K-ONE 기술 문서 #9

가상 네트워크의 접근 제어를 위한 Security-Mode ONOS

Document No. K-ONE #9
Version 1.0
Date 2016-04-29
Author(s) 이승현

K-ONE

ONK
OpenNetworkingKorea

■ 문서의 연혁

버전	날짜	작성자	내용
초안 - 0.1	2016. 04. 25	이승현	초안 작성

본 문서는 2015년도 정부(미래창조과학부)의 재원으로 정보통신 기술진흥센터의 지원을 받아 수행된 연구임 (No. B0190-15-2012, 글로벌 SDN/NFV 공개소프트웨어 핵심 모듈/기능 개발)

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No. B0190-15-2012, Global SDN/NFV OpenSource Software Core Module/Function Development)

기술문서 요약

새로운 네트워크의 페러다임이라고 불리우는 소프트웨어 정의 네트워킹은 기존 네트워킹 방법과는 다르게, 제어부와 데이터부를 분리시켜 네트워크의 단순화와 중앙집중식 네트워크 운용을 통하여 많은 기회를 창출하고 있다. 이러한 중앙집중식 네트워크의 운용을 위해 소프트웨어 정의 네트워크 상에서 데이터부와 제어부의 동작을 정의하는 프로토콜인 OpenFlow프로토콜을 기반으로, 해당 데이터부의 추상화를 담당하는 수많은 컨트롤러 들이 생겨나게 되었다. 대표적인 컨트롤러는 ONOS, Floodlight, OpenDayLight등이 있으며, 각 컨트롤러 별 장단점을 가지고 있다. 특히나 컨트롤러는 다양한 기능을 사용자에게 제공을 해 줘야 하기 때문에, Northbound API라는 여러 기능에 대한 API를 제공하게 된다. 기존 네트워크와 달리 소프트웨어 정의 네트워크기반의 네트워크는 이러한 API에 따라 동작하게 된다.

소프트웨어 정의 네트워크는 다양한 장점을 가지지만, 현재 컨트롤러에는 심각한 보안상의 문제를 내포하고 있다. 소프트웨어 정의네트워크 기반 애플리케이션이 악성코드나 버그가 포함될 수 있기 때문에, 단일 네트워크 애플리케이션이 전체 네트워크에 의도치 않은 영향을 줄 수 있다. 이러한 문제를 해결하기 위하여 ONOS기반의 Security-Mode ONOS가 제안되었고, 현재 많은 취약점 및 공격을 사전에 방지하고 있다. 하지만 Security-Mode ONOS에서 제어하지 못하는 데이터부는 아직도 취약한 상태로 남아있다.

본 기술문서에서는, Security-Mode ONOS가 제어하지 못하는, 가상네트워크 기반의 엑세스컨트롤 메커니즘을 실현 해 주는, SM ONOS-VN (Security-Mode ONOS for Virtual Network)을 제안한다. SM ONOS-VN은 Security-Mode ONOS가 제어하지 못하는 데이터부에 대한 최소권한메커니즘을 제공하여, Security-Mode ONOS와 더불어 전체 SDN의 스택을 보호한다. 메커니즘을 구현하여, Security-Mode ONOS의 데이터부의 확장모듈로서, 데이터부에 대한 제어메커니즘을 디자인하고 구축한다.

Contents

K-ONE #9. 가상 네트워크의 접근 제어를 위한 Security-Mode ONOS

그림 목차

표 목차

K-ONE #9. 가상네트워크의 접근 제어를 위한 Security-Mode ONOS

1. 서론

소포트웨어 정의 네트워크는 새로운 형태의 네트워킹 페러다임으로서, 기존 네트워크와는 다른 다양한 장점을 가지고 있다. 본 서론에서는 이러한 소프트웨어 정의 네트워크의 기본 지식과, 새로운 네트워크 기반에서 발생 할 수 있는 다양한 문제점들을 짚어본다. 그 후 널리 쓰이고 있는 컨트롤러 중 하나인 Open Network Operating System [1] 컨트롤러의 보안성을 향상시킨 Security-Mode ONOS에 대해 다루어 본 후, 본 연구의 목적인 가상네트워크에 대해 간략히 다룬다.

1.1. 배경지식

본 절에서는 해당 기술문서를 이해하기 위하여, 소프트웨어 정의 네트워킹, OpenFlow프로토콜, 그리고 Network Operating System에 대하여 간략하게 다룬다.

1.1.1. 소프트웨어 정의 네트워킹 (Software-Defined Networking)

소포트웨어 정의 네트워크는 새로운 네트워크의 페러다임으로서, 동적 네트워킹, 최소 비용 운용, 고 비용성의 네트워크 운용을 가능 해 주는 새로운 네트워크 아키텍쳐이다. SDN의 가장 큰 장점은 제어부(Control plane)와 데이터부(Data plane)을 독립시켜 네트워크의 중앙집중화 및 효율적인 네트워크 관리를 제공하는 것 이다. 제어부는 전체 네트워크의 핵심이 되는 다양한 컨트롤 로직이 동작하게 되고, 네트워크를 운용하는 오퍼레이터는 중앙집중식으로 네트워크를 관리 할 수 있다. 데이터부는 이와는 다르게, 제어부에서 요청하는 다양한 요청사항을 단순하게 처리만 하는 영역으로서, 기존네트워크와는 그 차별성을 두고 있다.

