오픈소스를 활용한 네트워크 보안 함수를 위한 프레임워크

김진용, 현대영, 홍동진, 정재훈
성균관대학교

요 약

본고에서는 국제 인터넷 표준화 기구(Internet Engineering Task Force, IETF) 해커톤(Hackathon)에서 오픈소스를 활용하여 구현한 네트워크 보안 함수 인터페이스(Interface to Network Security Functions, I2NSF) 프레임워크를 소개한다. 네트워크 공격이 복잡해지고 다양해짐에 따라 네트워크 통신의 무결성, 기밀성, 이용가능성을 보장하기 위하여 많은 보안 솔루션 제조사들은 다양한 네트워크 보안 함수(Network Security Functions, NSFs)들을 개발하고 있다. 이와 같이 다양한 보안 솔루션 제조사에서 개발된 네트워크 보안 함수들은 서로 다른 인터페이스로 이루어져있어 보안 서비스 관리자들은 다양한 네트워크 보안 함수들의 서로 다른 인터페이스로 인해 호환성의 문제를 겪고 있다. IETF의 I2NSF 워킹그룹(Working Group, WG)에서는 이러한 문제를 해결하고자 표준 인터페이스를 제정하고 있으며, 이를 활용하여 보안 서비스 관리자가 속한 환경에서 유연하게 관리할 수 있도록 네트워크 보안 함수를 위한 프레임워크를 설계하고 있다. 본 논문은 IETF I2NSF WG에서 제안하고 있는 프레임워크 기반으로 데이터 드리븐 보안 정책 관리(Data-driven Security Policy Management) 시스템을 Mininet, Confd, Apache2, MySQL, OpenDaylight, XSLT, Suricata 등의 오픈소스를 활용하여 구현하고, 방화벽과 웹 필터 보안 서비스 유스케이스를 보여준다.

Ⅰ. 서 론

네트워크 보안 함수(Network Security Function, NSF)는 네트워크 통신의 기밀성, 무결성, 이용가능성을 보장하기 위하여 원하지 않거나 혹은 의심이 가는 네트워크 활동을 감지 후 트래픽을 차단하거나 피해 정도를 약화하기 위하여 사용되는 기능을 말한다. 이러한 네트워크 보안 함수들은 보안 정책 및 규칙을 기반으로 동작하며 보안 솔루션 관리자들은 알맞은 보안 정책 및 규칙을 생성하여 네트워크 보안 함수에 적용 한다. 이와 같이 네트워크 보안 함수들은 적용된 보안 정책 및 규칙을 기반으로 악의적인 네트워크 트래픽들을 감지, 차단, 완화하는 보안 서비스들을 사용자에게 제공한다[1].

최근에 네트워크 공격이 복잡해지고 다양해짐에 따라 이러한 다양한 공격에 대비하기 위하여 네트워크 보안 함수들도 다양한 기능들이 요구되고 있으며 많은 보안 솔루션 제조사들에 의하여 개발되고 있다. 이와 같이 복잡해지고 다양해진 공격을 차단하기 위하여 보안 솔루션 소비자들은(회사 네트워크 관리자 혹은 서비스 제공자) 일반적으로 하나의 보안 솔루션을 이용하기보다는 다양한 제조사에서 개발된 하나 이상의 보안 솔루션들을 혼합하여 네트워크 보안 서비스들을 제공한다. 즉 서비스 제공자는 다양한 제조사에 의해 개발되거나, 오픈소스로 개발된 네트워크 보안 함수들의 혼합으로 자신의 고객들에게 네트워크 보안 서비스를 제공한다. 이와 같이 서비스 제공자는 다양한 제조사에 의해 개발된 네트워크 보안 함수들을 혼합하여 사용하므로 네트워크 보안 함수들을 감시 혹은 제어에 관한 표준 인터페이스 없이는 네트워크 보안 서비스들을 통합하여 한번에 관리하는 것은 사실상 불가능하다[1].

