(주) 이디 리서치

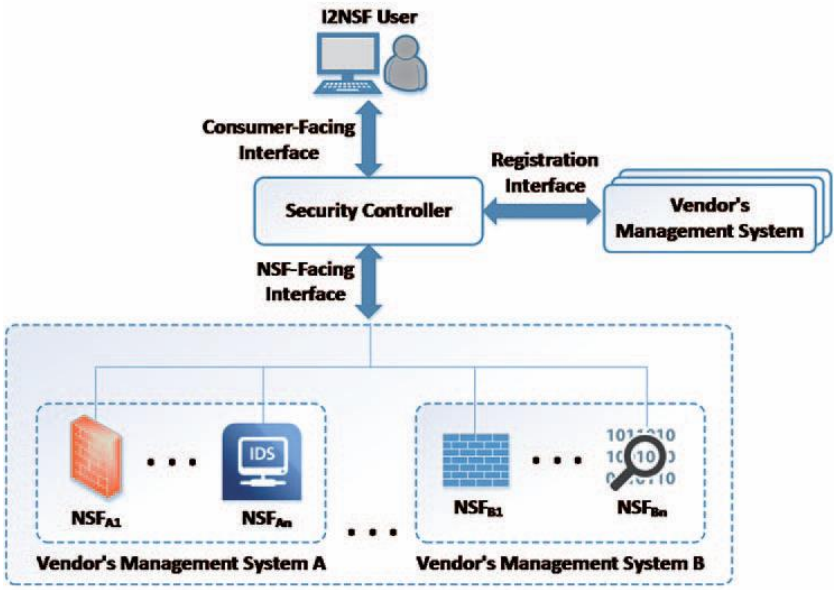
성균관대학교

다양한 네트워크 보안 함수를 위한 표준 인터페이스 및 프레임워크

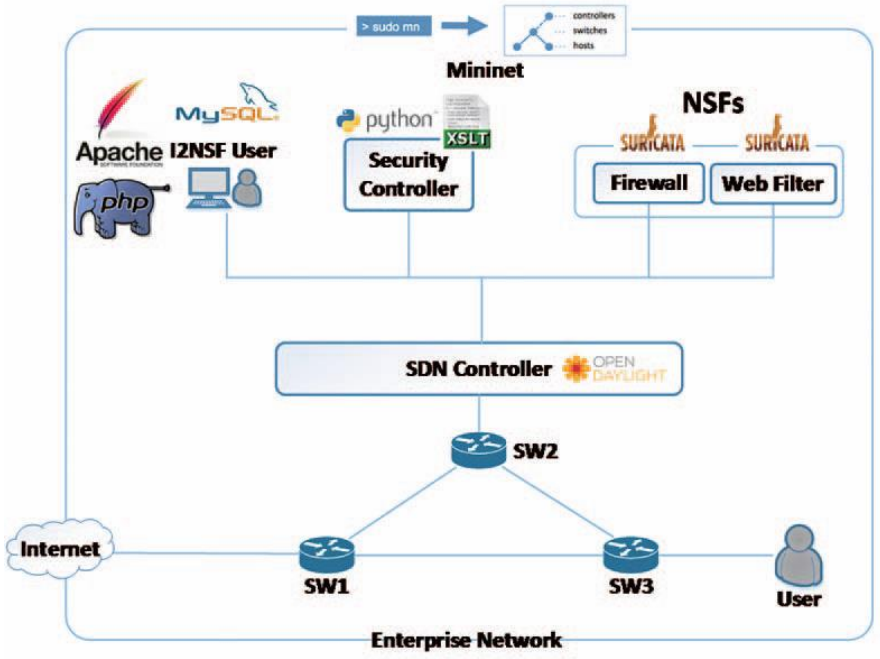
다양한 네트워크 보안 함수를 위한 표준 인터페이스 및 프레임워크

기술 개요

- 다양한 네트워크 보안 함수들의 서로 다른 인터페이스로 인해 많은 관리자 및 서비스 제공자들이 네트워크 보안 함수들을 관리하는데 있어 호환성 및 자동화의 어려움을 겪고 있으며 이를 해결하고자 **표준 인터페이스 제정 및 네트워크 보안 함수 프레임워크를 제공하여 관리의 효율성을 높이는 기술**



<네트워크 보안 함수를 위한 프레임워크>



<오픈소스를 활용하여 구현한 I2NSF 프레임워크>

다양한 네트워크 보안 함수를 위한 표준 인터페이스 및 프레임워크

세부 기술 내용

- **네트워크 보안 함수 인터페이스 프레임워크**
  - **소비자 정책 전달 인터페이스 표준 제정**
  - 네트워크 보안 함수 관리 자동화를 위하여 I2NSF(Interface to Network Security Functions) 유저(관리자)와 Security Controller간의 표준 인터페이스를 제정 및 제공
  - **네트워크 보안 함수 정책 전달 인터페이스 표준 제정**
  - 네트워크 보안 함수 관리 자동화를 위하여 Security Controller와 네트워크 보안 함수간의 표준 인터페이스를 제정 및 제공
  - **등록 인터페이스 표준 제정**
  - 네트워크 보안 함수 관리 자동화를 위하여 Vendor's Management System과 Security Controller간의 표준 인터페이스를 제정 및 제공
- **네트워크 보안 함수**
  - **방화벽 : SDN 및 오픈소스 Suricata를 활용하여 방화벽 기능을 제공하는 기술**
  - **DPI : Suricata를 활용하여 DPI 기능을 제공하는 기술**
- **네트워크 보안 함수 인터페이스 프레임워크에서의 진보된 기술**
  - **네트워크 보안 함수들의 모니터링 기술**
  - 네트워크 보안 함수들의 효율적인 관리 및 로드 밸런싱을 제공하는 모니터링 기술
  - **네트워크 보안 함수 기능에 따른 동적인 보안 정책 설정 기술**
  - 관리자들이 내린 보안 정책들을 네트워크 보안 함수 기능에 따라 네트워크 보안 함수에 동적으로 보안 정책이 설정될 수 있도록 하는 기술
  - **고수준 정책(High-level Policy)로부터 저수준 정책(Low-level Policy)으로 번역**
  - 관리자들로부터 받은 고수준 보안 정책들을 네트워크 보안 함수들이 이해할 수 있는 저수준 정책으로 번역하는 기술

다양한 네트워크 보안 함수를 위한 표준 인터페이스 및 프레임워크

기술의 특징

- 기존 네트워크 보안 함수들 간의 표준 인터페이스가 없어 호환성 및 관리의 어려움을 겪음
  - 네트워크 보안 함수들 간의 표준 인터페이스를 제공함으로써 호환성 및 관리의 효율성 증가
- 기존 네트워크 보안 함수들을 통합 관리하는 프레임워크가 없어 수많은 네트워크 보안 함수들의 관리의 효율성 떨어짐
  - 네트워크 보안 함수들 관리하는 프레임워크를 제공함으로써 네트워크 보안 함수 기반의 보안 서비스를 자동화함으로써 관리의 효율성 증가

관련 특허

No.	특허번호	발명의 명칭
1	10-2017-0079120	네트워크 가상화 환경에서 보안 관리를 위한 장치 및 방법
2	10-2018-0131256	네트워크 보안 서비스를 제공하기 위한 방법 및 이를 위한 장치
3	10-2019-0079436	I2NSF 네트워크 보안 기능에 직면한 인터페이스 YANG 데이터 모델
4	10-2019-0125454	네트워크 보안 기능 인터페이스를 위한 보안 정책 번역

활용분야

- 네트워크 보안 장비를 만드는 제조사
- 네트워크 보안 함수들을 활용하는 회사 및 서비스 제공자
- 네트워크 함수 가상화를 제공하는 가상화 환경

(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0003738
(43) 공개일자 2020년01월10일

(51) 국제특허분류(Int. Cl.)
H04L 29/06 (2006.01) *H04L 12/24* (2006.01)
(52) CPC특허분류
H04L 63/20 (2013.01)
H04L 41/145 (2013.01)
(21) 출원번호 10-2019-0079436
(22) 출원일자 2019년07월02일
심사청구일자 2019년07월02일
(30) 우선권주장
1020180076737 2018년07월02일 대한민국(KR)

(71) 출원인
성균관대학교산학협력단
경기도 수원시 장안구 서부로 2066 (천천동, 성균관대학교내)
(72) 발명자
정재훈
부산광역시 금정구 금강로 225, 207동 2203호(장전동, 벽산블루밍디자인시티)
김진용
경기도 수원시 장안구 서부로 2066, 311호 (천천동, 성균관대기숙사 의관)
(74) 대리인
특허법인로얄

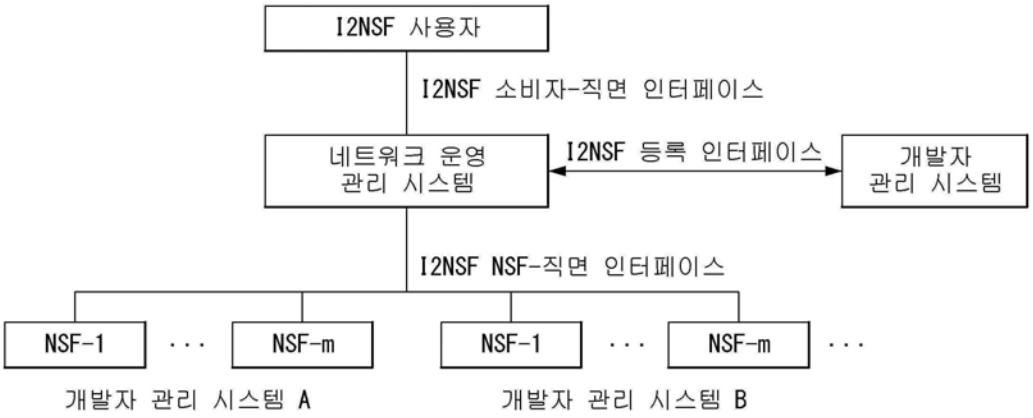
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 **I2NSF 네트워크 보안 기능에 직면한 인터페이스 YANG 데이터 모델**

(57) 요 약

보안 관리 시스템에서 네트워크 운영 관리 시스템이 보안 서비스를 제공하기 위한 방법 및 장치에 관한 것이다. 본 발명에 의하면, 보안 관리 시스템에서 네트워크 운영 관리 시스템이 보안 서비스를 제공하기 위한 방법에 있어서, I2NSF(Interface to Network Security Functions) 사용자로부터 상위 레벨(High-Level)의 제 1 보안 정책을 수신하고, 개발자 관리 시스템을 통해, 이용가능한 보안 서비스를 수신하며, 상기 보안 서비스에 근거하여, 상기 제 1 보안 정책에 대응되는 하위 레벨(Low-Level)의 제 2 보안 정책을 생성하고, 상기 생성된 제 2 보안 정책을 복수의 NSF(Network Security Function) 각각에게 설정하기 위한 상기 제 2 보안 정책을 포함하는 패킷을 전송하고, 상기 네트워크 운영 관리 시스템과 상기 복수의 NSF 각각은 I2NSF NSF-직면 인터페이스로 연결되며, 상기 제 2 보안 정책은 개발자 관리 시스템을 통해, 동적인 수명시간(life-cycle)을 갖을 수 있다.

대 표 도



이 발명을 지원한 국가연구개발사업
과제고유번호 2016-0-00078
부처명 과학기술정보통신부
연구관리전문기관 정보통신기술진흥센터
연구사업명 정보통신·방송연구개발사업
연구과제명 맞춤형 보안서비스 제공을 위한 클라우드 기반 지능형 보안 기술 개발
기 여 율 1/1
주관기관 성균관대학교
연구기간 2018.01.01 ~ 2018.12.31

명 세 서

청구범위

청구항 1

보안 관리 시스템에서 네트워크 운영 관리 시스템이 보안 서비스를 제공하기 위한 방법에 있어서,

I2NSF(Interface to Network Security Functions) 사용자로부터 상위 레벨(High-Level)의 제 1 보안 정책을 수신하는 단계;

개발자 관리 시스템을 통해, 이용가능한 보안 서비스를 수신하는 단계;

상기 보안 서비스에 근거하여, 상기 제 1 보안 정책에 대응되는 하위 레벨(Low-Level)의 제 2 보안 정책을 생성하는 단계; 및

상기 생성된 제 2 보안 정책을 복수의 NSF(Network Security Function) 각각에게 설정하기 위한 상기 제 2 보안 정책을 포함하는 패킷을 전송하는 단계;를 포함하되,

상기 네트워크 운영 관리 시스템과 상기 복수의 NSF 각각은 I2NSF NSF-직면 인터페이스로 연결되며, 상기 제 2 보안 정책은 상기 개발자 관리 시스템을 통해, 동적인 수명시간(life-cycle)을 갖는 방법.

청구항 2

제 1 항에 있어서,

상기 제 2 보안 정책은 적용되는 정책 규칙, 및 일반적인 보안 기능을 위한 동작을 나타나내는 기본 동작 정보를 포함하는 방법.

청구항 3

제 2 항에 있어서,

상기 정책 규칙은 정책 정보 및 규칙 정보를 포함하며,

상기 정책 정보 및 상기 규칙 정보는 시스템의 변경을 나타내는 이벤트 절(Event Clause), 정책 규칙의 적용 조건을 나타내는 조건 절(Condition Clause), 및 상기 이벤트 절 및 상기 조건 절을 만족할 때 수행되는 보안 기능을 나타내는 동작 절(Action Clause)을 포함하는 방법.

청구항 4

제 3 항에 있어서,

상기 정책 규칙은 상기 정책 규칙이 적용되는 시간을 나타내는 시간 정보를 더 포함하는 방법.

청구항 5

제 4 항에 있어서,

상기 시간 정보는 상기 정책 규칙이 적용되는 절대적인 시간을 나타내는 절대 시간 정보 또는 상기 정책 규칙이 적용되는 주기적인 시간을 나타내는 주기 정보 또는 상기 정책 규칙이 적용되는 지속 시간을 나타내는 지속 시간 정보를 포함하는 방법.

청구항 6

제 3 항에 있어서,

상기 이벤트 절은 상기 조건 절을 평가할 수 있는지 여부를 결정하는데 사용되는 방법.

