Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.29, NO.2, Apr. 2019 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2019.29.2.393

Software Defined Network(SDN) 환경에서 비인가 소프트웨어 차단 기법*

강 남 길,† 권 태 욱‡
국방대학교

Unauthorized Software Blocking Techniques in Software Defined Network (SDN) Environments*

Nam-Gil Kang,† Kwon TaeWook‡
Korea National Defense University

요 약

조직으로부터 인가받지 않고 내부로 반입한 비인가SW가 조직의 네트워크 보안에서 위협으로 대두되고 있는 상황에서 SDN(Software-Defined Network) 기반의 네트워크 환경이 구축된 조직에서는 별도의 보안장비를 설치하지 않고도 조직의 특성을 고려한 보안 어플리케이션 개발을 통해 네트워크보안을 강화할 수 있다. 기존 SDN 환경의 보안기술은 방화벽, 침입탐지시스템 등 외부 네트워크로부터 내부 네트워크를 보호하는 연구가 이루어져 왔으나 내부자 위협에 대해서는 부족하였다. 따라서 이러한 SDN 환경에서 조직 내부 위협 중 하나인 비인가SW로 부터 내부 네트워크를 보호할 수 있는 시스템을 제안한다.

ABSTRACT

In a situation where an unauthorized SW brought into the organization without being authorized is emerging as a threat to the network security, the security of the network based on the SDN(Software-Defined Network) can be strengthened through the development of the security application considering the organization's characteristics. Security technology of existing SDN environment has been studied to protect internal network from external networks such as firewalls and Intrusion Detection Systems, but the research for resolving insider threat was insufficient. Therefore, We propose a system that protects the internal network from unauthorized SW, which is one of the insider threats in the SDN environment.

Keywords: SDN, NAC, Network security, Unathorized SW

I. 서 론

조직 네트워크에 위협이 되는 원인은 다양하지만 주요 원인으로 불법SW 사용이 거론되고 있다. 불법SW는 명확한 정의는 없지만 일반적으로 법에 의해 보장되지 않은 방법으로 사용하는 SW이며, 비인가SW는 불법SW 뿐만 아니라 조직의 입장에서 IT부서나 보안부서로부터 인가받지 않고 반입한 SW다. 조직에서 운용하는 네트워크에 업무상 편의를 위해 CD, USB 등의 저장매체나 인터넷 다운로드를 통해 비인가SW를 업무PC에 설치하게 되면 조직 전체 네트워크의 안전을 위협하는 요소가 될 수 있다.

2018년 BSA(Business Software Alliance)[1] 자료[1]에 따르면 불법SW 사용률이 높은 국가일수

Received(10. 23. 2018), Modified(1st: 01. 15. 2019, 2nd: 04. 10. 2019), Accepted(04. 10. 2019)
* 본 논문은 2018년도 한국정보보호학회 영남지부 학술대회에서 발표한 논문을 확장한 논문입니다.
† 주저자, gaeshi@naver.com
‡ 교신저자, kwontw9042@kndu.ac.kr(Corresponding author)

록 악성코드 감염률도 높은 것으로 나타났으며, 불법 SW 설치로 인해 감염된 악성코드로 인해 발생하는 경비가 연간 3,590억 달러에 달하는 것으로 예상되는 등 사회적으로 막대한 피해를 주고 있는 것으로 나타났다. 따라서 조직내 반입되는 SW는 IT부서나 보안부서로 보안검증을 받아 비인가SW로부터 조직 네트워크의 안전을 보장할 수 있도록 해야 한다. 한편 네트워크 업계에서는 네트워크 장비에서 하드웨어 기능과 소프트웨어 기능을 분리하여 논리적으로 중앙 집중형태를 취하는 SDN이 기존 legacy 네트워크를 대체 할 수 있는 기술로 보고 많은 분야에서 연구가 진행 중이다[2]. 기존 네트워크는 복잡하며 유연성이 부족하여 수많은 네트워크 장비를 통제하기 어렵고, 새로운 사용자나 부서들이 생겨나거나 없어짐에 따라 네트워크 장비 설정을 수시로 변경해야 하는 등 번거로움이 많았는데 네트워크 장비에서 소프트웨어를 분리함으로써 네트워크의 유연한 운영 및 관리를 할 수 있는 SDN 기술이 보안분야에서도 각광을 받고 있다. 하지만 SDN의 특성을 이용한 방화벽[3]이나 DDoS 방어, 침입탐지기술[4] 등 외부 네트워크로부터 내부 네트워크 보호를 위한 연구가 대부분이며 NAC과 같이 내부자에 의한 보안위협을 차단하기 위한 기능을 구현하는 응용분야는 연구가 부족한 실정이다.

