디지털산업정보학회 논문지 제12권 제1호-2016년 3월
http://dx.doi.org/10.17662/ksdim.2016.12.1.089

오픈소스 IDS/IPS Snort와 Suricata의 탐지 성능에 대한 비교 연구*

석 진 욱*·최 문 석**·김 지 명***·박 종 순****

A Comparative Study on Performance of Open Source IDS/IPS Snort and Suricata

Seok Jinug ·Choi Moonseok ·Kim Jimyung ·Park Jonsung

〈Abstract〉

Recent growth of hacking threats and development in software and technology put Network security under threat, In addition, intrusion, malware and worm virus have been increased due to the existence of variety of sophisticated hacking methods.

The goal of this study is to compare Snort Alpha version with Suricata 2.0.11 version whereas previous study focuses on comparison between snort 2. x version under thread environment and Suricata under multi-threading environment.

This thesis' experiment environment is set as followed. Intel (R) Core (TM) i5-4690 3. 50GHz (4threads) of CPU, 16GB of RAM, 3TB of Seagate HDD, Ubuntu 14.04 are used.

According to the result, Snort Alpha version is superior to Suricata in performance, but Snort Alpha had some glitches when executing pcap files which created core dump errors. Therefore this experiment seeks to analyze which performs better between Snort Alpha version that supports multi packet processing threads and Suricata that supports multi-threading. Through this experiment, one can expect the better performance of beta and formal version of Snort in the future.

Key Words : Snort, Suricata, IDS, IPS, Comparison of Performance

Ⅰ. 서론

최근 3G부터 LTE 통신 및 집 혹은 외부 장소의 WiFi 무선 환경, IoT(Internet of Things) 기술 등 초고속 무선 통신이 더욱 발전함에 따라 네트워크에 연결되는 디바이스들은 다양해지고 네트워크와 연결되는 디바이스들의 수는 기하급수적으로 증가하고 있다[1, 2]. 또한 네트워크 해킹 위협이 점점 증가하고 소프트웨어 기술 및 하드웨어 기술이 급격하게 발전함에 따라 네트워크 보안 위협에 대한 대비가 시급한 실정이다. 고도화된 해킹 수법의 다양화로 시스템에 불법적인 침입과 악성코드 혹은 웜 바이러스 등의 해킹이 증가하고 있다. 과거 이러한 공격을 막기 위해 네트워크 방화벽을 사용하였고 네트워크 방화벽의 한계로 인하여 인증된 IP나 공개된 포트를 통한 공격

* 성균관대학교 정보통신대학원 석사과정(주저자)
** 성균관대학교 정보통신대학원 석사졸업
*** 성균관대학교 정보통신대학원 석사졸업
**** 서일대학교 인터넷정보과 교수(교신저자)

을 차단할 수 없었다. 또한 많은 사용자들은 해킹에 취약한 환경에 노출되어 있고 초고속 네트워크 망의 발전으로 인하여 IDS/IPS 등 네트워크 보안 장비들이 처리해야 하는 트래픽 양이 급격하게 늘어나고 있다. 이에 효율적이고 안전한 네트워크 환경을 위한 보안 솔루션들이 개발이 되고 있으며 이 중 하나가 오픈소스 침입탐지시스템인 Snort와 Suricata이다. Snort는 오픈 소스 침임 탐지 시스템으로 누구나 개발에 참여할 수 있고 내부규칙을 기반으로 규칙에 명시된 사항과 일치하는 패킷을 찾아내는 방식의 IDS이다[3, 4].

Suricata는 방화벽과 동시에 네트워크 패킷을 분석, 탐지, 차단이 모두 가능한 IDS/IPS 솔루션인 Snort보다 진보된 오픈소스 IDS/IPS이다. Snort의 장점 및 단점을 보완, 강화하여 개발이 되었으며 Multi-Threading 기능이 최대 장점으로 지목되고 있다[5].

