TTA Standard

정보통신단체표준(국문표준)
TTAx.xx-xx.xxxx/R1

제정일: 2019 년 xx 월 xx 일
개정일: 200x 년 xx 월 xx 일

SDN 기반의 네트워크 보안 기능의 인터페이스(I2NSF) 프레임워크 - 제 6부: 네트워크 보안 기능 모니터링 인터페이스

Interface to Network Security Functions (I2NSF) Framework Using Software-Defined Networking - Part6: Network Security Function Monitoring Interface

TTA 한국정보통신기술협회
Telecommunications Technology Association

표준초안 검토 위원회	**사이버보안 프로젝트그룹(PG503)**				
표준안 심의 위원회	**정보보호 기술위원회(TC5)**				
	성명	**소 속**	**직위**	**위원회 및 직위**	**표준번호**
표준(과제) 제안	정재훈	성균관대학교	부교수	PG503 - 위원	
표준 초안 작성자	정재훈	성균관대학교	부교수	PG503 - 위원	
	정재홍	성균관대학교	석사과정		
	김진용	성균관대학교	석박사과정		
사무국 담당	박수정	TTA	책임	PG503 - 사무국	

본 표준 발간 이전에 접수된 지식재산권 확약서 정보는 본 표준의 '부록(지식재산권 확약서 정보)'에 명시하고 있으며, 이후 접수된 지식재산권 확약서는 TTA 웹사이트에서 확인할 수 있습니다.
본 표준과 관련하여 접수된 확약서 외의 지식재산권이 존재할 수 있습니다.

발행인 : 한국정보통신기술협회 회장
발행처 : 한국정보통신기술협회
13591, 경기도 성남시 분당구 분당로 47
Tel : 031-724-0114, Fax : 031-724-0109
발행일 : 20xx.xx

서 문

1 표준의 목적

네트워크 보안 기능(Network Security Functions, NSF) 모니터링은 적시에 포괄적인 방식으로 NSF 상태 파악 및 보안 공격 탐지를 함으로써 I2NSF 프레임워크에서의 NSF의 안정적인 운용에 중요한 역할을 한다. 본 표준은 이러한 I2NSF 프레임워크에서 NSF의 안정적인 운용을 위해 보안 제어기(서비스 공급자의 관리 시스템)와 NSF들 사이의 표준 모니터링 인터페이스를 정의한다.

2 주요 내용 요약

본 표준은 I2NSF 프레임워크에서의 NSF 모니터링을 위한 표준 모니터링 인터페이스의 정보 모델 및 데이터 모델을 정의한다. 이러한 NSF 모니터링 인터페이스를 통해 I2NSF 프레임워크는 NSF의 리소스 상태를 확인하고, NSF를 대상으로 하는 비정상적인 행동이나 잠재적인 공격을 감지하고, 각 보안 정책이 NSF들을 통해 잘 적용되고 있는지 확인할 수 있다. 모니터링은 카운터(Counter), 알림(Notification), 로그(Log)로 구성된다. 카운터는 시스템 카운터와 NSF 카운터로 구성된다. 알림은 경고(Alarm)와 이벤트(Event)로 구성된다. 로그는 시스템 로그와 NSF 로그로 구성된다. 이와 같이 NSF 모니터링은 NSF 자체 관리뿐만 아니라 보안 공격을 탐지하여 대응할 수 있는 자율 보안 서비스를 제공할 수 있다.

3 인용 표준과의 비교

3.1 인용 표준과의 관련성

- 해당 사항 없음

3.2 인용 표준과 본 표준의 비교표

- 해당 사항 없음

Preface

1 Purpose

Network Security Functions (NSF) Monitoring plays an important role for operating the framework of Interface to Network Security Functions (I2NSF) called I2NSF framework stably by monitoring the states of NSFs and detecting security attacks in a timely and comprehensive way. This standard defines a standard monitoring interface between the Security Controller (i.e., service provider's management system) and NSFs in order to support the stable operations of NSFs in the I2NSF framework.

2 Summary

This standard defines an information model and the corresponding data model of a standard monitoring interface for NSF monitoring in an I2NSF framework. Through this NSF monitoring interface, the I2NSF framework can check the resource states of NSFs, detect abnormal activities and potential attacks for NSFs, and check whether each security policy is applied through NSFs or not. The Monitoring consists of Counter, Notification, and Log. The Counter consists of System Counter and NSF Counter. The Notification consists of Alarm and Event. The Log consists of System Log and NSF Log. Therefore, the NSF monitoring can not only manage NSFs, but also provide autonomous security services by detecting security attacks and taking actions.

3 Relationship to Reference Standards

N/A
(*국문 서문의 3.1항 부분을 작성하되, 필요 시 3.2항의 비교표 작성)

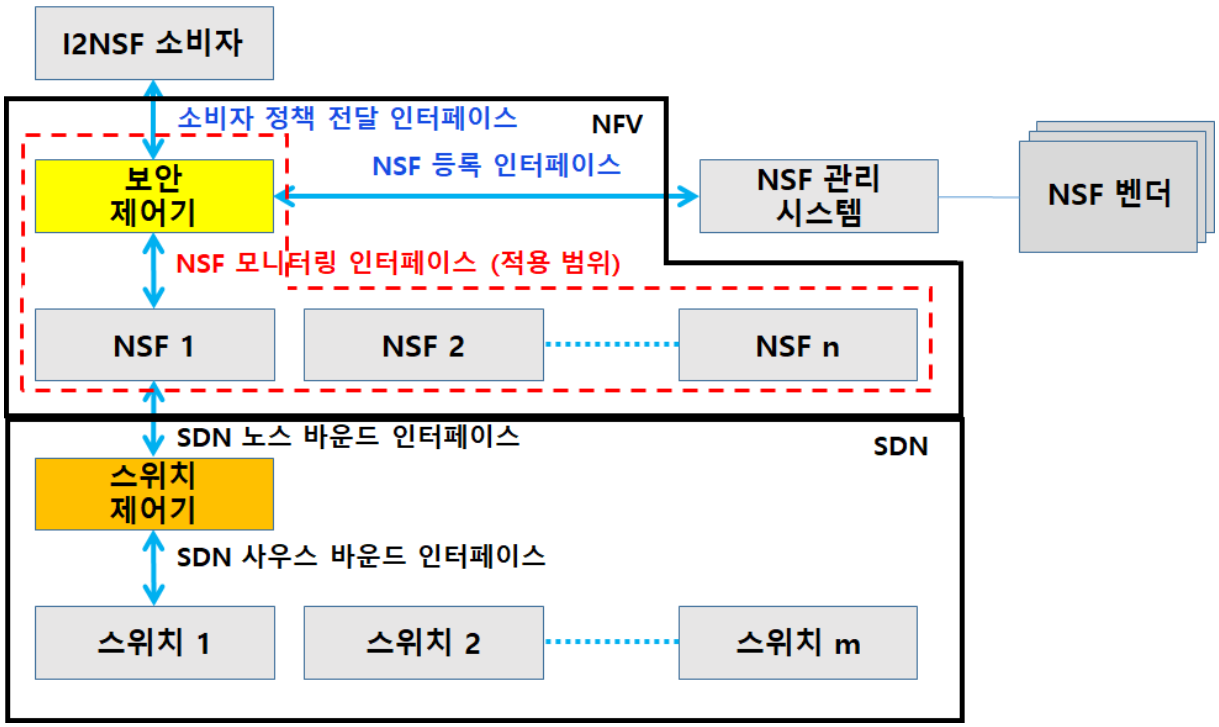
목 차

SDN 기반의 네트워크 보안 기능의 인터페이스(I2NSF) 프레임워크 - 제6부: 네트워크 보안 기능 모니터링 인터페이스

(Interface to Network Security Functions (I2NSF) Framework Using Software-Defined Networking – Part6: Network Security Function Monitoring Interface)

1 적용 범위

본 표준은 연계 표준 [II-3.1]에서 제안된 I2NSF 프레임워크[1]와 SDN이 결합된 보안 서비스 프레임워크의 NSF와 서비스 공급자의 관리 시스템 (즉 보안 제어기) 사이에서 NSF 모니터링을 위한 인터페이스[4]와 표준 모델을 제시하여, 다수의 보안 제조사에서 개발되어 프레임워크에 등록된 NSF들의 상태를 확인하고 NSF에 적용된 보안 정책의 상태 또한 확인 가능하게 한다. 본 표준에서 NSF 모니터링 인터페이스의 적용 범위는 (그림 1-1)과 같다.