본 논문에서는 향후 SDN 기술을 적용하는 조직이 늘어날 것으로 기대되는 상황에서 SDN 환경에서의 네트워크 보안을 위해 비인가SW, 즉 조직에서 반입 및 사용이 인가되지 않은 SW 차단을 위한 기법을 제시하고자 한다.

II. 관련 연구

2.1 SDN

SDN이란 기존 네트워크와는 달리 네트워크 제어 기능과 packet forwarding이 분리된 네트워크 아키텍처이다.

라우터로 패킷을 전달하기 위해 필요한 두 가지 요소는 트래픽 경로와 상대적 우선순위를 결정하는 제어 기능과 정책에 따라 데이터를 전달하는 기능인데 SDN이 등장하기 전에 이러한 기능들은 각 네트워크 장비(라우터, 브릿지, 스위치 등)에 통합되어 있었고 제어기능은 각 네트워크 노드에 구현된 라우팅과 제어 프로토콜이 담당하여 네트워크 관리 측면에서 어려움이 많았다. 그래서 점차 기존 간결한 네트워크를 손쉽게 제어하고 확장성이 용이한 새로운 네트워크가 요구되어 왔고 이런 욕구를 만족시켜줄 수 있는 SDN이 등장하게 되었다.

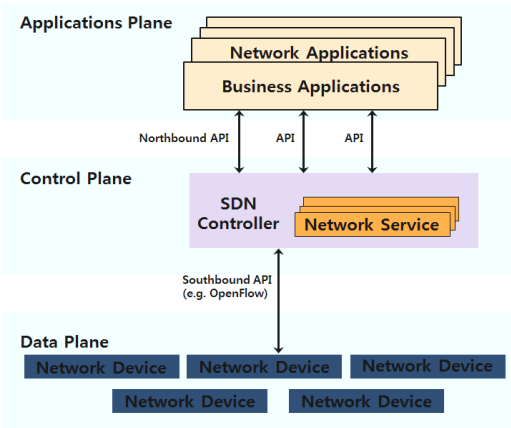

Fig. 1. SDN architecture

SDN의 기본적인 구조는 Fig. 1과 같으며 기본적으로 네트워크 지능화 기능을 담당하는 Application Plane, 운영체제 기능을 담당하는 Control Plane, 데이터 전송을 담당하는 Data Plane으로 나눌 수 있다.

Data Plane은 실제 네트워크 장비에 해당되는 영역이며, 스위치나 라우터가 바로 이 계층에 속한다. Flow Rule에 의해서 움직이며, Flow Rule의 집합인 Flow Table에 정의된 규칙에 의해 외부에 위치한 스위치로 전달하거나, 패킷을 수정하고 폐기하는 작업을 수행한다.

Control Plane 영역에서는 네트워크 장비를 제어하는 역할을 하는 Controller가 가장 중요한 핵심이라 할 수 있다. Controller는 Control Plane에서 트래픽 포워딩, 토폴로지 및 자원 상태 관리, 라우팅 제어 등 중앙집중적인 네트워크 환경 관리를 위한 기능을 수행한다. Controller 하위계층의 스위치와 라우터를 제어하며, 상위계층의 정책 요구에 따라 세분화된 트래픽 포워딩 및 패킷 처리를 결정한다.

