Software Defined Network의 보안 취약성* †

정제성[1], 김광조[1) 2)]

카이스트 정보보호대학원[1)] / 전산학과[1) 2)]

Security Issues in Software Defined Network

Je-Seong Jeong[1)], Kwangjo Kim[1) 2)]

Graduate School of Information Security[1)] / Dept. of Computer Science, KAIST[2)]

요 약

인터넷은 우리의 일상에서 이제 불가분의 중요한 역할을 하고 있으며 사물인터넷이 본격적으로 일상에 적용 될 시에는 이 역할은 더욱 커질 것이라 예상한다. 하지만 현재의 네트워크 장비는 벤더에 의해서 정해진 룰에 따라 작동이 되는 시스템으로 관리를 해야 하는데 이러한 방법은 관리 하는데 있어 어려움이 있으며 새로운 기능을 추가 할 시에는 연관 된 모든 장비를 업데이트 또는 교체해야하는 불편 사항이 있다. 또한, 각종 새로운 악성 공격으로부터도 보안 상의 취약성을 보이고 있다. 따라서 이를 해결하고자 등장한 것이 소프트웨어 정의 네트워크(Software Defined Network : SDN)로 기존의 네트워크 장비와는 달리 control plane과 data plane으로 나뉘어져 있으며 이로 인해 네트워크 구조가 단순화 되어 있으며 기존 네트워크보다 악성 공격에 대하여 일부 강점이 있으나 SDN도 보안에 관하여는 완벽한 해결책이 없으며 여전히 취약한 면이 있는 것도 사실이다. 따라서 본 논문에서는 SDN의 동작 원리를 설명 한 후에 현재까지 알려진 SDN에 관한 보안 취약점에 관하여 조사 분석하고, 해결 방안을 제시한다.

I. 서 론

현재 우리 사회는 누구나 손쉽게 인터넷을 통해 정보를 얻고 업무를 할 수 있는 사회이다. 그리고 스마트폰의 보급률이 증가함에 따라 데이터 사용량은 기하급수적으로 증가하고 있다. 또한 미래 산업으로 주목 받고 있는 사물인터넷(Internet of Things : IoT)의 사용이 증가할수록 이러한 추세는 더욱 증가할 것이라 예상된다. 따라서 이러한 데이터 사용이 원활히 이루어지게 하기 위해서는 안정적인 네트워크 망이 지원이 돼야 할 것이다. 하지만 현재의 네트워크망은 매우 복잡한 구조로 되어있어 이를 관리하는데 고비용을 초래하고 있다. 그리고 각종 악성 공격에도 취약한 모습을 보이고 있으며 이러한 사례로는 정부 기관에 대한 DDoS 공격, 금융 기관에서의 개인 정보 유출, 그리고 최근에 발생 했던 원전 해킹 시도 등이 있다. 따라서 이러한 문제점을 개선하고자 등장한 것이 소프트웨어 정의 네트워크(Software Defined Network : SDN)이다. SDN은 기존의 네트워크 장비와는 달리 control plane과 data plane으로 구분하여 네트워크를 통합 관리 할 수가 있다. 그리고 이를 통해 네트워크 구조를 단순화 시키고 데이터 분산 처리를 통해 네트워크의 사용 효율을 극대화 시킬 수 있다. 또한, SDN 컨트롤러를 이용하여 각종 악성 공격을 사전 방지 및 사후 조치를 할 수 있다. 하지만 이런 SDN도 여전히 보안상의 취약점이 있는 것은 사실이다. 따라서 본 논문에서 SDN의 보안 취약점에 대해서 설명하고 이를 극복할 수 있는 방안에 대해서 제안하고자 한다. 본 논문의 구성은 다음과 같다. 제2장에서는 SDN에 대해 설명하고 제3장에서는 SDN의 보안 취약점에 대해서 기술하고, 제4장에서는 보안상의 개선 방안을 제시하고 마지막으로 제5장에서는 결론을 맺는다.

* 본 연구는 미래부가 지원한 2014년 정보통신·방송(ICT) 연구개발사업의 연구결과[1391104001, 생체모방 알고리즘을 활용한 통신기술 연구]로 수행되었습니다.

† This research was supported by the KUSTAR-KAIST Institute, KAIST, Korea.

II. SDN

본 장에서는 SDN의 구조에 대해서 간략히 알아보며, SDN 인터페이스 중 하나인 OpenFlow가 무엇인지 살펴본다.

2.1 SDN 구조

SDN은 크게 데이터 계층(infrastructure layer)과, 제어 계층(control layer), 애플리케이션 계층(application layer)으로 나뉜다. 데이터 계층은 SDN의 특정 인터페이스를 통해 제어를 받는 계층으로서, 데이터 흐름의 전송을 담당한다. 제어 계층은 데이터의 흐름을 제어하는 계층으로서 애플리케이션과 네트워크 서비스를 통하여 데이터 흐름을 라우팅 할 것인지, 전달을 할 것인지, 거절 할 것인지를 결정한다. 또한 데이터 계층의 동작들을 정리하여 API(Application Programming Interface) 형태로 애플리케이션 계층에 전달한다. 마지막으로 애플리케이션 계층은 제어 계층에서 제공한 API들을 이용하여 네트워크의 다양한 기능들을 수행 할 수 있도록 한다. Fig.1은 SDN의 구조[1]를 간단히 도식화하였다.