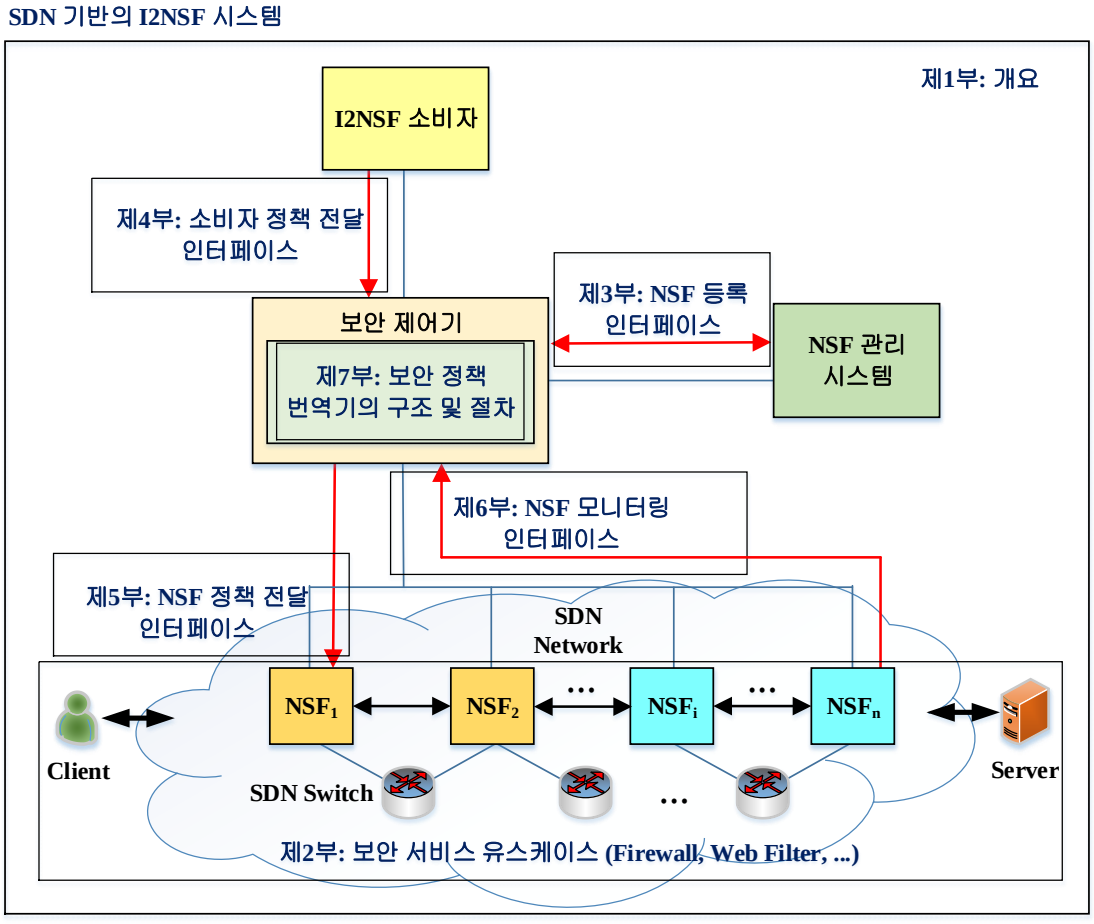


(그림 1-1) SDN 기반의 I2NSF 프레임워크에서의 적용 범위

<표 1-1>과 (그림 1-2)는 네트워크 보안 기능 모니터링 인터페이스의 연계(Family) 표준을 나타내는 표와 그림이다.

<표 1-1> SDN 기반의 I2NSF 시스템의 연계 표준 문서

순서	표준 번호	표준 제목
1	TTAK.KO-12.0314-Part1	제1부: 개요
2	TTAK.KO-12.0314-Part2	제2부: 보안 서비스 유스케이스
3	TTAK.KO-12.0314-Part3	제3부: NSF 등록 인터페이스
4	TTAK.KO-12.0314-Part4	제4부: 소비자 정책 전달 인터페이스
5	TTAK.KO-12.0314-Part5	제5부: NSF 정책 전달 인터페이스
6	TTAK.KO-12.0314-Part6 (예정)	제6부: NSF 모니터링 인터페이스
7	TTAK.KO-12.0314-Part7 (예정)	제7부: 보안 정책 번역기의 구조 및 절차



(그림 1-2) SDN 기반의 I2NSF 시스템의 연계 표준 적용 범위

2 인용 표준 (스타일 적용-대항목/소항목)

해당 없음.

3 용어 정의 (스타일 적용-대항목/소항목)

이 문서는 [3]에서 정의된 용어들을 사용한다.

4 약어

ACL	Access Control List
DDoS	Distributed Denial of Service
DM	Data Model
DoS	Denial of Service
ECA	Event-Condition-Action
GNSF	Generic Network Security Function
I2NSF	Interface to Network Security Function
IETF	Internet Engineering Task Force
IM	Information Model
IoT	Internet of Things
NFV	Network Functions Virtualization
NSF	Network Security Function
TCI	Tag Control Information

5 개요(Overview)

이 문서는 정책전달이 아닌 모니터링 부분에 중점을 두고 이를 위한 인터페이스와 NSF에 의해 생성되는 일련의 정보 요소 및 그 범위를 정의한다. NSF 모니터링 모델에 맞는 데이터가 NSF에 의해 생성되면 모니터링 인터페이스를 통해 보안 제어기로 전달됨으로써 모니터링을 수행한다.

6 NSF 모니터링 인터페이스

안정적인 보안 상태를 유지하려면 NSF 보안 정책을 구성할 뿐만 아니라 획득 가능하고 관찰 가능한 정보를 사용하여 NSF를 지속적으로 모니터링 해야 한다. 관리자가 NSF에 의해 생성된 모니터링 데이터를 확인하기 위해서는 모니터링 데이터를 보안 제어기에 전달하는 NSF 모니터링 인터페이스가 필요하다. 해당 인터페이스를 통해 전달되는 NSF의 이벤트 및 규격화된 활동 로그는 프레임워크 내부에서 발생하는 문제의 근본적인 원

인 분석에 도움이 될 수 있다.

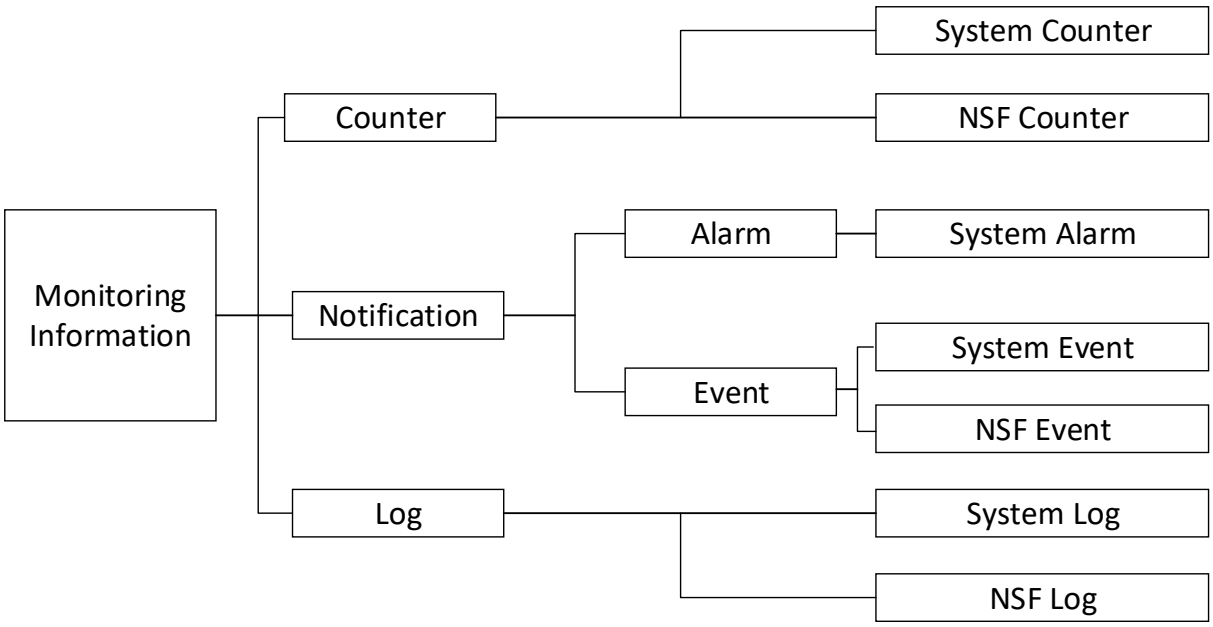
추가적으로, NSF에서 발생 가능한 연계 표준 [II-3.2]의 ECA(Event-Condition-Action) 정책 모델의 E(Event)의 트리거가 될 수 있는 목록을 구체화한다.

7 NSF 모니터링 정보 모델 디자인

NSF에서 생성된 일련의 모니터링 정보는 방출방법, 방출주기, 반복여부에 따라 전달된다. 이를 포함하여 기본적으로 NSF 모니터링 데이터에는 <표 7-1>과 같은 정보를 반드시 포함해야 한다.

<표 7-1> NSF 모니터링 데이터의 구성

구분	의미
방출방법	모니터링 정보가 방출되는 방법으로 값의 종류로는 subscription과 query가 있음.
방출주기	모니터링 정보가 방출되는 주기로 값의 종류로는 periodical과 on-change가 있음.
반복여부	모니터링 정보가 반복되어 생성되는지 여부로 값의 종류로는 no-dampening과 on-repetition가 있음.
메시지	모니터링 정보가 생성된 이유를 저장하기 위해 사용.
타임스탬프	모니터링 정보가 생성된 시간을 저장하기 위해 사용.
제조서명	NSF 공급업체의 이름을 저장하기 위해 사용.
NSF명	모니터링 정보를 생성하는 NSF의 이름 (또는 IP주소)을 저장하기 위해 사용.
모듈명	모니터링 정보를 출력하는 모듈 이름을 저장하기 위해 사용.
정도	모니터링 정보의 위험 수준을 저장하기 위해 사용. 총 8개의 레벨 (0에서 7까지)이 있으며 숫자가 작을수록 심각도가 높음.



(그림 7-1) NSF 모니터링 정보 모델

(그림 7-1)는 NSF 모니터링 정보의 분류에 따른 NSF 모니터링 정보 모델을 보여준다. I2NSF에 등록되어 동작하는 NSF는 여러 가지 정보를 기록할 수 있으며 생성된 모니터링 정보는 크게 카운터, 알림, 레코드로 나눌 수 있다.