Application Plane은 SDN에 있어 유연성이 가장 높은 영역이다. 네트워크 운영자는 조직의 네트워크 환경과 요구사항에 최적화된 어플리케이션을 적

1) 미국의 소프트웨어 저작권 보호 단체. '88년 MS社와 로터스社 등 소프트웨어 업체들 중심으로 설립

용해 네트워크를 유연하고 효과적으로 운영할 수 있다. NorthBound API로 서비스와 애플리케이션을 SDN Controller와 연계하여 네트워크를 구성하고, 기능을 제어할 수 있다. 이 계층은 구현 영역이 다양하고 자유롭기 때문에 SDN 네트워크 시장진입에 대한 기회와 가치를 창출해낼 수 있다. 다양한 보안 서비스, 네트워크 관리, Flow 제어 등 네트워크 사용자 및 관리자의 요구사항을 이 영역에서 쉽게 구현할 수 있다. 이런 특성을 이용해서 SDN환경에서의 방화벽, 침입탐지 등의 네트워크보안 기법들이 소개되었고 이외 다양한 보안 어플리케이션 개발이 가능하다.

2.2 NAC

불법SW에 의한 피해를 예방하기 위해 여러 기업이나 기관들에서는 조직 입장에서는 인가받지 않고 반입되어 설치되는 불법SW를 포함한 비인가SW 차단 할 수 있는 솔루션을 도입하여 운영하고 있다. NAC와 소프트웨어 자산관리 프로그램 등이 많이 사용되는 기술이다.

NAC는 네트워크에 접근하는 단말들의 검증하여 접속을 통제할 수 있는 보안 인프라이다. 사용 단말이 네트워크에 접근하기 전 보안 정책 준수 여부를 검사해 네트워크 접속을 통제한다. 최근 데스크탑 외에 노트북, 스마트폰, PDA 등 접속 단말이 다양화되고 모바일 환경 하에서의 조직 네트워크 접속이 증가함으로써 발생하는 취약점을 해소시키기 위해 많은 조직들이 도입하고 있다.

2005년 가트너 그룹은 NAC에 대한 참조모델을 발표[5]하였는데, 접근 단말에 대한 평가와 발생 할 수 있는 보안상황에 대한 대응, 네트워크 접근 허용 및 보안정책 준수에 대한 지속적인 모니터링과 실시간 대응 등에 관한 모델을 제시하고 있다. 가트너 그룹에서 제안하는 절차는 네트워크 접근을 위한 보안정책(Policy)을 정의 하는 것에서 부터 출발한다. 보안정책은 네트워크에 접속을 요청하는 단말에 강제하고자 하는 관리자가 설정한 보안요구사항이다. 이런 보안정책은 조직에 따라 특정 시스템이나 어플리케이션의 정책을 포함 할 수 있다.

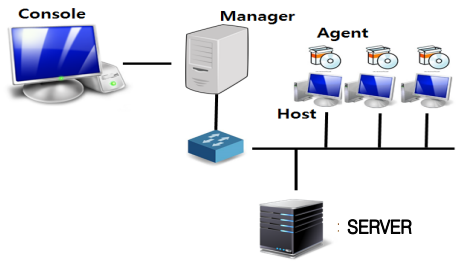

Fig. 2. Agent-based NAC

NAC에는 차단방식에 따라 802.1X, DHCP, VLAN, ARP 등과 같이 여러 방식이 존재한다. 현재 가장 널리 사용되고 있는 방식은 Fig. 2와 같은 Agent를 이용한 방식이다. 제조사에서 제어하고자 하는 단말기에 Agent를 설치, 관리자의 서버와의 통신을 통해 단말기의 네트워크를 효율적으로 제어할 수 있다. 보안정책 설정 및 감사를 담당하는 콘솔, 차단서버와 Agent를 관리하는 Manager, 실질적인 네트워크상의 장비 및 사용자를 통제하는 차단서버로 이루어져 기능을 수행한다.

