실시간 고속 네트워크 침입 탐지 엔진 설계 및 구현

조혜영° 김주홍 장종수* 김대영
한국정보통신대학교, 한국전자통신연구원
{hycho°, scarlet, kimd}@icu.ac.kr, jsjang@etri.re.kr

Design and Implementation of Real-Time and High Speed Detection Engine for Network Intrusion Detection System

Hye-Young Cho° Juho-Houng Kim Jong-Su Jang* Daeyoung Kim
Real-Time and Embedded Systems Lab., Information and Communications University, ETRI*

요 약

초고속 인터넷 망이 빠른 속도로 구축이 되고, 네트워크에 대한 해커나 침입자들의 수가 급속히 증가함에 따라, 실시간 고속 패킷 처리가 가능한 네트워크 침입 탐지 시스템이 요구되고 있다. 이러한 실시간 고속 네트워크 침입 탐지 시스템의 핵심 기술로써 수신된 패킷에서 침입 정보를 고속으로 탐지해내는 침입 탐지 엔진 기술은 필수적이다. 본 논문에서는 인텔의 IXP1200 네트워크 프로세서를 기반으로 하는 하드웨어 구조상에서 고성능 네트워크 침입 탐지 시스템을 위한 실시간 고속 탐지 엔진 구조와 프로그래밍 방법을 제안하였다.

1. 서 론

최근 네트워크 관련 기술의 급속한 성장으로 네트워크 침입 탐지 시스템의 속도 및 성능 향상이 요구되고 있으며, 침입 탐지 시스템의 성능 향상을 위한 알고리즘 개선, 소프트웨어의 ASIC화 등 다양한 연구들이 진행되고 있다. 그리고 다른 방법으로 기존의 소프트웨어 방식으로 구현된 침입 탐지 시스템을 네트워크 프로세서를 이용하여 재설계함으로써, 네트워크 침입 탐지 시스템의 성능을 향상 시킬 수 있다. 네트워크 프로세서는 고속의 패킷 처리에 뛰어난 성능을 가지고 있으며, 다양한 프로토콜을 수용할 수 있도록 설계된 프로그래밍 가능한 네트워크 전용 프로세서로서, 네트워크 상의 패킷들을 보다 신속하고 효율적으로 처리할 수 있다.[1][2]

본 논문에서는 현재 가장 널리 사용되고 있는 공개 소스 네트워크 침입 탐지 시스템(Network Intrusion Detection System, NIDS) 중의 하나인 Snort를 모델로 하여, 인텔 IXP1200 네트워크 프로세서를 이용한 고성능 네트워크 침입 탐지 시스템에서의 효율적인 침입 탐지 엔진 시스템 구조를 설계하고 소프트웨어 프로그래밍 방법을 제안하였다. [3][4]

본 논문의 구성은 다음과 같다. 2장에서는 네트워크 프로세서를 기반으로 하는 고속 네트워크 침입 탐지 시스템의 전체적인 구조를 제안하고, 3장에서는 침입 탐지 시스템에서 핵심 기술인 수신 패킷과 룰 시그너처를 비교하여 침입을 탐지하는 탐지 엔진을 설계한다. 마지막으로 4장에서는 결론 및 향후 연구 방향에 대하여 설명한다.

2. 고속 네트워크 침입 탐지 시스템

네트워크 프로세서를 이용한 침입 탐지 시스템은 기존에 호스트 프로세서에서 소프트웨어 방식으로 구현되던 기능들을 고속의 패킷 처리 능력을 가지고 있는 네트워크 프로세서를 활용하여 효율적으로 처리한다.