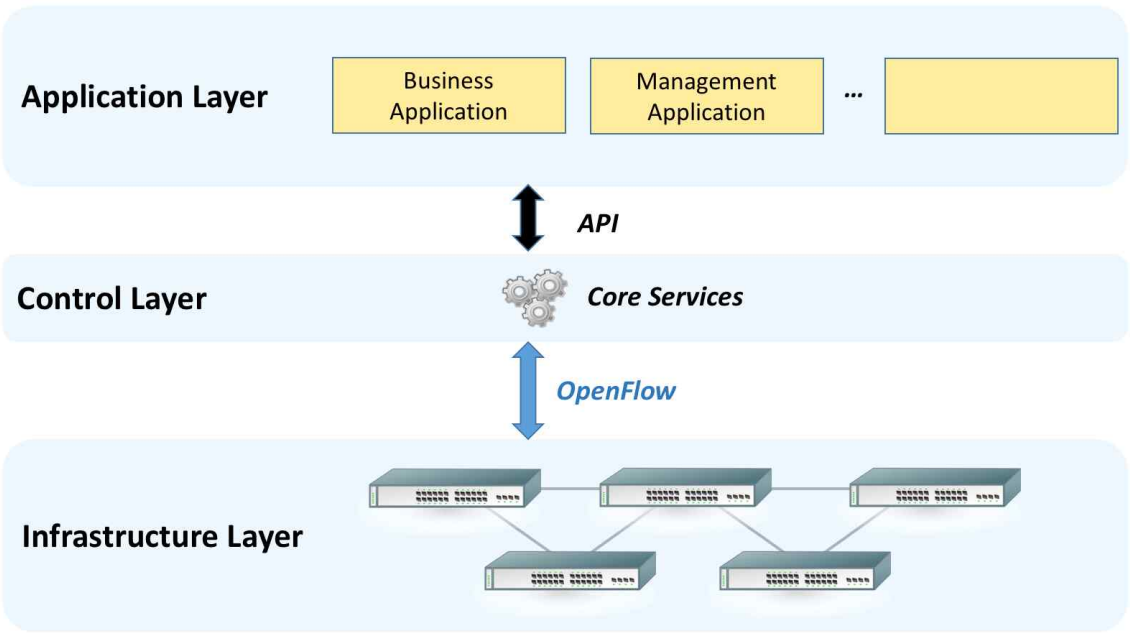

그림 1 소프트웨어 정의 네트워크 구조

그림 1은 SDN의 아키텍쳐를 보여준다. 가장 윗단의 애플리케이션 레이어는 비즈니스 운용에 필요한 다양한 컨트롤 로직이 올라가고, 중간의 컨트롤 레이어는 애플리케이션 레이어에게 다양한 기능을 제공 해 주고, 해당 요청을 데이터부에게 적용하는 역할을 하게 된다. 인프라스트럭쳐 레이어는 SDN에서 주로 쓰고있는 OpenFlow 프로토콜을 통하여 컨트롤 레이어로부터 명령을 받아 수행하게 된다.

1.1.2. OpenFlow

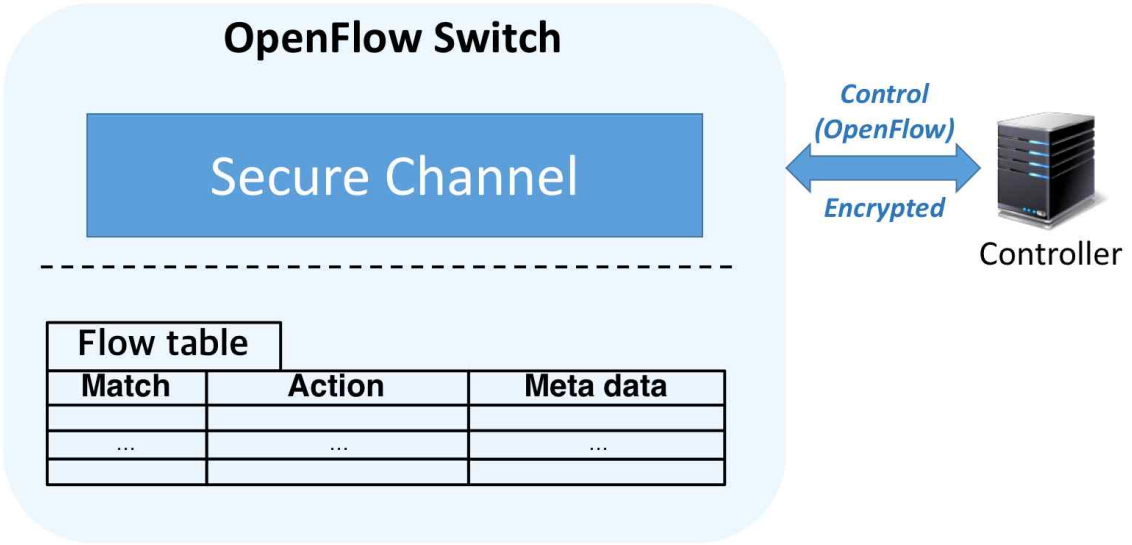

그림 2 OpenFlow 스위치 개념도

OpenFlow 프로토콜은 현재 널리쓰고있는 SDN기반의 네트워크상에서 제어부와 데이터부의 통신을 정의한 통신 규약이다. 2008년 스탠포드에서 제안된 해당 프로토콜은 현재 다양한 벤더들의 스위치 표준을 담당하고 있으며, 대부분의 SDN환경에서는 OpenFlow프로토콜을 사용하고 있다. OpenFlow의 화이트페이퍼에 기술되어 있는 OpenFlow의 스위치에 대한 다이어그램은 그림 2와 같다. 기본적으로 OpenFlow 스위치는 OpenFlow프로토콜을 정확하게 인지하고, 수행하는 것을 기본 원칙으로 하고, 제어부와 데이터부는 SSL과 같은 암호화 프로토콜로 암호화가 되어 있어, 안전한 통신을 원칙적으로 한다. 또한 내부에는 제어부의 동작을 받아들이는 부분과, 제어부의 동작에 따라 패킷을 제어하는 플로우 테이블로 이루어 져 있다. 플로우 테이블은 OpenFlow스위치의 기본으로서, 스위치로 들어오는 패킷은 해당 플로우 테이블에 따라 처리된다.

1.1.3. Network Operating System (Controller)

Network Operating System (Controller)는 SDN 네트워크의 핵심이다. 컨트롤러는 데이터부의 추상화를 통해 애플리케이션부에게 다양한 네트워크에 대한 기능을 제공한다. 컨트롤러는 일반적으로 OpenFlow프로토콜을 사용하여 데이터부를 제어하

게 된다. 기존네트워크에서는 이러한 제어를 할 수 있는 제어부가 스위치에 같이 있기 때문에, 중앙 집중화가 매우 어렵지만, SDN환경에서는 이러한 제어부가 스위치랑 분리가 되어 있기 때문에, 그림 3과 같이 중앙집중식으로 네트워크를 구성 할 수 있다.