청구항 7

제 3 항에 있어서,

상기 조건 절은 어플리케이션 실행과 관련된 규칙(rule)의 설정을 위한 상기 어플리케이션의 상태 정보를 포함하는 방법.

청구항 8

제 3 항에 있어서,

상기 동작 절은 입력 동작(ingress action), 전송 동작(egress action) 및 적용 프로파일 동작(application profile action)을 포함하는 방법.

청구항 9

제 8 항에 있어서,

상기 프로파일 동작은 응용 프로그램 계층에 적용되는 보안 기능을 나타내는 컨텐츠 보안 제어, 및 네트워크 공격의 탐지 및 완화를 위한 공격 완화 제어를 포함하는 방법.

청구항 10

제 3 항에 있어서,

상기 조건 절은 특정 URL(Uniform Resource Locator) 접속과 관련된 규칙(rule)의 설정을 위한 상기 URL의 상태 정보를 포함하는 방법.

청구항 11

보안 서비스를 제공하기 위한 보안 관리 시스템에 있어서,

상위 레벨(High-Level)의 제 1 보안 정책을 생성하는 I2NSF(Interface to Network Security Functions) 사용자;

보안 서비스를 제공하는 개발자 관리 시스템;

상기 I2NSF 사용자로부터 상기 제 1 보안 정책을 수신하고, 상기 개발자 관리 시스템으로부터 상기 보안 서비스를 수신하며, 상기 보안 서비스에 근거하여, 상기 제 1 보안 정책에 대응되는 하위 레벨(Low-Level)의 제 2 보안 정책을 생성하고, 상기 생성된 제 2 보안 정책을 복수의 NSF(Network Security Function) 각각에게 설정하기 위한 상기 제 2 보안 정책을 포함하는 패킷을 전송하는 네트워크 운영 관리 시스템; 및

상기 보안 관리 시스템으로부터 상기 제 2 보안 정책을 수신하는 복수의 NSF(Network Security Function)을 포함하되,

상기 네트워크 운영 관리 시스템과 상기 복수의 NSF 각각은 I2NSF NSF-직면 인터페이스로 연결되며, 상기 제 2 보안 정책은 개발자 관리 시스템을 통해, 동적인 수명시간(life-cycle)을 갖는 보안 관리 시스템.

청구항 12

제 11 항에 있어서,

상기 제 2 보안 정책은 적용되는 정책 규칙, 및 일반적인 보안 기능을 위한 동작을 나타나내는 기본 동작 정보를 포함하는 보안 관리 시스템.

청구항 13

제 12 항에 있어서,

상기 정책 규칙은 정책 정보 및 규칙 정보를 포함하며,

상기 정책 정보 및 상기 규칙 정보는 시스템의 변경을 나타내는 이벤트 절(Event Clause), 정책 규칙의 적용 조건을 나타내는 조건 절(Condition Clause), 및 상기 이벤트 절 및 상기 조건 절을 만족할 때 수행되는 보안 기능을 나타내는 동작 절(Action Clause)을 포함하는 보안 관리 시스템.

청구항 14

제 13 항에 있어서,

상기 정책 규칙은 상기 정책 규칙이 적용되는 시간을 나타내는 시간 정보를 더 포함하는 보안 관리 시스템.

청구항 15

제 14 항에 있어서,

상기 시간 정보는 상기 정책 규칙이 적용되는 절대적인 시간을 나타내는 절대 시간 정보 또는 상기 정책 규칙이 적용되는 주기적인 시간을 나타내는 주기 정보 또는 상기 정책 규칙이 적용되는 지속 시간을 나타내는 지속 시간 정보를 포함하는 보안 관리 시스템.

청구항 16

제 13 항에 있어서,

상기 이벤트 절은 상기 조건 절을 평가할 수 있는지 여부를 결정하는데 사용되는 보안 관리 시스템.

청구항 17

제 13 항에 있어서,

상기 조건 절은 어플리케이션 실행과 관련된 규칙(rule)의 설정을 위한 상기 어플리케이션의 상태 정보를 포함하는 보안 관리 시스템.

청구항 18

제 13 항에 있어서,

상기 동작 절은 입력 동작(ingress action), 전송 동작(egress action) 및 적용 프로파일 동작(application profile action)을 포함하는 보안 관리 시스템.

청구항 19

제 18 항에 있어서,

상기 프로파일 동작은 응용 프로그램 계층에 적용되는 보안 기능을 나타내는 컨텐츠 보안 제어, 및 네트워크 공격의 탐지 및 완화를 위한 공격 완화 제어를 포함하는 보안 관리 시스템.

청구항 20

제 13 항에 있어서,

상기 조건 절은 특정 URL(Uniform Resource Locator) 접속과 관련된 규칙(rule)의 설정을 위한 상기 URL의 상태 정보를 포함하는 보안 관리 시스템.

발명의 설명

기 술 분 야

[0001] 본 발명은 데이터 모델에 관한 것으로서, 보다 상세하게 I2NSF(Interface to Network Security Functions)에서 네트워크 보안 기능(Network Security Functions: NSF)에 직면한 인터페이스를 위한 정보 모델과 보안 서비스를 위한 YANG 데이터 모델을 정의하기 위한 것이다.

배 경 기 술

[0002] 네트워크를 전세계에 연결하면 지리적 거리에 관계없이 신속하게 정보에 액세스할 수 있다. 인터넷은 본질적으로 서로 다른 레벨들의 계층 구조가 서로 연결된 수많은 네트워크이다.

[0003] 인터넷은 IETF (Internet Engineering Task Force)에서 공표 한 TCP / IP (전송 제어 프로토콜/인터넷 프로토콜)에 따라 운영되며, TCP/IP는 RFC (Request For Comments) 703 및 IETF에서 발행 한 RFC 791에서 찾을 수 있다.

발명의 내용

해결하려는 과제

[0004] 본 발명의 목적은, I2NSF(Interface to Network Security Functions)에서 네트워크 보안 기능(Network Security Functions: NSF)에 직면한 인터페이스를 위한 정보 모델과 보안 서비스를 위한 YANG 데이터 모델을 설계하기 위한 방법을 제안한다.

[0005] 또한, 본 발명은 네트워크 보안 제어, 콘텐츠 보안 제어 및 공격 완화 제어와 같은 세 가지 보안 기능(예: 네트워크 보안 기능)에 대한 특정 정보 모델 및 해당 데이터 모델을 설계하기 위한 방법을 제안한다.

[0006] 본 명세서에서 이루고자 하는 기술적 과제들은 이상에서 언급한 기술적 과제들로 제한되지 않으며, 언급하지 않은 또 다른 기술적 과제들은 아래의 기재로부터 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0007] 본 발명의 일 양상은, 보안 관리 시스템에서 네트워크 운영 관리 시스템이 보안 서비스를 제공하기 위한 방법에 있어서, I2NSF(Interface to Network Security Functions) 사용자로부터 상위 레벨(High-Level)의 제 1 보안 정책을 수신하는 단계; 개발자 관리 시스템을 통해, 이용가능한 보안 서비스를 수신하는 단계; 상기 보안 서비스에 근거하여, 상기 제 1 보안 정책에 대응되는 하위 레벨(Low-Level)의 제 2 보안 정책을 생성하는 단계; 및 상기 생성된 제 2 보안 정책을 복수의 NSF(Network Security Function) 각각에게 설정하기 위한 상기 제 2 보안 정책을 포함하는 패킷을 전송하는 단계;를 포함하되, 상기 네트워크 운영 관리 시스템과 상기 복수의 NSF 각각은 I2NSF NSF-직면 인터페이스로 연결되며, 상기 제 2 보안 정책은 개발자 관리 시스템을 통해, 동적인 수명 시간(life-cycle)을 갖을 수 있다.

[0008] 또한, 상기 제 2 보안 정책은 적용되는 정책 규칙, 및 일반적인 보안 기능을 위한 동작을 나타나내는 기본 동작 정보를 포함할 수 있다.

[0009] 또한, 상기 정책 규칙은 정책 정보 및 규칙 정보를 포함하며, 상기 정책 정보 및 상기 규칙 정보는 시스템의 변경을 나타내는 이벤트 절(Event Clause), 정책 규칙의 적용 조건을 나타내는 조건 절(Condition Clause), 및 상기 이벤트 절 및 상기 조건 절을 만족할 때 수행되는 보안 기능을 나타내는 동작 절(Action Clause)을 포함할 수 있다.

[0010] 또한, 상기 정책 규칙은 상기 정책 규칙이 적용되는 시간을 나타내는 시간 정보를 더 포함할 수 있다.

[0011] 또한, 상기 시간 정보는 상기 정책 규칙이 적용되는 절대적인 시간을 나타내는 절대 시간 정보 또는 상기 정책 규칙이 적용되는 주기적인 시간을 나타내는 주기 정보 또는 상기 정책 규칙이 적용되는 지속 시간을 나타내는 지속 시간 정보를 포함할 수 있다.

[0012] 또한, 상기 이벤트 절은 상기 조건 절을 평가할 수 있는지 여부를 결정하는데 사용될 수 있다.

[0013] 또한, 상기 조건 절은 어플리케이션 실행과 관련된 규칙(rule)의 설정을 위한 상기 어플리케이션의 상태 정보를 포함할 수 있다.

[0014] 또한, 상기 동작 절은 입력 동작(ingress action), 전송 동작(egress action) 및 적용 프로파일 동작(application profile action)을 포함할 수 있다.

[0015] 또한, 상기 프로파일 동작은 응용 프로그램 계층에 적용되는 보안 기능을 나타내는 컨텐츠 보안 제어, 및 네트워크 공격의 탐지 및 완화를 위한 공격 완화 제어를 포함할 수 있다.

[0016] 또한, 상기 조건 절은 특정 URL(Uniform Resource Locator) 접속과 관련된 규칙(rule)의 설정을 위한 상기 URL의 상태 정보를 포함할 수 있다.

[0017] 본 발명의 또 다른 일 양상은, 보안 서비스를 제공하기 위한 보안 관리 시스템에 있어서, 상위 레벨(High-Level)의 제 1 보안 정책을 생성하는 I2NSF(Interface to Network Security Functions) 사용자; 상기 I2NSF 사용자로부터 상기 제 1 보안 정책을 수신하고, 상기 제 1 보안 정책에 대응되는 하위 레벨(Low-Level)의 제 2 보안 정책을 생성하며, 상기 생성된 제 2 보안 정책을 복수의 NSF(Network Security Function) 각각에게 설정하기 위한 상기 제 2 보안 정책을 포함하는 패킷을 전송하는 네트워크 운영 관리 시스템; 및 상기 보안 관리 시스템

으로부터 상기 제 2 보안 정책을 수신하는 복수의 NSF(Network Security Function)을 포함하되, 상기 네트워크 운영 관리 시스템과 상기 복수의 NSF 각각은 I2NSF NSF-직면 인터페이스로 연결되며, 상기 제 2 보안 정책은 개발자 관리 시스템을 통해, 동적인 수명시간(life-cycle)을 갖을 수 있다.

[0018] 또한, 상기 제 2 보안 정책은 적용되는 정책 규칙, 및 일반적인 보안 기능을 위한 동작을 나타나내는 기본 동작 정보를 포함할 수 있다.

[0019] 또한, 상기 정책 규칙은 정책 정보 및 규칙 정보를 포함하며, 상기 정책 정보 및 상기 규칙 정보는 시스템의 변경을 나타내는 이벤트 절(Event Clause), 정책 규칙의 적용 조건을 나타내는 조건 절(Condition Clause), 및 상기 이벤트 절 및 상기 조건 절을 만족할 때 수행되는 보안 기능을 나타내는 동작 절(Action Clause)을 포함할 수 있다.

[0020] 또한, 상기 정책 규칙은 상기 정책 규칙이 적용되는 시간을 나타내는 시간 정보를 더 포함할 수 있다.

[0021] 또한, 상기 시간 정보는 상기 정책 규칙이 적용되는 절대적인 시간을 나타내는 절대 시간 정보 또는 상기 정책 규칙이 적용되는 주기적인 시간을 나타내는 주기 정보 또는 상기 정책 규칙이 적용되는 지속 시간을 나타내는 지속 시간 정보를 포함할 수 있다.

[0022] 또한, 상기 이벤트 절은 상기 조건 절을 평가할 수 있는지 여부를 결정하는데 사용될 수 있다.

[0023] 또한, 상기 조건 절은 어플리케이션 실행과 관련된 규칙(rule)의 설정을 위한 상기 어플리케이션의 상태 정보를 포함할 수 있다.

[0024] 또한, 상기 동작 절은 입력 동작(ingress action), 전송 동작(egress action) 및 적용 프로파일 동작(application profile action)을 포함할 수 있다.

[0025] 또한, 상기 프로파일 동작은 응용 프로그램 계층에 적용되는 보안 기능을 나타내는 컨텐츠 보안 제어, 및 네트워크 공격의 탐지 및 완화를 위한 공격 완화 제어를 포함할 수 있다.

[0026] 또한, 상기 조건 절은 특정 URL(Uniform Resource Locator) 접속과 관련된 규칙(rule)의 설정을 위한 상기 URL의 상태 정보를 포함할 수 있다.

발명의 효과

[0027] 본 발명의 실시 예에 따르면, I2NSF(Interface to Network Security Functions)에서 네트워크 보안 기능(Network Security Functions: NSF)에 직면한 인터페이스를 위한 정보 모델과 보안 서비스를 위한 YANG 데이터 모델을 설계할 수 있다.

[0028] 또한, 본 발명의 실시 예에 따르면, 네트워크 보안 제어, 콘텐츠 보안 제어 및 공격 완화 제어와 같은 세 가지 보안 기능(예: 네트워크 보안 기능)에 대한 특정 정보 모델 및 해당 데이터 모델을 설계할 수 있다.

[0029] 또한, 본 발명의 실시 예에 따르면, I2NSF 컨트롤러는 NSF의 기능을 제어 할 수 있다.

[0030] 본 명세서에서 얻을 수 있는 효과는 이상에서 언급한 효과들로 제한되지 않으며, 언급하지 않은 또 다른 효과들은 아래의 기재로부터 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

[0031] 본 발명에 관한 이해를 돕기 위해 상세한 설명의 일부로 포함되는, 첨부 도면은 본 발명에 대한 실시 예를 제공하고, 상세한 설명과 함께 본 발명의 기술적 특징을 설명한다.