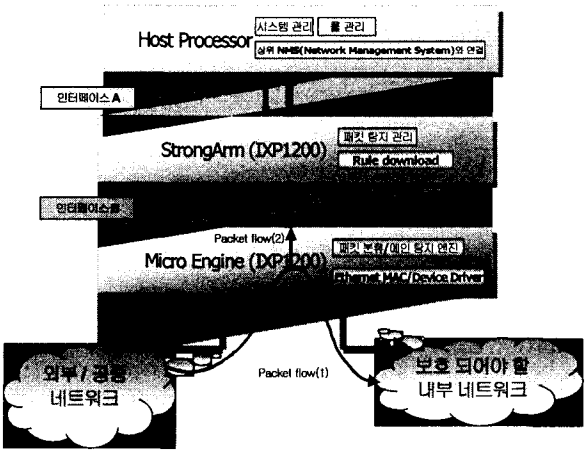

그림 1. IXP1200 네트워크 프로세서 기반 침입 탐지 시스템의 구조

인텔 IXP1200 네트워크 프로세서를 활용한 침입 탐지 시스템의 구조는 호스트 프로세서, StrongArm(IXP1200), 마이크로엔진(IXP1200)의 3단계 계층 구조를 가진다. 호스트 프로세서는 시스템과 전체 룰 시그너처를 관리하고 상위 네트워크 관리 시스템과의 인터페이스를 제공한다. StrongArm에서는 호스트 프로세서와 마이크로엔진 사이에서 중재 기능을 담당하고, 호스트 프로세서로부터 다운받은 룰 시그너처를 마이크로엔진 구조에 알맞게 분류 관리하다. 또한 마이크로엔진을 관리하고 보조하는 역할을 한다. 마이크로엔진에서는 이더넷 프레임의 수신, 송신, 에러 체크 등 이더넷 MAC, 디바이스 드라이버의 역할과 IP 패킷의 수신 및 송신, 에러 체크 등을 담당한다. 또한

침입 탐지를 위한 시그너처를 호스트 프로세서와 StrongArm을 통해서 받고, 이 시그너처 데이터베이스와 네트워크 상에 수신되어 분류된 패킷을 비교하여, 침입을 탐지한다.

IXP12000 네트워크 프로세서를 기반으로 하는 침입 탐지 시스템(Network Processor-based Intrusion Detection System, NP-IDS)은 침입을 탐지하고자 하는 대상 네트워크 상에 존재하면서 수상한 패킷이 있는지 탐지한다. 네트워크 프로세서의 MAC 디바이스로부터 수신된 패킷을 저장된 룰 시그너처 데이터베이스에서 검색하여, 침입 여부를 판정한다. 이상이 없을시(그림1의 Packet flow(1)), 내부 망으로 패킷을 수정 없이 전달하고, 침입 탐지시(그림1의 Packet flow(2)), StrongArm이 호스트에 경고 메시지를 보낸다.

3. 고성능 침입 탐지 시스템을 위한 탐지 엔진 설계

침입 탐지 엔진은 네트워크 프로세서를 기반으로 하는 침입 탐지 시스템에서 시그너처 데이터베이스 관리와 시그너처를 검색하여 직접 침입을 탐지하는 기능을 담당한다. 본 장에서는 마이크로엔진의 침입 탐지 엔진에서 네트워크 상의 TCP, IP, UDP, ICMP 등의 모든 프로토콜에 대해서 침입을 탐지하고 관리하느냐, 아니면 탐지 엔진을 프로토콜별로 나누어 침입을 탐지하고 관리하느냐에 따라, 통합형 침입 탐지 엔진 시스템과 분산형 침입 탐지 엔진 시스템을 설계하였다.