7.1 카운터(Counter)

높은 빈도로 발생하는 정보 요소의 연속적인 값 변경에 대한 특정 표현으로 NSF의 동작 상태를 주기적으로 기록한다. 또한 NSF 정책 전달 인터페이스를 통해 설정된 보안 정책을 거쳐간 패킷의 양을 기록하여 보안 정책의 수정 또는 추가적인 보안 정책의 구성을 결정하기 위한 기준으로 사용된다.

7.1.1 시스템 카운터

NSF에서 생성된 모니터링 정보 중 시스템 카운터는 정보 전달을 위해 다음과 같은 속성을 가진다. (방출방법: subscription or query, 방출주기: periodical, 반복여부: none) 시스템 카운터의 대표적인 예시로는 네트워크 인터페이스 카운터가 있고 트래픽에 대한 각종 정보를 가지고 있다.

7.1.2 NSF 카운터

NSF에서 생성된 모니터링 정보 중 NSF 카운터는 정보 전달을 위해 다음과 같은 속성을 가진다. (방출방법: subscription or query, 방출주기: periodical, 반복여부: none)
NSF 카운터는 NSF 이벤트가 발생하여 관련 보안 정책이 적용되는 횟수를 센다.

7.2 알림(Notification)

알림은 I2NSF 에이전트에 의해 직접 또는 간접적으로 관찰될 수 있고 보안 정책의 이벤트 절에 필요한 입력으로 사용될 수 있는 시스템 또는 NSF 의 변경된 구성, 설정, 상태이다. 알림을 통해 연계 표준 [II-3.2]의 ECA 정책 모델의 E(Event)를 감지하고 매칭되는 E 가 존재한다면 C(Condition)를 트리거 시킨다.

7.2.1 경고(Alarm)

알림의 종류 중 경고는 사전에 관리자에 의해 설정된 임계 값을 넘어가는 경우 생성되고 주로 리소스의 상태에 대한 정보를 알린다.

7.2.1.1 시스템 경고

NSF 에서 생성된 모니터링 정보 중 시스템 경고는 정보 전달을 위해 다음과 같은 속성을 가진다. (방출방법: subscription, 방출주기: on-change, 반복여부: none)
종류로는 메모리 경고, CPU 경고, 디스크 경고, 하드웨어 경고, 인터페이스 경고가 있다.

7.2.2 이벤트(Event)

알림의 종류 중 이벤트는 연계 표준 [II-3.2]의 ECA 정책 모델 중 E(Event)로 본 절에서는 이벤트의 종류를 구체화한다.

7.2.2.1 시스템 이벤트

NSF에서 생성된 모니터링 정보 중 시스템 이벤트는 정보 전달을 위해 다음과 같은 속성을 가진다. (방출방법: subscription, 방출주기: on-change, 반복여부: on-repetition)
종류로는 접근 위반, 설정 변경이 있다.

7.2.2.2 NSF 이벤트

NSF에서 생성된 모니터링 정보 중 NSF 이벤트는 정보 전달을 위해 다음과 같은 속성을

가진다. (방출방법: subscription, 방출주기: on-change, 반복여부: none) 종류로는 분산된 서비스 거부 이벤트, 세션 테이블 이벤트, 바이러스 이벤트, 침입 이벤트, 봇넷 이벤트, 웹 공격 이벤트가 있다.

7.3 로그(Log)

NSF의 로그 파일 또는 데이터베이스에 저장되는 정보로서 즉각적인 주의를 필요로 하지 않지만 적절하게 분석된다면 사이버포렌식에 도움을 줄 수 있다. 로그 파일 형태의 로그는 일반적으로 구조가 정해져 있지 않지만 시스템의 변경된 사항과 관련하여 잠재적으로 보다 자세한 정보를 포함합니다. 반면에 데이터베이스는 보다 엄격한 구조나 데이터 모델을 사용하기 때문에 규격화된 정보를 제공할 수 있지만 해당 구조나 모델과 일치하지 않는 정보를 저장하지는 못한다. 따라서, 로그는 즉각적인 주의를 필요로 하지 않지만 I2NSF 프레임워크에 안정적인 동작을 위한 분석에 필요한 정보로 파일이나 데이터베이스에 저장된다.

7.3.1 시스템 로그

NSF에서 생성된 모니터링 정보 중 시스템 로그는 정보 전달을 위해 다음과 같은 속성을 가진다. (방출방법: subscription, 방출주기: on-change, 반복여부: on-repetition) 종류로는 접근 로그, 리소스 사용 로그, 유저 활동 로그가 있다.

7.3.2 NSF 로그

NSF에서 생성된 모니터링 정보 중 NSF 로그는 정보 전달을 위해 다음과 같은 속성을 가진다. (방출방법: subscription, 방출주기: on-change, 반복여부: on-repetition) 종류로는 분산된 서비스 거부 로그, 바이러스 로그, 침입 로그, 봇넷 로그, 심층 패킷 분석 로그, 취약성 분석 로그가 있다.

8 NSF 모니터링 데이터 모델 디자인

이 절에서는 NSF 모니터링 정보 모델에 기반한 NSF 모니터링 인터페이스를 정의하는 데이터 모델링을 위한 YANG [2] 데이터 모델을 보여준다.

8.1 YANG 데이터 모델 심볼

이 섹션에서는 데이터 모델 트리에서 사용된 심볼들을 정의한다. 데이터 모델 트리에서 사용된 심볼들의 정의는 다음과 같다.

- "[", "]"는 list 키의 괄호이다.
- "rw"는 읽고 쓸 수 있는 설정 데이터를 의미하며 "ro"는 오직 읽을 수만 있는 상태 데이터를 의미한다.
- "?"는 선택적인 노드를 의미한다.
- "*"는 다수의 노드를 의미하며 list나 leaf-list를 의미한다.
- ":"는 choice-case 구문을 의미한다.

8.2 NSF 모니터링 데이터 모델 트리

이 절에서는 NSF 모니터링 데이터 모델에 관한 모델 트리를 보여준다.

8.2.1 카운터 데이터 모델 트리

```
module: ietf-i2nsf-monitor
        +--rw counters
           +--rw system-interface
           |  +--rw acquisition-method?          identityref
           |  +--rw emission-type?               identityref
           |  +--rw dampening-type?              identityref
           |  +--rw interface-name?              string
           |  +--rw in-total-traffic-pkts?       uint32
           |  +--rw out-total-traffic-pkts?      uint32
           |  +--rw in-total-traffic-bytes?      uint32
           |  +--rw out-total-traffic-bytes?     uint32
           |  +--rw in-drop-traffic-pkts?        uint32
           |  +--rw out-drop-traffic-pkts?       uint32
           |  +--rw in-drop-traffic-bytes?       uint32
```

```
|  +--rw out-drop-traffic-bytes?        uint32
|  +--rw total-traffic?                 uint32
|  +--rw in-traffic-ave-rate?           uint32
|  +--rw in-traffic-peak-rate?          uint32
|  +--rw in-traffic-ave-speed?          uint32
|  +--rw in-traffic-peak-speed?         uint32
|  +--rw out-traffic-ave-rate?          uint32
|  +--rw out-traffic-peak-rate?         uint32
|  +--rw out-traffic-ave-speed?         uint32
|  +--rw out-traffic-peak-speed?        uint32
|  +--rw message?                       string
|  +--rw time-stamp?                    yang:date-and-time
|  +--rw vendor-name?                   string
|  +--rw nsf-name?                      string
|  +--rw module-name?                   string
|  +--rw severity?                      severity
+--rw nsf-firewall
|  +--rw acquisition-method?            identityref
|  +--rw emission-type?                 identityref
|  +--rw dampening-type?                identityref
|  +--rw src-ip?                        inet:ipv4-address
|  +--rw dst-ip?                        inet:ipv4-address
|  +--rw src-port?                      inet:port-number
|  +--rw dst-port?                      inet:port-number
|  +--rw src-zone?                      string
|  +--rw dst-zone?                      string
|  +--rw src-region?                    string
|  +--rw dst-region?                    string
|  +--rw policy-id?                     uint8
|  +--rw policy-name?                   string
|  +--rw src-user?                      string
|  +--rw protocol?                      identityref
```

```
|  +--rw app?                          string
|  +--rw total-traffic?                uint32
|  +--rw in-traffic-ave-rate?          uint32
|  +--rw in-traffic-peak-rate?         uint32
|  +--rw in-traffic-ave-speed?         uint32
|  +--rw in-traffic-peak-speed?        uint32
|  +--rw out-traffic-ave-rate?         uint32
|  +--rw out-traffic-peak-rate?        uint32
|  +--rw out-traffic-ave-speed?        uint32
|  +--rw out-traffic-peak-speed?       uint32
+--rw nsf-policy-hits
   +--rw acquisition-method?           identityref
   +--rw emission-type?                identityref
   +--rw dampening-type?               identityref
   +--rw src-ip?                       inet:ipv4-address
   +--rw dst-ip?                       inet:ipv4-address
   +--rw src-port?                     inet:port-number
   +--rw dst-port?                     inet:port-number
   +--rw src-zone?                     string
   +--rw dst-zone?                     string
   +--rw src-region?                   string
   +--rw dst-region?                   string
   +--rw policy-id?                    uint8
   +--rw policy-name?                  string
   +--rw src-user?                     string
   +--rw protocol?                     identityref
   +--rw app?                          string
   +--rw message?                      string
   +--rw time-stamp?                   yang:date-and-time
   +--rw vendor-name?                  string
   +--rw nsf-name?                     string
   +--rw module-name?                  string
```

```
        +--rw severity?                    severity
        +--rw hit-times?                   uint32
```