도 1은 본 발명의 일 실시 예에 따른 I2NSF(Interface to Network Security Functions) 시스템을 예시한다.

도 2는 본 발명의 일 실시 예에 따른 I2NSF 시스템의 아키텍처를 예시한다.

도 3은 본 발명이 적용될 수 있는 전체 I2NSF 정보 모델 디자인의 일 예를 나타낸다.

도 4는 본 발명이 적용될 수 있는 네트워크 보안 정보 하위 모델 개요의 일 예를 나타낸다.

도 5은 본 발명이 적용될 수 있는 네트워크 보안 정보 하위 모델의 확장의 일 예를 나타낸다.

도 6는 본 발명이 적용될 수 있는 네트워크 보안 정보 하위 모델 이벤트 클래스의 확장의 일 예를 나타낸다.

도 7는 본 발명이 적용될 수 있는 네트워크 보안 정보 하위 모델 컨디션 클래스의 확장의 일 예를 나타낸다.

도 8은 본 발명이 적용될 수 있는 네트워크 보안 정보 하위 모델 액션의 확장의 일 예를 나타낸다.

도 9은 본 발명이 적용될 수 있는 I2NSF 보안 기능의 상위 레벨 모델의 일 예를 나타낸다.

도 10은 본 발명이 적용될 수 있는 네트워크 보안 기능 정보 모델의 일 예를 나타낸다.

도 11는 본 발명이 적용될 수 있는 공격 완화 기능 정보 모델의 일 예를 나타낸다.

도 12는 본 발명의 일 실시 예에 따른 네트워크 보안 정책 식별을 위한 데이터 모델 구조를 예시한다.

도 13은 본 발명의 일 실시 예에 따른 이벤트 규칙을 위한 데이터 모델 구조를 예시한다.

도 14a 내지 도 14d는 본 발명의 일 실시 예에 따른 컨디션 규칙을 위한 데이터 모델 구조를 예시한다.

도 15는 본 발명의 일 실시 예에 따른 액션 규칙을 위한 데이터 모델 구조를 예시한다.

도 16a 내지 도 18j는 본 발명의 일 실시 예에 따른 I2NSF NSF-Facing Interface의 YANG 데이터 모듈을 예시한다.

도 19a 내지 도 19j는 본 발명의 일 실시 예에 따른 NSF 모니터링을 위한 데이터 모델을 예시한다.

도 20a 내지 도 21i는 본 발명의 일 실시 예에 따른 모니터링을 위한 YANG 데이터 모델을 예시한다.

발명을 실시하기 위한 구체적인 내용

[0032] 이하, 본 발명에 따른 바람직한 실시 형태를 첨부된 도면을 참조하여 상세하게 설명한다. 첨부된 도면과 함께 이하에 개시될 상세한 설명은 본 발명의 예시적인 실시형태를 설명하고자 하는 것이며, 본 발명이 실시될 수 있는 유일한 실시형태를 나타내고자 하는 것이 아니다. 이하의 상세한 설명은 본 발명의 완전한 이해를 제공하기 위해서 구체적 세부사항을 포함한다. 그러나, 당업자는 본 발명이 이러한 구체적 세부사항 없이도 실시될 수 있음을 안다.

[0033] 몇몇 경우, 본 발명의 개념이 모호해지는 것을 피하기 위하여 공지의 구조 및 장치는 생략되거나, 각 구조 및 장치의 핵심기능을 중심으로 한 블록도 형식으로 도시될 수 있다.

[0034] 이하의 설명에서 사용되는 특정 용어들은 본 발명의 이해를 돕기 위해서 제공된 것이며, 이러한 특정 용어의 사용은 본 발명의 기술적 사상을 벗어나지 않는 범위에서 다른 형태로 변경될 수 있다.

[0036] 최근에는, Network Functions Virtualization(NFV)-based security function을 위한 기본 표준 인터페이스가 I2NSF(Interface to Network Security Functions) 워킹 그룹에 의해 개발되고 있다. 이는 인터넷 엔지니어링 태스크 포스(IETF: Internet Engineering Task Force)로 불리는 국제 인터넷 표준 기구의 일부이다.

[0037] I2NSF의 목적은 다수의 보안 솔루션 벤더(security solution vendor)들에 의해 제공되는 이종의(heterogeneous) 네트워크 보안 기능(들)(NSF: network security function)을 위한 표준화된 인터페이스를 정의하기 위함이다.

[0038] I2NSF 아키텍처(architecture)에서, NSF(들)의 관리에 대하여 상세히 고려할 필요 없이(NSF의 관리는 결국 보안 정책의 시행(enforce)을 요구한다), 사용자는 사용자의 네트워크 시스템 내 네트워크 자원을 보호하기 위한 보호 정책을 정의할 수 있다. 또한, 다수의 vendor들로부터 NSF(들)로의 표준화된 인터페이스는 이종의 NSF(들)에 대한 태스크(task)의 설정 및 관리를 단순화할 수 있다.

[0039] 도 1은 본 발명의 일 실시 예에 따른 I2NSF(Interface to Network Security Functions) 시스템을 예시한다.

[0040] 도 1을 참조하면, I2NSF 시스템은 I2NSF 사용자(user), 네트워크 운영 관리 시스템(Network Operator Management System), 개발자 관리 시스템(Developer's Management System) 및/또는 적어도 하나의 NSF(Network Security Function)을 포함한다.

[0041] I2NSF 사용자는 I2NSF 소비자-직면 인터페이스(I2NSF Consumer-Facing Interface)를 통해 네트워크 운영 관리 시스템과 통신한다. 네트워크 운영 관리 시스템은 I2NSF NSF-직면 인터페이스(I2NSF NSF-Facing Interface)를 통해 NFC(들)과 통신한다. 개발자 관리 시스템은 I2NSF 등록 인터페이스(I2NSF Registration Interface)를 통

해 네트워크 운영 관리 시스템과 통신한다. 이하에서는 I2NSF 시스템의 각 컴포넌트(I2NSF 컴포넌트) 및 각 인터페이스(I2NSF 인터페이스)에 설명한다.

[0042] **I2NSF 사용자**

[0043] I2NSF 사용자는 다른 I2NSF 컴포넌트(예컨대, 네트워크 운영 관리 시스템)에서 정보를 요청하거나 및/또는 다른 I2NSF 컴포넌트(예컨대, 개발자 관리 시스템)에 의해 제공되는 서비스(예컨대, 네트워크 보안 서비스)를 사용하는 I2NSF 컴포넌트이다. 예를 들면, I2NSF 사용자는 오버레이 네트워크 관리 시스템, 기업 네트워크 관리자 시스템, 다른 네트워크 도메인 관리자 등일 수 있다.

[0044] 이러한 I2NSF 사용자 컴포넌트에 할당된 역할을 수행하는 대상은 I2NSF 소비자로 지칭될 수 있다. I2NSF 소비자의 예로는, 일정 기간(time span) 동안 패킷의 특정 필드에 기초하여 흐름을 허용, 속도-제한(rate-limit), 또는 거부하기 위해 언더레이 네트워크(underlay network)에 동적으로 알릴 필요가 있는 화상 회의 네트워크 관리자(video-conference network manager), 특정 흐름에 대한 특정 I2NSF 정책을 시행(enforce)하기 위해 제공자 네트워크를 요청할 필요가 있는 기업 네트워크 관리자(Enterprise network administrators) 및 관리 시스템(management systems), 특정 조건의 세트와 일치하는 흐름을 차단하기 위해 언더레이 네트워크에 요청을 전송하는 IoT 관리 시스템(IoT management system)가 포함될 수 있다.

[0045] I2NSF 사용자는 고수준(high-level) 보안 정책(security policy)을 생성 및 배포할 수 있다. 구체적으로 설명하면, I2NSF 사용자는 다양한 악의적인(malicious) 공격으로부터 network 트래픽(traffic)을 보호하기 위하여 네트워크 보안 서비스(network security service)를 이용할 필요가 있다. 이 보안 서비스를 요청하기 위하여, I2NSF 사용자는 자신이 원하는 보안 서비스에 대한 고수준 보안 정책을 생성하고 네트워크 운영 관리 시스템에게 이를 알릴 수 있다.

[0046] 한편, 고수준 보안 정책을 준비하는 과정에서, I2NSF 사용자는 각 NSF(들)를 위한 보안 서비스 또는 보안 정책 규칙 구성(security policy rule configuration)을 실현하기 위하여 요구되는 NSF(들)의 타입에 대하여 고려하지 않을 수 있다.

[0047] 또한, I2NSF 사용자는 네트워크 운영 관리 시스템에 의해 기본적인(underlying) NSF(들) 내에서 발생되는 보안 이벤트(들)(security event)를 통지 받을 수 있다. 이들의 보안 이벤트(들)을 분석함으로써, I2NSF 사용자는 새로운 공격을 식별하고, 새로운 공격에 대처하기 위한 고수준 보안 정책을 업데이트(또는 생성)할 수 있다. 이와 같이, I2NSF 사용자는 보안 정책을 정의, 관리 및 모니터링할 수 있다.

[0048] **네트워크 운영 관리 시스템**

[0049] 네트워크 운영 관리 시스템은 보안 제공, 모니터링 및 기타 동작을 위한 수집(collection) 및 배포(distribution) 지점(point)의 역할을 수행하는 컴포넌트이다. 예를 들면, 네트워크 운영 관리 시스템은 보안 제어기(Security Controller)일 수 있다. 이러한 네트워크 운영 관리 시스템은 네트워크 보안 관리자에 의해 관리될 수 있고, I2NSF 관리 시스템으로 지칭될 수도 있다.

[0050] 네트워크 운영 관리 시스템(또는 보안 제어기)의 주요한 역할 중 하나는 I2NSF 사용자로부터의 고수준 보안 정책(또는 정책 규칙)을 특정 NSF(들)을 위한 저수준(low-level) 보안 정책 규칙으로 번역(translate)하는 것이다. 네트워크 운영 관리 시스템(또는 보안 제어기)은 고수준 보안 정책을 I2NSF 사용자로부터 수신한 후, 우선 I2NSF 사용자에 의해 요구되는 정책을 시행하기 위하여 요구되는 NSF(들)의 타입을 결정할 수 있다. 그리고, 네트워크 운영 관리 시스템(또는 보안 제어기)은 요구되는 각 NSF(들)을 위한 저수준(low-level) 보안 정책을 생성할 수 있다. 결국, 네트워크 운영 관리 시스템(또는 보안 제어기)은 생성된 저수준 보안 정책을 각 NSF(들)에게 설정할 수 있다.

[0051] 또한, 네트워크 운영 관리 시스템(또는 보안 제어기)은 시스템 내 구동 중인 NSF(들)을 모니터링하고, 각 NSF(들)에 대한 다양한 정보(예를 들어, 네트워크 액세스(access) 정보 및 작업로드(workload) 상태 등)를 유지할 수 있다. 또한, 네트워크 운영 관리 시스템(또는 보안 제어기)은 개발자 관리 시스템의 도움을 받아 NSF 인스턴스의 동적인 수명시간(life-cycle) 관리를 통해 NSF 인스턴스(instance)의 풀(pool)을 동적으로 관리할 수 있다.

[0052] **NSF**

[0053] NSF는 보안 관련 서비스를 제공하는 논리적 엔티티(logical entity) 또는 소프트웨어 컴포넌트이다. 예를 들면, NFC는 저수준 보안정책을 수신하고, 이에 기초하여 악의적인 네트워크 트래픽을 감지하고, 이를 차단하거나 완

화할 수 있다. 이를 통해, 네트워크 통신 스트림의 무결성(integrity) 및 기밀성(confidentiality)이 보장될 수 있다.

[0054] **개발자 관리 시스템**

[0055] 개발자 관리 시스템은 다른 I2NSF 컴포넌트(예컨대, I2NSF 사용자, 네트워크 운영 관리 시스템)으로 정보를 보내거나, 및/또는 서비스(예컨대, 네트워크 보안 서비스)를 제공하는 I2NSF 컴포넌트이다. 개발자 관리 시스템은 벤더 관리 시스템(Vendor's Management System)으로 지칭될 수도 있다. 이러한 개발자 관리 시스템에 할당된 역할을 수행하는 대상은 I2NSF 생산자(producer)로 지칭될 수 있다.

[0056] 개발자 관리 시스템은 네트워크 운영 관리 시스템에게 NSF(들)을 제공하는 제3자(third-party) 보안 벤더에 의해 관리될 수 있다. 다양한 보안 벤더의 다수의 개발자 관리 시스템(들)이 존재할 수 있다.

[0057] I2NSF 소비자-직면 인터페이스(간단히, 소비자-직면 인터페이스(CFI))

[0058] CFI는 I2NSF 사용자와 네트워크 운영 관리 시스템 사이에 위치하는, 사용자의 I2NSF 시스템으로의 인터페이스이다. 이렇게 설계됨으로써, 하위(underlying) NSF(들)의 상세한 내용을 숨기고, 사용자에게 NSF(들)의 추상적인 시각(abstract view)만을 제공한다.

[0059] 이 CFI는 주어진 I2NSF 시스템의 상이한 사용자가 관리 도메인 내의 특정 흐름(flow)에 대한 보안 정책을 정의, 관리 및 모니터링할 수 있게 하기 위해 사용될 수 있다. I2NSF 사용자에 의해 생성된 고수준 보안 정책(또는 정책 규칙)은 이 CFI를 통해 네트워크 운영 관리 시스템으로 전달될 수 있다.

[0060] **I2NSF NSF-직면 인터페이스(간단히, NSF-직면 인터페이스(NFI))**

[0061] NFI는 네트워크 운영 관리 시스템(또는 보안 제어기)과 NSF(들) 사이에 위치하는 인터페이스이다.