3.1 통합형 침입 탐지 엔진 시스템 구조

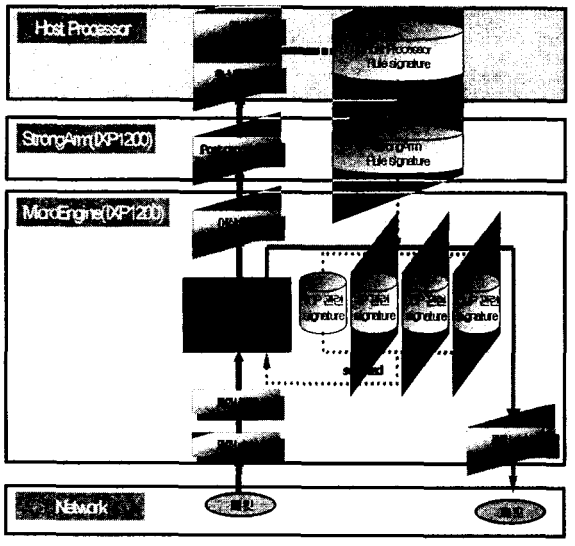

그림 2. 통합형 침입 탐지 엔진 방식 패킷 처리 절차도

통합형 침입 탐지 엔진 구조에서는 모든 탐지 엔진이 TCP, IP, UDP, ICMP 등 모든 프로토콜에 대해서 침입을 탐지하고 관리한다. 그러므로 통합형 침입 탐지 엔진 방식의 마이크로엔진들에는 동일한 마이크로엔진 프로그램이 수행된다. 그림2는 통합형 침입 탐지 엔진 구조에서의 패킷 처리 절차를 나타낸 것이다. 마이크로엔진은 IXP1200의 MAC 디바이스로부터 패킷을 수신하여 TCP, IP, UDP, ICMP 프로토콜 종류에 따라 패킷을 분류한다. 마이크로엔진은 수신된 패킷의 프로토콜 종류에 해당하는 룰 시그너처를 차례로 읽어서 패킷의 정보와 비교한다. 룰 시그너처 데이터베이스에서 수신된 패킷의 정보와 일치하는 패킷을 발견하면 StrongArm에게 침입탐지를 알린다. 이때 해당 룰 번호와 패킷의 내용을 StrongArm에게 알린다. StrongArm은 이 내용을 다시 호스트 프로세서에게 알린다. 호스트 프로세서는 받은 룰 번호와 패킷 정보를 이용하여 로그를 남기고, 메시지를 출력하던지, 상위 NMS(Network Management System)에 알리는 등 적절한 액션을 취한다.

통합형 침입 탐지 엔진 방식은 탐색 엔진이 모든 프로토콜에 대해서 침입을 탐지하고 관리하기 때문에 work conserving하다는 장점이 있다. 즉, 유휴(idle)한 마이크로엔진이 있는한 수신 패킷은 바로 서비스를 받을 수 있다. 반면, TCP, IP, UDP, ICMP 등 여러 종류의 프로토콜에 대해서 침입 탐지 룰을 수용할 수 있어야 하기 때문에, 시그너처 데이터베이스의 구조가 복잡하고, 수신 패킷당 처리 시간이 길다는 단점이 있다.

3.2 분산형 침입 탐지 엔진 시스템 구조

분산형 침입 탐지 엔진 방식은 프로토콜별로 전용 마이크로엔진 쓰레드를 할당하여 탐지 엔진을 운영하는 방식이다. Snort의 경우 TCP, IP, UDP, ICMP 네 가지 종류의 탐지 엔진이 독립적으로 구현되어 있다.[5] IXP1200에는 6개의 마이크로엔진이 있고, 각 마이크로엔진은 4개의 하드웨어 쓰레드를 제공한다.[2] 분산형은 각 쓰레드에 프로토콜 탐지 엔진을 할당할 때, 전체 네트워크 트래픽 중 각 프로토콜이 차지하는 비율(네트워크 트래픽의 프로토콜 비율)이나 전체 룰 시그너처에서 각 프로토콜 시그너처가 차지하는 비율(프로토콜 룰 시그너처 비율)에 따라 각 탐지 엔진을 균형 분배하는 방법, 네트워크 트래픽의 프로토콜 비율과 프로토콜 룰 시그너처 비율에 가중치(weight)를 주어 분배하는 방법 등을 사용할 수 있다.