이러한 문제를 해결하고자 국제 인터넷 표준화 기구(Internet Engineering Task Force, IETF) 네트워크 보안 함수 인터페이스(Interface to Network Security Functions, I2NSF) 워킹그룹(Working Group, WG)[1]에서는 네트워크 보안 서비스들을 효율적으로 관리하기 위하여 네트워크 보안 함수들을 감시 혹은 제어에 관한 소프트웨어 인터페이스[2][3][4] 및 표준 데이터 모델[5][6][7]을 제정하고 있으며 이와 관련된 네트워크 보안 함수에 관한 프레임워크[8]를 설계하고 있다. 네트워크 보안 함수를 위한 프레임워크는 I2NSF 유저(I2NSF User), 보안 제어기(Security Controller), 벤더 관리 시스템(Vendor's Management System, VMS), 네트워크 보안 함수들로 이루어져 있으며 구성요소들간의 인터페이스로는 Consumer-Facing Interface[2], NSF-Facing Interface[3], Registration Interface[4]로 이루어져 있다. I2NSF WG에서는 이와 같은 표준 인터페이스를 위하여 YANG 데이터 모델[9]을 이용하고 있

으며 RESTCONF[10], NETCONF[11]와 같은 프로토콜들을 사용하여 통신하고 있다.

본 논문에서는 IETF I2NSF WG에서 제안하고 있는 I2NSF 프레임워크 기반으로 데이터 드리븐 보안 정책 관리(Data-driven Security Policy Management) 시스템을 살펴보고 이를 바탕으로 Mininet[12], ConfD[13], Apache2[14], MySQL[15], OpenDaylight[16], XSLT[17], Suricata[18] 등의 오픈소스를 활용하여 구현한 I2NSF 프레임워크를 소개한다. 소개된 I2NSF 프레임워크는 IETF 해커톤[19]에서 개발되었으며 Best University Award[20]를 수상하였다. 본 논문은 이전 학술대회 논문의 향상된 버전이다[21].

Ⅱ. 본 론

1. 네트워크 보안 함수를 위한 프레임워크

본 섹션에서는 보안 솔루션 관리자 혹은 서비스 제공자는 네트워크 보안 서비스들을 효율적으로 관리하기 위하여 IETF I2NSF WG 에서 설계한 네트워크 보안 함수를 위한 프레임워크를 소개한다.

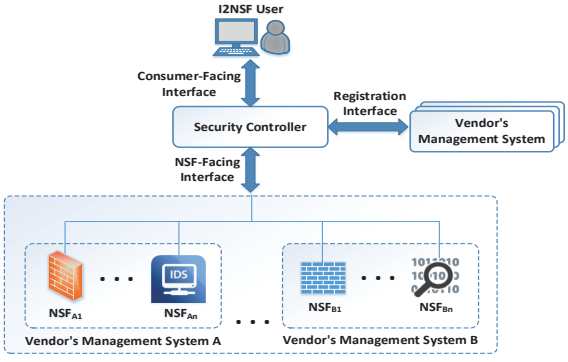

그림 1. 네트워크 보안 함수를 위한 프레임워크

〈그림 1〉은 IETF I2NSF WG에서 설계하고 있는 네트워크 보안 함수를 위한 프레임워크를 보여준다.

1.1. 구성요소

본 섹션에서는 I2NSF 프레임워크의 구성요소를 소개한다. 〈그림 1〉에서 보이는 것과 같이 I2NSF 프레임워크는 I2NSF 유저, 보안 제어기, 벤더 관리 시스템, 네트워크 보안 함수들로 4개의 구성요소로 이루어진다.

1.1.1. I2NSF 유저(I2NSF User)

I2NSF 유저는 회사 네트워크의 보안 관리자 혹은 서비스 제공자가 보안 정책들을 생성하거나 관리하기 위하여 사용된다. I2NSF 유저는 저수준(Low-Level)의 네트워크 보안 함수들의 성질을 고려하지 않고 고수준(High-Level) 보안 정책(Policy)을 생성하여 Consumer-Facing Interface를 통하여 보안 제어기에게 전달 한다.

1.1.2. 보안 제어기(Security Controller)

보안 제어기는 I2NSF 프레임워크에서 가장 중요한 구성요소이다. I2NSF 유저로부터 받은 고수준 보안 정책들을 저수준 보안 정책들로 번역하며 또한 벤더 관리 시스템에서 등록된 네트워크 보안 함수들의 기능들을 관리한다. 즉 I2NSF 유저로부터 받은 고수준 보안 정책을 등록된 네트워크 보안 함수들의 기능들에 따라 알맞은 저수준 보안 정책으로 번역하는 역할을 한다. 이와 같이 번역된 저수준 보안 정책들은 등록된 네트워크 보안 함수들이 제공하는 기능들에 따라 NSF-Facing Interface를 통하여 알맞은 네트워크 보안 함수들에게 전달한다.