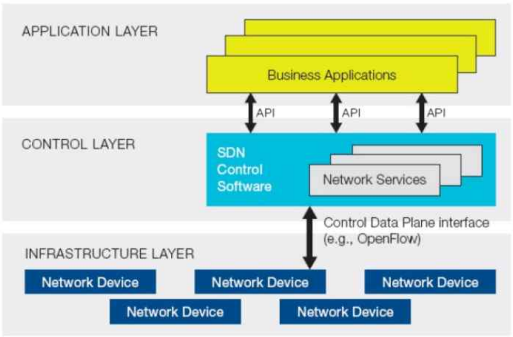

Figure. 1. Software Defined Network Architecture

2.2 OpenFlow

현재 OpenFlow는 데이터 계층을 제어하기 위해 사용되는 인터페이스 중 하나이며 가장 널리 사용되고 있다. OpenFlow는 SDN의 첫 번째 표준으로서 OpenFlow를 지원하는 스위치는 Flow Table를 가지고 있으며, 컨트롤러와 별도의 채널을 통해 연결되며 Fig.2는 OpenFlow 구조이다[2].

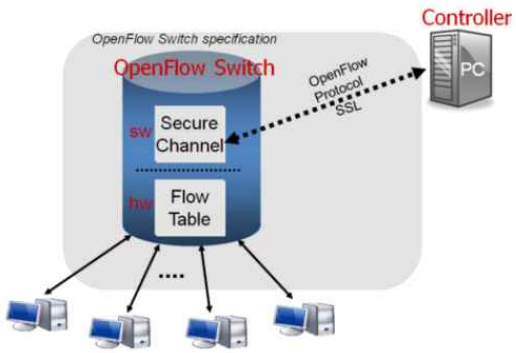

Figure 2. OpenFlow structure

2.2.1 OpenFlow Switch

기존 네트워크 장비에서는 벤더 간에 Flow Table이 달라 서로 정보 공유가 되지 않았다면 OpenFlow 스위치는 공통 된 Flow Table을 개발 적용하여 네트워크 장비 간 서로 정보를 공유할 수 있게 되었다. 이를 통해 개발자는 새로운 라우팅 프로토콜과 보안 모델, 주소 체계를 만들어 전체 네트워크에 적용할 수 있다. 이런 OpenFlow 스위치는 다음과 같이 3부분으로 구성되어 있다. 첫째, Flow 처리를 어떻게 해야 되는지 스위치에 전달하는 Flow Table이 있다. 둘째, 스위치와 원격의 컨트롤러를 연결하여 둘 사이에서 패킷 처리 방법과 패킷이 오고 갈 수 있도록 해주는 보안 채널이 있다. 셋째, 컨트롤러가 스위치와 통신 할 수 있도록 개방형 기준을 제공하는 OpenFlow 프로토콜이다. 이를 통해 Flow Table을 네트워크 전체에 적용 하여 개발자가 스위치 프로그램을 만들 필요를 없게 하였다[3].

III. SDN 취약점

본 장에서는 차세대 네트워크 기술로 주목 받는 SDN의 취약점을 컨트롤러와 OpenFlow로 나눠서 설명한다[3].

3.1 컨트롤러 취약점

SDN은 컨트롤러를 통해 네트워크의 흐름을 제어할 수 있고 이를 통해 네트워크를 단순화 할 수 있는 것이다. 그러나, 컨트롤러로 인해 발생 가능한 보안 문제점은 다음과 같다. 첫째, 컨트롤러가 악성코드에 감염 된 사례이다. 만약 컨트롤러가 악성코드에 감염이 된다면 공격자는 프로그램을 재설치 하여 네트워크 상에 있는 데이터 스니핑(sniffing) 또는 드로핑(dropping) 할 수 있을 것이다. 둘째, 관리자가 나쁜 의도를 가졌을 경우이다. 컨트롤러의 룰을 작성하는 관리자가 나쁜 의도를 가지고 컨트롤러의 룰을 변경한다면 네트워크가 정지가 될 수도 있으며 정보를 유출 할 수도 있을 것이다. 셋째, control plane과 data plane 사이의 DDoS 공격이다. 이 공격은 컨트롤러가 정상적으로 데이터 계층에 지시를 내리지 못하여 하여 정상적인 작동을 방해한다[4].