[0063] *71이 NFI는 하나 이상의 NSF에 의해 시행되는 흐름-기반(flow-based) 보안 정책을 지정하고 모니터링하기 위해 사용될 수 있다. 예를 들면, I2NSF 시스템은 흐름-기반 NSF를 사용할 수 있다. 여기서, 흐름-기반 NSF는 보안 특성을 강화하기 위해 정책의 세트에 따라 네트워크 흐름을 검사하는 NSF이다. 이러한 흐름-기반 NSF에 의한 흐름-기반 보안은 수신된 순서대로 패킷들이 검사되고, 검사 프로세스에 따라 패킷에 대한 수정이 없는 것을 의미한다. 흐름-기반 NSF에 대한 인터페이스는 다음과 같이 분류될 수 있다:

[0064] - NSF 운영 및 관리 인터페이스(NSF Operational and Administrative Interface): NSF의 운영 상태를 프로그래밍하기 위해 I2NSF 관리 시스템에 의해 사용되는 인터페이스 그룹; 이 인터페이스 그룹은 또한 관리 제어 기능을 포함한다. I2NSF 정책 규칙은 일관된 방식으로 이 인터페이스 그룹을 변경하는 한가지 방법을 나타낸다. 어플리케이션 및 I2NSF 컴포넌트가 그들이 송신 및 수신하는 트래픽의 동작을 동적으로 제어할 필요가 있기 때문에, I2NSF 노력(effort)의 대부분이 이 인터페이스 그룹에 집중된다.

[0065] - 모니터링 인터페이스(Monitoring Interface): 하나 이상의 선택된 NSF로부터의 모니터링 정보를 획득하기 위해 I2NSF 관리 시스템에 의해 사용되는 인터페이스 그룹; 이 인터페이스 그룹의 각 인터페이스는 쿼리 또는 리포트 기반 인터페이스일 수 있다. 둘 사이의 차이점은 쿼리 기반 인터페이스는 정보를 획득하기 위해 I2NSF 관리 시스템에 의해 사용되고, 이에 반하여 리포트 기반 인터페이스는 정보를 제공하기 위해 NSF에 의해 사용된다는 것이다. 이 인터페이스 그룹의 기능은 또한 SYSLOG 및 DOTS와 같은 다른 프로토콜에 의해 정의될 수 있다. I2NSF 관리 시스템은 정보의 수신에 기초하여 하나 이상의 동작(action)을 취할 수 있다. 이는 I2NSF 정책 규칙에 의해 지정되어야 한다. 이 인터페이스 그룹은 NSF의 운영 상태를 변경하지 않는다.

[0066] 이와 같이, NFI는 흐름-기반 패러다임을 사용하여 개발될 수 있다. 흐름-기반 NSF의 공통 특성(common trait)은 수신된 패킷의 콘텐츠(예컨대, 헤더/페이로드) 및/또는 컨텍스트(예컨대, 세션 상태 및 인증 상태)에 기초하여 패킷을 처리하는 것이다. 이 특징은 I2NSF 시스템의 동작을 정의하기 위한 요구사항(requirement) 중 하나이다.

[0067] 한편, I2NSF 관리 시스템은 주어진 NSF의 모든 기능들을 사용할 필요가 없으며, 모든 사용 가능한 NSF들을 사용할 필요도 없다. 따라서, 이 추상화(abstraction)는 NSF 특징(feature)을 NSF 시스템에 의해 빌딩 블록(building block)으로 취급될 수 있게 해준다. 그러므로, 개발자는 벤더 및 기술에 독립적인 NSF에 의해 정의되는 보안 기능을 자유롭게 사용할 수 있게 된다.

[0068] **I2NSF 등록 인터페이스(간단히, 등록 인터페이스(RI))**

[0069] RI는 네트워크 운영 관리 시스템 및 개발자 관리 시스템 사이에 위치하는 인터페이스이다. 상이한 벤더에 의해 제공되는 NSF는 상이한 기능(capability)을 가질 수 있다. 따라서, 상이한 벤더에 의해 제공되는 여러 유형의 보안 기능을 이용하는 프로세스를 자동화하기 위해, 벤더가 그들의 NSF의 기능을 정의하기 위한 전용 인터페이스를 가질 필요가 있다. 이러한 전용 인터페이스는 I2NSF 등록 인터페이스(RI)로 지칭될 수 있다.

[0070] NSF의 기능은 미리 구성되거나 또는 I2NSF 등록 인터페이스를 통해 동적으로 검색될 수 있다. 만일 소비자에게 노출되는 새로운 기능이 NSF에 추가된다면, 관심 있는(interested) 관리 및 제어 엔티티가 그것들을 알 수 있도록, 그 새로운 기능의 capability가 I2NSF 등록 인터페이스를 통해 I2NSF 레지스트리(registry)에 등록될 필요가 있다.

[0072] 도 2는 본 발명의 일 실시예에 따른 I2NSF 시스템의 아키텍처를 예시한다. 도 2의 I2NSF 시스템은 도 1의 I2NSF 시스템에 비하여 I2NSF 사용자 및 네트워크 운영 관리 시스템의 구성을 더 구체적으로 나타낸다. 도 2에서는 도 1에서 상술한 설명과 중복된 설명은 생략한다.

[0073] 도 2를 참조하면, I2NSF 시스템은 I2NSF 사용자, 보안 관리 시스템(Security Management System), 및 NSF 인스턴스(instances) 계층을 포함한다. I2NSF 사용자 계층은 어플리케이션 로직(Application Logic), 정책 업데이터(Policy Updater), 및 이벤트 수집기(Event Collector)을 컴포넌트로서 포함한다. 보안 관리 시스템 계층은 보안 제어기 및 개발자 관리 시스템을 포함한다. 보안 관리 시스템 계층의 보안 제어기는 보안 정책 관리자(Security policy manager) 및 NSF 기능 관리자(NSF capability manager)를 컴포넌트로서 포함한다.

[0074] I2NSF 사용자 계층은 소비자-직면 인터페이스를 통해 보안 관리 시스템 계층과 통신한다. 예를 들면, I2NSF 사용자 계층의 정책 업데이터 및 이벤트 수집기는 소비자-직면 인터페이스를 통해 보안 관리 시스템 계층의 보안 제어기와 통신한다. 또한, 보안 관리 시스템 계층은 NSF-직면 인터페이스를 통해 NSF 인스턴스 계층과 통신한다. 예를 들면, 보안 관리 시스템 계층의 보안 제어기는 NSF-직면 인터페이스를 통해 NSF 인스턴스 계층의 NSF 인스턴스(들)과 통신한다. 또한, 보안 관리 시스템 계층의 개발자 관리 시스템은 등록 인터페이스를 통해 보안 관리 시스템 계층의 보안 제어기와 통신한다.

[0075] 도 2의 I2NSF 사용자 계층, 보안 관리 시스템 계층의 보안 제어기 컴포넌트, 보안 관리 시스템 계층의 개발자 관리 시스템 컴포넌트 및 NSF 인스턴스 계층은 각각 도 1의 I2NSF 사용자 컴포넌트, 네트워크 운영 관리 시스템 컴포넌트, 개발자 관리 시스템 컴포넌트 및 NSF 컴포넌트에 대응된다. 또한, 도 2의 소비자-직면 인터페이스, NSF-직면 인터페이스 및 등록 인터페이스는 도 1의 소비자-직면 인터페이스, NSF-직면 인터페이스 및 등록 인터페이스에 대응된다. 이하에서는, 각 계층에 포함된 새로 정의된 컴포넌트들에 대하여 설명한다.

[0076] **I2NSF 사용자**

[0077] 상술한 것처럼, I2NSF 사용자 계층은 다음 3 개의 컴포넌트를 포함한다: 어플리케이션 로직(Application Logic), 정책 업데이터(Policy Updater), 및 이벤트 수집기(Event Collector). 각각의 역할 및 동작을 설명하면 다음과 같다.

[0078] 어플리케이션 로직은 고수준 보안 정책을 생성하는 컴포넌트이다. 이를 위해, 어플리케이션 로직은 이벤트 수집기로부터 고수준 정책을 업데이트(또는 생성)하기 위한 이벤트를 수신하고, 수집된 이벤트에 기초하여 고수준 정책을 업데이트(또는 생성)한다. 그 이후에, 고수준 정책은 보안 제어기로 배포하기 위해 정책 업데이터로 보내진다. 고수준 정책을 업데이트(또는 생성)하기 위해, 이벤트 수집기는 보안 수집기에 의해 보내진 이벤트를 수신하고, 그들을 어플리케이션 로직으로 보낸다. 이 피드백에 기초하여, 어플리케이션 로직은 고수준 보안 정책을 업데이트(또는 생성)할 수 있다.

[0079] 도 2에서는, 어플리케이션 로직, 정책 업데이터 및 이벤트 수집기를 각각 별도의 구성으로 도시하고 있으나, 본 발명의 이에 한정되지 않는다. 다시 말해, 각각은 논리적인 컴포넌트로서, I2NSF 시스템에서 하나 또는 2 개의 컴포넌트로 구현될 수도 있다.

[0081] ***보안 관리 시스템**

[0082] 상술한 것처럼, 보안 관리 시스템 계층의 보안 제어기는 보안 정책 관리자(Security policy manager) 및 NSF 기능 관리자(NSF capability manager)와 같은 2개의 컴포넌트를 포함한다

[0083] 보안 정책 관리자는 CFI를 통해 정책 업데이터로부터 고수준 정책을 수신하고, 이 정책을 여러 저수준 정책으로 맵핑할 수 있다. 이 저수준 정책은 NSF 기능 관리자에 등록된 주어진 NSF 기능과 관련된다. 또한, 보안 정책 관리자는 이 정책을 NFI를 통해 NSF(들)로 전달할 수 있다.

[0084] NSF 기능 관리자는 주어진 NSF 기능과 관련된 저수준 정책을 생성하기 위해, 개발자 관리 시스템에 의해 등록된 NSF의 기능을 지정하고, 그것을 보안 정책 관리자와 공유할 수 있다. 새로운 NSF가 등록될 때마다, NSF 기능 관리자는 등록 인터페이스를 통해 NSF 기능 관리자의 관리 테이블에 NSF의 기능을 등록하도록 개발자 관리 시스템에 요청할 수 있다. 개발자 관리 시스템은 새로운 NSF의 기능을 NSF 기능 관리자로 등록하기 위한 보안 관리 시스템의 다른 부분에 해당한다.

[0085] 도 2에서는, 보안 정책 관리자 및 NSF 기능 관리자를 각각 별도의 구성으로 도시하고 있으나, 본 발명의 이에 한정되지 않는다. 다시 말해, 각각은 논리적인 컴포넌트로서, I2NSF 시스템에서 하나의 컴포넌트로 구현될 수도 있다.

[0086] **NSF 인스턴스(NSF Instances)**

[0087] 도 2에 도시된 것처럼, NSF 인스턴스 계층은 NSF들을 포함한다. 이때, 모든 NSF들은 이 NSF 인스턴스 계층에 위치된다. 한편, 고수준 정책을 저수준 정책에 맵핑한 후에, 보안 정책 관리자는 NFI를 통해 정책을 NSF(들)로 전달한다. 이 경우, NSF는 수신된 저수준 보안 정책에 기초하여 악의적인 네트워크 트래픽을 감지하고, 이를 차단하거나 완화할 수 있다.

[0088] 가상화 시스템의 신속한 개발을 위해서는 다양한 시나리오에서 고급 보안 기능이 필요하다(예를 들면, 엔터프라이즈 네트워크의 네트워크 장치, 모바일 네트워크의 사용자 장비, 인터넷의 장치 또는 거주자 액세스 사용자 등).

[0089] 여러 보안 업체에서 생산한 NSF는 고객에게 다양한 보안 기능을 제공할 수 있다. 즉, NSF는 물리적 또는 가상 기능으로 구현되었는지 여부와 관계없이 여러 NSF가 함께 결합되어 주어진 네트워크 트래픽에 대한 보안 서비스를 제공 할 수 있다.

[0090] 보안 기능은 보안 정책 시행 목적으로 사용할 수 있는 일련의 네트워크의 보안과 관련된 기능을 말한다. 보안 기능은 실제 구현되는 보안 제어 메커니즘과는 독립적이며, 모든 NSF는 NSF에서 제공할 수 있는 기능들의 세트가 등록되어 있다.

[0091] 보안 기능(security capability)은 특정 NSF가 제공하는 보안 기능을 모호하지 않게 설명함으로써 맞춤형 보안 보호를 정의 할 수 있는 기능 명세(capability specification)를 제공한다. 또한, 보안 기능을 통해 보안 기능의 공급 업체의 중립적인 방식으로 설명 할 수 있다.

[0092] 즉, 네트워크를 설계할 때 특정 제품을 언급할 필요가 없으며, 기능별로 특징이 고려될 수 있다.

[0093] 앞에서 살펴본 바와 같이 보안 정책 제공에 사용될 수 있는 I2NSF 인터페이스는 아래와 같이 두 가지 유형이 존재할 수 있다.

[0095] *101- I2NSF 사용자와 응용 프로그램 간의 인터페이스 및 보안 컨트롤러 (Consumer-Facing Interface): NSF 데이터 및 서비스 사용자와 네트워크 운영 관리시스템(또는 보안 제어기) 사이에 통신 채널을 제공하는 소비자 지향 인터페이스.

[0096] I2NSF Consumer-Facing Interface는 보안 정보가 다양한 애플리케이션(예를 들면: OpenStack 또는 다양한 BSS / OSS 구성 요소)과 보안 컨트롤러간의 교환에 사용될 수 있다. Consumer-Facing Interface의 설계 목표는 보안 서비스의 스펙을 구현과 분리하는데 있다.

[0097] - NSF 간의 인터페이스(예를 들면: 방화벽, 침입 방지 또는 안티 바이러스) 및 보안 컨트롤러 (NSF-Facing Interface): NSF-Facing Interface는 보안 관리 체계를 NSF 집합과 여러 가지 구현에서 분리하는 데 사용되며 NSF가 구현되는 방식(예를 들면: 가상 머신 또는 실제 appliances 등)에서 독립적이다.

[0098] 이하, 연관된 I2NSF 정책 객체와 함께 네트워크 보안, 콘텐츠 보안 및 공격 완화 기능에 대한 객체 지향 정보 모델에 대해 살펴보도록 한다.