8.2.2 알림 데이터 모델 트리

```
module: ietf-i2nsf-nsf-monitor
+--rw counters
     | …
notifications:
       +---n system-detection-alarm
       |  +--ro alarm-category?            identityref
       |  +--ro acquisition-method?        identityref
       |  +--ro emission-type?             identityref
       |  +--ro dampening-type?            identityref
       |  +--ro usage?                     uint8
       |  +--ro threshold?                 uint8
       |  +--ro message?                   string
       |  +--ro time-stamp?                yang:date-and-time
       |  +--ro vendor-name?               string
       |  +--ro nsf-name?                  string
       |  +--ro module-name?               string
       |  +--ro severity?                  severity
       +---n system-detection-event
       |  +--ro event-category?            identityref
       |  +--ro acquisition-method?        identityref
       |  +--ro emission-type?             identityref
       |  +--ro dampening-type?            identityref
       |  +--ro user                       string
       |  +--ro group                      string
       |  +--ro login-ip-addr              inet:ipv4-address
       |  +--ro authentication?            identityref
       |  +--ro message?                   string
```

```
|   +--ro time-stamp?                    yang:date-and-time
|   +--ro vendor-name?                   string
|   +--ro nsf-name?                      string
|   +--ro module-name?                   string
|   +--ro severity?                      severity
+---n nsf-detection-flood
|   +--ro event-name?                    identityref
|   +--ro dst-ip?                        inet:ipv4-address
|   +--ro dst-port?                      inet:port-number
|   +--ro rule-id                        uint8
|   +--ro rule-name                      string
|   +--ro profile?                       string
|   +--ro raw-info?                      string
|   +--ro sub-attack-type?               identityref
|   +--ro start-time                     yang:date-and-time
|   +--ro end-time                       yang:date-and-time
|   +--ro attack-rate?                   uint32
|   +--ro attack-speed?                  uint32
|   +--ro message?                       string
|   +--ro time-stamp?                    yang:date-and-time
|   +--ro vendor-name?                   string
|   +--ro nsf-name?                      string
|   +--ro module-name?                   string
|   +--ro severity?                      severity
+---n nsf-detection-session-table
|   +--ro current-session?               uint8
|   +--ro maximum-session?               uint8
|   +--ro threshold?                     uint8
|   +--ro message?                       string
|   +--ro time-stamp?                    yang:date-and-time
|   +--ro vendor-name?                   string
|   +--ro nsf-name?                      string
```

```
|   +--ro module-name?                    string
|   +--ro severity?                       severity
+---n nsf-detection-virus
|   +--ro src-ip?                         inet:ipv4-address
|   +--ro dst-ip?                         inet:ipv4-address
|   +--ro src-port?                       inet:port-number
|   +--ro dst-port?                       inet:port-number
|   +--ro src-zone?                       string
|   +--ro dst-zone?                       string
|   +--ro rule-id                         uint8
|   +--ro rule-name                       string
|   +--ro profile?                        string
|   +--ro raw-info?                       string
|   +--ro virus?                          identityref
|   +--ro virus-name?                     string
|   +--ro file-type?                      string
|   +--ro file-name?                      string
|   +--ro message?                        string
|   +--ro time-stamp?                     yang:date-and-time
|   +--ro vendor-name?                    string
|   +--ro nsf-name?                       string
|   +--ro module-name?                    string
|   +--ro severity?                       severity
+---n nsf-detection-intrusion
|   +--ro src-ip?                         inet:ipv4-address
|   +--ro dst-ip?                         inet:ipv4-address
|   +--ro src-port?                       inet:port-number
|   +--ro dst-port?                       inet:port-number
|   +--ro src-zone?                       string
|   +--ro dst-zone?                       string
|   +--ro rule-id                         uint8
|   +--ro rule-name                       string
```

```
|   +--ro profile?                    string
|   +--ro raw-info?                   string
|   +--ro protocol?                   identityref
|   +--ro app?                        string
|   +--ro sub-attack-type?            identityref
|   +--ro message?                    string
|   +--ro time-stamp?                 yang:date-and-time
|   +--ro vendor-name?                string
|   +--ro nsf-name?                   string
|   +--ro module-name?                string
|   +--ro severity?                   severity
+---n nsf-detection-botnet
|   +--ro src-ip?                     inet:ipv4-address
|   +--ro dst-ip?                     inet:ipv4-address
|   +--ro src-port?                   inet:port-number
|   +--ro dst-port?                   inet:port-number
|   +--ro src-zone?                   string
|   +--ro dst-zone?                   string
|   +--ro rule-id                     uint8
|   +--ro rule-name                   string
|   +--ro profile?                    string
|   +--ro raw-info?                   string
|   +--ro attack-type?                identityref
|   +--ro protocol?                   identityref
|   +--ro botnet-name?                string
|   +--ro role?                       string
|   +--ro message?                    string
|   +--ro time-stamp?                 yang:date-and-time
|   +--ro vendor-name?                string
|   +--ro nsf-name?                   string
|   +--ro module-name?                string
|   +--ro severity?                   severity
```

```
+---n nsf-detection-web-attack
|   +--ro src-ip?                    inet:ipv4-address
|   +--ro dst-ip?                    inet:ipv4-address
|   +--ro src-port?                  inet:port-number
|   +--ro dst-port?                  inet:port-number
|   +--ro src-zone?                  string
|   +--ro dst-zone?                  string
|   +--ro rule-id                    uint8
|   +--ro rule-name                  string
|   +--ro profile?                   string
|   +--ro raw-info?                  string
|   +--ro sub-attack-type?           identityref
|   +--ro request-method?            identityref
|   +--ro req-uri?                   string
|   +--ro uri-category?              string
|   +--ro filtering-type*            identityref
|   +--ro message?                   string
|   +--ro time-stamp?                yang:date-and-time
|   +--ro vendor-name?               string
|   +--ro nsf-name?                  string
|   +--ro module-name?               string
|   +--ro severity?                  severity
+---n system-access-log
|   +--ro login-ip                   inet:ipv4-address
|   +--ro administrator?             string
|   +--ro login-mode?                login-mode
|   +--ro operation-type?            operation-type
|   +--ro result?                    string
|   +--ro content?                   string
|   +--ro acquisition-method?        identityref
|   +--ro emission-type?             identityref
|   +--ro dampening-type?            identityref
```

```
+---n system-res-util-log
|  +--ro system-status?                  string
|  +--ro cpu-usage?                      uint8
|  +--ro memory-usage?                   uint8
|  +--ro disk-usage?                     uint8
|  +--ro disk-left?                      uint8
|  +--ro session-num?                    uint8
|  +--ro process-num?                    uint8
|  +--ro in-traffic-rate?                uint32
|  +--ro out-traffic-rate?               uint32
|  +--ro in-traffic-speed?               uint32
|  +--ro out-traffic-speed?              uint32
|  +--ro acquisition-method?             identityref
|  +--ro emission-type?                  identityref
|  +--ro dampening-type?                 identityref
+---n system-user-activity-log
|  +--ro acquisition-method?             identityref
|  +--ro emission-type?                  identityref
|  +--ro dampening-type?                 identityref
|  +--ro user                            string
|  +--ro group                           string
|  +--ro login-ip-addr                   inet:ipv4-address
|  +--ro authentication?                 identityref
|  +--ro access?                         identityref
|  +--ro online-duration?                string
|  +--ro logout-duration?                string
|  +--ro additional-info?                string
+---n nsf-log-ddos
|  +--ro attack-type?                    identityref
|  +--ro attack-ave-rate?                uint32
|  +--ro attack-ave-speed?               uint32
|  +--ro attack-pkt-num?                 uint32
```

```
|   +--ro attack-src-ip?              inet:ipv4-address
|   +--ro action?                     log-action
|   +--ro acquisition-method?         identityref
|   +--ro emission-type?              identityref
|   +--ro dampening-type?             identityref
|   +--ro message?                    string
|   +--ro time-stamp?                 yang:date-and-time
|   +--ro vendor-name?                string
|   +--ro nsf-name?                   string
|   +--ro module-name?                string
|   +--ro severity?                   severity
+---n nsf-log-virus
|   +--ro attack-type?                identityref
|   +--ro action?                     log-action
|   +--ro os?                         string
|   +--ro time                        yang:date-and-time
|   +--ro acquisition-method?         identityref
|   +--ro emission-type?              Identityref
|   +--ro dampening-type?             identityref
|   +--ro message?                    string
|   +--ro time-stamp?                 yang:date-and-time
|   +--ro vendor-name?                string
|   +--ro nsf-name?                   string
|   +--ro module-name?                string
|   +--ro severity?                   severity
+---n nsf-log-intrusion
|   +--ro attack-type?                identityref
|   +--ro action?                     log-action
|   +--ro time                        yang:date-and-time
|   +--ro attack-rate?                uint32
|   +--ro attack-speed?               uint32
|   +--ro acquisition-method?         identityref
```

```
|   +--ro emission-type?                identityref
|   +--ro dampening-type?               identityref
|   +--ro message?                      string
|   +--ro time-stamp?                   yang:date-and-time
|   +--ro vendor-name?                  string
|   +--ro nsf-name?                     string
|   +--ro module-name?                  string
|   +--ro severity?                     severity
+---n nsf-log-botnet
|   +--ro attack-type?                  identityref
|   +--ro action?                       log-action
|   +--ro botnet-pkt-num?               uint8
|   +--ro os?                           string
|   +--ro acquisition-method?           identityref
|   +--ro emission-type?                identityref
|   +--ro dampening-type?               identityref
|   +--ro message?                      string
|   +--ro time-stamp?                   yang:date-and-time
|   +--ro vendor-name?                  string
|   +--ro nsf-name?                     string
|   +--ro module-name?                  string
|   +--ro severity?                     severity
+---n nsf-log-dpi
|   +--ro attack-type?                  dpi-type
|   +--ro acquisition-method?           identityref
|   +--ro emission-type?                identityref
|   +--ro dampening-type?               identityref
|   +--ro src-ip?                       inet:ipv4-address
|   +--ro dst-ip?                       inet:ipv4-address
|   +--ro src-port?                     inet:port-number
|   +--ro dst-port?                     inet:port-number
|   +--ro src-zone?                     string
```

```
|   +--ro dst-zone?                  string
|   +--ro src-region?                string
|   +--ro dst-region?                string
|   +--ro policy-id?                 uint8
|   +--ro policy-name?               string
|   +--ro src-user?                  string
|   +--ro protocol?                  identityref
|   +--ro app?                       string
|   +--ro message?                   string
|   +--ro time-stamp?                yang:date-and-time
|   +--ro vendor-name?               string
|   +--ro nsf-name?                  string
|   +--ro module-name?               string
|   +--ro severity?                  severity
+---n nsf-log-vuln-scan
|   +--ro vulnerability-id?          uint8
|   +--ro victim-ip?                 inet:ipv4-address
|   +--ro protocol?                  identityref
|   +--ro port-num?                  inet:port-number
|   +--ro level?                     severity
|   +--ro os?                        string
|   +--ro vulnerability-info?        string
|   +--ro fix-suggestion?            string
|   +--ro service?                   string
|   +--ro acquisition-method?        identityref
|   +--ro emission-type?             identityref
|   +--ro dampening-type?            identityref
|   +--ro message?                   string
|   +--ro time-stamp?                yang:date-and-time
|   +--ro vendor-name?               string
|   +--ro nsf-name?                  string
|   +--ro module-name?               string
```

```
|   +--ro severity?                    severity
+---n nsf-log-web-attack
    +--ro attack-type?                 identityref
    +--ro rsp-code?                    string
    +--ro req-clientapp?               string
    +--ro req-cookies?                 string
    +--ro req-host?                    string
    +--ro raw-info?                    string
    +--ro acquisition-method?          identityref
    +--ro emission-type?               identityref
    +--ro dampening-type?              identityref
    +--ro message?                     string
    +--ro time-stamp?                  yang:date-and-time
    +--ro vendor-name?                 string
    +--ro nsf-name?                    string
    +--ro module-name?                 string
    +--ro severity?                    Severity
```