2009년 미국 국토안보부 지원하에 비영리단체인 OISF(Open Information Security Foundation)를 설립하였고 Snort의 기술적인 한계로 인하여 2010년 Suricata를 개발하였다. Snort를 바탕으로 개발된 Suricata는 Snort와 동작 방식이 비슷하고 기존 Snort에서 사용되고 있는 룰셋을 사용할 수 있다[6]. 박우진[3]의 연구에서는 오픈소스 네트워크 침입 차단 시스템인 Suricata의 Multi-Threading 성능에 대한 한계를 실험을 진행하였고 정명기[4]의 연구에서는 Snort와 Suricata의 탐지 기능과 성능에 대한 비교 연구를 하여 Snort 보다 Suricata가 기능적인 부분과 성능이 뛰어나다는 것을 시사하였다. 또한 유상규[5]의 연구에서는 다중 큐를 이용하여 Multi-Thread 기반 Suricata 시스템을 설계 구현하여 성능비교를 하였고 Single-Threading 기반 Snort, Mutlti-Threading 기반 Suricata, 다중 큐를 구현한 Multi-Core/Multi-Threading 기반 Suricata 모델 순으로 성능이 개선됨을 증명하였다.

Suricata의 경우 Multi-Threading 환경에서 동작하며 Snort 의 경우 2. x 버전은 Sing-Threading 환경에서 동작한다. 하지만 최근 Snort 3.0 Alpha 2 버전은 Multi packet process threading를 한다고 홈페이지에 공시되었고 현재 지속적으로 새로운 버전이 개발이 되고 있다.

이를 통해 연구의 차별화는 첫째, Snort와 Suricata를 이용하여 효과적인 성능을 내는지에 대하여 제언하고자 한다. 둘째, Snort와 Suricata의 성능 비교를 통해 증명하고자 한다. 셋째, Snort와 Suricata의 성능 비교를 통해 침입탐지시스템이 발전할 수 있는 방안을 제언하고자 한다.

따라서 본 연구의 목적은 기존 연구에서 Single-Threading 환경의 Snort 2. x 버전과 Suricata Multi-Threading 환경에서의 성능 비교를 하였지만 본 논문에서는 대표적인 침입 탐지 시스템인 Snort alpha 버전과 Suricata 2.0. 11 버전을 설치하여 탐지 기능 측면 및 성능 비교에 관해 분석하고자 한다.

Ⅱ. 관련연구

2.1 IDS/IPS

2.1.1 IDS

네트워크 보안 솔루션으로 현재까지 널리 사용되고 있는 침입 차단 시스템은 해커에 의한 비정상적인 침입이 발생하지 않도록 네트워크를 제어하는 기능을 수행하고 인증 받지 않은 접근의 시도는 차단할 수 있다고 하지만 인가된 사용자나 이를 가장한 침입자에 대한 공격은 취약한 것이 사실이다. 내부인 혹은 외부인에 의해 발생하는 네트워크 및 시스템 침입은 즉각적으로 대처 및 탐지하는 기술이 요구된다.

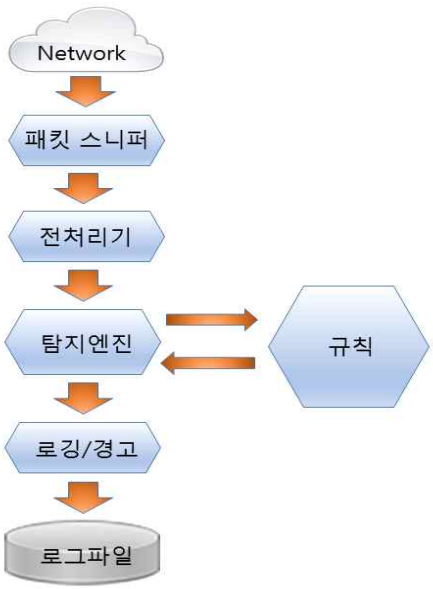
침입 탐지 시스템은 이러한 요구에 맞추어진 보안 솔루션으로 네트워크로부터 보안 관련 정보를 분석 및 수집하여 침입 또는 오용을 탐지하며 침입에 대한 대응기능을 포함하는 시스템이다[7].

2.1.2 IPS

IPS는 네트워크에서 공격 서명을 찾아내어 자동으로 조치를 취하고 비정상적인 트래픽을 차단시키는 보안 솔루션이다. 수동적인 방어의 침입 탐지 시스템과는 다르게 침입 경고 이전에 공격을 차단시키는 데 중점을 두고 있다. 자동 대처 기능과 침입 유도 기능이 합쳐진 솔루션으로 IDS의 대안으로 만들어졌다. 또한 서버의 비정상적인 행동에 따른 정보의 유출을 탐지하여 차단함으로써 인가자의 비정상 행위를 통제할 수 있다[8].