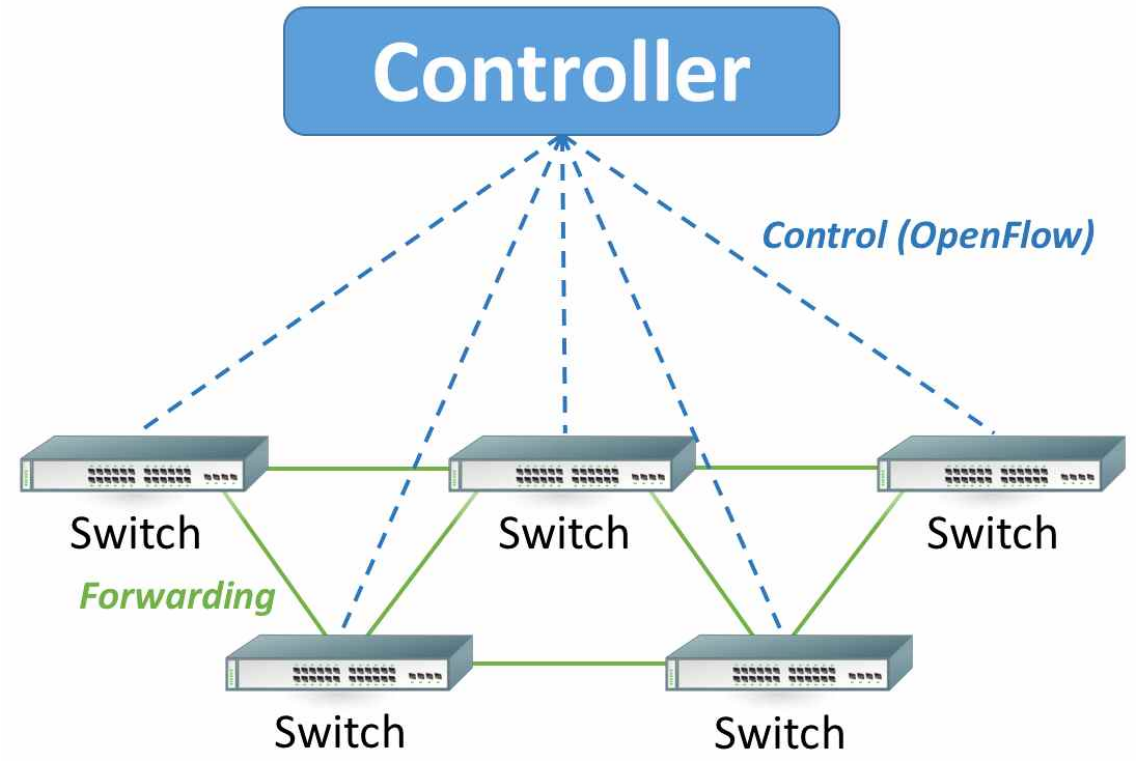

그림 3 SDN명령어 전달 구조

컨트롤러는 현재 다양하게 제안이 되고 있고, 대규모 네트워크를 처리하기 위한 분산형 컨트롤러도 제안되고 있다. 캐리어 네트워크의 운용을 담당하고, 현재 널리 쓰이고있는 ONOS컨트롤러는 확장성과, 간결한 코드로 널리 쓰이고 있고, 다양한 프로젝트도 진행되고 있다. 컨트롤러는 일반적으로 Southbound와 Northbound로 구성된다. Southbound는 데이터부를 제어하는 컴포넌트로 이루어 져 있고, 일반적으로 OpenFlow 프로토콜을 지원하게 된다. 또한 Northbound의 기능을 도와주기 위한 다양한 하위단의 기능들이 제공이 된다. Northbound는 잘 정의된 API의 집합으로서, 애플리케이션 부에게 데이터 부에 대한 추상화를 제공 해 주고, 통합적인 기능을 제공 해 준다.

1.2. NOS의 보안 문제점

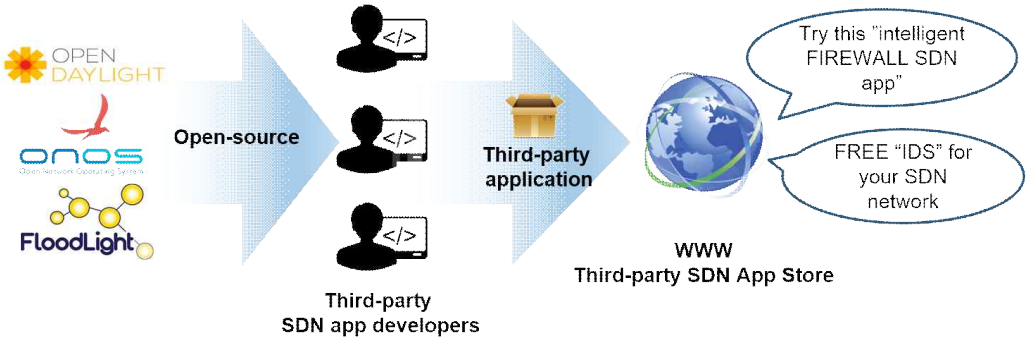

그림 4 오픈소스 기반 SDN 애플리케이션 배포과정

SDN은 새로운 네트워크 페러다임으로서 네트워크 운용에 대양한 장점을 제공 해 주지만, 편의성과 더불어 다양한 보안문제를 내포하고 있다. 보안 문제에서 가장 큰 문제점은, 현재 컨트롤러에서 제공되는 기능에 대한 어떠한 제약도 없기 때문에, 악성 애플리케이션이나, 사용자의 프로그래밍 실패는 전체 네트워크에 대한 실패로 직결 될 수 있다. Northbound에서 제공되는 다양한 API들은 전체 네트워크를 중앙에서 관리하고 제어 할 수 있기 때문에, 애플리케이션의 실패는 전체 네트워크의 마비를 불러이르키고, 네트워크 운용이 관리자의 의도와 다르게 수행 될 수 있다. 또한, 사용자가 인지하지 못하는 다양한 문제점을 이르킬 수 있고, 심지어는 컨트롤러 자체를 마비시킬 수 있다.

이러한 문제의 근본적인 원인은 그림4와 같은 오픈소스 기반 SDN 애플리케이션의 배포과정에 대한 구조 때문이다. 현재 널리쓰이고 있는 컨트롤러는 대부분이 오픈소스로서, Nox [4], ONOS [1], Beacon [5] 등이 있다. 기본적으로 올라가는 애플리케이션과 달리, 사용자는 잘 정의된 각각의 컨트롤러에서 제공되는 Northbound API를 사용하여 자신이 원하는 애플리케이션을 제작하게 되고, 해당 애플리케이션을 네트워크 관리에 사용되게 된다. 이러한 운용은 검증되지 않은 애플리케이션의 오동작을 보장 할 수 없기 때문에, 잠재적인 위협이 될 수 밖에 없다.

1.3. Security-Mode ONOS

앞서 언급한 문제를 해결하기 위해 ONOS의 프로젝트 중 하나인 Security-Mode ONOS는 애플리케이션을 위한 최소권한 모델을 디자인하고 수행하고 있다. 최소권한 전략은 널리 쓰이는 기술 중 하나로서, 대상에 대한 권한 요구사항을 명백히 분석하고, 해당 권한을 기반으로 대상이 할 수 있는 행위를 제안하는 기술이다.