[0099] 본 발명에서 정보 모델에 사용되는 용어는 다음과 같이 정의될 수 있다

[0100] AAA: Access control, Authorization, Authentication

[0101] ACL: Access Control List

[0102] (D)DoD: (Distributed) Denial of Service (attack)

[0103] ECA: Event-Condition-Action

[0104] FMR: First Matching Rule (resolution strategy)

[0105] FW: Firewall

[0106] GNSF: Generic Network Security Function

[0107] HTTP: HyperText Transfer Protocol

[0108] I2NSF: Interface to Network Security Functions

[0109] IPS: Intrusion Prevention System

[0110] LMR: Last Matching Rule (resolution strategy)

[0111] MIME: Multipurpose Internet Mail Extensions

[0112] NAT: Network Address Translation

[0113] NSF: Network Security Function

[0114] RPC: Remote Procedure Call

[0115] SMA: String Matching Algorithm

[0116] URL: Uniform Resource Locator

[0117] VPN: Virtual Private Network

[0119] **정보 모델 설계**

[0120] 기능 정보 모델(Capability Information Model)의 설계의 출발점은 보안 기능의 유형을 분류하는 것이다. 예를 들어, "IPS", "안티 바이러스" 및 "VPN 집중 장치"와 같은 보안 기능의 유형을 분류하는 것이다.

[0121] 또는, "패킷 필터"는 다양한 조건(예를 들면: 발신 및 수신 IP 주소, 발신 및 수신 포트 및 IP 프로토콜 유형 필드 등)에 따라 패킷 전달을 허용하거나 거부 할 수 있는 저장 장치로 분류될 수 있다.

[0123] 그러나, 상태 기반 방화벽이나 응용 프로그램 계층 필터와 같은 다른 장치의 경우 더 많은 정보가 필요하다. 이러한 장치는 패킷이나 통신을 필터링하지만 패킷과 통신들을 카테고리화하고 유지하는 상태에서 차이가 있다.

[0124] 아날로그적 고려사항은 채널 보호 프로토콜들에서 고려될 수 있다. 여기서 채널 보호 프로토콜들은 비대칭 암호로 협상될 수 있는 대칭 알고리즘을 통해 패킷을 보호할 수 있으며, 서로 다른 계층에서 작동하고 서로 다른 알고리즘과 프로토콜을 지원할 수 있다.

[0125] 안전한 보호를 위해 이러한 프로토콜은 무결성, 선택적으로 기밀성, anti-reply 보호 및 피어 인증이 적용되어야 한다.

[0127] **기능 정보 모델 오버뷰(Capability Information Model Overview)**

[0128] 기능 정보 모델은 NSF의 자동 관리를 위한 토대를 제공하는 보안 기능 모델을 정의한다. 기능 정보 모델은 보안 컨트롤러가 NSF를 적절하게 식별 및 관리 할 수 있도록 하고, NSF가 기능들을 올바른 방법으로 사용할 수 있도록 적절하게 선언하는 것을 허용하는 것도 포함한다.

[0130] 보안을 위한 몇 가지 기본 설계 원칙 및 이를 관리해야 하는 시스템은 다음과 같다.

[0132] - 독립성(Independence): 각 보안 기능은 다른 기능에 최소한의 중첩 또는 종속성을 갖는 독립적인 기능이어야 한다. 이를 통해 각 보안 기능을 자유롭게 사용 및 조합 할 수 있다. 더 중요한 것은, 하나의 기능으로의 변경이 다른 기능에 영향을 미치지 않는다는 것이다.

[0133] 이것은 Single Responsibility Principle [Martin] [OODSRP]을 따른다.

[0134] - 추상성(Abstraction): 각 기능은 벤더 독립적인 방식으로 정의되어야 하며 잘 알려진 인터페이스와 연결되어 처리 결과를 기술하고 보고할 수 있는 표준화 된 기능을 제공해야 한다. 따라서, 다중 공급 벤더와의 상호 운용성이 향상될 수 있다.

[0135] - 자동화(Automation): 시스템은 보안 기능(즉, 사용자 개입없이)을 자동 검색, 자동 협상 및 자동 업데이트 할 수 있어야 한다. 이러한 자동화 기능은 다수의 NSF를 관리하는 데 특히 유용하다.

[0136] 채택 된 보안 체계에 대한 스마트 서비스(예를 들면: 분석, 정제, 기능 추론 및 최적화)를 추가하는 것은 필수적이다. 이러한 기능은 Observer Pattern [OODOP], Mediator Pattern [OODMP] 및 Message Exchange Patterns [Hohpe]와 같은 많은 디자인 패턴에서 지원된다.

[0137] - 확장성: 관리 시스템에는 scale up/down 또는 scale in/out 기능이 있어야 한다. 따라서, 이러한 확장성으로 인하여 변경 가능한 네트워크 트래픽 또는 서비스 요청에서 파생된 다양한 성능 요구 사항을 충족 할 수 있다. 또한, 확장성의 영향을 받는 보안 기능은 보안 컨트롤러에 통계보고를 지원해야 스케일링을 호출해야 하는지 여부를 결정하는 데 도움이 될 수 있다.

[0139] 위의 원칙에 따라 표준 인터페이스를 갖춘 추상 및 벤더 중립 기능 집합이 정의될 수 있다. 이것은 주어진 시간에 필요한 NSF 세트를 사용할 수 있게 해주는 Capability 모델과 사용된 NSF 세트에 의해 제공되는 보안의 모호하지 않도록 정의를 제공한다.

[0140] 보안 컨트롤러는 사용자 및 응용 프로그램의 요구 사항을 현재 사용할 수 있는 기능 집합과 비교하여 해당 요구 사항을 충족하는데 필요한 NSF를 선택한다.

[0141] 또한, NSF에 의해 알려지지 않은 위협(예를 들어, zero-day exploits 및 unknown malware)이 보고 될 때, 새로운 기능이 생성될 수 있고 및/또는 기존의 기능이 업데이트 될 수 있다(예를 들어, 그의 서명 및 알고리즘을 업데이트함으로써). 그 결과 새로운 위협에 대처하기 위해 기존의 NSF를 강화(및/또는 새로운 NSF를 생성)하게 된다.

[0142] 새로운 기능은 중앙 리포지토리에 전송되어 저장되거나 벤더의 로컬 리포지토리에 개별적으로 저장 될 수 있다. 두 경우 모두 표준 인터페이스가 업데이트 프로세스가 용이하게 수행되도록 한다.

[0143] **ECA 정책 모델 오버뷰(ECA Policy Model Overview)**

[0144] "Event-Condition-Action"(ECA) 정책 모델은 I2NSF 정책 규칙의 설계를 위한 기초로 사용된다. 이때, I2NSF 정책과 관련된 용어는 아래와 같이 정의될 수 있다([I-D.draft-ietf-i2nsf-terminology] 참조):

[0146] *- 이벤트: 이벤트는 관리되는 시스템이 변경될 때 및/또는 관리되는 시스템의 환경에서 중요한 시점에 발생한다. 이벤트는 I2NSF 정책 규칙의 컨텍스트에서 사용될 때 I2NSF 정책 규칙의 조건 절을 평가할 수 있는지 여부를 결정하는 데 사용될 수 있다. I2NSF 이벤트의 예로는 시간 및 사용자 동작(예를 들면: 로그온, 로그 오프 및 ACL을 위반하는 동작)이 있을 수 있다.

[0147] - 조건(Condition): 조건은 알려진 속성, 특징 및/또는 값의 세트와 비교될 속성, 기능 및/또는 값의 집합으로 정의되어 그(명령형) I2NSF 정책 규칙을 실행하거나 실행하지 않을 수 있다. I2NSF 조건의 예에는 패킷 또는 흐름의 일치하는 속성과 NSF의 내부 상태를 원하는 상태와 비교하는 것이 포함될 수 있다.

[0148] - 동작(Action): 동작은 이벤트 및 조건 절이 충족될 때 흐름 기반 NSF의 측면을 제어하고 모니터링하는데 사용된다. NSF는 다양한 액션을 실행하여 보안 기능을 제공한다. I2NSF 작업의 예에는 침입 탐지 및 / 또는 보호,

웹 및 플로우 필터링, 패킷 및 플로우에 대한 심층 패킷 검사 제공이 포함될 수 있다.

[0150] I2NSF 정책 규칙은 Event 절, Condition 절 및 Action 절의 세 가지 Boolean 절로 구성된다.

[0151] Boolean 절은 TRUE 또는 FALSE로 평가되는 논리문을 의미하며, 하나 이상의 용어로 구성 될 수 있습니다.

[0152] 두 개 이상의 용어가 있는 경우 Boolean 절은 논리 연결 요소(즉, AND, OR 및 NOT)를 사용하여 용어를 연결한다. 이때, 논리적 연결 요소는 아래의 표 1과 같은 의미를 가질 수 있다.

표 1

[0153]

```
IF <event-clause> is TRUE
        IF <condition-clause> is TRUE
            THEN execute <action-clause>
        END-IF
    END-IF
```

[0154] 기술적으로 "정책 규칙"은 실제로 메타 데이터뿐 아니라 앞에서 설명한 “이벤트”, “동작” 및 “조건”을 집계하는 컨테이너 역할을 수행할 수 있다.

[0155] 앞에서 설명한 ECA 정책 모델은 매우 일반적이며 쉽게 확장 할 수 있으며 일반 보안 기능 구현을 제한 할 수 있는 잠재적 제약을 피할 수 있다.

[0156] **외부 정보 모델과의 관계**

[0157] 도 3은 본 발명이 적용될 수 있는 전체 I2NSF 정보 모델 디자인의 일 예를 나타낸다.

[0158] I2NSF NSF-Facing Interface는 NSF의 기능을 사용하여 NSF를 선택 및 관리하며, 이는 아래와 같은 접근법을 이용하여 수행된다.

[0159] 1) 각 NSF는 "참여"할 때 관리 시스템에 기능을 등록하므로 관리 시스템에서 해당 기능을 사용할 수 있다.

[0160] 2) 보안 컨트롤러는 관리하는 모든 사용 가능한 NSF에서 보안 서비스의 요구 사항을 충족시키는 데 필요한 기능 집합을 선택한다.

[0161] 3) 보안 컨트롤러는 Capability 정보 모델을 사용하여 선택한 기능을 공급 업체와 독립적인 NSF로 일치시킨다.

[0162] 4) 보안 컨트롤러는 위의 정보를 가져 와서 기능 정보 모델의 하나 이상의 데이터 모델을 생성 또는 사용하여 NSF를 관리합니다.

[0163] 5) 제어 및 모니터링을 시작할 수 있습니다.

[0164] 이러한 접근법은 외부 정보 모델이 ECA 정책 규칙 및 그 구성 요소(예를 들면: 이벤트, 조건 및 조치 객체 등)의 개념을 정의하는 데 사용된다고 가정할 수 있다. 이를 통해 외부 정보 모델로부터 I2NSF 정책 규칙을 하위 클래스로 분류 할 수 있다(I-D.draft-ietf-i2nsf-terminology 참조).

[0165] 본 발명에서 데이터 모델은 데이터의 저장소, 데이터 정의 언어, 쿼리 언어, 구현 언어 및 프로토콜에 의존하는 형식으로 환경에 대한 관심의 컨셉을 나타낸 것이다.

[0167] *

[0168] *171또한, 정보 모델은 데이터 저장소, 데이터 정의 언어, 쿼리 언어, 구현 언어 및 프로토콜과 독립적인 형태로 환경에 대한 관심 컨셉을 나타낸 것이다.

[0169] 기능은 클래스(예를 들면: 공통된 특성 및 행동 집합을 나타내는 객체의 집합)로 정의될 수 있다(I-D.draft-ietf-supa-generic-policy-info-model 참조).

[0170] 각 기능은 시스템의 다른 모든 객체와 구별되는 하나 이상의 모델 요소(예를 들면: 속성, 메소드 또는 관계)로 구성될 수 있다. 기능은 일반적으로 일종의 메타 데이터(즉, 객체의 행동을 설명 및 / 또는 처방하는 정보)이다.

[0171] 따라서, 각 기능은 외부 정보 모델이 메타 데이터를 정의하는데 사용될 수 있다(클래스 계층 구조의 형태가 바람직함). 따라서, 기능들은 외부 메타 데이터 모델에서 하위 클래스로 분류될 수 있다.

[0172] 기능 하위 모델은 NSF가 포함된 장치의 유형 및 공급 업체와 독립적인 특정 보안 기능 세트를 광고, 생성, 선택 및 관리하는데 사용된다.

[0173] 즉, NSF-Facing Interface의 사용자는 NFC가 가상화 되거나 호스팅되는지, NSF 공급 업체가 누구인지, NSF가 통신하는 엔티티 세트(예를 들면, 방화벽 또는 IPS)를 고려하지 않는다.

[0174] 대신 사용자는 NSF가 가지고 있는 패킷 필터링이나 딥 패킷 검사와 같은 기능 세트만을 고려한다.

[0175] 이러한 전체 ISNSF 정보 모델의 설계는 도 3과 같다.

[0176] 도 3에 도시된 외부 모델은 모두 SUPA 정보 모델을 기반으로 할 수 있다(I-D.draft-ietf-supa-generic-policy-info-model 참조). 기능 하위 모델의 클래스는 외부 메타 데이터 정보 모델에서 메타 데이터 집계(AggregatesMetadata)의 집합을 이어받는다.

[0177] 도 3에 도시된 외부 ECA 정보 모델은 일반 ECA 정책 규칙을 나타내는 최소한의 클래스 집합과 일반 ECA 정책 규칙에 의해 집계 될 수 있는 이벤트, 조건 및 동작을 나타내는 클래스 집합을 제공한다.

[0178] 이를 통해, I2NSF는 이러한 일반 모델을 다른 목적으로 재사용 할 수 있을 뿐만 아니라 I2NSF 관련 개념을 표현하기 위해 새로운 하위 클래스를 생성하거나 속성 및 관계를 추가 할 수 있다.

[0179] 본 발명에서 외부 ECA 정보 모델은 메타 데이터를 수집하는 기능을 가지고 있다고 가정한다. 기능들은 외부 메타 데이터 정보 모델의 적절한 클래스에서 하위 클래스로 분류될 수 있다.

[0180] 이는 ECA 개체가 메타 데이터와 기존의 집계를 사용하여 메타 데이터를 적절한 ECA 개체에 추가할 수 있게 한다.