부 록 Ⅰ-1

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

YANG 데이터 모듈 (YANG Data Modules)

이 부록에서는 NSF 모니터링 정보 모델 YANG 모듈을 정의한다.

```
<CODE BEGINS> file "ietf-i2nsf-monitor@2019-03-11.yang"
    module ietf-i2nsf-monitor {
        yang-version 1.1;
        namespace
          "urn:ietf:params:xml:ns:yang:ietf-i2nsf-monitor";
        prefix
          iim;
        import ietf-inet-types{
          prefix inet;
          reference
            "Section 4 of RFC 6991";
        }
        import ietf-yang-types {
          prefix yang;
          reference
            "Section 3 of RFC 6991";
        }
        organization
          "IETF I2NSF (Interface to Network Security Functions)
           Working Group";
        contact
          "WG Web: <http://tools.ietf.org/wg/i2nsf>
           WG List: <mailto:i2nsf@ietf.org>

           WG Chair: Linda Dunbar
           <mailto:Linda.duhbar@huawei.com>

           Editor: Jaehoon Paul Jeong
           <mailto:pauljeong@skku.edu>

           Editor: Chaehong Chung
```

```
    <mailto:darkhong@skku.edu>";

  description
    "This module is a YANG module for monitoring NSFs.

    Copyright (c) 2018 IETF Trust and the persons identified as
    authors of the code. All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject
    to the license terms contained in, the Simplified BSD License
    set forth in Section 4.c of the IETF Trust's Legal Provisions
    Relating to IETF Documents
    (http://trustee.ietf.org/license-info).

    This version of this YANG module is part of RFC 6087; see
    the RFC itself for full legal notices.";

  revision "2019-03-11" {
    description "First revision";
    reference
      "RFC XXXX: I2NSF NSF Monitoring YANG Data Model";
  }

  typedef severity {
    type enumeration {
      enum high {
        description
          "high-level";
      }
      enum middle {
        description
          "middle-level";
      }
      enum low {
        description
          "low-level";
      }
    }
    description
```

```
      "An indicator representing severity";
  }
  typedef log-action {
    type enumeration {
      enum allow {
        description
          "If action is allowed";
      }
      enum alert {
        description
          "If action is alert";
      }
      enum block {
        description
          "If action is block";
      }
      enum discard {
        description
          "If action is discarded";
      }
      enum declare {
        description
          "If action is declared";
      }
      enum block-ip {
        description
          "If action is block-ip";
      }
      enum block-service{
        description
          "If action is block-service";
      }
    }
    description
      "This is used for protocol";
  }
  typedef dpi-type{
    type enumeration {
      enum file-blocking{
        description
```

```
            "DPI for blocking file";
        }
        enum data-filtering{
          description
            "DPI for filtering data";
        }
        enum application-behavior-control{
          description
            "DPI for controlling application behavior";
        }
      }
      description
        "This is used for dpi type";
    }
    typedef operation-type{
      type enumeration {
        enum login{
          description
            "Login operation";
        }
        enum logout{
          description
            "Logout operation";
        }
        enum configuration{
          description
            "Configuration operation";
        }
      }
      description
        "An indicator representing operation-type";
    }
    typedef login-mode{
      type enumeration {
        enum root{
          description
            "Root login-mode";
        }
        enum user{
          description
```

```
            "User login-mode";
        }
        enum guest{
          description
            "Guest login-mode";
        }
    }
    description
      "An indicator representing login-mode";
}

identity characteristics {
    description
    "Base identity for monitoring information
    characteristics";
}
identity acquisition-method {
    base characteristics;
    description
    "The type of acquisition-method. Can be multiple
    types at once.";
}
identity subscription {
    base acquisition-method;
    description
    "The acquisition-method type is subscription";
}
identity query {
    base acquisition-method;
    description
    "The acquisition-method type is query";
}
identity emission-type {
    base characteristics;
    description
    "The type of emission-type.";
}
identity periodical {
    base emission-type;
    description
```

```
    "The emission-type type is periodical.";
  }
  identity on-change {
    base emission-type;
    description
    "The emission-type type is on-change.";
  }
  identity dampening-type {
    base characteristics;
    description
    "The type of dampening-type.";
  }
  identity no-dampening {
    base dampening-type;
    description
    "The dampening-type is no-dampening.";
  }
  identity on-repetition {
    base dampening-type;
    description
    "The dampening-type is on-repetition.";
  }
  identity none {
    base dampening-type;
    description
    "The dampening-type is none.";
  }

  identity authentication-mode {
    description
      "User authentication mode types:
       e.g., Local Authentication,
       Third-Party Server Authentication,
       Authentication Exemption, or Single Sign-On (SSO)
       Authentication.";
  }
  identity local-authentication {
    base authentication-mode;
    description
      "Authentication-mode : local authentication.";
```

```
}
identity third-party-server-authentication {
  base authentication-mode;
  description
    "If authentication-mode is
    third-part-server-authentication";
}
identity exemption-authentication {
  base authentication-mode;
  description
  "If authentication-mode is
  exemption-authentication";
}
identity sso-authentication {
  base authentication-mode;
  description
  "If authentication-mode is
  sso-authentication";
}

identity alarm-type {
  description
    "Base identity for detectable alarm types";
}
identity MEM-USAGE-ALARM {
  base alarm-type;
  description
  "A memory alarm is alerted";
}
identity CPU-USAGE-ALARM {
  base alarm-type;
  description
  "A CPU alarm is alerted";
}
identity DISK-USAGE-ALARM {
  base alarm-type;
  description
  "A disk alarm is alerted";
}
identity HW-FAILURE-ALARM {
```

```
  base alarm-type;
  description
  "A hardware alarm is alerted";
}
identity IFNET-STATE-ALARM {
  base alarm-type;
  description
  "An interface alarm is alerted";
}
identity event-type {
  description
    "Base identity for detectable event types";
}
identity ACCESS-DENIED {
  base event-type;
  description
  "The system event is access-denied.";
}
identity CONFIG-CHANGE {
  base event-type;
  description
  "The system event is config-change.";
}

identity flood-type {
  description
    "Base identity for detectable flood types";
}
identity syn-flood {
  base flood-type;
  description
    "A SYN flood is detected";
}
identity ack-flood {
  base flood-type;
  description
    "An ACK flood is detected";
}
identity syn-ack-flood {
  base flood-type;
```

```
  description
    "An SYN-ACK flood is detected";
}
identity fin-rst-flood {
  base flood-type;
  description
    "A FIN-RST flood is detected";
}
identity tcp-con-flood {
  base flood-type;
  description
    "A TCP connection flood is detected";
}
identity udp-flood {
  base flood-type;
  description
    "A UDP flood is detected";
}
identity icmp-flood {
  base flood-type;
  description
    "An ICMP flood is detected";
}
identity https-flood {
  base flood-type;
  description
    "A HTTPS flood is detected";
}
identity http-flood {
  base flood-type;
  description
    "A HTTP flood is detected";
}
identity dns-reply-flood {
  base flood-type;
  description
    "A DNS reply flood is detected";
}
identity dns-query-flood {
  base flood-type;
```

```
  description
    "A DNS query flood is detected";
}
identity sip-flood {
  base flood-type;
  description
    "A SIP flood is detected";
}

identity nsf-event-name {
  description
  "Base identity for detectable nsf event types";
}
identity SEC-EVENT-DDOS {
  base nsf-event-name;
  description
  "The nsf event is sec-event-ddos.";
}
identity SESSION-USAGE-HIGH {
  base nsf-event-name;
  description
  "The nsf event is session-usage-high";
}
identity SEC-EVENT-VIRUS {
  base nsf-event-name;
  description
  "The nsf event is sec-event-virus";
}
identity SEC-EVENT-INTRUSION {
  base nsf-event-name;
  description
  "The nsf event is sec-event-intrusion";
}
identity SEC-EVENT-BOTNET {
  base nsf-event-name;
  description
  "The nsf event is sec-event-botnet";
}
identity SEC-EVENT-WEBATTACK {
  base nsf-event-name;
```

```
  description
  "The nsf event is sec-event-webattack";
}
identity attack-type {
  description
    "The root ID of attack-based notification
    in the notification taxonomy";
}
identity system-attack-type {
  base attack-type;
  description
    "This ID is intended to be used
    in the context of system events";
}
identity nsf-attack-type {
  base attack-type;
  description
    "This ID is intended to be used
     in the context of nsf event";
}
identity botnet-attack-type {
  base nsf-attack-type;
  description
    "This is an ID stub limited to indicating
    that this attack type is botnet.
    The usual semantic and taxonomy is missing
    and name is used.";
}
identity virus-type {
  base nsf-attack-type;
  description
    "The type of virus. Can be multiple types at once.
     This attack type is associated with a detected
     system-log virus-attack";
}
identity trojan {
  base virus-type;
  description
    "The detected virus type is trojan";
}
```

```
identity worm {
  base virus-type;
  description
    "The detected virus type is worm";
}
identity macro {
  base virus-type;
  description
    "The detected virus type is macro";
}
identity intrusion-attack-type {
  base nsf-attack-type;
  description
    "The attack type is associated with
    a detected system-log intrusion";
}
identity brute-force {
  base intrusion-attack-type;
  description
    "The intrusion type is brute-force";
}
identity buffer-overflow {
  base intrusion-attack-type;
  description
    "The intrusion type is buffer-overflow";
}
identity web-attack-type {
  base nsf-attack-type;
  description
    "The attack type associated with
    a detected system-log web-attack";
}
identity command-injection {
  base web-attack-type;
  description
    "The detected web attack type is command injection";
}
identity xss {
  base web-attack-type;
  description
```

```
      "The detected web attack type is XSS";
}
identity csrf {
  base web-attack-type;
  description
    "The detected web attack type is CSRF";
}
identity ddos-attack-type {
  base nsf-attack-type;
  description
    "The attack type is associated with a detected
    nsf-log event";
}

identity req-method {
  description
    "A set of request types (if applicable).
     For instance, PUT or GET in HTTP";
}
identity put-req {
  base req-method;
  description
    "The detected request type is PUT";
}
identity get-req {
  base req-method;
  description
    "The detected request type is GET";
}

identity filter-type {
  description
    "The type of filter used to detect, for example,
     a web-attack. Can be applicable to more than
     web-attacks. Can be more than one type.";
}
identity whitelist {
  base filter-type;
  description
    "The applied filter type is whitelist";
```

```
}
identity blacklist {
  base filter-type;
  description
    "The applied filter type is blacklist";
}
identity user-defined {
  base filter-type;
  description
    "The applied filter type is user-defined";
}
identity balicious-category {
  base filter-type;
  description
    "The applied filter is balicious category";
}
identity unknown-filter {
  base filter-type;
  description
    "The applied filter is unknown";
}

identity access-mode {
  description
    "Base identity for detectable access mode.";
}
identity ppp {
  base access-mode;
  description
    "Access-mode : ppp";
}
identity svn {
  base access-mode;
  description
    "Access-mode : svn";
}
identity local {
  base access-mode;
  description
    "Access-mode : local";
```

```
}

identity protocol-type {
  description
    "An identity used to enable type choices in leaves
     and leaflists wrt protocol metadata.";
}
identity tcp {
  base ipv4;
  base ipv6;
  description
    "TCP protocol type.";
    reference
      "RFC 793: Transmission Control Protocol";
}
identity udp {
  base ipv4;
  base ipv6;
  description
  "UDP protocol type.";
  reference
    "RFC 768: User Datagram Protocol";
}
identity icmp {
  base ipv4;
  base ipv6;
  description
    "General ICMP protocol type.";