1.1.3. 벤더 관리 시스템(Vendor's Management System)

벤더 관리 시스템은 네트워크 보안 함수를 개발 및 제공하는 보안 솔루션 벤더들에 의하여 관리된다. 다양한 벤더에서 개발된 네트워크 보안 함수들은 벤더 관리 시스템을 통하여 네트워크 보안 함수를 위한 프레임워크에 등록될 수 있다. 이와 같은 등록은 Registration Interface를 통하여 이루어진다.

1.1.4. 네트워크 보안 함수(NSFs)

방화벽(Firewall), 심층 패킷 분석(Deep Packet Inspection, DPI) DDoS 공격 약화(DDoS Attack Mitigation)와 같은 네트워크 보안 함수들은 보안 정책 규칙에 따라 원치 않은 네트워크 트래픽을 차단하거나 약화시킨다. I2NSF 프레임워크에서는 우선 패킷 헤더를 분석 할 수 있는 방화벽과 같은 기본적인 네트워크 보안 함수들을 활용하여 패킷들을 검사한다. 만약 추가적인 검사가 필요하다고 판단되면, 심층 패킷 분석, 웹 필터, DDoS 공격 약화와 같은 다른 네트워크 보안 함수들에게 패킷들을 전달하여 추가적인 검사를 수행하게 한다.

1.2. 인터페이스

본 섹션에서는 I2NSF 프레임워크의 인터페이스를 소개한다. 〈그림 1〉에서 보이는 것과 같이 I2NSF 프레임워크는 Consumer-Facing Interface, NSF-Facing Interface, Registration Interface로 3개의 인터페이스로 이루어진다.

1.2.1. Consumer-Facing Interface

Consumer-Facing Interface는 I2NSF 유저와 보안 제어기 사이의 인터페이스를 말한다. 본 인터페이스를 통해 보안 관리

자들은 네트워크 보안 함수들의 성질 및 기능들을 고려할 필요 없이 보안 관리자들에게 친숙한 고수준 보안 정책들을 생성할 수 있으며 생성된 고수준 보안 정책들을 보안 제어기에 전달할 수 있다. 이때 보안 제어기에서 전달 받은 고수준 보안 정책들을 저수준 보안 정책들로의 번역을 자동화하기 위하여 I2NSF WG에서는 Consumer-Facing Interface를 제정하고 있으며 표준 데이터 모델을 위하여 YANG 데이터 모델을 이용하고 있다. 또한 I2NSF 유저에서 Consumer-Facing Interface에 맞춰 생성된 고수준 보안 정책들은 RESTCONF 프로토콜을 통해 보안 제어기에게 전달된다.

1.2.2. NSF-Facing Interface

NSF-Facing Interface는 보안 제어기와 네트워크 보안 함수들 사이의 인터페이스를 말한다. 본 인터페이스를 통해 보안 제어기는 번역된 저수준 보안 정책들을 등록된 네트워크 보안 함수들이 제공하는 기능에 따라 알맞은 네트워크 보안 함수들에게 전달할 수 있다. 이때 다양한 벤더에서 개발된 각기 다른 네트워크 보안 함수들의 인터페이스로 인해 발생하는 관리적인 측면의 문제를 해결하고 네트워크 보안 함수들의 효율적인 관리를 위해 I2NSF WG에서는 NSF-Facing Interface를 제정하고 있으며 표준 데이터 모델을 위하여 YANG 데이터 모델을 이용하고 있다. 또한 보안 제어기에서 NSF-Facing Interface에 맞춰 번역된 저수준 보안 정책들을 NETCONF 프로토콜을 통해 네트워크 보안 함수들에게 전달된다.

1.2.3. Registration Interface

Registration Interface는 보안 제어기와 벤더 관리 시스템 사이의 인터페이스를 말한다. 본 인터페이스를 통해 보안 솔루션 벤더들은 자사에서 개발한 네트워크 보안 함수들을 보안 함수들이 제공할 수 있는 기능들과 함께 등록할 수 있다. 이때 다양한 벤더에서의 각기 다른 인터페이스로 인해 I2NSF WG에서는 Registration Interface를 제정하고 있으며 표준 데이터 모델을 위하여 YANG 데이터 모델을 이용하였다. 이와 같이 벤더 관리 시스템에서 Registration Interface에 맞춰 생성된 네트워크 보안 함수와 보안 함수들이 제공할 수 있는 기능들이 나열된 정책들은 NETCONF 프로토콜을 통해 보안 제어기에게 전달 된다.