[0181] 이하 정보 모델의 각 부분에 대해서 살펴보도록 한다.

[0182] I2NSF 기능 정보 모델: 운영 이론(I2NSF Capability Information Model: Theory of Operation)

[0183] 기능은 일반적으로 호출할 수 있는 NSF 함수를 나타내는 데 사용된다. 기능은 객체이므로 I2NSF ECA 정책 규칙의 이벤트, 조건 및/또는 액션을 설명하는 절에서 사용할 수 있다.

[0184] I2NSF 기능 정보 모델은 사전 정의된 메타 데이터 모델을 구체화한다. I2NSF 기능의 적용은 기능 집합을 사용, 관리 또는 조작하는 방법을 정의하는 사전 정의된 ECA 정책 규칙 정보 모델을 수정함으로써 수행될 수 있다. 이러한 접근법에서 I2NSF 정책 규칙은 이벤트 절, 조건 절 및 작업 절의 세 가지 절로 구성된 컨테이너 역할을 수행할 수 있다.

[0185] I2NSF 정책 엔진이 일련의 이벤트를 수신하면 해당 이벤트를 활성 ECA 정책 규칙의 이벤트와 일치시킨다. 이벤트가 일치하면 일치하는 I2NSF 정책 규칙의 조건절의 평가를 트리거한다. 조건 절이 평가되고, 이것이 일치하는 경우, 일치하는 I2NSF 정책 규칙에 있는 일련의 행동이 실행될 수 있다.

[0186] **초기 NSFs 기능 카테고리(Initial NSFs Capability Categories)**

[0187] 이하, 네트워크 보안, 콘텐츠 보안 및 공격 완화의 세 가지 일반적인 기능에 대해서 살펴본다. 본 발명에서 살펴보는 특정 카테고리 내의 카테고리 수와 기능 유형은 모두 확장될 수 있다.

[0188] **네트워크 보안 기능(Network Security Capabilities)**

[0189] 네트워크 보안은 미리 정의 된 보안 정책을 사용하여 네트워크 트래픽을 검사하고 처리하는 방법을 설명하기 위한 카테고리이다.

[0190] 검사 부분은 직접적으로 또는 패킷이 연관된 흐름의 맥락에서 네트워크를 통과하는 패킷을 검사하는 패킷 처리

엔진일 수 있다. 패킷 처리의 관점에서 볼 때 구현할 수 있는 패킷 헤더 및/또는 페이로드의 내용, 유지할 수 있는 다양한 흐름 및 컨텍스트 상태, 패킷 또는 흐름에 적용 할 수 있는 동작이 구현에 따라 달라질 수 있다.

[0191] **콘텐츠 보안 기능(Content Security Capabilities)**

[0192] 콘텐츠 보안은 응용 프로그램 계층에 적용되는 보안 기능의 또 다른 카테고리이다. 예를 들어, 응용 프로그램 계층에서 전달되는 트래픽 내용을 분석하여 콘텐츠 보안 기능을 사용함으로써 필요한 다양한 보안 기능을 식별할 수 있다.

[0193] 여기에는 침입에 대한 방어, 바이러스 검사, 악의적 인 URL 또는 정크 메일 필터링, 불법적 인 웹 액세스 차단 또는 악의적인 데이터 검색 방지가 포함될 수 있다.

[0194] 일반적으로 콘텐츠 보안의 각 위협 유형에는 고유한 특성 집합이 있으며 해당 유형의 콘텐츠에 고유한 메서드 집합을 사용하여 처리해야 한다. 따라서 이러한 기능은 고유한 콘텐츠 별 보안 기능을 특징으로 한다.

[0195] **공격 완화 기능(Attack Mitigation Capabilities)**

[0196] 공격 완화 기능은 다양한 유형의 네트워크 공격을 탐지하고 완화하는데 사용된다. 오늘날 일반적인 네트워크 공격은 아래와 같이 정의될 수 있다.

[0197] - DDoS 공격:

[0198] 네트워크 계층 DDoS 공격: SYN flood, UDP flood, ICMP flood, IP fragment flood, IPv6 routing header attack 및 IPv6 duplicate address detection 공격을 예로 들 수 있다.

[0199] 응용 프로그램 계층 DDoS 공격: 예를 들어 HTTP flood, https flood, 캐시 우회 HTTP floods, WordPress XML RPC floods 및 ssl DDoS가 있습니다.

[0200] - 단일 패킷 공격:

[0201] 스캐닝(scanning) 및 스니핑(sniffing) 공격: IP 스윕(sweep), 포트 스캐닝 등

[0202] 잘못된 패킷 공격: Ping of Death, Teardrop 등

[0203] 특별 패킷 공격: 특대 ICMP, Tracert, IP 타임 스탬프 옵션 패킷 등

[0204] 각 유형의 네트워크 공격에는 고유한 네트워크 동작 및 패킷/흐름 특성이 있다. 따라서, 각 유형의 공격에는 탐지 및 완화를 위해 기능 집합으로 알리는 특수 보안 기능이 필요하다. 이러한 보안 범주의 구현 및 관리 공격 완화 제어 기능은 콘텐츠 보안 제어 범주와 매우 유사할 수 있다.

[0205] 네트워크 보안 기능을 위한 정보 하부 모델(Information Sub-Model for Network Security Capabilities)

[0206] 기능 정보 하위 모델의 목적은 기능의 개념을 정의하고 기능들을 적절한 객체에 집계 할 수 있게 하는 것이다. 이하, 네트워크 보안, 콘텐츠 보안 및 공격 완화 기능 하위 모델에 대해 설명하도록 한다.

[0207] 네트워크 보안을 위한 정보 하위 모델(Information Sub-Model for Network Security)

[0208] 도 4는 본 발명이 적용될 수 있는 네트워크 보안 정보 하위 모델 개요의 일 예를 나타낸다.

[0209] 네트워크 보안 정보 하위 모델의 목적은 네트워크 트래픽을 정의하는 방법을 정의하고 하나 이상의 네트워크 보안 기능을 트래픽에 적용해야 하는지 여부를 결정하기 위한 것이다.

[0210] 도 4에서 ECA정책규칙은 이벤트, 조건 및 동작 객체와 함께 외부 ECA 정보 모델에 정의되어 있다. 네트워크 보안 하위 모델은 보안 관련 ECA 정책 규칙 및 (일반)이벤트, 조건 및 조치 개체에 대한 확장을 정의하기 위해 이러한 모든 개체를 확장할 수 있다.

[0211] I2NSF 정책 규칙은 이벤트 조건 동작 (ECA) 형식의 특수한 유형의 정책 규칙이다. 정책 규칙, 정책 규칙의 구성 요소(예를 들면: 이벤트, 조건, 작업 및 해결 정책, 기본 작업 및 외부 데이터와 같은 일부 확장자) 및 선택적으로 메타 데이터로 구성될 수 있으며, NSF를 통한 단방향 및 양방향 트래픽에 모두 적용될 수 있다.

[0212] **네트워크 보안 정책 규칙 확장(Network Security Policy Rule Extensions)**

[0213] 도 5은 본 발명이 적용될 수 있는 네트워크 보안 정보 하위 모델의 확장의 일 예를 나타낸다.

[0214] 도 5는 네트워크 보안 정보 하위 모델에 포함된 ECA 정책 규칙 하위 클래스의 보다 자세한 디자인의 일 예를 나

타낸다. 이는 보다 구체적인 네트워크 보안 정책들이 SecurityECAPolicyRule 클래스에서 이전되고 확장되는 방법을 보여준다.

[0215] 다음과 같은 패턴의 클래스 설계를 따르면 새로운 종류의 특정 네트워크 보안 정책을 생성 할 수 있다.

[0216] SecurityECAPolicyRule은 I2NSF ECA 정책 규칙 계층의 맨 위에 위치한다. 이 규칙은 (외부) 일반 ECA 정책 규칙에서 이전되며 보안 관련 ECA 정책 규칙을 추가하기 위한 이러한 일반 ECA 정책 규칙의 특수화를 나타낸다.

[0217] SecurityECAPolicyRule은 슈퍼 클래스에 정의된 모든 속성, 메소드 및 관계를 포함하며 네트워크 보안에 필요한 추가 개념을 추가한다.

[0218] 6 개의 SecurityECAPolicyRule 서브 클래스는 SecurityECAPolicyRule 클래스를 확장하여 6 가지 유형의 Network Security ECA Policy Rules를 나타낸다. (외부) 일반 ECAPolicyRule 클래스는 설명 및 기타 필요한 정보뿐만 아니라 고유한 객체 ID와 같은 속성의 형태로 기본 정보를 정의할 수 있다.

[0219] 네트워크 보안 정책 규칙 동작(Network Security Policy Rule Operation)

[0220] 네트워크 보안 정책은 위에서 설명한 정보 모델로 구성된 하나 이상의 ECA 정책 규칙으로 구성된다. 이벤트 및 조건 절이 변경되지 않은 간단한 경우에는 한 정책 규칙의 작업이 다른 정책 규칙에서 추가 네트워크 보안 작업을 호출 할 수 있다. 네트워크 보안 정책은 다음과 같이 트래픽을 검사하고 기본 처리를 수행한다.

[0221] 1. NSF는 주어진 SecurityECAPolicyRule의 이벤트 절을 평가한다(도 3에 도시된 바와 같이 보안에 일반적이거나 특정 일 수 있음). 보안 이벤트 객체를 사용하여 아래 설명할 평가의 전부 또는 일부를 수행 할 수 있다.

[0222] Event 절이 TRUE로 평가되면 이 SecurityECAPolicyRule의 조건 절이 평가된다. 그렇지 않으면 SecurityECAPolicyRule의 실행이 중지되고 다음 SecurityECAPolicyRule 이 평가될 수 있다.

[0223] 2. 이후, 조건 절이 평가될 수 있다. 보안 요구 사항 객체를 사용하여 아래에서 설명할 평가의 전부 또는 일부가 수행될 수 있다. 조건 절이 TRUE로 평가되면 SecurityECAPolicyRule과 "일치"하는 것으로 정의된다. 그렇지 않으면 SecurityECAPolicyRule의 실행이 중지되고 다음 SecurityECAPolicyRule이 평가될 수 있다.

[0224] 3. 실행될 일련의 작업이 검색되고, 해결 전략이 실행 순서를 정의하는 데 사용된다. Step 3)에서 프로세스에는 SecurityECAPolicyRule과 관련된 선택적 외부 데이터 사용이 포함될 수 있다.

[0225] 4. 실행은 다음 세 가지 형식 중 하나를 취합니다.

[0226] a. 하나 이상의 행동이 선택되면, NSF는 해결 전략에 의해 정의된 행동을 수행 할 수 있다. 예를 들어, 해결 전략은 단일 액션 (예를 들면: FMR 또는 LMR) 만 실행되도록 허용하거나 모든 액션이 실행되도록 허용 할 수 있다(선택적으로 또는 특정 순서로).

[0227] 이러한 경우와 다른 경우 NSF 기능은 실행 방법을 명확하게 정의해야 한다.

[0228] 보안 액션 객체를 사용하여 아래에서 설명하는 실행의 전부 또는 일부를 수행 할 수 있습니다. 기본 액션이 허가 또는 미러인 경우 NSF는 먼저 해당 기능을 수행 한 다음 특정 보안 기능이 규칙에서 참조되는지 여부를 확인한다. 만약 “Yes” 인 경우, Step 5로 이동한다. No인 경우, 트래픽이 허용된다.

[0229] b. 선택된 동작이 없고 기본 동작이 있는 경우, 기본 동작이 수행될 수 있다. 그렇지 않으면 아무 작업도 수행되지 않는다.

[0230] c. 그렇지 않으면 트래픽이 거부됩니다.

[0232] 5. SecurityECAPolicyRule의 동작 집합에서 다른 보안 기능(예를 들면: 바이러스 백신 또는 IPS 프로파일 NSF가 암시하는 조건 및 / 또는 동작)이 참조되는 경우 NSF는 참조 된 보안 기능을 사용하도록 구성 할 수 있다 (예를 들면: check 조건 또는 행동 집행).

[0233] 이후, 실행이 종료될 수 있다.

[0234] **네트워크 보안 이벤트 하위 서브 모델(Network Security Event Sub-Model)**

[0235] 도 6는 본 발명이 적용될 수 있는 네트워크 보안 정보 하위 모델 이벤트 클래스의 확장의 일 예를 나타낸다.

[0236] 도 6은 네트워크 보안 정보 하위모델에 포함된 이벤트 하위 클래스의 디자인의 일 예를 나타낸다.

[0237] 도 6의 네 가지 Event 클래스는 (외부) 일반 Event 클래스를 확장하여 네트워크 보안에서 중요한 이벤트를 나타낸다. (외부) 일반 Event 클래스는 고유 이벤트 ID, 설명 및 이벤트가 발생한 날짜 및 시간과 같은 속성 양식의 기본 이벤트 정보를 정의한다고 가정할 수 있다.

[0238] **네트워크 보안 조건 하위 서브 모델(Network Security Condition Sub-Model)**

[0239] 도 7는 본 발명이 적용될 수 있는 네트워크 보안 정보 하위 모델 조건 클래스의 확장의 일 예를 나타낸다.

[0240] 도 7은 네트워크 보안 정보 하위 모델에 포함된 조건 하위 클래스의 보다 상세한 디자인을 나타낸다.

[0241] 도 7에 표시된 여섯 가지 조건 클래스는 (외부) 일반 조건 클래스를 확장하여 네트워크 보안과 관련된 조건을 나타낸다. (외부) 일반 조건 클래스는 추상적이므로 데이터 모델 최적화가 정의 될 수 있다고 가정한다.

[0242] 일반 조건 클래스는 고유 한 객체 ID, 설명 및 0 개 이상의 메타 데이터 객체를 연결하는 메커니즘과 같은 속성의 형태로 기본 조건 정보를 정의한다고 가정한다.

[0243] **네트워크 보안 동작 서브 모델(Network Security Action Sub-Model)**

[0244] 도 8은 본 발명이 적용될 수 있는 네트워크 보안 정보 하위 모델 액션의 확장의 일 예를 나타낸다.