    reference
      "RFC 792: Internet Control Message Protocol";
}
identity icmpv4 {
  base ipv4;
  description
    "ICMPv4 protocol type.";
}
identity icmpv6 {
  base ipv6;
  description
    "ICMPv6 protocol type.";
```

```
}
identity ip {
  base protocol-type;
  description
    "General IP protocol type.";
    reference
      "RFC 791: Internet Protocol
      RFC 2460: Internet Protocol, Version 6 (IPv6)";
}
identity ipv4 {
  base ip;
  description
    "IPv4 protocol type.";
    reference
      "RFC 791: Internet Protocol";
}
identity ipv6 {
  base ip;
  description
    "IPv6 protocol type.";
    reference
      "RFC 2460: Internet Protocol, Version 6 (IPv6)";
}
identity http {
  base tcp;
  description
    "HTPP protocol type.";
    reference
      "RFC 2616: Hypertext Transfer Protocol";
}
identity ftp {
  base tcp;
  description
   "FTP protocol type.";
   reference
     "RFC 959: File Transfer Protocol";
}
grouping common-monitoring-data {
  description
  "The data set of common monitoring";
```

```
    leaf message {
      type string;
      description
        "This is a freetext annotation of
        monitoring notification content";
    }
    leaf time-stamp {
      type yang:date-and-time;
      description
        "Indicates the time of message generation";
    }
    leaf vendor-name {
      type string;
      description
        "The name of the NSF vendor";
    }
    leaf nsf-name {
      type string;
      description
        "The name (or IP) of the NSF
        generating the message";
    }
    leaf module-name {
      type string;
      description
        "The module name outputting the message";
    }
    leaf severity {
      type severity;
      description
        "The severity of the alarm such
        as critical, high, middle, low.";
    }
  }
  grouping characteristics{
    description
      "A set of monitoring information characteristics";
    leaf acquisition-method {
      type identityref {
        base acquisition-method;
```

```
      }
      description
        "The acquisition-method for characteristics";
    }
    leaf emission-type {
      type identityref {
        base emission-type;
      }
      description
        "The emission-type for characteristics";
    }
    leaf dampening-type {
      type identityref {
        base dampening-type;
      }
      description
        "The dampening-type for characteristics";
    }
  }
  grouping i2nsf-system-alarm-type-content {
    description
      "A set of system alarm type contents";
    leaf usage {
      type uint8;
      description
        "specifies the amount of usage";
    }
    leaf threshold {
      type uint8;
      description
        "The threshold triggering the alarm or the event";
    }
  }
  grouping i2nsf-system-event-type-content {
    description
      "System event metadata associated
       with system events caused by user activity.";
    leaf user {
      type string;
      mandatory true;
```

```
      description
        "Name of a user";
    }
    leaf group {
      type string;
      mandatory true;
      description
        "Group to which a user belongs.";
    }
    leaf login-ip-addr {
      type inet:ipv4-address;
      mandatory true;
      description
        "Login IP address of a user.";
    }
    leaf authentication {
      type identityref {
        base authentication-mode;
      }
      description
        "The authentication-mode for authentication";
    }
  }
  grouping i2nsf-nsf-event-type-content-extend {
    description
      "A set of common IPv4-related NSF event
      content elements";
    leaf src-ip {
      type inet:ipv4-address;
      description
        "The source IP address of the packet";
    }
    leaf dst-ip {
      type inet:ipv4-address;
      description
        "The destination IP address of the packet";
    }
    leaf src-port {
      type inet:port-number;
      description
```

```
      "The source port of the packet";
  }
  leaf dst-port {
    type inet:port-number;
    description
      "The destination port of the packet";
  }
  leaf src-zone {
    type string;
    description
      "The source security zone of the packet";
  }
  leaf dst-zone {
    type string;
    description
      "The destination security zone of the packet";
  }
  leaf rule-id {
    type uint8;
    mandatory true;
    description
      "The ID of the rule being triggered";
  }
  leaf rule-name {
    type string;
    mandatory true;
    description
      "The name of the rule being triggered";
  }
  leaf profile {
    type string;
    description
      "Security profile that traffic matches.";
  }
  leaf raw-info {
    type string;
    description
      "The information describing the packet
      triggering the event.";
  }
```

```
}
grouping i2nsf-nsf-event-type-content {
  description
    "A set of common IPv4-related NSF event
    content elements";
  leaf dst-ip {
    type inet:ipv4-address;
    description
      "The destination IP address of the packet";
  }
  leaf dst-port {
    type inet:port-number;
    description
      "The destination port of the packet";
  }
  leaf rule-id {
    type uint8;
    mandatory true;
    description
      "The ID of the rule being triggered";
  }
  leaf rule-name {
    type string;
    mandatory true;
    description
      "The name of the rule being triggered";
  }
  leaf profile {
    type string;
    description
      "Security profile that traffic matches.";
  }
  leaf raw-info {
    type string;
    description
      "The information describing the packet
      triggering the event.";
  }
}
grouping traffic-rates {
```

```
description
  "A set of traffic rates
  for statistics data";
leaf total-traffic {
  type uint32;
  description
    "Total traffic";
}
leaf in-traffic-ave-rate {
  type uint32;
  description
    "Inbound traffic average rate in pps";
}
leaf in-traffic-peak-rate {
  type uint32;
  description
    "Inbound traffic peak rate in pps";
}
leaf in-traffic-ave-speed {
  type uint32;
  description
    "Inbound traffic average speed in bps";
}
leaf in-traffic-peak-speed {
  type uint32;
  description
    "Inbound traffic peak speed in bps";
}
leaf out-traffic-ave-rate {
  type uint32;
  description
    "Outbound traffic average rate in pps";
}
leaf out-traffic-peak-rate {
  type uint32;
  description
    "Outbound traffic peak rate in pps";
}
leaf out-traffic-ave-speed {
  type uint32;
```

```
      description
        "Outbound traffic average speed in bps";
    }
    leaf out-traffic-peak-speed {
      type uint32;
      description
        "Outbound traffic peak speed in bps";
    }
  }
  grouping i2nsf-system-counter-type-content{
    description
      "A set of system counter type contents";
    leaf interface-name {
      type string;
      description
        "Network interface name configured in NSF";
    }
    leaf in-total-traffic-pkts {
      type uint32;
      description
        "Total inbound packets";
    }
    leaf out-total-traffic-pkts {
      type uint32;
      description
        "Total outbound packets";
    }
    leaf in-total-traffic-bytes {
      type uint32;
      description
        "Total inbound bytes";
    }
    leaf out-total-traffic-bytes {
      type uint32;
      description
        "Total outbound bytes";
    }
    leaf in-drop-traffic-pkts {
      type uint32;
      description
```

```
        "Total inbound drop packets";
    }
    leaf out-drop-traffic-pkts {
      type uint32;
      description
        "Total outbound drop packets";
    }
    leaf in-drop-traffic-bytes {
      type uint32;
      description
        "Total inbound drop bytes";
    }
    leaf out-drop-traffic-bytes {
      type uint32;
      description
        "Total outbound drop bytes";
    }
    uses traffic-rates;
}
grouping i2nsf-nsf-counters-type-content{
    description
      "A set of nsf counters type contents";
    leaf src-ip {
      type inet:ipv4-address;
      description
        "The source IP address of the packet";
    }
    leaf dst-ip {
      type inet:ipv4-address;
      description
        "The destination IP address of the packet";
    }
    leaf src-port {
      type inet:port-number;
      description
        "The source port of the packet";
    }
    leaf dst-port {
      type inet:port-number;
      description
```

```
      "The destination port of the packet";
  }
  leaf src-zone {
    type string;
    description
      "The source security zone of the packet";
  }
  leaf dst-zone {
    type string;
    description
      "The destination security zone of the packet";
  }
  leaf src-region {
    type string;
    description
      "Source region of the traffic";
  }
  leaf dst-region{
    type string;
    description
      "Destination region of the traffic";
  }
  leaf policy-id {
    type uint8;
    description
      "The ID of the policy being triggered";
  }
  leaf policy-name {
    type string;
    description
      "The name of the policy being triggered";
  }
  leaf src-user{
    type string;
    description
      "User who generates traffic";
  }
  leaf protocol {
    type identityref {
      base protocol-type;
```

```
      }
      description
        "Protocol type of traffic";
    }
    leaf app {
      type string;
      description
        "Application type of traffic";
    }
}

notification system-detection-alarm {
    description
      "This notification is sent, when a system alarm
      is detected.";
    leaf alarm-category {
      type identityref {
        base alarm-type;
      }
      description
        "The alarm category for
        system-detection-alarm notification";
    }
    uses characteristics;
    uses i2nsf-system-alarm-type-content;
    uses common-monitoring-data;
}
notification system-detection-event {
    description
      "This notification is sent, when a security-sensitive
       authentication action fails.";
    leaf event-category {
      type identityref {
        base event-type;
      }
      description
        "The event category for system-detection-event";
    }
    uses characteristics;
    uses i2nsf-system-event-type-content;
```

```
    uses common-monitoring-data;
  }
  notification nsf-detection-flood {
    description
      "This notification is sent,
      when a specific flood type is detected";
    leaf event-name {
      type identityref {
       base SEC-EVENT-DDOS;
      }
      description
      "The event name for nsf-detection-flood";
    }
    uses i2nsf-nsf-event-type-content;
    leaf sub-attack-type {
      type identityref {
        base flood-type;
      }
      description
        "Any one of Syn flood, ACK flood, SYN-ACK flood,
         FIN/RST flood, TCP Connection flood, UDP flood,
         Icmp flood, HTTPS flood, HTTP flood, DNS query flood,
         DNS reply flood, SIP flood, etc.";
    }
    leaf start-time {
      type yang:date-and-time;
      mandatory true;
      description
        "The time stamp indicating when the attack started";
    }
    leaf end-time {
      type yang:date-and-time;
      mandatory true;
      description
        "The time stamp indicating when the attack ended";
    }
    leaf attack-rate {
      type uint32;
      description
        "The PPS rate of attack traffic";
```

```
    }
    leaf attack-speed {
      type uint32;
      description
        "The BPS speed of attack traffic";
    }
    uses common-monitoring-data;
  }
  notification nsf-detection-session-table {
    description
      "This notification is sent, when a session table
      event is detected";
    leaf current-session {
      type uint8;
      description
        "The number of concurrent sessions";
    }
    leaf maximum-session {
      type uint8;
      description
        "The maximum number of sessions that the session
         table can support";
    }
    leaf threshold {
      type uint8;
      description
        "The threshold triggering the event";
    }
    uses common-monitoring-data;
  }
  notification nsf-detection-virus {
    description
      "This notification is sent, when a virus is detected";
    uses i2nsf-nsf-event-type-content-extend;
    leaf virus {
      type identityref {
        base virus-type;
      }
      description
        "The virus type for nsf-detection-virus notification";
```

```
    }
    leaf virus-name {
      type string;
      description
      "The name of the detected virus";
    }

    leaf file-type {
      type string;
      description
        "The type of file virus code
        is found in (if applicable).";
    }
    leaf file-name {
      type string;
      description
          "The name of file virus code
          is found in (if applicable).";
    }
    uses common-monitoring-data;
  }
  notification nsf-detection-intrusion {
    description
      "This notification is sent, when an intrusion event
       is detected.";
    uses i2nsf-nsf-event-type-content-extend;