그림3은 분산형 침입 탐지 엔진 방식에서의 패킷 처리 절차를 그림으로 나타낸 것이다. 마이크로엔진은 IXP1200의 MAC 디바이스로부터 수신된 패킷을 프로토콜 종류에 따라 TCP, IP, UDP, ICMP로 분류한다. 각 프로토콜마다 하나 이상씩 존재하는 탐지 엔진 하드웨어 쓰레드가 StrongArm으로부터 받은 미리 저장된 룰 시그너처와 패킷의 정보를 비교하여 수상한 패킷을 탐지한다. 마이크로엔진은 수상한 패킷을 탐지하면 StrongArm에게 탐지된 패킷의 룰 번호와 패킷의 정보를 알린다. StrongArm은 마이크로엔진으로부터 전달 받은 침입 탐지된 패킷의 룰 번호와 패킷 정보를 호스트에 알리고, 호스트는 그 룰에 해당하는 액션에 따라 syslog, tcpdump 포맷의 로그, 윈도우 팝업 메시지 등과 같은 적절한 행동을 취한다.

분산형 침입 탐지 엔진 방식은 프로토콜별로 다른 룰 시그너처를 사용하기 때문에 룰 시그너처 데이터베이스 구조가 간단하고, 각 프로토콜에 따라 최적화될 수 있다는 장점이 있다. 또한 프로토콜별 룰 시그너처에 따라 탐지 엔진이 최적화 되어서 수신 패킷을 처리하는 속도가 빠르다. 반면, 프로토콜별로 전용

마이크로엔진 쓰레드를 할당하여 탐지 엔진을 운영하기 때문에, work conserving이 아니다. 즉, 해당 프로토콜 처리를 위한 유휴(idle)한 탐지 엔진 쓰레드가 없을 시, 다른 마이크로엔진이 유휴(idle)하더라도 대기해야 한다.

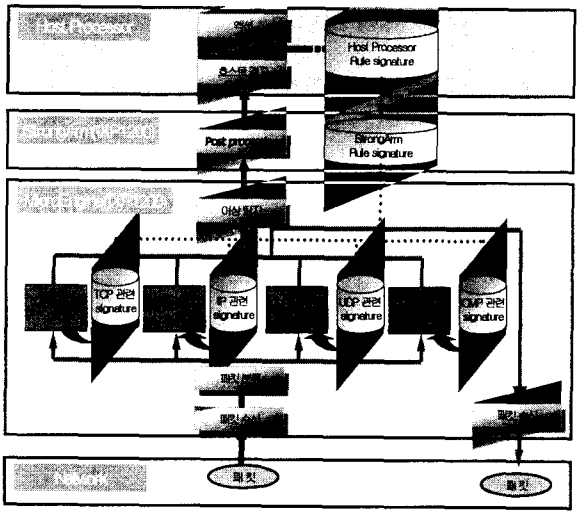

그림 3. 분산형 침입 탐지 엔진 방식 패킷처리 절차도

아래에는 분산형 침입 탐지 엔진 시스템에서 마이크로 엔진 쓰레드에 탐지 엔진을 할당하는 방법을 제안한다.

3.2.1 네트워크 프로토콜 트래픽 비율에 기반한 할당 방법

네트워크 프로토콜 트래픽 비율에 기반한 할당 방법은 전체 네트워크 트래픽 중 각 프로토콜이 차지하는 비율에 따라 탐지 엔진 쓰레드를 분배하는 방식이다. 전체 네트워크 트래픽에서 TCP, IP, UDP, ICMP 등 각 프로토콜이 차지하는 비율은 차이가 있다. 침입 탐지 대상 네트워크의 프로토콜 트래픽 비율을 분석하여, 마이크로엔진의 각 쓰레드에 프로토콜 탐지 엔진을 할당할 때, 각 프로토콜이 차지하는 비율에 따라 프로토콜별 전용 마이크로엔진 쓰레드의 수를 조절하여 균형 분배한다.