[0245] 도 8은 네트워크 보안 정보 하부 모델에 포함 된 조치 서브 클래스의 보다 자세한 설계를 나타낸다. 도 8의 네 가지 동작 클래스는 (외부)일반 동작 클래스를 확장하여 네트워크 보안 제어 기능을 수행하는 작업을 나타낸다.

[0246] 도 8의 세 가지 동작 클래스는 (외부) 일반 동작 클래스를 확장하여 네트워크 보안과 관련된 작업을 나타낸다. (외부) Generic Action 클래스는 추상적이므로 데이터 모델 최적화가 정의 될 수 있다.

[0247] 일반적인 동작 클래스는 고유 한 객체 ID, 설명 및 0 개 이상의 메타 데이터 객체를 첨부하는 메커니즘과 같은 속성 형식의 기본 동작 정보를 정의한다고 가정한다.

[0248] **I2NSF 기능을 위한 정보 모델(Information Model for I2NSF Capabilities)**

[0249] 도 9은 본 발명이 적용될 수 있는 I2NSF 보안 기능의 상위 레벨 모델의 일 예를 나타낸다.

[0250] 도 9에 도시된 바와 같이 I2NSF 기능 모델은 다양한 콘텐츠 보안 및 공격 완화 기능을 나타내는 많은 기능으로 구성된다. 각 기능은 응용 프로그램 계층에서 특정 유형의 위협으로부터 보호한다.

[0251] 도 9는 SecurityCapability라고 하는 일반적인 I2NSF 보안 기능 클래스를 도시한다. 이를 통해 외부 메타 데이터 정보 모델의 디자인에 영향을 주지 않으면 서 이 클래스에 공통 속성, 관계 및 동작을 추가 할 수 있다. 모든 I2NSF 보안 기능은 SecuritCapability 클래스에서 서브 클래싱된다.

[0252] 컨텐츠 보안 기능을 위한 정보 모델(Information Model for Content Security Capabilities)

[0253] 도 10은 본 발명이 적용될 수 있는 네트워크 보안 기능 정보 모델의 일 예를 나타낸다.

[0254] 도 10은 콘텐츠 보안 GNSF(Generic Network Security Function)의 예시적인 유형들을 도시한다.

[0255] 도 10에 도시된 바와 같이 콘텐츠 보안은 여러 가지 고유 한 보안 기능으로 구성될 수 있다. 이러한 각 기능은 응용 프로그램 계층에서 특정 유형의 위협으로부터 콘텐츠를 보호할 수 있다.

[0256] 콘텐츠 보안은 도 10에 도시된 바와 같이 GNSF (Generic Network Security Function) 유형일 수 있다.

[0257] 공격완화 기능을 위한 정보 모델(Information Model for Attack Mitigation Capabilities)

[0258] 도 11는 본 발명이 적용될 수 있는 공격 완화 기능 정보 모델의 일 예를 나타낸다.

[0259] 도 11에 도시된 바와 같이 공격 완화는 여러 GNSF로 구성될 수 있다. 각각은 특정 유형의 네트워크 공격으로부터 컨텐츠를 보호합니다. 공격 완화 (Acknowledge mitigation) 보안은 잘 정의 된 보안 기능을 요약 한 GNSF 유형이다.

[0261] **I2NSF 보안 정책의 구조와 목적**

[0262] 1.I2NSF 보안 정책 규칙(I2NSF Security Policy Rule)

[0263] I2NSF 보안 정책 규칙은 일반 네트워크 보안 기능에 대한 정책 규칙을 나타낸다. 정책 규칙의 객체는 정책 정보 및 규칙 정보로 정의될 수 있다. 여기에는 Event Clause Objects, Condition Clause Objects, Action Clause Objects, Resolution Strategy 및 Default Action과 같은 ECA 정책 규칙이 포함될 수 있다.

[0264] 2.Event Clause

[0265] 이벤트는 앞에서 살펴본 바와 같이 관리되는 시스템이 변경 될 때 및/또는 관리되는 시스템의 환경에서 중요한 시점에 발생할 수 있다.

[0266] Event Clause Objects는 I2NSF 정책 규칙의 컨텍스트에서 사용될 때 I2NSF 정책 규칙의 조건 절을 평가할 수 있는지 여부를 결정하는 데 사용될 수 있다. 이벤트 절의 대상은 사용자 보안 이벤트, 장치 보안 이벤트, 시스템 보안 이벤트 및 시간 보안 이벤트로 정의될 수 있다. 이벤트 조항의 대상은 특정 공급 업체 이벤트 기능에 따라 확장될 수 있다.

[0267] 3.Condition Clause

[0268] 조건은 앞에서 살펴본 바와 같이 알려진 속성, 특징 및/또는 값의 세트와 비교 될 속성, 기능 및/또는 값의 집합으로 정의되어 그 (명령형) I2NSF 정책 규칙을 실행하거나 실행하지 않을 수 있다.

[0269] 이러한 object는 패킷 보안 조건, 패킷 페이로드 보안 조건, 대상 보안 조건, 사용자 보안 조건, 컨텍스트 조건 및 일반 컨텍스트 조건으로 정의될 수 있다.

[0270] Action 조항의 오브젝트는 특정 공급 업체 조건 기능에 따라 확장될 수 있다.

[0271] 4.Action Clause

[0272] 동작은 이벤트 및 조건 절이 충족될 때 흐름 기반 NSF의 측면을 제어하고 모니터링 하는데 사용된다. NSF는 다양한 액션을 실행하여 보안 기능을 제공한다. 동작 절의 오브젝트는 입력 동작, 송신 동작 및 적용 프로파일 동작으로 정의될 수 있으며, 동작 절의 오브젝트는 특정 벤더 조치 기능에 따라 확장될 수 있다.

[0274] **데이터 모델 구조**

[0275] 이하, 본 발명에서 제안하는 데이터 모델에 대해 살펴보도록 한다.

[0276] 본 발명에서 제안하는 데이터 모델의 구조는 아래와 같은 사항이 고려되었다.

[0277] - Event, Condition, Action 절 집계에 의한 ECA 정책 모델의 고찰

[0278] - 기능 대수의 고려.

[0279] - NSF 기능 카테고리 (예 : 네트워크 보안, 컨텐츠 보안 및 공격 완화 기능) 고려.

[0280] - 네트워크 보안 이벤트 클래스, 네트워크 보안 조건 클래스 및 네트워크 보안 작업 클래스에 대한 정의.

[0282] 도 12는 본 발명의 일 실시 예에 따른 네트워크 보안 정책 식별을 위한 데이터 모델 구조를 예시한다.

[0283] 네트워크 보안 정책을 식별하기 위한 데이터 모델은 도 12에 도시된 바와 같은 구조로 구성될 수 있다

[0284] 네트워크 보안 정책을 식별하기 위한 데이터 모델은 보안 정책, 이벤트 절 컨테이너, 조건 절 컨테이너 및 동작 절 컨테이너로 구성될 수 있다.

[0285] 보안 정책의 데이터 필드는 정책 이름, 규칙들, 해결 전략, 고정 동작 및 규칙(rule) 그룹으로 구성될 수 있다.

[0286] 규칙들은 규칙들을 식별하기 위한 이름, 규칙을 설명하기 위한 description, priority, enable, session-aging-time, long-connection, policy-event-clause-agg-ptr*, policy-condition-clause-agg-ptr*, policy-action-clause-agg-ptr* 및 time-zone으로 구성될 수 있다.

[0287] long-connection은 규칙이 적용될 수 있는 지속시간을 설정할 수 있도록 enable 및 during을 포함할 수 있다.

[0288] 또한, time-zone은 적용되는 룰의 절대적인 시간 외에 주기적인 시간을 설정할 수 있도록 absolute-time-zone

및 periodic-time-zone을 포함할 수 있다.

[0289] absolute-time-zone는 룰이 적용되는 절대적인 시간 또는 날짜를 설정하기 위해서 시작 시간 및 종료 시간을 설정하기 위한 start-time?, end-time? 및 날짜를 설정하기 위한 absolute-date*를 포함할 수 있다.

[0290] periodic-time-zone은 룰이 적용되는 주기적인 시간을 설정하기 위한 day 및 month를 포함할 수 있다.

[0291] resolution-strategy은 룰을 위한 해결 전략을 설정하기 위해서 전략의 타입을 설정하기 위한 (resolution-strategy-type)?, 첫 번째로 매칭되는 룰을 설정하기 위한 first-matching-rule? 및 마지막으로 매칭되는 룰을 설정하기 위한 last-matching-rule?을 포함할 수 있다.

[0292] default-action은 선택된 동작이 없는 경우 수행될 수 있는 동작을 설정하기 위한 필드로써 동작의 타입을 설정할 수 있다.

[0293] rule-group은 규칙들이 그룹화되어 관리될 수 있는 그룹들로 구성되며, 각 그룹에 대한 데이터 필드는 group-name, rule-range, enable, description을 포함한다.

[0294] event-clause-container, condition-clause-container 및 action-clause-container는 정책 규칙이 “이벤트”, “동작” 및 “조건”을 집계하기 위해서 사용될 수 있다.

[0296] 도 13은 본 발명의 일 실시 예에 따른 이벤트 규칙을 위한 데이터 모델 구조를 예시한다.

[0297] 이벤트는 앞에서 살펴본 바와 같이 관리되는 시스템이 변경될 때 및/또는 관리되는 시스템의 환경에서 중요한 시점에 발생하는 사건을 의미한다.

[0298] 도 13에 도시된 이벤트 절을 위한 오브젝트들은 사용자 보안 이벤트, 장치 보안 이벤트, 시스템 보안 이벤트 및 시간 보안 이벤트로 정의될 수 있다. 이러한 개체는 특정 공급 업체 이벤트 기능에 따라 확장될 수 있으며, 보다 일반적인 네트워크 보안 기능을 위한 추가 이벤트 객체가 추가될 수 있다.

[0300] 도 14a 내지 도 14d는 본 발명의 일 실시 예에 따른 컨디션 규칙을 위한 데이터 모델 구조를 예시한다.

[0301] 조건은 앞에서 살펴본 바와 같이 알려진 속성, 특징 및/또는 값의 세트와 비교될 속성, 기능 및/또는 값의 집합으로 정의되어 그 (명령형) I2NSF 정책 규칙을 실행하거나 실행하지 않을 수 있다.

[0302] 컨디션 규칙을 위한 객체는 패킷 보안 조건, 패킷 페이로드 보안 조건, 대상 보안 조건, 사용자 보안 조건, 컨텍스트 조건 및 일반 컨텍스트 조건으로 정의될 수 있다.

[0303] 이러한 컨디션 규칙을 위한 개체는 특정 공급 업체 조건 기능에 따라 확장 될 수 있으며, 보다 일반적인 네트워크 보안 기능을 위한 조건 개체를 추가 할 수 있다.

[0304] 또한, 도 14c에 도시된 바와 같이 컨디션 규칙을 위한 데이터 모델 구조는 pkt-sec-cond-tcp-src-port*, pkt-sec-cond-tcp-dest-port*, pkt-sec-cond-udp-src-port* 및 pkt-sec-cond-udp-dest-port*를 통해 포트 번호와 관련된 룰을 설정할 수 있다.

[0305] 도 14d에서는 규칙이 적용될 수 있는 어플리케이션의 상태를 관리하기 위해, application-condition의 데이터 필드는 application-description?, application-object*, application-group*, application-label*, category를 포함한다.

[0306] 또한, 규칙들은 URL(Uniform Resource Locator)에 따라 적용여부가 설정될 수 있으며, 이를 위해 url-category-condition의 데이터 필드는 pre-defined-category*, user-defined-category*를 포함한다.

[0307] 도 15는 본 발명의 일 실시 예에 따른 액션 규칙을 위한 데이터 모델 구조를 예시한다.

[0309] *동작은 이벤트 및 조건 절이 충족될 때 흐름 기반 NSF의 측면을 제어하고 모니터링 하는데 사용된다.

[0310] 이러한 개체는 수신 동작, 송신 동작 및 적용 프로필 동작으로 정의될 수 있다. 이러한 개체는 특정 공급 업체 작업 기능에 따라 확장될 수 있으며, 보다 일반적인 네트워크 보안 기능을 위한 액션 객체를 추가 할 수 있다.

[0312] 도 14a 내지 도 15에 도시된 컨디션 규칙 및 액션 규칙을 위한 데이터 모델의 구조는 컨테이너 구조가 사용되기 때문에 다중의 컨디션을 적용할 수 있다.

[0314] 도 16a 내지 도 19j는 본 발명의 일 실시 예에 따른 I2NSF NSF-Facing Interface의 YANG 데이터 모듈을 예시한다.

[0315] 도 16a 내지 도 19j를 참조하면, 도 12a 내지 도 15b에서 설명한 데이터 모델을 이용하여 네트워크 보안 기능들의 정보 모델을 위한 YANG 데이터 모델을 설정할 수 있다.

[0317] 도 16a 내지 19j에 도시된 모듈은 네트워크 보안 기능들을 위한 양 데이터 모듈로 정의될 수 있다.

[0318] 이하, NSF 모니터링을 위한 정보 모델에 대해 살펴보도록 한다.

[0319] 보안 기능을 구성하기 위해 관리 엔티티(예를 들면: NMS, 보안 컨트롤러)에 NSF(예를 들면: FW, IPS, Anti-DDOS 또는 Anti-Virus 기능)가 제공하는 인터페이스 NSF에서 모니터링하고 NSF를 모니터링하는 것을 "I2NSF NSF-Facing Interface"라고 한다(ID.ietf-i2nsf-terminology 참조).

[0320] 모니터링 부분은 NSF에 관한 중요한 정보를 획득하는 것을 의미한다. 알림, 이벤트, 레코드, 카운터. 시의 적절하고 포괄적인 방식으로 수행되면 NSF 모니터링은 전반적인 보안 프레임 워크에서 매우 중요한 역할을 한다. NSF에 의해 생성된 모니터링 정보는 악의적인 활동 또는 비정상적인 행동 또는 서비스 거부 공격의 잠재적 징후의 조기 표시일 수 있다.

[0321] NSF 모니터링 데이터는 아래와 같은 상황에서 사용될 수 있다.