이러한 NAC는 보안관리에 매우 효과적이지만 시스템을 구축하고 유지하는데 경제적 비용이 발생하게 된다. 2016년 지란지교(정보보안업체)에서 458개의 중소기업을 대상으로 한 설문조사[5] 결과 절반에 가까운 49.4%의 기업이 정보보호에 투자하지 않는 이유가 비싼 솔루션 비용 때문이라 응답하였다.

따라서 Agent를 이용한 NAC 기법을 SDN의 소프트웨어적인 영역을 활용하여 구현하면 별도의 차단서버와 Manager가 없어도 해당기능을 동일하게 구현할 수 있어 경제적, 관리적으로 부담을 줄임으로써 중소기업 등과 같은 소규모 네트워크에도 적용 가능하다.

III. SDN 환경에서 비인가SW 차단 시스템

3.1 시스템 요구사항

SDN환경에서 비인가SW를 사용하는 호스트를 식별하기 위해서는 Agent 기반의 NAC와 마찬가지로 호스트의 프로세스 정보가 요구된다. Agent는 이러한 정보를 스캔하여 SDN Controller로 전송하는 역할을 수행한다. 이러한 행위는 정기적으로 발생하도록 하여 간단없이 호스트 프로세스 정보를 모니터링 하여 비인가SW 사용여부를 지속 확인해야한다. 비인가SW를 사용하게 되면 최단시간에 네트워크에서 논리적으로 분리시켜 네트워크의 안정성을 보장해 줄 수 있도록 한다. 하지만 해당 네트워크에서 분리시킨 호스트가 이후 비인가SW에 해당하는 프로

세스를 중단시키면 다시 네트워크에 연결시켜 가용성이 보장되어야 한다. 이를 위해 Agent는 호스트가 네트워크로부터 논리적으로 분리되어 있어도 SDN Controller와 지속 통신을 유지하여야 한다.

3.2 시스템 구조

비인가SW 차단 시스템은 SDN 환경을 기반으로 하기 때문에 SDN의 기본구조를 따른다. 즉 SDN의 구성요소인 Data Plane, Control Plane, Application Plane을 그대로 따르고 이 안에서 일부 계층과 호스트에 소프트웨어적인 기능을 통해 구현한다. 제안하는 SDN 환경에서의 비인가SW 차단 시스템 구조는 Fig. 3과 같은 구조로 도식화 할 수 있다. SDN 계층을 이루는 하위계층부터 상향식 접근법을 통해 각 Plane에 대해 살펴보면, 가장 하위계층인 Data Plane에는 OpenFlow를 지원하는 스위치들과 이에 연결되어있는 호스트들로 구성되어 있다. 비인가SW를 사용하는 호스트를 확인하기 위해 각 호스트마다 비인가SW를 탐지하는 Agent를 설치한다. 각 Agent들은 호스트에서 작동하는 프로세스를 탐지하여 목록화 하고 이를 주기적으로 Controller로 전송하는 역할을 한다.

Control Plane 계층에는 Controller가 DB서버와 연동하여 Open vSwitch(OpenFlow를 지원하는 스위치)를 통제하게 된다. DB서버는 호스트에 반드시 설치되어 있어야 하거나 일반적으로 호스트에 설치가 가능하도록 인가된 SW의 프로세스 목록인 White-List를 보관한다. 프로세스 목록을 White-List 기반으로 하는 목적은 개인이 일반적으로 사용하는 단말이 아니라 조직에서 업무용으로 사용하는 단말이기 때문이다. 조직에서 사용하는 프로그램은 업무를 위해 사용하기 때문에 사용하는 SW가 정해져 있고 프리웨어가 아니라 라이센스를 구매해서 사용한다. 따라서 조직에서 단말기에 설치되는 SW에 대한 예측이 가능하기 때문에 조직에서 정한 SW 제품만 동작이 가능하도록 함으로써 강력한 보안을 갖출 수 있을 뿐만 아니라 저작권 측면에서도 조직을 보호할 수 있다.

Application Plane 계층에서는 DB서버와 논리적으로 통신하며 DB서버내 White-List 목록과 맞지 않은 프로세스 목록을 보낸 호스트와의 접속을 차단하도록 Controller를 통제하는 '비인가SW 차단 어플리케이션'이 설치된다.