1.3. I2NSF 프레임워크 보안 서비스 절차

본 섹션에서는 I2NSF 프레임워크에서의 보안 서비스 절차를 설명한다. <그림 2>는 I2NSF 프레임워크에서의 보안 서비스 절차를 보여준다. 절차는 아래 순서와 같다.

1) 다양한 벤더들은 자사에서 개발한 네트워크 보안 함수들을 I2NSF 프레임워크에 등록하기 위하여 그 보안 함수들이 제공할 수 있는 기능들에 관한 정책들을 생성하여 벤더 관리 시스템에 등록한다. 이때 생성된 네트워크 보안 함수들의 기능들에 관한 정책들은 Registration Interface를 통해 보안 제어기에 등록된다.

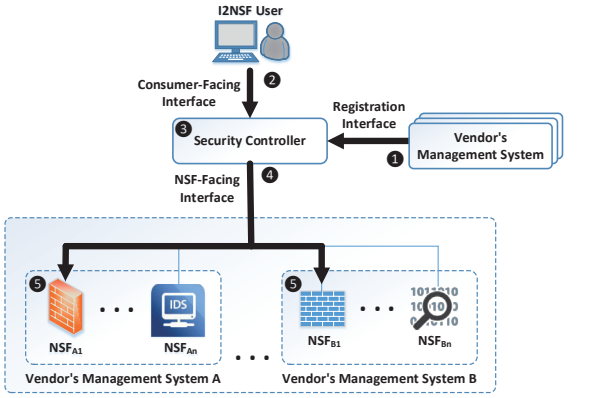

그림 2. I2NSF 프레임워크에서의 보안 서비스 절차

2) 보안 관리자는 네트워크 보안 함수들을 설정하기 위하여 유저 인터페이스인 I2NSF 유저를 통해 고수준 보안 정책들을 생성하여 Consumer-Facing Interface를 통해 보안 제어기에게 전달한다.

3) 보안 제어기는 전달받은 고수준 보안 정책들을 보안 제어기에 등록된 네트워크 보안 함수들의 기능들에 따라 알맞은 저수준 보안 정책들로 번역한다.

4) 보안 제어기는 번역된 저수준 보안 정책들을 NSF-Facing Interface를 통해 보안 제어기에 등록되어 있는 네트워크 보안 함수들 중에서 그 정책들을 수용할 수 있는 네트워크 보안 함수들에게 전달한다.

5) 네트워크 보안 함수들은 전달받은 저수준 보안 정책들을 자신의 시스템에 설정하여 요구된 보안 서비스(예, 방화벽, 심층 패킷 분석, DDoS 공격 약화)를 수행한다.

2. 네트워크 보안 함수를 위한 프레임워크

본 섹션에서는 IETF I2NSF WG에서 제안하고 있는 프레임워크 타당성을 검증하기 위해 구현한 내용을 보여준다. 실제 네트워크 환경 속에서 어떻게 적용이 될 수 있는 지를 회사 네트워크 시나리오에 적용하여 I2NSF 프레임워크의 타당성을 검증하였으며, 본 구현은 Mininet[12], ConfD[13], Apache2[14], MySQL[15], OpenDaylight[16], XSLT[17], Suricata[18] 등의 오픈소스들을 활용하여 구현하였다.

2.1. 시나리오 구성

우리는 실제 네트워크 환경 속에서 I2NSF 프레임워크가 어떻게 적용이 될 수 있는지를 보여주기 위하여 회사 내부 네트워크에서의 원치 않은 네트워크 활동들을 차단하는 시나리오를 적용하였다.

