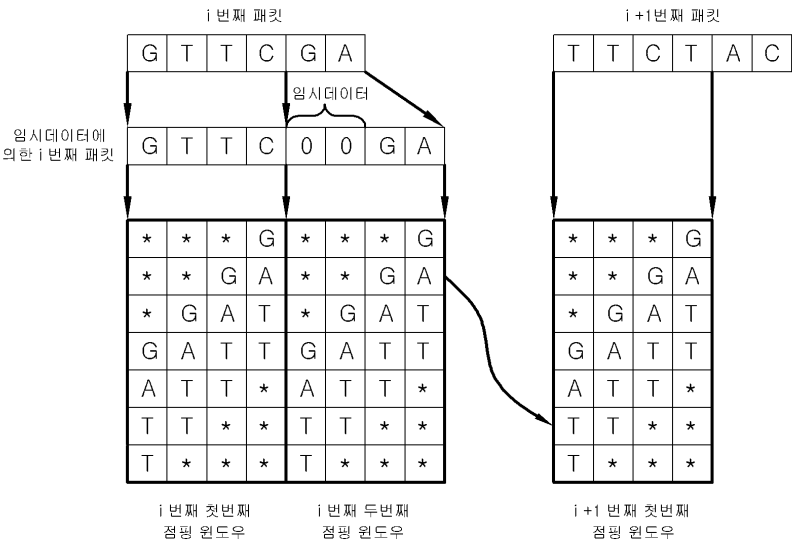
도면6



도면7a



도면7b



도면8