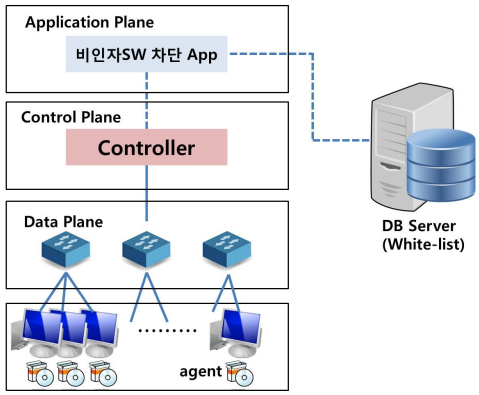

Fig. 3. Unauthorized SW blocking in Software Defined Network environment

3.3 동작 절차

본 논문에서 제안하는 시스템의 동작 절차는 호스트의 Agent에서부터 시작된다. Fig. 4처럼 호스트에 설치된 Agent가 호스트내에서 동작 중인 프로세스 목록을 수집하고 실시간으로 Open vSwitch를 통해 Controller로 전송한다. 정상적으로 동작하는 Agent는 수집된 프로세스 리스트를 Controller로 전송하지만 만약 Agent가 어떠한 이유로든 동작을 멈추어 Controller와 통신이 되지 않는 상황이 되면 즉각 보안위반행위로 간주하여 네트워크에서는 해당 IP의 패킷을 스위치에서 Drop 시킨다. 이는

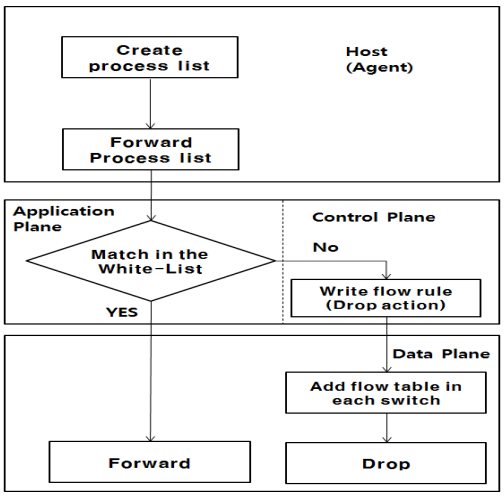

Fig. 4. Unauthorized SW blocking system in SDN environment algorithm

Agent가 사용자 의지와 상관없이 지속적으로 동작하여야 감시가 가능하기 때문이며 이를 통해 호스트에서 비인가 행위를 하는 것을 지속적으로 모니터링하여 네트워크를 보호 할 수 있다.

Controller의 Application Plane에서 Agent로부터 전송받은 프로세스 목록을 DB서버에서 관리중인 White-List와 비교한다. 비교한 후 DB서버 내에 저장된 White-List와 맞지 않는 프로세스 이름이 있다면 해당 프로세스 목록을 전송한 호스트와의 접속을 차단하도록 Flow Rule을 생성한다. 이 Rule을 수신한 스위치는 해당 IP의 패킷을 전달하지 않고 Drop하여 네트워크에서 논리적으로 분리시킨다. 이를 통해 비인가SW에 의해 네트워크가 위험에 노출되는 것을 차단할 수 있다.

비인가SW를 실행시킨 호스트가 네트워크에서 단절되어도 Controller와 해당 호스트의 Agent사이에서는 지속적으로 통신이 이루어진다. Flood Rule을 통해 Agent와의 통신은 지속적으로 유지될 수 있도록 하였으며, 이를 통해 호스트에서 비인가SW의 프로세스를 정지시키면 다시 네트워크에 논리적으로 연결하여 가용성을 보장한다.

IV. 시스템 구현 및 실험

4.1 시스템 구현

제안한 비인가SW 차단 시스템을 가상 네트워크 환경에 구현하여 실제 동작여부와 성능을 평가하였다. 구현을 위한 환경은 Table 1과 같고 Agent는 Python 코드로 구현하였고, Controller는 Python 기반의 POX를 사용하였다. Controller의 Application Plane에 구현되는 어플리케이션도 Python 코드로 제작하였다.