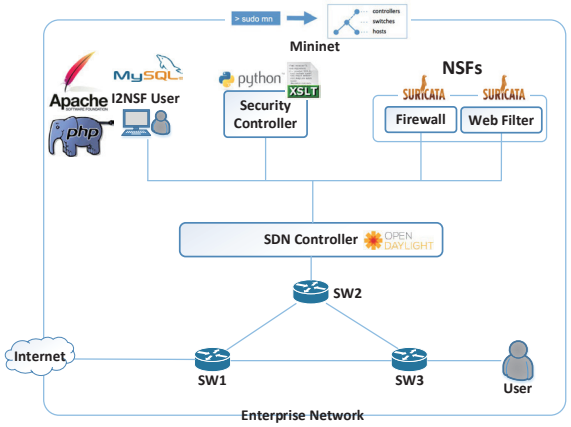

그림 3. 오픈소스를 활용하여 구현한 I2NSF 프레임워크

〈그림 3〉는 회사 네트워크에 적용한 I2NSF 프레임워크를 보여준다. 그림 3에서 보이는 것과 같이 이와 같은 I2NSF 프레임워크를 구축하기 위하여 오직 오픈소스만을 활용하여 구현하였다. 〈표 1〉은 I2NSF 프레임워크를 구축하기 위하여 사용된 오픈소스들을 보여준다.

표 1. I2NSF 시스템에 활용된 오픈소스

오픈소스	역할
Mininet[12]	가상 네트워크 환경 구축
Apache2[14]	웹 서버 구축
PHP[22]	서버 스크립팅 언어
MySQL[15]	데이터 베이스
Python[23]	고수준 프로그래밍 언어
XSLT[17]	문서 변환
Suricata[18]	IDS/IPS
Jetconf[24]	RESTCONF 프로토콜 통신
ConfD[13]	NETCONF 프로토콜 통신
YANG[9]	표준 데이터 모델
OpenDaylight[16]	SDN 컨트롤러, 패킷 플로우 제어

가상의 회사 네트워크를 구축하기 위하여 오픈소스인 Mininet을 활용하여 구축하였다. I2NSF 유저는 웹 기반 어플리케이션으로 구현되었으며 보안 관리자들은 이를 활용하여 회사에서의 원치 않은 네트워크 활동들에 대한 고수준 보안 정책들을 생성하여 차단할 수 있다. 이와 같이 웹 기반 어플리케이션으로 구현된 I2NSF 유저는 오픈소스인 Apache2, PHP, MySQL들을 활용하여 구현하였다. 고수준 보안 정책들을 저수준 보안 정책들로 번역하고 네트워크 보안 함수들 및 그 기능들을 관리하는 보안 제어기는 파이썬을 통해 구현하였다. 또한 보안 제어기에서 고수준 보안 정책들을 저수준 보안 정책들로 번역하기 위하여 XSLT 프로그램을 사용하여 번역하였다. 네트워크 보안 함수로써는 방화벽 보안 서비스와 웹 필터 보안 서비스를 제공하는 네트워크 보안 함수를 구현하였으며, 이는 IDS/IPS로 사용되는 오픈소스인 Suricata를 활용하여 구현하였다.

인터페이스로는 RESTCONF 프로토콜에 기반한 Consumer-Facing Interface는 파이썬의 오픈 API인 Jetconf API를 활용하여 구현하였으며, NETCONF 프로토콜에 기반한 NSF-Facing Interface와 Registration Interface는 Cisco에서 개발한 오픈소스인 ConfD를 활용하여 구현하였다. 또한 표준 인터페이스 데이터 모델을 위하여 IETF에서 개발한 YANG 데이터 모델을 활용하여 구현하였다. 마지막으로 Mininet에서의 패킷 흐름을 제어하기 위하여 오픈 소스인 OpenDaylight를 활용하여 상이한 위치에서 발생하는 패킷 흐름을 제어하였다.

I2NSF 프레임워크 소스 코드와 문서들은 프로그래머를 위한 소셜 코딩 공간인 깃허브(GitHub) https://github.com/kimjinyong/i2nsf-framework 에서 접근 가능하며 이와 관련된 비디오 데모는 유튜브(YouTube) https://www.youtube.com/watch?v=fRCnQX2aFa4 에서 접근 가능하다.

2.1.1. 방화벽 보안 서비스 시나리오

방화벽 보안 서비스 시나리오는 회사 내부 네트워크에서의 원치 않은 네트워크 서비스를 차단하기 위하여 패킷 헤더의 포트 번호를 검사하여 차단한다.