2.2 Snort

Snort는 오픈소스 NIDS에서 가장 대중화 되어 있고 가장 많이 이용하고 있는 네트워크 기반 침입 탐지 시스템으로 1998년 Martion Roesch에 의해 처음으로 만들어 졌다. Snort는 내부 규칙을 가지고 시스템으로 통과하는 트래픽 패킷을 비교 분석하여 침입을 탐지하고 IP 네트워크 상 패킷 로깅 작업과 실시간 트래픽 분석 작업을 수행한다. 또한 프로토콜 분석, 컨텐츠 비교 및 검색 기능을 하며 다양한 공격과 스캔(스텔스 포트 스캔, 버퍼 오버플로우 SMB스캔, OS 핑거프린팅, CGI 공격)을 탐지할 수 있다. Snort는 크게 세가지 모드(네트워크 침입탐지, 스니퍼 모드, 패킷 로거)로 다양한 설정이 가능하며 스니퍼 모드에서는 네트워크에서 패킷을 읽어 들여 출력을 한다. 패킷 로거 모드에서는 패킷을 저장매체에 로그 형태로 저장한다. 네트워크 침입 탐지 모드는 사용자에 의해 설정된 규칙을 가지고 네트워크 트래픽을 모니터링 하며 분석을 한다. <그림 1>과 같이 Snort의 내부 동작 단계의 구성을 살펴보면 패킷 스니퍼, 전처리기, 탐지엔진, 로깅/경고, 로그파일/데이터베이스로 구성되어 있는 것을 알 수 있다. 먼저 패킷 스니퍼는 네트워크로부터 패킷을 받아들이며 이 패킷을 전처리기에서 Snort의 탐지 엔진에 도달하기 전에 악의적인 패킷인지 올바른 패킷인지 구별하는 과정을 거친다. 다음으로 탐지 엔진을 통해 사용자에 의하여 설정된 규칙을 가지고 비교하여 악의적인 침입을 탐지 한다. 마지막으로 탐지 엔진에서 나온 결과를 바탕으로 보안 관리자에게 경보 및 로그를 기록하도록 하여 로그 파일과 데이터베이스의 형태로 탐지 기록을 저장하도록 한다[9].



<그림 1> Snort Architecture

2.2 Suricata

Snort의 Single-Threading 방식과 달리 Suricata는

Multi-Threading 환경을 통하여 탐지 성능 향상에 노력을 하고 있다는 특징이 있다. Suricata와 Snrot는 여러 가지 기능적인 부분에서 비슷한 점이 있지만 처리 방식은 다른 부분이 있으며 가장 대표적인 특징 중 하나는 구동 방식의 차이인데, Snort의 Single-Threading 구동과 Suricata의 Multi-Threading 구동 방식의 차이이다. 또한 모든 시그니처 룰셋 등은 Snort와 호환 가능하며, Snort의 룰셋 형식에 대하여 자세히 알지 못하여도 스크립트만으로 룰셋을 정의할 수 있는 장점이 있다[5].

Suricata의 장점으로는 첫째, Multi-Threading 에서 운영 가능한 장점을 가지고 있는 Suricata는 최대 10Gigabits의 속도를 지원한다. 둘째, Suricata는 다양한 프로토콜을 손쉽게 인식할 수 있는 능력을 갖추었다. 셋째, Suricata는 네트워크에서 여러 형태의 파일들이 통신되고 있을 때 에도 인식을 할 수 있다. 넷째, Snort에서 지원하지 않고 있는 Protocol Identification과 HTTP Normalizer&Parser, 그리고 File Identification 기능이 추가 되었다. 이 뿐만 아니라 기존의 Snort에서 사용되고 있는 VRT Rules, ET-Pro 룰셋 그리고 ET-Open 룰셋을 사용할 수 있다[4-9-10].

Ⅲ. 실험환경과 연구결과

3.1 실험 환경

Suricata와 Snort의 객관적인 성능 측정을 하고자 OS 환경 및 실험에 사용된 PC 사양은 동일하게 맞춰주었다. CPU는 Intel(R) Core(TM) i5-4690 3. 50GHz(4threads), RAM 16GB, Ubuntu 14.04을 사용하였다.