Table 1. Emulation environment

	HW / SW
CPU	Intel(R) Core(TM) i5 2.67GHz
RAM	6GB
Virtual machine	VMware® Workstation14 Pro (Ver. 14.0.0)
OS	Linux(Ubuntu 14.04.4)
Emulator	Mininet(2.2.2)
Controller	POX
SDN Switch	Open vSwitch

4.2 실험 방법 및 결과

실험은 세 가지 방법으로 수행하였다. 먼저 시스템이 정상적으로 동작을 하는지 여부를 확인하기 위한 실험(실험 1)과 각 호스트의 수가 시스템의 차단 성능(Agent가 프로세스 리스트를 보내는 순간부터 Controller에 의해 차단되는 시간)에 미치는 영향을 확인하는 실험(실험 2)을 진행하였다. 마지막 실험은 토폴로지를 변경하여 비인가SW 차단 기법이 네트워크의 성능에 미치는 영향을 확인하기 위한 실험(실험 3)이다.

실험 1의 토폴로지는 Fig. 5와 같이 Controller 1, Open vSwitch 2, host 6로 구성하였다. 스위치 1에 연결되어 있는 host 1(IP : 10.0.0.1), host 2(IP : 10.0.0.2)를 대상으로 host 1은 임의로 동작중인 프로세스에 'malware'라는 프로세스를 포함시키고, host 2는 현재 호스트에서 동작 중인 프로세스를 목록화하여 Controller에 전송한다. 스위치 2에서도 host 4(IP : 10.0.0.4)과 5(IP :

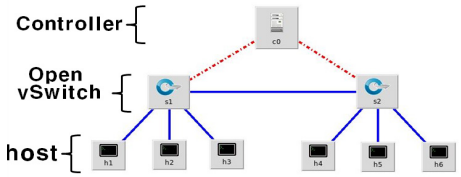

Fig. 5. Network topology

Table 2. Switch 1,2 Flow Rule

Switch 1(s1)
NXST_FLOW reply (xid=0x4): cookie=0x0, duration=86.356s, table=0, n_packets=460, n_bytes=102028, idle_age=1, priority=60000,tcp,tp_dst=6974 actions=FLOOD cookie=0x0, duration=0.55s, table=0, n_packets=0, n_bytes=0, idle_age=0, ip,nw_src=10.0.0.1 actions=drop cookie=0x0, duration=0.791s, table=0, n_packets=0, n_bytes=0, idle_age=0, ip,nw_src=10.0.0.4 actions=drop cookie=0x0, duration=86.346s, table=0, n_packets=376, n_bytes=25360, idle_age=0, priority=100 actions=FLOOD
Switch 2(s2)
NXST_FLOW reply (xid=0x4): cookie=0x0, duration=90.522s, table=0, n_packets=490, n_bytes=108674, idle_age=1, priority=60000,tcp,tp_dst=6974 actions=FLOOD cookie=0x0, duration=1.591s, table=0, n_packets=0, n_bytes=0, idle_age=1, ip,nw_src=10.0.0.1 actions=drop cookie=0x0, duration=2.106s, table=0, n_packets=0, n_bytes=0, idle_age=2, ip,nw_src=10.0.0.4 actions=drop cookie=0x0, duration=90.512s, table=0, n_packets=401, n_bytes=27066, idle_age=0, priority=100 actions=FLOOD

10.0.0.5)를 스위치 1과 동일하게 진행하였다. 각 호스트들의 Agent를 동작 시킨 후 각 스위치에 설치된 Flow Rule을 보면 Table 2와 같이 호스트 1, 4의 패킷이 Drop 되어 네트워크로부터 논리적으로 분리되었단 것을 알 수 있다.