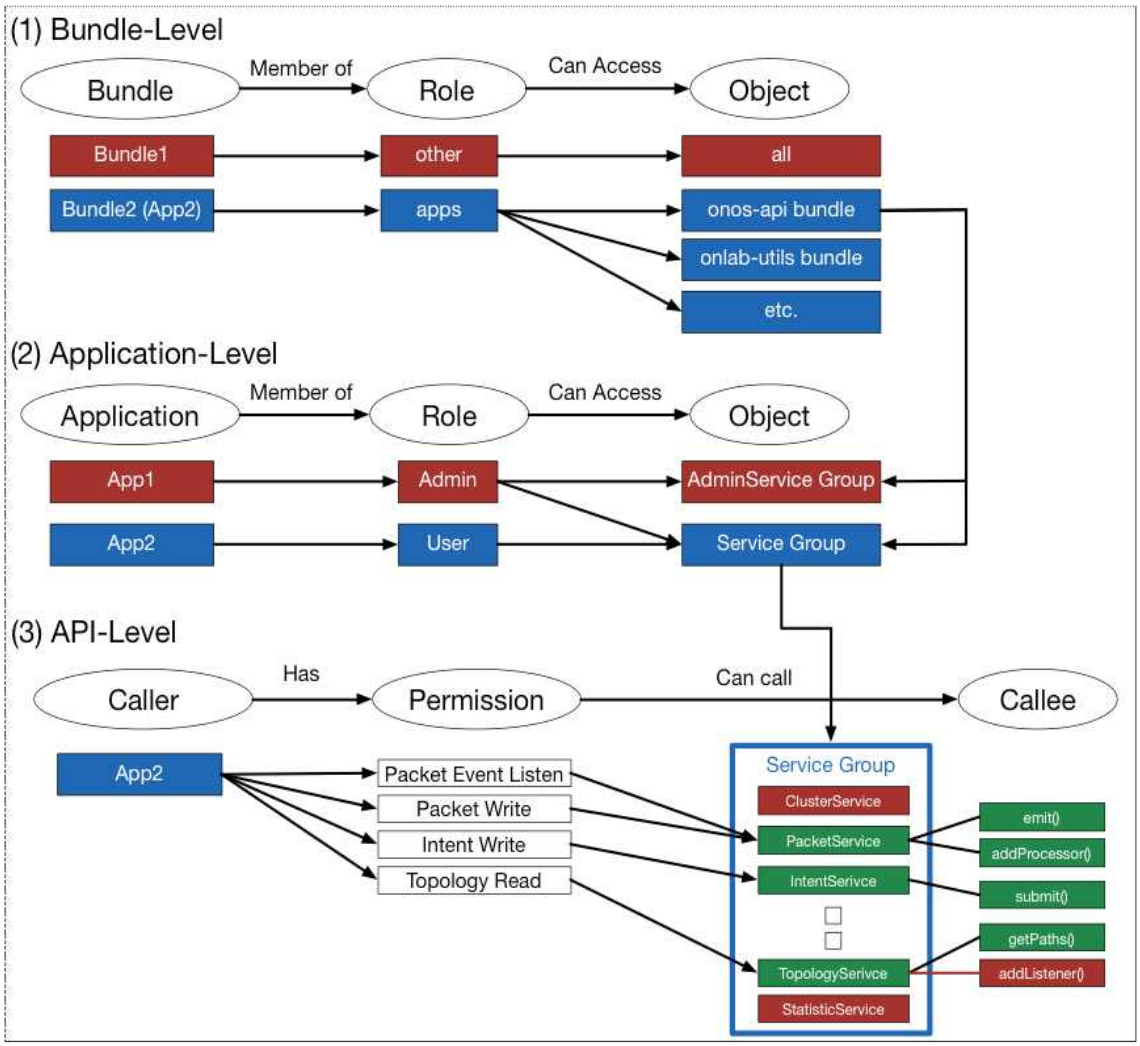

그림 5 Security-Mode ONOS 구조 [1]

그림 5는 Security-Mode ONOS의 기본 구조이다. 기본적으로 Security-Mode ONOS는 OSGI기반으로 동작하는 ONOS 컨트롤러의 기본 단위인 번들에 대한 권한 부여를 하는 것 이다. ONOS는 다양한 기능을 기능집합군인 번들을 통하여 움직이게 되고, 해당 번들은 동적으로 올라가게 된다. 번들보다는 조금 더 깊게 제어 할 수 있는 (Fine-grained) API수준의 권한제어도 제공하고 있다. 또한 애플리케이션 단위로 전반적인(Corase-grained) 권한을 부여 할 수 있는 시스템을 가지고 있다.

1.4. 기존 연구

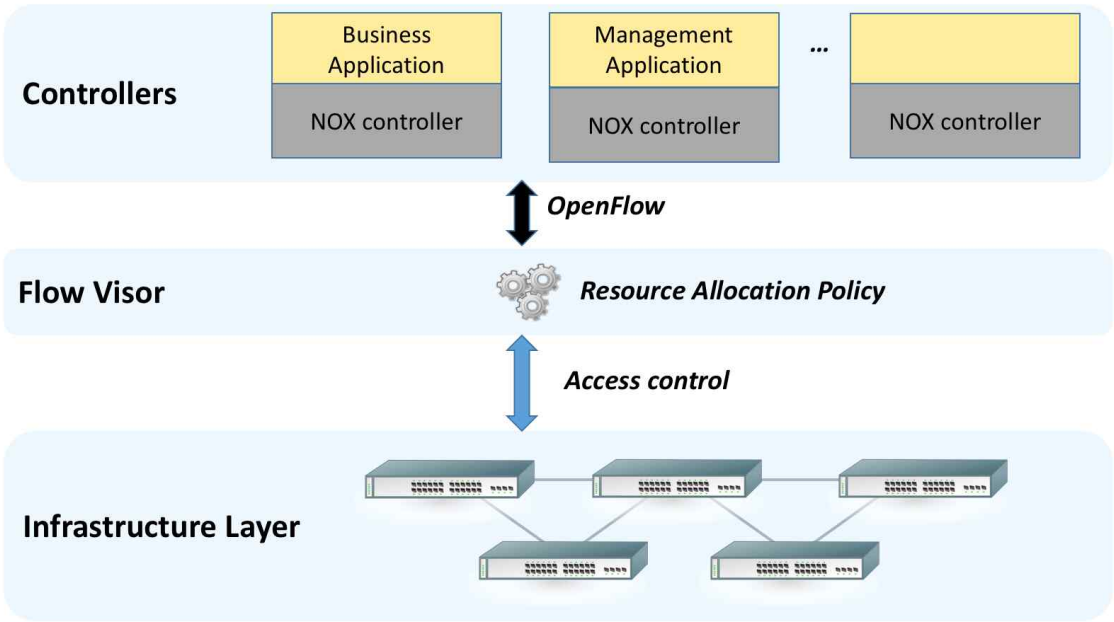

그림 6 FlowVisor 도식도

FlowVisor [2]는 SDN환경에서 데이터부를 분리시키려고 한 최초의 연구이다. 해당 연구에서는 다양한 네트워크 컨트롤러가 대상 네트워크를 위해 동작 할 때, 각각의 네트워크 컨트롤러가 영향을 미칠 수 있는 영역을 Flow Space로 나누어 관리를 하게 된다. 각각의 컨트롤러들은 아랫단의 FlowVisor를 신경쓰지 않고, OpenFlow로 룰을 내리는 것과 동일하게 룰을 내리면, 사전에 정의된 정책 구성에 따라 FlowVisor가 정절히 권한을 분배하여 데이터부에 영향을 주게 된다. 해당 기술은 기존에 존재하는 다양한 가상 네트워크 기술 기반의 네트워크 분리는 아니고, 논리적인 식별자를 부여하는 방법으로서, 기존과는 다른 차이점이 있다.