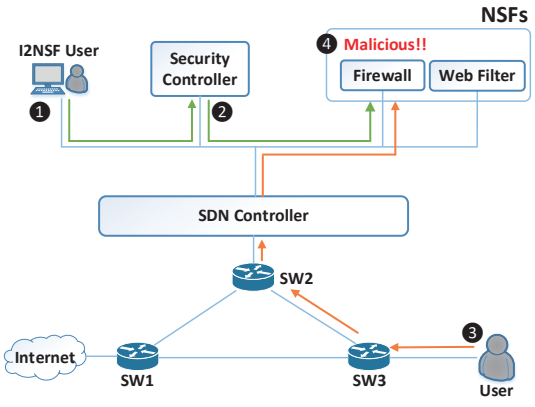

그림 4. 방화벽 보안 서비스 절차

〈그림 4〉는 I2NSF 프레임워크에서의 방화벽 보안 서비스 절

차를 보여준다. 절차는 아래 순서와 같다.

1) 보안 관리자는 원치 않은 포트를 사용하는 유저의 패킷을 차단하기 위하여 이와 관련된 고수준 보안 정책들을 생성하여 I2NSF 유저를 통해 보안 제어기에게 전달한다.
2) 고수준 보안 정책들을 받은 보안 제어기는 저수준 보안 정책들로 번역 후 보안 제어기에 등록된 방화벽 보안 서비스에게 저수준 보안 정책들을 전달하며, 전달받은 방화벽 보안 서비스는 해당 보안 정책들을 자신의 시스템에 설정 한다.
3) 만약 회사 내부에 있는 유저가 회사에서 원치 않은 포트를 사용하고자 한다면 해당 패킷은 SDN 컨트롤러를 통해 방화벽 보안 서비스 함수에게 전달된다.
4) 전달받은 허용되지 않는 포트 번호를 사용하는 패킷들은 방화벽 보안 서비스 함수에 설정되어 있는 보안 정책에 따라 차단 된다.

2.1.2. 웹 필터 보안 서비스 시나리오

웹 필터 보안 서비스 시나리오는 회사 내부 네트워크에서의 원치 않은 웹 사이트로의 접속을 차단하기 위하여 패킷의 데이터를 검사하여 차단한다.

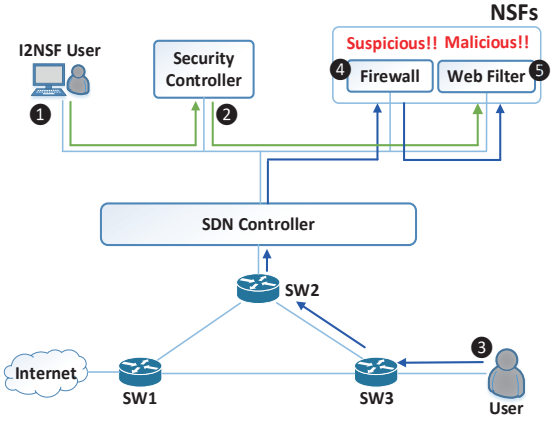

그림 5. 웹 필터 보안 서비스 절차

〈그림 5〉는 I2NSF 프레임워크에서의 웹 필터 보안 서비스 절차를 보여준다. 절차는 아래 순서와 같다.

1) 보안 관리자는 원치 않은 웹 사이트로 접속을 시도하는 유저의 패킷을 차단하기 위하여 이와 관련된 고수준 보안 정책들을 생성하여 I2NSF 유저를 통해 보안 제어기에게 전달한다.
2) 고수준 보안 정책들을 받은 보안 제어기는 저수준 보안 정책으로 번역 후 보안 제어기에 등록된 웹 필터 보안 서비스에게 저수준 보안 정책들을 전달하며 전달 받은 웹 필터 보안 서비스는 해당 보안 정책들을 자신의 시스템에 설정 한다.
3) 만약 회사 내부에 있는 유저가 회사에서 원치 않은 웹 사이트로 접속을 시도하고자 한다면 해당 패킷은 SDN 컨트롤러를 통해 방화벽 보안 서비스 함수에게 전달된다.
4) 방화벽 보안 서비스 함수는 해당 패킷의 헤더를 검사 후 해당 패킷이 웹 브라우저와 관련된 프로토콜이나 포트를 사용 시 웹 필터 보안 서비스 함수에게 해당 패킷들을 전달한다.
5) 전달된 원치 않은 웹 사이트로의 접속을 시도하는 패킷들은 웹 필터 보안 서비스 함수에 설정되어 있는 보안 정책에 따라 차단된다.