실험 2는 스위치 2개에 균등하게 호스트들을 추가하여 호스트 수가 차단성능(Agent가 프로세스 리스트를 보내는 순간부터 Controller에 의해 차단되는 시간)에 어떠한 영향을 미치는지 확인하였다. 각 호스트는 모두 'malware'를 포함한 프로세스 리스트를 전송하여 스위치에 의해 차단되도록 설정하였고, 호스트의 수는 10개, 20개, 30개, 40개씩 늘려가며 차단되는 시간을 각각 200회씩 측정하였다. 실험결과 Fig. 6에서와 같이 호스트의 수가 10대 인 경우에는 0.2sec 내외로 대부분이 차단되었고 나머지도 대부분이0.2~0.4sec 내외로 차단되었다. 또한 호스트가 20대에서 40대로 늘어남에도 차단시간에는 큰 변화가 없었다. 또한 모든 호스트들이 비인가SW를 사용하는 것으로 가정하여 실험했음에도 90% 이상이 0.2~0.6sec 내에 차단 된 것으로 보아 본 시스템이 네트워크에 큰 부하를 주지 않고 동작하고 있음을 알 수 있다.

실험 3에서는 본 논문에서 제안하는 기법이 네트워크 성능에 얼마나 영향을 미치는지를 확인하는 실험이다. 이를 위해서 네트워크 성능을 판단하는 요소 중 지연(Latency)을 측정하였다. Controller 1대에 Open vSwitch 2대, 각 스위치당 host 5대가 연결 된 네트워크 토폴로지에서 실험을 진행하였으며, 비인가SW 차단 시스템 적용 전후를 비교하여 성능을 비교하였다.

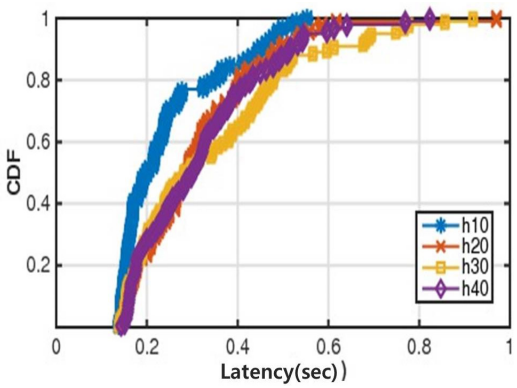

Fig. 6. Blocking time according to the number of hosts

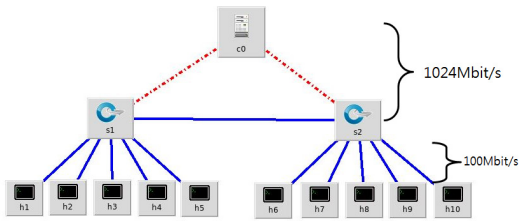

Fig. 7. Network Topology

토폴로지는 Fig. 7에서처럼 스위치단은 일반적으로 광케이블로 되어 있으므로 1024Mbit/sec, 스위치와 호스트는 UTP 케이블로 가정하여 100Mbit/sec로 대역폭을 설정하였다. 네트워크에서 지연을 발생시키는 원인인 전파지연, 큐 지연, 전송지연 중 네트워크 상 부하가 많은 경우에 발생할 수 있는 지연인 큐 지연에 의해 발생하는 지연을 평가하였다.

비인가SW 차단 시스템을 전 호스트에서 동작시키고 h1에서 h10으로 Ping 패킷을 60회 보낸 후 지연 시간을 측정 후 측정값은 CDF로 분석하였다. 비인가SW 차단 시스템을 동작시키기 전의 결과를 보면 Fig. 8에서와 같이 약 90%가 0.1~0.2msec 사이의 값으로 측정되었다. 일부 큰 값(약0.6msec)은 처음 통신할 때 Flow Rule이 설치되지 않은 상태에서 Controller를 경유하여 생기는 지연이다. 처음에 스위치는 Flow Table에 없는 경로는 Controller를 경유하여 Controller로부터 Flow Rule을 받고 이를 Table에 설치함으로써 경로를 설정하여 패킷을 전달할 수 있기 때문이다. 비인가SW 차단 시스템을 동작시키기 전 실험과 마찬가지로 일부 높은 값들은 Controller를 경유하여 생긴 지연임을 감안하여도 비인가SW 차단 시스템 동작 전 보다 0.3~0.4msec 지연이 더 발생한 것을 알 수 있다. 하지만 실제 우리가 인지할 수 있는 만큼의