Snort의 Alpha 버전인 Snort 3.0.0-a2-172를 설치하였으며 Suricata는 2.0. 11버전을 설치하여 실험을 진행하였다.

실험을 진행하기 위하여 Suricata와 Snort의 룰셋 개수를 맞춰주었으며 Emerging Threats 홈페이지[11] 내 ET-Open 룰셋을 다운받아 각각의 IDS에 적용하려 하였으나 Snort의 경우 Alpha 버전에서 탐지 에러가 발생한 룰셋은 삭제하여 진행하였고 총 룰셋의 개수는 총 2만여개에서 1794개로 줄였다. 이 때 사용 불가능한 미호환 룰셋 옵션에는 threshold, depth, offset, fast_pattern, stream-event, app-layer-event, flowint, user-rules, within, urilen, uricontent, fileext, filestore, filemagic, noalert, ls. fingerprint, uricontent 가 있었으며 탐지 가능한 옵션에는 $HOME_NET, IP, Port, Flow, Content, Http_uri가 있었다.

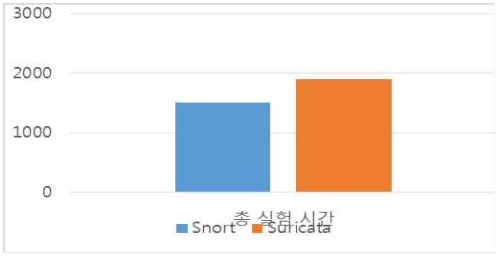
Threads 설정 부분에서는 Snort 실행 명령어 부분에 --max-packet-thread 4를 사용하였고, Suricata의 경우 Suricata. yaml의 detect-cpu-set에서 0-3으로 설정하여 4개의 threads를 사용하도록 설정하였다.

다양한 용량 및 악성코드 트래픽을 포함하고 있는 Pcap파일을 제공하는 CTU[12] pcap 파일을 다운 받아 실험을 하였고 실험에서 사용 가능한 파일의 개수는 62개였으며, 파일의 총 용량은 106. 5gb이다. Pcap 파일을 이용한 탐지가 끝난 후에는 캐쉬 메모리 초기화 명령어를 사용하여 다음 실험을 진행하였다.

3.2 Snort와 Suricata 성능 분석 비교

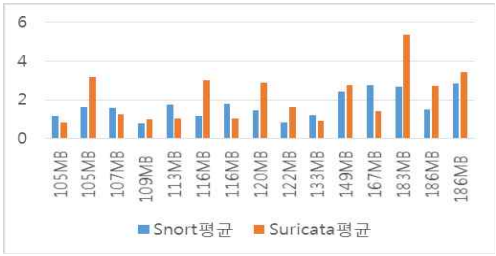
Linux top 명령어를 사용하여 CPU/Core 사용률을 모니터링 하였고 <표 1>에서는 Snort CPU 사용률을, <표 2>에서는 Suricata CPU 사용률을 나타내었다. 표에서 나타내는 id값은 CPU의 Idle값을 의미하고 100-id를 하여 CPU 사용률을 확인할 수 있다. <표 1>에서 보는바와 같이 Snort의 경우 내부 부하 분산 환경은 작동하지 않는 것으로 보였고, <표 2>의 Suricata의 경우 균등하게 Thread를 사용하고 있는