이와 비슷한 연구로서 OpenVirtex [3]가 있다. OpenVirtex는 FlowVisor와 유사한 구조도를 가지게 된다. 해당 연구는 FlowVisor의 후속연구로서, vSDN이라는 영역으로 각각의 다양한 컨트롤러에 따라 서로다른 네트워크 구성을 보여주는데 의의가 있다. 기존 FlowVisor에서는 NOX [4] 컨트롤러를 지원하지만 OpenVirtex는 완전한 가상네트워크로서, 다양한 컨트롤러 (Floodlight [7], Beacon [5], Pox [6])를 지원한다.

2. 본론

본 절에서는, 컨트롤러의 보안성을 향상시키고, SDN 구조상의 문제점을 해결하기 위한 Security-Mode ONOS가 제어하지 못하는 데이터부에 대한 최소권한 정책에 대해 다룬다. 본 절은 Security-Mode ONOS의 확장모듈로서, 데이터부의 가상네트워크에 대한 권한 부여 및 설계에 대하여 다룬다.

2.1. Security-Mode ONOS for Virtual Network (SM ONOS-VN)란?

SDN구조의 가장 큰 장점이자 취약점은, 중앙집중형 네트워크 관리 체계이다. 중앙집중식 네트워크 관리체계는 기존네트워크에서 하기 힘들었던, 빠르고 효율적인 네트워크 운용을 할 수 있는 것 인데, 네트워크 애플리케이션 자체가 많은 권한을 가지기 때문에 이로인해 다양한 보안 문제점이 발생하게 된다. 근본적인 문제점은 애플리케이션이 컨트롤러에서 제공하는 모든 기능을 아무런 제약없이 사용 할 수 있다는 것이 문제이다. 이러한 문제를 해결하기 위해 Security-Mode ONOS가 탄생하게 되었고, 애플리케이션은 사전 검증과정과 더불어 최소권한 정책으로 필요한 최소한의 기능만을 가지기 때문에, 보안성에 위배 될 수 있는 다양한 문제점들을 해결하게 되었다.

하지만 Security-Mode ONOS는 컨트롤러에서 제공하는 기능에 대한 최소 권한 정책을 부여하기 때문에, 기능상의 제약이 있다. 현대 엔터프라이즈 네트워크는 매우 큰 규모를 가지기 때문에, 단일 네트워크 상황에서 쓰기에는 제약이 많다. 이를 극복하기 위해 물리 네트워크상에 가상네트워크 개념을 두고, 각 테넌트별로 가상네트워크를 부여할 수 있는 시스템을 고안하게 되었다. 현재 Security-Mode ONOS는 이러한 가상네트워크를 전혀 염두해두고 있지 않기 때문에, 같은 권한을 가진 애플리케이션은 논리적으로 분리되어야 할 다른 가상네트워크 환경에 영향을 줄 수 있는 심각한 문제점을 가지고 있다.

본 기술문서에서는, 최소권한 정책의 맹점인 가상네트워크에 대한 최소권한 전략을 통해 기존의 Security-Mode ONOS에서 극복하지 못한 데이터부의 최소권한 정책을 다룬다. Security-Mode ONOS for Virtual Network (SM ONOS-VN)의 주된 목적은 데이터부에 대한 적절한 권한 제어로, 논리적으로 분리된 다양한 네트워크에 대한 최소 권한 정책을 부여해, 사용자로하여금 최소권한으로 자신이 관리하는 네트워크에 대한 권한을 부여하는 것 이다. 이를 통하여 네트워크 관리자는 특정 가상네트워크에 한정하여, 다양한 애플리케이션을 동작할 수 있고, 애플리케이션에 대한 다양한 취약점 및 문제점이 다른 가상네트워크에 영향을 줄 수 없는 환경을 구축 할 수 있다.

2.2. SM ONOS-VN 요구사항

본 절에서는 Security-Mode ONOS의 확장 모듈로서, 데이터부의 최소권한 정책을 부여하는 가상네트워크를 위한 SM ONOS-VN가 풀려고하는 구체적인 문제의 범위와, 문제를 풀기 위한 구체적인 요구사항에 대해 정리한다.

SM ONOS-VM는 데이터부에 구현되어 있는 가상 네트워크에 대한 엑세스 컨트롤을 하는 것 이다. 가상네트워크는 하나의 물리네트워크 상에서, 논리적으로 네트워크를 여러개로 나누어, 각 관리자에 따라 논리적으로 구분된 네트워크를 부여 하는 것 이다. 본 절에서 다룰 논리 네트워크는 그림 6과 같다.