RP_{TCP}: TCP 프로토콜 트래픽 비율, RP_{IP}: IP 프로토콜 트래픽 비율
RP_{UDP}: UDP 프로토콜 트래픽 비율, RP_{ICMP}: ICMP 프로토콜 트래픽 비율
N : # of threads, T_{TCP}: TCP 프로토콜 전용 마이크로엔진 쓰레드 수
$T_{TCP} = N \cdot RP_{TCP}$ (예)

3.2.2 프로토콜 룰 시그너처 비율에 기반한 할당 방법

프로토콜 룰 시그너처 비율에 기반한 할당 방법은 전체 룰 시그너처 중 각 프로토콜이 차지하는 비율을 고려하여 탐지 엔진 쓰레드를 분배하는 방법이다. 표1은 Snort(버전1.8.6)에서 기본 룰 시그너처 중 각 프로토콜이 차지하는 비율을 나타낸 것이다. UDP 룰 시그너처와 ICMP 룰 시그너처는 전체 룰 시그너처의 10%정도로 비슷한 비율을 차지하고, TCP 룰 시그너처는 전체 룰의 76.72%, IP 룰 시그너처는 2.45%로, 가장 높은 비율을 차지하는 TCP 룰 시그너처는 가장 낮은 비율을 차지하는 IP 룰 시그너처의 31배가 넘는다.[5]

	TCP	IP	UDP	ICMP	합계
룰 갯수	972	31	136	128	1267
룰 비율(RR)	76.72%	2.45%	10.73%	10.10%	100%

표 1. Snort의 프로토콜 룰 시그너처 비율(버전1.8.6)

마이크로엔진의 각 쓰레드에 프로토콜 탐지 엔진을 할당할 때, 이와 같은 프로토콜 룰 시그너처 비율에 따라, 프로토콜별 전용 마이크로엔진 쓰레드의 수를 조절하고 분배함으로써, 더욱 효율적인 탐지 엔진을 설계할 수 있다.

RR_{TCP}: TCP 룰 시그너처 비율, RR_{IP}: IP 룰 시그너처 비율
RR_{UDP}: UDP 룰 시그너처 비율, RR_{ICMP}: ICMP 룰 시그너처 비율
$T_{TCP} = N \cdot RR_{TCP}$ (예)

3.2.3 가중치(weight)에 기반한 할당 방법

가중치에 따른 할당 방식은 상기 두가지 방법에 가중치를 부여하여 마이크로 엔진 쓰레드를 분배하는 방식이다. 네트워크 프로토콜 트래픽 비율과 프로토콜 룰 시그너처 비율의 가중치를 조절하여 보다 최적화된 침입 탐지 엔진 시스템의 구현이 가능하다.

WP: 프로토콜 트래픽 비율 가중치, WR: 룰 시그너처 비율 가중치
$T_{TCP} = N \cdot [(WP \cdot RP_{TCP}) + (WR \cdot RR_{TCP})]$ (예)

4. 결론 및 향후연구

본 논문에서는 네트워크 프로세서를 기반으로한 고성능 네트워크 침입 탐지 시스템에서의 효율적인 탐지 엔진 구조를 설계하였다. 특히 인텔 네트워크 프로세서 IXP1200을 활용한 고성능 침입 탐지 시스템의 구조를 설계하고, 침입 탐지 시스템의 핵심 부분인 침입 탐지 엔진의 효율적인 구조를 제안하였다.

현재 본 논문에서 제안한 침입 탐지 엔진을 활용하여, 고성능 네트워크 침입 탐지 시스템을 구현하고 있다. 향후 네트워크 프로세서에 기반한 침입 탐지 시스템과 기존의 소프트웨어 방식의 침입 탐지 시스템과의 성능을 비교 분석하고, 또한 본 논문에서 제안한 여러 가지 탐지 엔진 구조 및 할당 방법간의 성능비교 분석을 통하여, 효과적인 탐지 엔진 실현 방법을 제시할 것이다.

참고 문헌

[1] 조혜영 외 3인, "네트워크 프로세서를 이용한 초고속 침입 탐지 시스템 설계 및 구현," 한국정보과학회, 2002 가을.

[2] Intel Corporation, "Intel IXP1200 Network Processor Family Hardware Reference Manual", December 2001.

[3] Neil Desai, "Increasing Performance in High Speed NIDS," A look at Snort's Internals, 2002.

[4] Martin Roesch and Chris Green, "Snort User Manual", pp.1-40, version 1.9.x, www.snort.org. 2002.

[5] Martin Roesch, snort source, version 1.8.6, www.snort.org , 2002.