것으로 보였다. 그러나 <그림 2>의 그래프에서 보이는 바와 같이 Snort의 총 탐색 시간은 1515.338398초였으며, Suricata의 총 탐색 시간은 1905.02초 였다. <그림 3>의 경우Snort가 62개의 파일 중 34개의 파일이 더 높은 성능을 보였으며 나머지 28개의 파일은 Suricata가 높은 성능을 보였다. <그림 4> 의 경우 총 15개 pcap파일로 이루어져 있으며 105mb 용량부터 186mb 용량의 pcap 파일을 실험하였다. 105mb, 107mb, 113mb, 116mb, 167mb 파일을 제외한 10개의 pcap 파일이 Snort에서 더 나은 성능을 보였다. <그림 5>의 경우 총 12개 pcap파일로 이루어져 있으며 212mb부터 283mb용량의 pcap 파일을 실험하였다. 두 개의 다른 254mb, 273mb, 283mb 파일을 제외한 나머지 9개의 pcap파일이 Snort에서 더 나은 성능을 보였다. <그림 6>의 경우 총 13개의 pcap파일로 이루어져 있으며 305mb부터 898mb 용량의 pcap파일로 실험을 하였다. 316mb, 366mb, 747mb, 898mb 파일을 제외한 나머지 9개의 pcap파일이 Snort에서 더 나은 성능을 보였다. <그림 7> 의 경우 17개의 pcap파일로 이루어져 있으며 1.04gb부터 3.79gb까지의 pcap 파일을 실험하였다. 1.08gb, 1.32gb, 1.37gb, 1.59gb, 2.44gb 파일을 제외한 나머지 12개의 pcap파일이 Snort에서 더 나은 성능을 보였다. <그림 8>의 경우 5개의 pcap파일로 이루어져 있으며 8.17gb부터 26.6gb의 고용량의 pcap파일로 실험을 하였다. 고용량의 pcap 파일의 경우 모든 pcap파일에서 Snort가 더 나은 성능을 보였다.

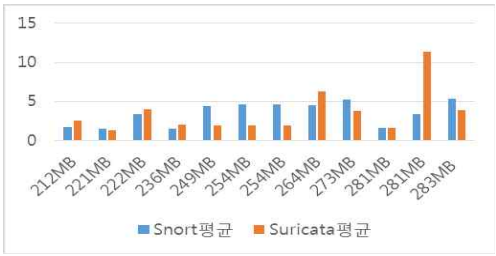


<그림 2> Snort와 Suricata 총 탐색 시간 비교

<그림 3> Snort와 Suricata 높은 성능을 보인 파일의 수



<그림 4> Snort와 Suricata 탐색 시간 비교



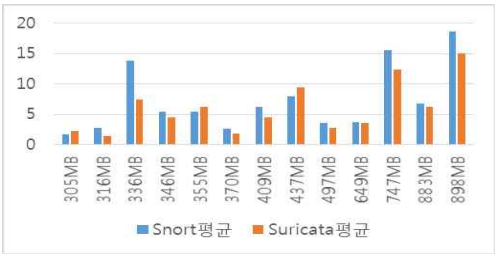
<그림 5> Snort와 Suricata탐색 시간 비교

<표 1> Snort threads 사용률

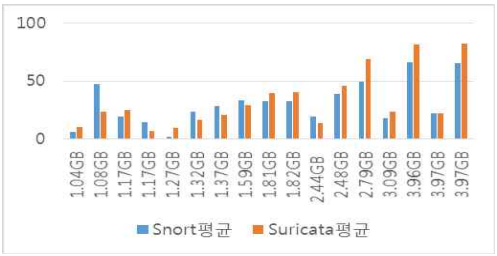
	us	Sy	ni	id
Cpu0	4.4%	1.7%	0.0%	93.9%
Cpu1	87.0%	7.6%	0.0%	4.7%
Cpu2	0.0%	0.0%	0.0%	100.0%
Cpu3	1.3%	2.0%	0.0%	96.7%

<표 2> Suricata threads 사용률

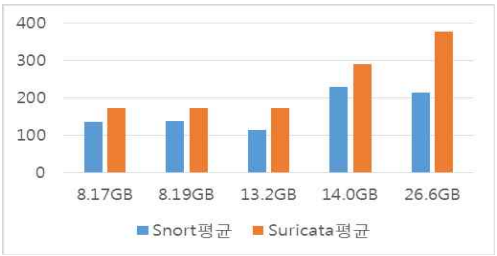
	us	Sy	ni	id
Cpu0	56.5%	11.4%	25.8%	0.0%
Cpu1	57.0%	4.0%	0.0%	39.0%
Cpu2	50.5%	6.0%	0.0%	42.8%
Cpu3	50.0%	3.3%	0.0%	46.5%