3.2 OpenFlow 취약점

OpenFlow는 SDN의 대표적인 인터페이스로서 control plane과 data plane 사이의 가교 역할을 통해 원활히 SDN 네트워크가 작동되게 하는 중요

한 역할을 하고 있다. 하지만 이 역시 악성 공격에 취약한 면이 있다. 첫째, MAC 주소, IP 주소, 포트 등 네트워크 통신과 관련된 정보를 속여 공격 대상자의 정보를 획득 또는 정상적인 서비스를 제공하지 못하게 하는 공격하는 스푸핑(spoofing) 공격이 있다. 이러한 공격 기술을 OpenFlow에 적용을 한다면 공격자가 컨트롤러인 것처럼 행동을 하여 OpenFlow 스위치의 행동을 조작하여 데이터 스니핑 혹은 드로핑 공격을 할 수 있으며 또한 네트워크 자체를 정지시킬 수도 있다. 둘째, 시스템의 정상적인 기능을 변경하여 보안 기능의 약화를 초래하거나 기능을 하지 못하게 하는 공격으로서 OpenFlow를 통해 data plane의 장비들이 이상 동작하게 만드는 tampering 공격이 있다. 셋째, 통신의 모든 또는 부분에 관여하여 송수신 사실을 부인하는 공격인 repudiation 공격이 있으며 이 역시 OpenFlow를 통해서 조작이 가능 하겠다. 넷째, 시스템을 악의적으로 공격해 해당 시스템의 자원을 부족하게 만들어 정상적인 동작을 하지 못하게 하는 분산 서비스 거부 공격(DDoS : Distributed Denial of Service)이 있다. 만약 이러한 공격을 통해 OpenFlow 스위치에 위조된 패킷을 대량으로 전달한다면 SDN 네트워크는 과부하가 걸려 정상적인 작동이 제한될 것이다. 다섯째, 시스템을 속여 임의적으로 인증되지 않은 권한을 갖도록 하는 권한 상승(elevation of privileges) 공격이 있으며 이 역시 OpenFlow 상에서 가능한 공격이다[5].

위와 같이 SDN은 기존의 네트워크를 효율적으로 운영하기 위해 control plane과 data plane으로 나누어 관리를 하고 있지만 이로 인해서 보안 취약점이 발생하는 것을 알 수 있으며. 그밖에 SDN은 Transport Layer Security(TLS) 보안 프로토콜 적용 시 발생하는 속도지연으로 인해 TCP를 사용함으로써 중간자 공격 등에 취약하다.

IV. 향후과제

SDN은 control plane과 data plane을 구분함으로써 네트워크 구조를 단순화 시켰으며 컨트롤러를 통해 네트워크를 통제함으로써 효율성을 극대화 시켰다. 또한 애플리케이션을 제작, 적용 할 수 있어 사용자의 요구에 맞는 네트워크 환경을 구성 할 수 있다. 하지만 SDN도 이미 기술한 바와 같은 취약점이 있다. 이를 해결하기 위해서는 다음과 같은 조치가 필요하다. 첫째, 컨트롤러가 악성코드에 감염이 되지 않도록 IDS/IPS를 control plane과 data plane에 두어 방어 할 수 있도록 해야 한다. 또한, 컨트롤러 프로그램이 의도하지 않게 변경되었을 시는 자동으로 시스템을 재부팅시키고 이전으로 돌아 갈 수 있는 시스템을 만들어야 한다. 둘째, SDN 네트워크에 적합한 보안 프로토콜의 개발이 필요하다. TLS를 사용할 때 네트워크의 과부화가 걸린다는 이유로 현재 SDN에 TLS를 사용하지 않은 상태로 TCP를 사용해서 서로를 확인하고 있다. 따라서 control plane과 data plane 사이에서 가능한 보안 프로토콜을 개발하여 이를 보완해야 한다. 셋째, Open-Switch의 보안성을 강화하는 방안으로 악성 코드를 자체적으로 방어 할 수 있는 시스템을 갖춰야겠다.

V. 결론

네트워크의 사용이 날로 중요해져가는 현 시점에 미래의 안전한 네트워크 시스템을 구성하는 것은 매우 중요하다. 그리고 이러한 요구에 SDN은 적합한 미래 기술이라고 할 수 있으나 본 고에서 기술한 바와 같이 SDN은 아직 보안상 취약점이 있다. 따라서 이를 해결하는 것은 앞으로 SDN을 실제 적용하는데 있어 매우 중요하다 할 수 있다.

References

[1] Taejune Park, Seungsoo Lee, Seoungwon Shin, "A reflectornet Based on Software Defined Network", The Journal of Korea Information and Communications Society '14-06 Vol.39B No.06

[2] OpenFlow, "How does OpenFlow Work", http://yuba.stanford.edu/cs24- 4 wiki/index.php/Overview

[3] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, Jonathan Turner, "OpenFlow: Enabling Innovation in Campus Networks", ACM SIGCOMM Computer Communication Review, Number 2, Apr. 2008

[4] Diego Kreutz, Fernando M. V. Ramos, Paulo Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, Steve Uhlig, "Software-Defined Networking: A Comprehensive Survey", arXiv :1406.0440v3 [cs.NI] 8 Oct. 2014

[5] Rowan Kloti, Vasileios Kotronis, Paul Smith, "OpenFlow: A Security Analysis", IEEE, 2013