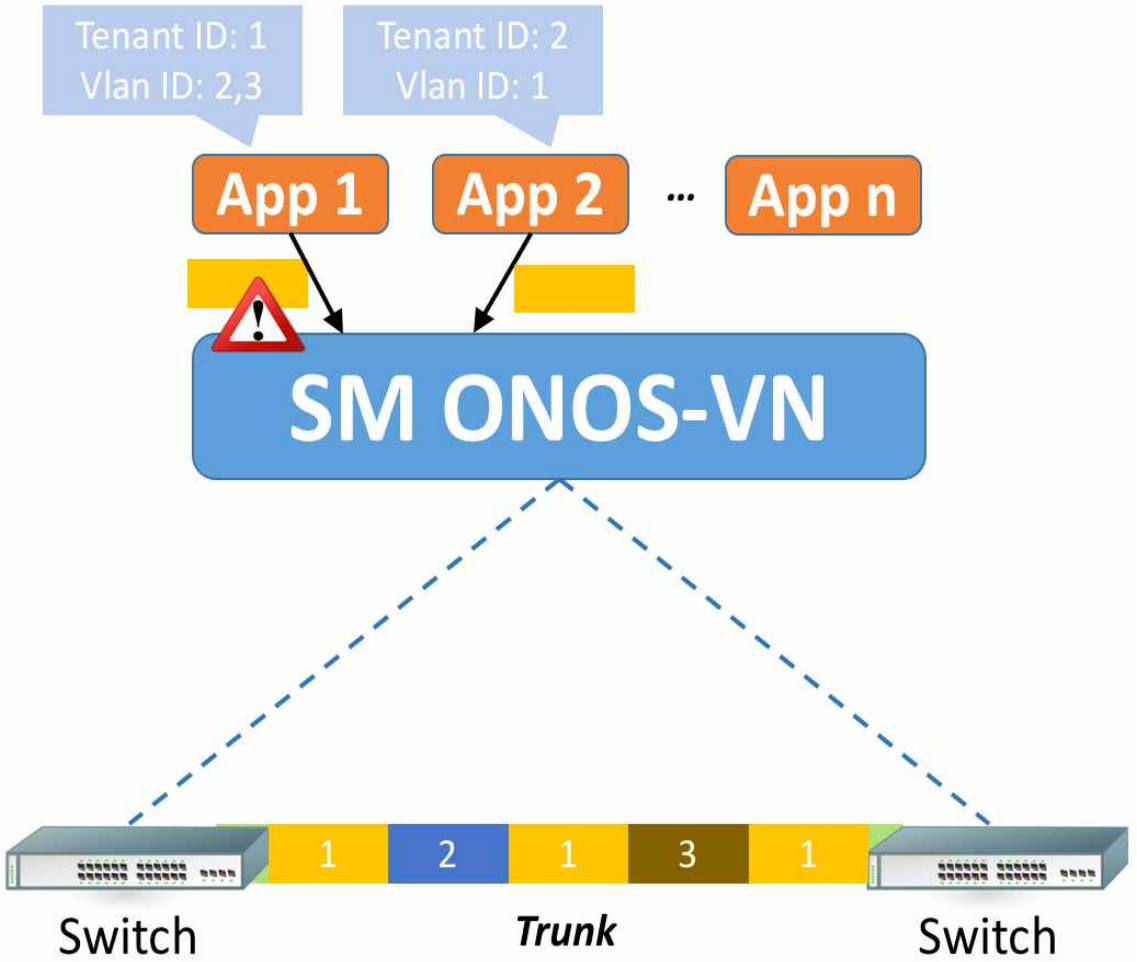

그림 7 SM ONOS-VN의 동작 시나리오

그림 6은 SM ONOS-VN의 동작 시나리오를 나타내고 있다. 각 스위치는 물리 망으로 연결이 되어 있고, 하나의 포트를 이용해 Trunk방식으로 가상네트워크를 만든 것을 확인 할 수 있다. 각각의 가상네트워크는 물리네트워크 망을 공유하고 논리적으로 분리된 가상네트워크 망을 이루게 된다. 각각의 애플리케이션은 서로다른 가

상네트워크에 접근을 시도하려고 한다. 이때 각각의 애플리케이션은 SM ONOS와 같이, 자신이 가지고 있는 애플리케이션의 아이디에 따라 각각의 가상네트워크에 접근 할 수 있는 권한을, 권한파일(Policy file)에 명시를 해야 한다. 해당 번호를 기반으로 SM ONOS-VN은 각 가상네트워크 별로 애플리케이션이 하는 행위를 제한하게 된다. 시나리오에서 보면 애플리케이션 1번은 VN(Virtual Network) 1번 프레임에 대한 권한이 없는데, VN 1에 접근을 하는 것을 볼 수 있다. 이 경우 SM ONOS-VN은 이를 감지하고, 사전에 접근을 통제하게 된다. 이와 반대로 애플리케이션 2번은 해당 VN에 대한 적절한 권한이 있기 때문에 VN 1에 대해 SM ONOS-VN은 접근을 허가 해 준다.

이와같이 SM ONOS-VN의 궁극적인 목적은 데이터부에서 동작하는 가상네트워크에 대하여 적절한 퍼미션을 부여하고, 해당 퍼미션을 기반으로 데이터부를 안전하게 보호하는 것 이다. 포로토타입에서는 호스트를 고려하지 않고, 네트워크 자체에 대한 (VN 자체에 대한) 최소권한 부여 정책을 목표로 한다.

2.3. SM ONOS-VN 권한 목록

SM ONOS-VN은 SM ONOS의 확장모듈로서 SM ONOS에서 사용되는 권한을 그대로 승계하게 된다. 표 1은 SM ONOS에 가상네트워크를 위한 권한이 추가된 전체 표를 볼 수 있다.

Permission Type	Permission Description
APP_READ	Permission to read various information about installed applications
APP_WRITE	Permission to register new application
APP_EVENT	Permission to receive application life cycle events
CONFIG_READ	Permission to read configuration properties
CONFIG_WRITE	Permission to write configuration properties
CODEC_READ	Permission to read codec information
CODEC_WRITE	Permission to add/remove entity class from codecs
CLOCK_WRITE	Permission to write clock properties
CLUSTER_READ	Permission to read cluster information
CLUSTER_EVENT	Permission receive cluster events
DEVICE_READ	Permission to read device information
DEVICE_EVENT	Permission receive device events
DRIVER_READ	Permission to get driver instances
DRIVER_WRITE	Permission to create a new driver handler
DEVICE_KEY_READ	Permission to read device key
EVENT_READ	Permission to read event properties
EVENT_WRITE	Permission to write event properties
FLOWRULE_READ	Permission to read flow rule information
FLOWRULE_WRITE	Permission to add/remove flow rules
FLOWRULE_EVENT	Permission receive flow rule events
GROUP_READ	Permission to read group information
GROUP_WRITE	Permission to modify groups
GROUP_EVENT	Permission to receive group events
HOST_READ	Permission to read host information

HOST_WRITE	Permission to modify host
HOST_EVENT	Permission receive host events
INTENT_READ	Permission to read intent information
INTENT_WRITE	Permission to add/remove intents
INTENT_EVENT	Permission receive intent events
LINK_READ	Permission to read link information
LINK_WRITE	Permission to modify link information
LINK_EVENT	Permission receive link events
MUTEX_WRITE	Permission to execute mutex task
PACKET_READ	Permission to read packet information
PACKET_WRITE	Permission to send/block packet
PACKET_EVENT	Permission to handle packet events
PARTITION_READ	Permission to read partition information
PARTITION_EVENT	Permission to handle partition events
PERSISTENCE_WRITE	Permission to create persistent builder
REGION_READ	Permission to read region information
RESOURCE_READ	Permission to read resource information
RESOURCE_WRITE	Permission to allocate/release resource
RESOURCE_EVENT	Permission to handle resource events
STATISTIC_READ	Permission to access flow statistic information
TOPOLOGY_READ	Permission to read path and topology information
TOPOLOGY_EVENT	Permission to handle topology events
TUNNEL_READ	Permission to read tunnel information
TUNNEL_WRITE	Permission to create tunnels
TUNNEL_EVENT	Permission to receive tunnel events
UI_READ	Permission to read UI information
UI_WRITE	Permission to create/remove UI service
VN_CREATE	***Permission to create a virtual network***
VN_REMOVE	***Permission to remove a virtual network***
VN_READ	**Permission to read a list of virtual networks**
VN_EVENT	**Permission to receive events from a certain virutal network**
VN_WRITE	**Permission to access a certain virtual network**

표 1 SM ONOS-VN의 권한이 추가된 전체 권한 목록.