<그림 6> Snort와 Suricata탐색 시간 비교



<그림 7> Snort와 Suricata탐색 시간 비교



<그림 8> Snort와 Suricata탐색 시간 비교

Ⅳ. 결 론

본 논문에서는 IDS/IPS 분야에서 다양하게 사용하고 있는 Snort와 Suricata 성능 비교 분석을 실험 결과를 통하여 증명하였고, 현재 Snort Alpha 알파 버전의 한계로 인하여 내부 부하 분산 기능이 구현이 되지 않았음에도 불구하고 성능 측정 실험 결과 Alpha 버전의 Snort가 더 나은 성능을 보였다. Snort 실험 중 몇 개의 pcap 파일은 core Dump 에러가 발생하여 실험을 진행하지 못하였다. 이 부분은 현재 Alpha 버전의 문제로써 향후 개선이 될 필요성 있다. 향후 연구로는 Snort의 업데이트 베타버전 혹은 정식 버전이 배포가 된다면 다양한 룰셋 및 실제 네트워크 환경에서 성능측정을 연구해야할 것이며 Suricata는 Multi-Threading 방식과 더불어 GPU의 병렬컴퓨팅 아키텍처를 지원하는 Nvida 사의 CUDA를 사용하여 침임 탐지 시스템을 구현할 수 있으며 향후 본 논문의 결과와 비교 분석하여 오픈소스인 IDS/IPS Snort와 Multi-Threading의 Suricata, CUDA를 적용한 Suricata의 성능 비교 분석 연구가 필요할 것이다.

참고문헌

[1] 왕종수 · 서두옥, “Sparse M2M 환경을 위한 DTMNs 라우팅 프로토콜,” 디지털산업정보학회 논문지, 제10권, 제4호, 2014, p.12.

[2] 최희식 · 조양현, “사물인터넷 보안 문제제기와 대안,” 디지털산업정보학회 논문지, 제11권, 제1호, 2015, p.69.

[3] 박우진 · 최석환 · 최윤호, “Suricata의 Multi-Threading 효율성에 관한 실험적 연구,” 한국통신학회 하계종합학술발표회, Vol 2015, No 6, 2015, pp.874-875.

[4] 정명기 · 안성진 · 박원형, “Snort와 Suricata의 탐지 기능과 성능에 대한 비교 연구,” Convergence Security Journal, Vol 14, No 5, 2014, pp.4-8.

[5] 유상규, “멀티코어 환경에서 다중 큐를 이용한 멀티 스레드 기반 IPS 시스템의 설계 및 구현,” 서강대학교 정보통신대학원 석사학위 논문, 2013, pp 1-32.

[6] Albin Eugene, “A comparative analysis of the Snort and Suricata intrusion-detection systems,”

Master's thesis NAVAL POSTGRADUATE SCHOOL, 2011, pp.1-13.

[7] Denning, D. E, "An intrusion-detection model," IEEE Transactions on Software Engineering, 1987, pp.1-16.

[8] 안성진・이경호・박원형, 보안관제학, 이한미디어, 고양, 2014, p. 223.

[9] Jay Beale・James C. Foster・Jeffrey Posluns・Brian Caswell, 스노트 2.0 마술상자, 에이콘, 서울, 2003, p. 30.

[9] Suricata, www.suricata-ids.org

[10] Snort, www.snort.org

[11] Emerging Threats ET Rule, https://rules.emergingthreats.net/open/suricata

[12] MCFP CTU PCAP, https://mcfp.felk.cvut.cz/publicDatasets

■ 저자소개 ■

석 진 욱
Seok Jinug

2013 3월~현재 성균관대학교 정보통신대학원 정보보호학과 석사과정

관심분야 : 네트워크 보안, 시스템보안, 디지털 포렌식
E-mail : jinugi6@skku.edu

최 문 석
Choi Moonseok

2016 2월 성균관대학교 정보통신대학원 정보보호학과(공학석사)

관심분야 : 네트워크 보안, 정보보안
E-mail : msoki@skku.edu

김 지 명
Kim Jimyung

2016 2월 성균관대학교 정보통신대학원 정보보호학과(공학석사)

관심분야 : 네트워크 보안, 사물인터넷 보안
E-mail : bugeking@skku.edu

박 종 순
Park Jongsoon

1993년 3월~현재 서일대학교 인터넷정보과 교수
2005년 2월 한국외국어대학교 경영학박사
1990년 2월 한국외국어대학교 경영학석사
1985년 2월 성균관대학교 행정학사

관심분야 : 모바일 비즈니스, 정보기술
E-mail : jsoonpark@lycos.co.kr

논문접수일: 2016년 2월 28일
수 정 일: 2016년 3월 17일
게재확정일: 2016년 3월 20일