    leaf protocol {
      type identityref {
        base protocol-type;
      }
      description
        "The protocol type for
        nsf-detection-intrusion notification";
    }
    leaf app {
      type string;
      description
        "The employed application layer protocol";
    }
    leaf sub-attack-type {
```

```
      type identityref {
        base intrusion-attack-type;
      }
      description
        "The sub attack type for intrusion attack";
    }
    uses common-monitoring-data;
  }
  notification nsf-detection-botnet {
    description
      "This notification is sent, when a botnet event is
       detected";
    uses i2nsf-nsf-event-type-content-extend;
    leaf attack-type {
      type identityref {
        base botnet-attack-type;
      }
      description
        "The attack type for botnet attack";
    }
    leaf protocol {
      type identityref {
        base protocol-type;
      }
      description
        "The protocol type for nsf-detection-botnet notification";
    }
    leaf botnet-name {
      type string;
      description
        "The name of the detected botnet";
    }
    leaf role {
      type string;
      description
        "The role of the communicating
         parties within the botnet";
    }
    uses common-monitoring-data;
  }
```

```
notification nsf-detection-web-attack {
  description
    "This notification is sent, when an attack event is
     detected";
  uses i2nsf-nsf-event-type-content-extend;
  leaf sub-attack-type {
    type identityref {
      base web-attack-type;
    }
    description
      "Concrete web attack type, e.g., sql injection,
       command injection, XSS, CSRF";
  }
  leaf request-method {
    type identityref {
      base req-method;
    }
    description
      "The method of requirement. For instance, PUT or
      GET in HTTP";
  }
  leaf req-uri {
    type string;
    description
    "Requested URI";
  }
  leaf uri-category {
    type string;
    description
    "Matched URI category";
  }
  leaf-list filtering-type {
    type identityref {
      base filter-type;
    }
    description
      "URL filtering type, e.g., Blacklist, Whitelist,
       User-Defined, Predefined, Malicious Category,
       Unknown";
  }
```

```
  uses common-monitoring-data;
}
notification system-access-log {
  description
    "The notification is sent, if there is
    a new system log entry about
     a system access event";
  leaf login-ip {
    type inet:ipv4-address;
    mandatory true;
    description
      "Login IP address of a user";
  }
  leaf administrator {
    type string;
    description
      "Administrator that maintains the device";
  }
  leaf login-mode {
    type login-mode;
    description
      "Specifies the administrator log-in mode";
  }
  leaf operation-type {
    type operation-type;
    description
      "The operation type that the administrator executes";
  }
  leaf result {
    type string;
    description
      "Command execution result";
  }
  leaf content {
    type string;
    description
      "The Operation performed by an administrator
      after login";
  }
  uses characteristics;
```

```
}
notification system-res-util-log {
  description
    "This notification is sent, if there is
    a new log entry representing resource
     utilization updates.";
  leaf system-status {
     type string;
     description
       "The current systems
       running status";
  }
  leaf cpu-usage {
    type uint8;
    description
      "Specifies the relative amount of
      cpu usage wrt platform resources";
  }
  leaf memory-usage {
     type uint8;
     description
       "Specifies the amount of memory usage";
  }
  leaf disk-usage {
    type uint8;
    description
      "Specifies the amount of disk usage";
  }
  leaf disk-left {
     type uint8;
     description
       "Specifies the amount of disk left";
  }
  leaf session-num {
    type uint8;
    description
      "The total number of sessions";
  }
  leaf process-num {
    type uint8;
```

```
      description
        "The total number of process";
    }
    leaf in-traffic-rate {
      type uint32;
      description
        "The total inbound traffic rate in pps";
    }
    leaf out-traffic-rate {
      type uint32;
      description
         "The total outbound traffic rate in pps";
    }
    leaf in-traffic-speed {
      type uint32;
      description
        "The total inbound traffic speed in bps";
    }
    leaf out-traffic-speed {
      type uint32;
      description
        "The total outbound traffic speed in bps";
    }
    uses characteristics;
  }
  notification system-user-activity-log {
    description
      "This notification is sent, if there is
      a new user activity log entry";
    uses characteristics;
    uses i2nsf-system-event-type-content;
    leaf access {
      type identityref {
        base access-mode;
      }
      description
        "The access type for
        system-user-activity-log notification";
    }
    leaf online-duration {
```

```
      type string;
      description
        "Online duration";
    }
    leaf logout-duration {
      type string;
      description
        "Lockout duration";
    }
    leaf addtional-info {
      type string;
      description
        "User activities. e.g., Successful
        User Login, Failed Login attempts,
        User Logout, Successful User
        Password Change, Failed User
        Password Change, User Lockout,
        User Unlocking, Unknown";
    }
  }
  notification nsf-log-ddos {
    description
      "This notification is sent, if there is
      a new DDoS event log entry in the nsf log";
    leaf attack-type {
      type identityref {
        base ddos-attack-type;
      }
      description
        "The ddos attack type for
        nsf-log-ddos notification";
    }
    leaf attack-ave-rate {
      type uint32;
      description
        "The ave PPS of attack traffic";
    }
    leaf attack-ave-speed {
      type uint32;
      description
```

```
        "the ave bps of attack traffic";
    }
    leaf attack-pkt-num {
      type uint32;
      description
        "the number of attack packets";
    }
    leaf attack-src-ip {
      type inet:ipv4-address;
      description
        "The source IP addresses of attack
         traffics. If there are a large
         amount of IP addresses, then
         pick a certain number of resources
         according to different rules.";
    }
    leaf action {
      type log-action;
      description
        "Action type: allow, alert,
        block, discard, declare,
        block-ip, block-service";
    }
    uses characteristics;
    uses common-monitoring-data;
  }
  notification nsf-log-virus {
    description
      "This notification is sent, if there is
      a new virus event log entry in the nsf log";
    leaf attack-type {
      type identityref {
        base virus-type;
      }
      description
        "The virus type for nsf-log-virus notification";
    }
    leaf action {
      type log-action;
      description
```

```
        "Action type: allow, alert,
        block, discard, declare,
        block-ip, block-service";
    }
    leaf os{
      type string;
      description
        "simple os information";
    }
    leaf time {
      type yang:date-and-time;
      mandatory true;
      description
        "Indicate the time when the message
        is generated";
    }
    uses characteristics;
    uses common-monitoring-data;
  }
  notification nsf-log-intrusion {
    description
      "This notification is sent, if there is
      a new intrusion event log entry in the nsf log";
    leaf attack-type {
      type identityref {
        base intrusion-attack-type;
      }
      description
        "The intrusion attack type for
        nsf-log-intrusion notification";
    }
    leaf action {
      type log-action;
      description
        "Action type: allow, alert,
        block, discard, declare,
        block-ip, block-service";
    }
    leaf time {
      type yang:date-and-time;
```

```
      mandatory true;
      description
        "Indicate the time when the message
        is generated";
    }
    leaf attack-rate {
      type uint32;
      description
        "The PPS of attack traffic";
    }
    leaf attack-speed {
      type uint32;
      description
        "The bps of attack traffic";
    }
    uses characteristics;
    uses common-monitoring-data;
  }
  notification nsf-log-botnet {
    description
      "This notification is sent, if there is
      a new botnet event log in the nsf log";
    leaf attack-type {
      type identityref {
        base botnet-attack-type;
      }
      description
        "The botnet attack type for
        nsf-log-botnet notification";
    }
    leaf action {
      type log-action;
      description
        "Action type: allow, alert,
        block, discard, declare,
        block-ip, block-service";
    }
    leaf botnet-pkt-num{
      type uint8;
      description
```

```
        "The number of the packets sent to
        or from the detected botnet";
    }
    leaf os{
      type string;
      description
        "simple os information";
    }
    uses characteristics;
    uses common-monitoring-data;
  }
  notification nsf-log-dpi {
    description
      "This notification is sent, if there is
      a new dpi event in the nsf log";
    leaf attack-type {
      type dpi-type;
      description
        "The type of the dpi";
    }
    uses characteristics;
    uses i2nsf-nsf-counters-type-content;
    uses common-monitoring-data;
  }
  notification nsf-log-vuln-scan {
    description
      "This notification is sent, if there is
      a new vulnerability-scan report in the nsf log";
    leaf vulnerability-id {
      type uint8;
      description
        "The vulnerability id";
    }
    leaf victim-ip {
      type inet:ipv4-address;
      description
        "IP address of the victim host
        which has vulnerabilities";
    }
    leaf protocol {
```

```
      type identityref {
        base protocol-type;
      }
      description
        "The protocol type for
        nsf-log-vuln-scan notification";
    }
    leaf port-num {
      type inet:port-number;
        description
          "The port number";
    }
    leaf level {
      type severity;
      description
        "The vulnerability severity";
    }
    leaf os {
      type string;
      description
        "simple os information";
    }
    leaf vulnerability-info {
      type string;
      description
        "The information about the vulnerability";
    }
    leaf fix-suggestion {
      type string;
      description
        "The fix suggestion to the vulnerability";
    }
    leaf service {
      type string;
      description
        "The service which has vulnerability in the victim host";
    }
    uses characteristics;
    uses common-monitoring-data;
}
```

```
notification nsf-log-web-attack {
  description
    "This notification is sent, if there is
    a new web-attack event in the nsf log";
  leaf attack-type {
    type identityref {
      base web-attack-type;
    }
    description
      "The web attack type for
      nsf-log-web-attack notification";
  }
  leaf rsp-code {
    type string;
    description
      "Response code";
  }
  leaf req-clientapp {
    type string;
    description
      "The client application";
  }
  leaf req-cookies {
    type string;
    description
      "Cookies";
  }
  leaf req-host {
    type string;
    description
      "The domain name of the requested host";
  }
  leaf raw-info {
    type string;
    description
      "The information describing
      the packet triggering the event.";
  }
  uses characteristics;
  uses common-monitoring-data;
```

```
    }
    container counters {
      description
        "This is probably better covered by an import
        as this will not be notifications.
        Counter are not very suitable as telemetry, maybe
        via periodic subscriptions, which would still
        violate principle of least surprise.";
      container system-interface {
        description
          "The system counter type is interface counter";
        uses characteristics;
        uses i2nsf-system-counter-type-content;
        uses common-monitoring-data;
      }
      container nsf-firewall {
        description
          "The nsf counter type is firewall counter";
        uses characteristics;
        uses i2nsf-nsf-counters-type-content;
        uses traffic-rates;
      }
      container nsf-policy-hits {
        description
          "The counters of policy hit";
        uses characteristics;
        uses i2nsf-nsf-counters-type-content;
        uses common-monitoring-data;
        leaf hit-times {
          type uint32;
          description
            "The hit times for policy";
        }
      }
    }
  }
<CODE ENDS>
```