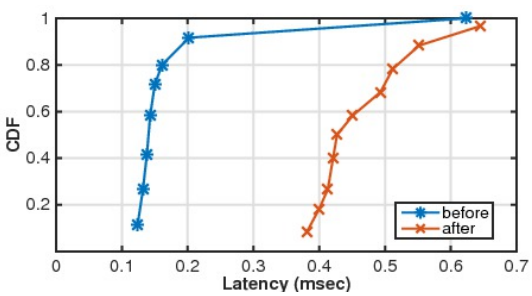

Fig. 8. Comparison of Delay before and after System operation

차이는 아니었으므로 사용자 입장에서의 가용성에 크게 영향을 미치는 수준은 아니었다.

V. 결 론

본 논문에서는 SDN 환경에서 비인가SW 차단을 통해 네트워크보안을 강화 할 수 있는 시스템을 제안하고 이를 구현하였다. 기존 SDN의 소프트웨어적인 측면을 활용하여 방화벽, IPS 등과 같은 외부 네트워크로부터 SDN기반의 네트워크를 보호하는 연구가 아닌 호스트에 설치된 Agent와 연계된 시스템을 통해 내부자 위협에 대응할 수 있게 하였으며 이를 통해 악성코드 감염의 원인이 될 수 있는 비인가SW 사용을 차단함으로써 보안성을 높여 안전한 네트워크를 구축할 수 있을 뿐만 아니라 저작권 논란도 해소해줄 수 있을 것이다. 향후에는 Agent에 프로세스 목록 뿐 아니라 호스트의 보안위반 행위를 감시할 수 있는 다양한 기능을 부여하여 조직의 특성에 맞는 맞춤형 보안을 제공 할 수도 있다. SDN은 앞으로 네트워크 시장을 선도할 것이고 확장성과 유연성이 있는 만큼 SDN의 특성을 최대한 활용할 수 있는 연구가 필요하다.

References

[1] BSA Global Software Survey, "Software Management: Security Imperative, Business Opportunity," https://www.bsa.org/~/media/Files/StudiesDownload/2018_BSA_GSS_Report_en.pdf, Jun. 2018

[2] Lee, Bum-Chul, Yang, Hee-Hee, and Byung-Sun Lee, "SDN / NFV / Cloud Trends", Electronic Telecommunications Trends Vol. 30, No. 1, pp.88-90, Feb. 2015

[3] Seungwon Shin and Guofei Gu. "CloudWatcher:Network Security Monitoring Using Openflow in Dynamic Cloud Networks", Proceedings of the ICNP affiliated 7th Workshop on Secure Network Protocols(ICNP-NPSec), Oct. 2012

[4] Alsmadi I.M. and AlEroud A., "SDN-Based Real-Time IDS/IPS Alerting System," Information Fusion for Cyber-Security Analytics, vol. 691, pp. 297-306, Oct. 2016.

[5] 'Gartner's Network Access Control Model', Gartner, Lawrence Orans, Aug. 2005

[6] Why hackers are targeting small businesses, ITWorld, http://www.itworld.co.kr/insight/91446, Jan. 16 2015(Oct. 17 2018)

〈저자소개〉

강 남 길 (Nam-Gil Kang) 정회원
2007년 3월 육군사관학교 토목공학과 졸업
2019년 1월: 국방대학교 컴퓨터공학과 석사
〈관심분야〉 정보보호 SDN

권 태 욱 (Kwon TaeWook) 정회원
1986년 3월: 육군사관학교 전산학과 졸업
1995년 9월: 미 해군대학원 컴퓨터공학 석사
2001년 2월: 연세대학교 컴퓨터공학 박사
2007년 6월~현재: 국방대학교 컴퓨터공학과 교수
〈관심분야〉 네트워크, 센서네트워킹, SDN