표 1은 SM ONOS-VN의 권한이 추가된 전체 권한 목록으로서, SM ONOS-VN은 총 5개의 권한을 사용하게 된다. 이텔릭체로 표시된 권한은 관리자를 위한 권한으로서, 가상네트워크에 대한 수정 및 삭제를 행하는 권한이다. 해당 권한은 SM ONOS의 관리자(Admin permission)가 가지는 권한으로서, 가상네트워크를 직접 제어하는 권한을 의미한다. 나머지 세가지의 권한은 사용자(User permission)이 가지는 권한으로서, 특정 가상네트워크에 대한 접근권한이다. VN_READ는 가상네트워크에 대한 정보를 열람할 수 있는 지에 대한 유무를 표현 한 것이다. 해당 권한은 데이터부에 직접적인 영향은 줄 수 없지만, 가상네트워크를 구성하는 다양한 정보를 열람 할 수 있기 때문에 해당 권한의 유무로, 가상네트워크 정보의 유출을 방지할 수 있다. VN_EVENT는 특정 가상네트워크에서 발생되는 다양한 네트워크 정보들 (Packet_IN, Flow statistics, ...)에 대한 제어메시지를 받을 수 있는지에 대한 여부이다. 예를들어 특정 가상네트워크에 대한 권한이 있는 경우, SM ONOS-VN은 발생하는 이벤트가 어느 가상네트워크 상에서 발생되어 있는지를 확인 한 후, 적절

한 권한이 있는 애플리케이션에게 해당 정보를 넘겨주게 된다. VN_WRITE권한은 데이터부에 직접적으로 영향을 줄 수 있는 권한으로서, 특정 가상네트워크에 대하여 SM ONOS에서 정의한 데이터부에 영향을 줄 수 있는 권한을 사용 할 수 있는 권한이다. 예를 들어 특정네트워크에 대하여 해당 권한을 가진 경우에는, 애플리케이션이 대상 가상네트워크에 네트워크 룰도 내릴 수 있고, 물리네트워크를 관리할 때처럼 다양한 행위를 할 수 있다.

2.4. SM ONOS-VN 동작 시나리오

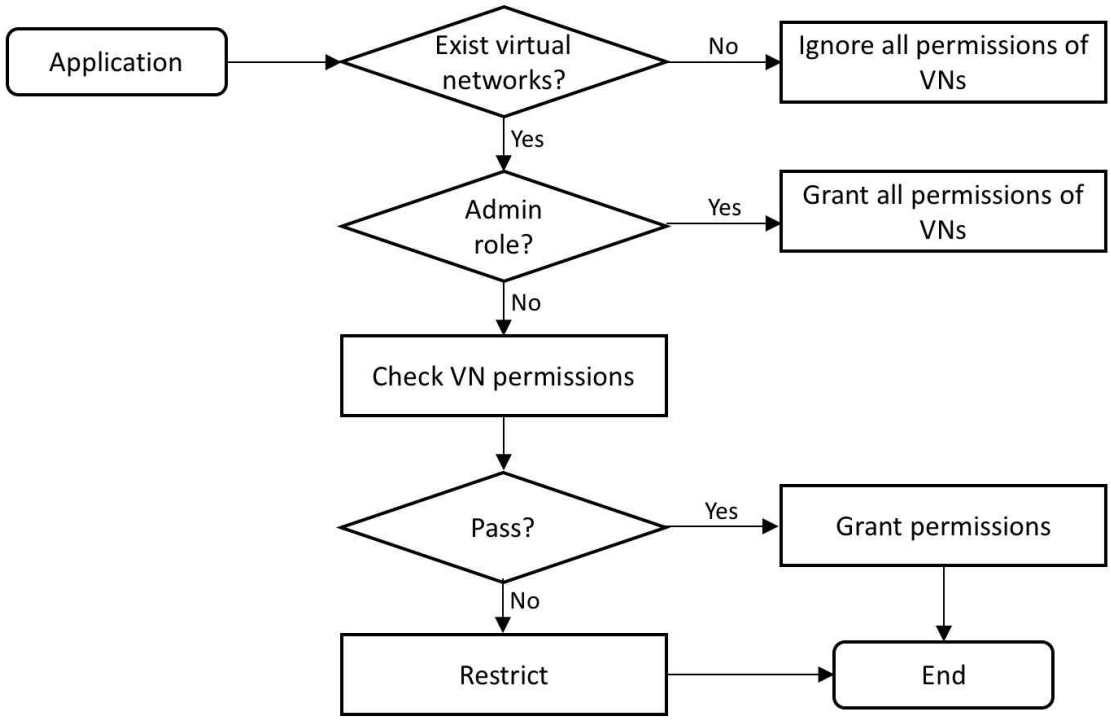

그림 8 SM ONOS-VN 퍼미션 모델 플로우 차트

그림 7은 SM ONOS-VN의 퍼미션 모델 플로우 차트를 도식화 한 그림이다. 특정 이벤트 (가상네트워크 접근, 제어, 등등)가 발생 한 경우 먼저 SM ONOS-VN은 현재 가상 네트워크가 있는지 없는지를 먼저 판단 한 후, 없는 경우에는 가상네트워크에 대한 모든 권한을 무시하게 된다. 그 다음 관리자역할인지 확인을 한 후 만약 관리자 권한을 가진다면 모든 권한을 부여하게 된다. 관리자 권한이 없는 경우에는 해당 애플리케이션은 최소 권한 전략 (Fine-grained access control mechanism)을 취하고 있는 것이므로, 권한을 상세히 확인하는 프로세스를 진행하게 된다. SM ONOS-VN의 권한파일에 정의 되어 있는 다양한 권한을 확인 한 뒤, 만약 해당 권한을 애플리케이션이 가지고 있으면, 권한을 주고 아니면 권한을 부여하지 않는다.