부 록 II-1

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

지식재산권 확약서 정보

(※ 본 표준 발간 이전에 접수된 지식재산권 확약서가 있는 경우에 작성하며, 해당 사항이 없는 경우, 각 항목별로 '해당 사항 없음'으로 기재하고, 본 양식을 삭제하지 않음)

- 발명의 명칭

  I2NSF NSF 모니터링 YANG 데이터 모델

- 권리자의 성명

  정재훈, 정재홍

- 출원 번호

  10-2019-0027215

- 출원 연월일

  2019. 03. 08

- 실시조건

- 확약서 접수일

부 록 II-2

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

시험인증 관련 사항

- 해당 사항 없음

부 록 II-3

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

본 표준의 연계(family) 표준

II-3.1 정재훈, 현상원, 김형식, 김진용, 김은수, 위사랑, 박정수, “SDN 기반의 네트워크 보안 기능의 인터페이스(I2NSF) 프레임워크 - 제 1 부: 개요”, TTAK.KO-12.0314, 2017 년 12 월 13 일.

II-3.2 정재훈, 김진용, 박정수, "SDN 기반의 네트워크 보안기능의 인터페이스(I2NSF) 프레임워크 - 제 5 부 : NSF 정책 전달 인터페이스", TTA Standard Draft, 2019 년 4 월 5 일.

부 록 II-4

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

참고 문헌

1. 규정 참조 표준

[1] D. Lopez, E. Lopez, L. Dunbar, J Strassner, and R. Kumar, "Framework for Interface to Network Security Functions", RFC 8329, February 2018.

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

2. 정보 참조 표준

[3] S. Hares, J. Strassner, D. Lopez, L. Xia, and H. Birkholz, "Interface to Network Security Functions (I2NSF) Terminology", draft-ietf-i2nsf-terminology-08, July 2019.

[4] L. Xia, D. Zhang, Y. Wu, R. Kumar, A. Lohiya, and H. Birkholz, "An Information Model for the Monitoring of Network Security Functions (NSF)", draft-zhang-i2nsf-info-model-monitoring-06, May 2018.

부 록 II-5

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

영문표준 해설서

- 해당 사항 없음

부 록 II-6

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

표준의 이력

판수	채택일	표준번호	내용	담당 위원회