Ⅲ. 결 론

본고에서는 IETF I2NSF WG에서 제안한 네트워크 보안 서비스를 위한 프레임워크를 살펴보았고, 데이터 드리븐 보안 정책 관리(Data-driven Security Policy Management)의 타당성을 검증하기 위하여 실제 네트워크 환경 속에서 어떻게 적용이 될 수 있는 지를 회사 네트워크 시나리오에 적용하여 타당성을 검증하였다. 본 구현은 Mininet[12], ConfD[13], Apache2[14], MySQL[15], OpenDaylight[16], XSLT[17], Suricata[18] 등 오픈소스를 활용하여 구현되었다. 향후 연구로써 Mininet 환경이 아닌 오픈스택(OpenStack)[25]을 활용하여 ETSI의 NFV 환경 환경에서 I2NSF 기반 보안 서비스 시스템을 구현할 예정이다.

Acknowledgement

"본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터육성지원사업의 연구결과로 수행되었음" (IITP-2017-2017-0-01633).

참 고 문 헌

[1] IETF I2NSF Working Group, "Interface to Network Security Functions (I2NSF)", http://datatracker.ietf.org/wg/i2nsf

[2] R. Kumar, A. Lohiya, D. Qi, N. Bitar, S. Palislamovic, L. Xia, and J. Jeong, "Information

model for Client-Facing Interface to Security Controller", IETF Internet Draft, draft-kumar-i2nsf-client-facing-interface-im-04, October 2017.

[3] L. Xia, J. Strassner, C. Basile, and D. Lopez, "Information Model of NSFs Capabilities", IETF Internet Draft, draft-ietf-i2nsf-capability-00, September 2017.

[4] S. Hyun, T. Roh, S. Wi, J. Jeong, and J. Park, "Registration Interface Information Model", IETF Internet Draft, draft-hyun-i2nsf-registration-interface-im-03, October 2017.

[5] J. Jeong, E. Kim, T. Ahn, R. Kumar, and S. Hares, "I2NSF Consumer-Facing Interface YANG Data Model", IETF Internet Draft, draft-jeong-i2nsf-consumer-facing-interface-dm-04, October 2017.

[6] J. Kim, J. Jeong, J. Park, S. Hares, and Q. Lin, "I2NSF Network Security Functions-Facing Interface YANG Data Model", IETF Internet Draft, draft-kim-i2nsf-nsf-facing-interface-data-model-04, October 2017.

[7] S. Hares, J. Jeong, J. Kim, R. Moskowitz, and Q. Lin, "I2NSF Capability YANG Data Model", IETF Internet Draft, draft-hares-i2nsf-capability-data-model-05, October 2017.

[8] D. Lopez, E. Lopez, L. Dunbar, J. Strassner, and R. Kumar, "Framework for Interface to Network Security Functions", IETF Internet Draft, draft-ietf-i2nsf-framework-08, October 2017.

[9] M. Bjorklund, "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[10] A. Bierman, M. Bjorklund, and K. Watsen, "RESTCONF Protocol", RFC 8040, January 2017.

[11] R. Enns, Ed, M. Bjorklund, J. Schoenwaelder, and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

[12] Mininet, "An instant virtual network on your laptop", http://mininet.org

[13] ConfD, "ConfD is a lightweight configuration management tool", http://www.confd.io/

[14] Apache2, "HTTP Server Project", https://httpd.apache.org/

[15] MySQL, "An open-source relational database management system", https://www.mysql.com/

[16] OpenDaylight, "Collaborative open source project hosted by the Linux Foundation", https://www.opendaylight.org/

[17] XSLT, "A language for transforming XML documents into other XML documents", https://www.w3schools.com/xml/xsl_intro.asp

[18] Suricata, "A free and open source, mature, fast and robust network threat detection engine", https://suricata-ids.org/

[19] IETF Hackathon, "IETF Hackathon Program Contest", https://www.ietf.org/hackathon/

[20] Hackathon Award, "Best University Award", https://communities.cisco.com/community/developer/opensource/blog/2017/07/23/running-code-is-king-at-ietf-99-in-prague/

[21] J. Kim and J. Jeong, "Framework Development for Network Security Functions", SWCC 2017-Summer, August 2017.

[22] PHP, "PHP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language", http://php.net/

[23] Python, "Python is a widely used high-level programming language for general-purpose programming", https://www.python.org/

[24] Jetconf, "JetConf is an implementation of the RESTCONF protocol written in Python 3", https://pypi.python.org/pypi/jetconf/0.3.4

[25] OpenStack, "Open source software for creating private and public cloud", https://www.openstack.org/