기본적인 SM ONOS-VN의 퍼미션 플로우 차트는 기존 SM ONOS 퍼미션모델과 비슷하다. 차이점은 대상이 제어부에 대한 최소 권한 전략에서 데이터부를 위한 최

소권한 전략으로 바뀐게 가장 큰 차이점이다.

2.5. SM ONOS-VN 권한 파일 형식

```
<security>
    <role>USER</role>

    <VNs>
        <VN>
            <VNID>5</VNID>
            <VN-perm>VN_READ</VN-perm>
        </VN>
        <VN>
            <VNID>2</VNID>
            <VN-perm>VN_READ</VN-perm>
            <VN-perm>VN_EVENT</VN-perm>
            <VN-perm>VN_WRITE</VN-perm>
        </VN>
    </VNs>

    <permissions>
        <app-perm>VN_CREATE</app-perm>
        <app-perm>VN_REMOVE</app-perm>
        <app-perm>DEVICE_READ</app-perm>
        <app-perm>PACKET_WRITE</app-perm>
        <app-perm>HOST_READ</app-perm>
        <java-perm>
            <classname>org.osgi.framework.AdminPermission</classname>
            <name>*</name>
            <actions>metadata></actions>
        </java-perm>
        <java-perm>
            <classname>org.lang.RuntimePermission</classname>
            <name>modifyThread</name>
        </java-perm>
    </permissions>
</security>
```

Permissions per VN

SM ONOS permissions

그림 9 SM ONOS-VN이 추가된 권한 파일 형식

그림 8은 SM ONOS-VN의 권한이 추가된 권한 형석을 나타내고 있다. 기본적으로 SM ONOS-VN의 권한파일은 SM ONOS에서 제공하는 권한파일을 그대로 계승하고, 가상네트워크에 대한 권한이 추가된 점을 알 수 있다. SM ONOS에서 사용되는 role도 동일하게 적용이 된다. User 역할 일 경우, 가상네트워크에 대한 관리 권한 및 사용 권한은 최소 권한 부여 전략 (Fine-grained permission grant)을 취할 수 있고, 가상네트워크 관리 자체에 대한 권한과, 가상 네트워크 별 권한을 부여 할 수 있다. 만약 Admin인 경우에는 전체 가상네트워크에 대한 모든 권한을 가질 수 있다. 예를들어 Admin권한을 가진 애플리케이션은 모든 가상네트워크에 대한

VN_READ, VN_EVENT, VN_WRITE권한을 가지게 되고, 관리자로서 모든 가상네트워크에 대한 삭제 및 생성을 할 수 있다.

VN테그는 새롭게 추가된 권한으로서, 특정 가상네트워크에 대한 최소 권한을 부여 할 수 있는 태그이다. 각 가상 네트워크에 따라 서로다른 권한을 부여 할 수 있다. 특히 VN_WRITE권한은 SM ONOS에서 정의 한 데이터부에 영향을 줄 수 있는 권한들을 app-perm테그에 정의되어 있는 권한에 한해 수행 할 수 있도록 한다. 마찬가지로 VN_READ는 가상네트워크에 대한 전반적인 정보와 해당 가상네트워크에 대한 정보를 열람 할 수 있다. 이 경우에는 데이터부에 직접적인 영향을 주지 않으므로 VN_WRITE보다 덜 중요한 권한이 된다. VN_EVENT같은 경우에는, 해당 가상네트워크에서 오는 이벤트정보를 받을 수 있는 권한이다. app-perm테그에는 가상네트워크 관리에 대한 권한을 부여 할 수 있다. 가상네트워크 자체에 대한 권한은 가상네트워크를 생성하거나 삭제할 수 있는 권한으로서 Admin권한에 속하지만, 최소권한 정책으로 좀더 상세히 정의 할 수 있다.

이와 같이 SM ONOS-VN은 SM ONOS 권한 파일을 그대로 승계하여, 가상네트워크에 대한 부분을 추가하였고, 하위호환성을 고려하여 디자인되었다.

3. 결론

SM ONOS의 등장으로 네트워크 운영체제에 대한 최소권한 모델의 효용성이 입증되었고, 네트워크 운영체제에 대한 고 가용성 확보 및 다양한 보안문제를 해결 할 수 있다. 하지만 SM ONOS의 단점인 데이터부에 대한 적절한 권한 부여 방식의 부재는 네트워크 관리자로 하여금 제한된 형태의 권한 부여시스템을 가질 수 밖에 없었다. 본 연구에서는 이러한 SM ONOS의 한계를 극복하기 위한 초석으로, 데이터부에 대한 적절한 권한 부여시스템 전략을 통해 데이터부 또한 최소 권한 메커니즘으로 구현 할 수 있다는 것을 보였고, 본 연구는 데이터부에 대한 권한 부여의 시작으로 자리 메김 할 것을 고대하고 있다.

본 기술문서는 가상네트워크에 대한 SM ONOS의 확장에 대한 기술문서로서, 데이터부에 대한 적절한 권한 부여 시스템의 효용성을 보여주고, 데이터부에 대한 권한 시스템의 필요성에 대해 기술하였다. 본 기술문서를 기반으로 전 방위적으로 통합된 보안 프레임워크를 설계하여, 종래에 문제가 되고 있는 SDN환경에서의 보안문제를 해결 할 계획이다.

References

[1] Onlab, “Security-Mode ONOS”, https://wiki.onosproject.org/display/ONOS/introduction

[2] Sherwood, Rob, et al. "Flowvisor: A network virtualization layer." OpenFlow Switch Consortium, Tech. Rep (2009): 1-13.

[3] Al-Shabibi, Ali, et al. "OpenVirteX: Make your virtual SDNs programmable."Proceedings of the third workshop on Hot topics in software defined networking. ACM, 2014.

[4] Gude, Natasha, et al. "NOX: towards an operating system for networks."ACM SIGCOMM Computer Communication Review 38.3 (2008): 105-110.

[5] Erickson, David. "The beacon openflow controller." Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking. ACM, 2013.

[6] POX controller, “POX Wiki”, https://openflow.stanford.edu/display/ONL/POX+Wiki

[7] Floodlight Project, “Floodlight Is an Open SDN Controller”, http://www.projectfloodlight.org/floodlight/

K-ONE 기술 문서

- 이 문서의 기술적인 내용은 프로젝트의 진행과 함께 별도의 예고 없이 변경될 수 있습니다.

- 본 문서와 관련된 문의 사항은 아래의 정보를 참조하시길 바랍니다.
  (Homepage: http://opennetworking.kr/projects/k-one-collaboration-project/wiki, E-mail: k1@opennetworking.kr)

작성기관: K-ONE Consortium
작성년월: 2016/04