[0322] 위에서 설명한 바와 같이 모니터링은 전반적인 보안 프레임 워크에서 매우 중요한 역할을 한다. NSF를 모니터링하면 규정된 보안 상태를 유지하는 데 있어 보안 컨트롤러에 매우 중요한 정보가 제공된다. 이 외에도 아래와 같이 NSF를 모니터링 할 수 있는 다른 이유가 있다.

[0323] - 보안 관리자는 NSF 또는 네트워크에서 발생한 특정 이벤트에서 트리거 되는 정책을 구성할 수 있다. 보안 컨트롤러는 지정된 이벤트를 모니터링하고 이벤트가 발생하면 정책에 따라 추가 보안 기능을 구성한다.

[0324] - 보안 정책 위반의 결과로 NSF에 의해 촉발된 사건은 의심스러운 활동을 탐지하기 위해 SIEM에 의해 사용될 수 있다.

[0325] - NSF의 이벤트 및 활동 로그를 사용하여 동작 및 예측과 같은 고급 분석을 구축하여 보안 상태를 개선할 수 있다.

[0326] - 보안 컨트롤러는 고 가용성을 달성하기 위해 NSF의 이벤트를 사용할 수 있다. 실패한 NSF 재시작, NSF 수평 확장 등의 수정 조치를 취할 수 있다.

[0327] - NSF의 이벤트 및 활동 로그는 운영 문제의 디버깅 및 근본 원인 분석에 도움이 될 수 있다.

[0328] - NSF의 활동 기록은 운영 및 비즈니스상의 이유로 기록 데이터를 작성하는 데 사용될 수 있다.

[0329] **NSF 모니터링 데이터의 분류**

[0330] 강력한 보안 상태를 유지하려면 NSF 보안 정책을 구성 할뿐만 아니라 관찰 가능한 정보를 소비하여 NSF를 지속적으로 모니터링 해야 한다. 이를 통해 보안 관리자는 적시에 네트워크에서 어떤 일이 일어나고 있는지 평가할 수 있다.

[0331] 정적 보안 상태에 기반하여 모든 내부 및 외부 위협을 차단하는 것은 불가능하다. 이 목표를 달성하려면 일정한 가시성을 가진 매우 역동적인 자세가 필요하다. 본 발명은 NSF에서 얻을 수 있고 모니터링 정보로 사용될 수 있는 일련의 정보 요소(및 그 범위)를 정의할 수 있다.

[0332] 본질적으로 이러한 유형의 모니터링 정보는 여러 수준의 세밀성에 대한 지속적인 가시성을 지원하기 위해 활용 될 수 있으며 해당 기능에 의해 소비 될 수 있다

[0333] 이하, 모든 모니터링 데이터를 위한 기본적인 정보 모델에 대해 살펴보도록 한다.

[0334] 모든 모니터링 데이터를 위한 기본적인 정보 모델(Basic Information Model for All Monitoring Data)

[0335] - message_version: 데이터 형식의 버전을 나타내며 01에서 시작하는 2 자리 10 진수.

[0336] - message_type : 이벤트, 경고, 알람, 로그, 카운터 등

[0337] - time_stamp: 메시지가 생성 된 시간을 나타냄.

[0338] - vendor_name: NSF 공급 업체의 이름.

[0339] - NSF_name: 메시지를 생성하는 NSF의 이름 (또는 IP).

[0340] - Module_name: 메시지를 출력하는 모듈 이름

[0341] - Severity: 로그의 레벨을 나타냄. 총 8 개의 레벨 (0에서 7까지)이 존재하며, 숫자가 작을수록 심각도가 높다.

[0342] 모니터링 데이터를위한 확장 정보 모델(Extended Information Model for Monitoring Data)

[0343] 확장 정보 모델은 알람과 같은 구조화 된 데이터에만 사용됩니다. 구조화되지 않은 데이터는 기본 정보 모델로만 지정된다.

[0344] **시스템 알람(System Alarm)**

[0345] 메모리 알람(Memory Alarm)

[0346] 다음 정보가 메모리 알람에 포함되어야 한다.

[0347] - event_name: 'MEM_USAGE_ALARM'

[0348] - module_name:알람 생성을 담당하는 NSF 모듈을 나타냄.

[0349] - usage: 사용 된 메모리 양을 지정함.

[0350] - 임계 값: 경보를 트리거 하는 임계 값

[0351] - 심각도: 위험 수준 (예를 들면: 위험 수준, 높음, 보통, 낮음)

[0352] - 메시지: '메모리 사용량이 임계 값을 초과했습니다.'와 같은 메시지를 출력함.

[0354] CPU 알람(CPU Alarm)

[0355] 다음과 같은 정보가 CPU 알람에 포함될 수 있다.

[0356] - event_name: 'CPU_USAGE_ALARM'

[0357] - usage: 사용 된 CPU의 양을 지정합니다.

[0358] - threshold: 이벤트를 트리거 하는 임계 값

[0359] - 심각도: 위험 수준 (예를 들면: 위험 수준, 높음, 보통, 낮음)

[0360] - 메시지: 'CPU 사용량이 임계 값을 초과했습니다.' 와 같은 메시지를 출력함.

[0362] 디스크 알람(Disk Alarm)

[0363] 다음과 같은 정보가 디스크 알람에 포함될 수 있다.

[0364] - event_name: 'DISK_USAGE_ALARM'

[0365] - usage: 사용 된 디스크 공간의 양을 지정합니다.

[0366] - threshold: 이벤트를 트리거 하는 임계 값

[0367] - 심각도: 위험 수준 (예를 들면: 위험 수준, 높음, 보통, 낮음)

[0368] - 메시지: '디스크 사용량이 임계 값을 초과했습니다.' 와 같은 메시지를 출력함.

[0370] 하드웨어 알람(Hardware Alarm)

[0371] 다음과 같은 정보가 하드웨어 알람에 포함될 수 있다.

[0372] - event_name: 'HW_FAILURE_ALARM'

[0373] - component_name:이 알람을 생성하는 HW 구성 요소를 나타냅니다.

[0374] - 임계 값: 경보를 트리거 하는 임계 값

[0375] - 심각도: 위험 수준 (예를 들면: 위험 수준, 높음, 보통, 낮음)

[0376] - 메시지: '하드웨어 구성 요소가 고장 났거나 성능이 저하되었습니다.'와 같은 메시지를 출력함.

[0378] 인터페이스 알람(Interface Alarm)

[0379] 다음과 같은 정보가 인터페이스 알람에 포함될 수 있다.

[0380] - event_name: 'IFNET_STATE_ALARM'

[0381] - interface_Name: 인터페이스 이름

[0382] - interface_state: 'UP', 'DOWN', 'CONGESTED'

[0383] - threshold: 이벤트를 트리거 하는 임계 값

[0384] - 심각도: 위험 수준 (예를 들면: 위험 수준, 높음, 보통, 낮음)

[0385] - 메시지: '현재 인터페이스 상태'를 출력함.

[0387] **시스템 이벤트(System Events)**

[0388] 액세스 위반(Access Violation)

[0389] 다음과 같은 정보가 이벤트에 포함될 수 있다.

[0390] - event_name: 'ACCESS_DENIED'

[0391] - user: 사용자 이름

[0392] - group: 사용자가 속한 그룹

[0393] - login_ip_address: 사용자의 로그인 IP 주소

[0394] - authentication_mode: 사용자 인증 모드. 예를 들면: 로컬 인증, 제 3 자 서버 인증, 인증 면제, SSO 인증

[0396] *- 메시지: '액세스가 거부되었습니다.' 와 같은 메시지를 출력함.

[0398] 구성 변경(Configuration Change)

[0399] 다음과 같은 정보가 이벤트에 포함될 수 있다.

[0400] - event_name: 'CONFIG_CHANGE'

[0401] - user: 사용자 이름

[0402] - group: 사용자가 속한 그룹

[0403] - login_ip_address: 사용자의 로그인 IP 주소

[0404] - authentication_mode: 사용자 인증 모드. 예를 들면: 로컬 인증, 제 3 자 서버 인증, 인증 면제, SSO 인증

[0405] - 메시지: '구성이 수정되었습니다' 와 같은 메시지를 출력함.

[0406] **시스템 로그(System Log)**

[0407] 접속 로그(Access Logs)

[0408] 액세스 로그는 관리자의 로그인, 로그 아웃 및 장치 작동을 기록하고, 이를 분석하여 보안 취약성을 식별 할 수 있다. 운영 보고서에는 아래와 같은 정보가 포함될 수 있다.

[0409] - 관리자: 장치에서 작동하는 관리자

[0410] - login_ip_address : 관리자가 로그인 할 때 사용하는 IP 주소

[0411] - login_mode : 관리자 로그인 모드를 지정합니다.(예를 들면: 뿌리, 사용자)

[0412] - operation_type : 관리자가 수행하는 조작 유형(예를 들면: 로그인, 로그 아웃, 구성 등)

[0413] - 결과: 명령 실행 결과

[0414] - content: 로그인 후 관리자가 수행 한 작업.

[0415] 자원 사용률 로그(Resource Utilization Logs)

[0417] *403실행중인 보고서는 장치 시스템의 실행 상태를 기록하며 이는 장치 모니터링에 유용하다. 실행 보고서는 다음과 같은 정보를 포함할 수 있다.

[0418] - system_status: 현재 시스템의 실행 상태

[0419] - CPU_usage: CPU 사용량을 지정합니다.

[0420] - memory_usage: 메모리 사용량을 지정합니다.

[0421] - disk_usage: 디스크 사용량을 지정합니다.

[0422] - disk_left: 사용 가능한 디스크 공간을 지정합니다.

[0423] - session_number: 총 동시 세션 수를 지정합니다.

[0424] - process_number: 총 시스템 프로세스 수를 지정합니다.

[0425] - in_traffic_rate: 총 인바운드 트래픽 속도 (pps)

[0426] - out_traffic_rate: 총 아웃 바운드 트래픽 속도 (pps)

[0427] - in_traffic_speed: 총 인바운드 트래픽 속도 (bps)

[0428] - out_traffic_speed: 총 아웃 바운드 트래픽 속도 (bps)

[0429] 사용자 활동 로그(User Activity Logs)

[0430] 사용자 활동 기록은 사용자의 온라인 기록 (로그인 시간, 온라인 / 잠금 기간 및 로그인 IP 주소)과 사용자가 수행하는 작업에 대한 가시성을 제공한다. 사용자 활동 보고서는 사용자 로그인 및 네트워크 액세스 활동 중 예외를 식별하는 데 유용하다.

[0431] - group: 사용자가 속한 그룹

[0432] - login_ip_address: 사용자의 로그인 IP 주소

[0433] - authentication_mode: 사용자 인증 모드. 예를 들면: 로컬 인증, 제 3 자 서버 인증, 인증 면제, SSO 인증

[0434] - access_mode: 사용자 액세스 모드. 예를 들면: PPP, SVN, LOCAL

[0435] - online_duration: 온라인 기간

[0436] - lockout_duration: 잠금 기간

[0437] - 유형: 사용자 활동. 성공한 사용자 로그인, 실패한 로그인 시도, 사용자 로그 아웃, 성공한 사용자 비밀번호 변경, 실패한 사용자 비밀번호 변경, 사용자 잠금, 사용자 잠금 해제, 알 수 없음

[0438] - 원인: 사용자 작업에 실패했습니다.

[0439] **시스템 카운터(System Counter)**

[0440] 인터페이스 카운터(Interface counters)

[0441] 인터페이스 카운터는 NSF로 들어오고 나가는 트래픽, 대역폭 사용에 대한 가시성을 제공한다.

[0442] - interface_name : NSF에서 구성된 네트워크 인터페이스 이름

[0443] - in_total_traffic_pkts : 전체 인바운드 패킷

[0444] - out_total_traffic_pkts : 총 아웃 바운드 패킷

[0445] - in_total_traffic_bytes : 총 인바운드 바이트

[0447] *- out_total_traffic_bytes : 총 아웃 바운드 바이트

[0448] - in_drop_traffic_pkts : 총 인바운드 드롭 패킷

[0449] - out_drop_traffic_pkts : 총 아웃 바운드 드롭 패킷

[0450] - in_drop_traffic_bytes : 총 인바운드 드롭 바이트

[0451] - out_drop_traffic_bytes : 총 아웃 바운드 삭제 바이트

[0452] - in_traffic_ave_rate : 인바운드 트래픽 평균 요금 (pps)

[0453] - in_traffic_peak_rate : 인바운드 트래픽 피크 속도 (pps)

[0455] *- in_traffic_ave_speed : 인바운드 트래픽 평균 속도 (bps)

[0456] - in_traffic_peak_speed : 인바운드 트래픽 최고 속도 (bps)

[0457] - out_traffic_ave_rate : 아웃 바운드 트래픽 평균 요금 (pps)

[0458] - out_traffic_peak_rate : 아웃 바운드 트래픽 피크 속도 (pps)

[0459] - out_traffic_ave_speed : 아웃 바운드 트래픽 평균 속도 (bps)

[0460] - out_traffic_peak_speed : 아웃 바운드 트래픽 최고 속도 (bps)

[0461] NSF 이벤트(NSF Events)

[0462] DDos 이벤트는 다음과 같은 정보를 포함할 수 있다.

[0463] - event_name : 'SEC_EVENT_DDoS'

[0464] - sub_attack_type : Syn flood, ACK flood, SYN-ACK flood, FIN / RST flood, TCP 연결 flood, UDP flood, Icmp flood, HTTPS flood, HTTP flood, DNS query flood, DNS reply flood, SIP flood 등

[0465] - dst_ip: 공격 받고있는 victum의 IP 주소

[0466] - dst_port: 트래픽을 목표로 삼고있는 포트 번호.

[0467] - start_time: 공격이 시작된 시간을 나타내는 타임 스탬프

[0468] - end_time: 공격이 종료 된 시간을 나타내는 타임 스탬프. 경보를 전송할 때 공격이 계속 발생하면이 필드는 비어있을 수 있습니다.

[0469] - attack_rate: 공격 트래픽의 PPS

[0470] - attack_speed: 공격 트래픽의 bps

[0471] - rule_id: 트리거되는 규칙의 ID입니다.

[0472] - rule_name: 트리거되는 규칙의 이름

[0473] - 프로필: 트래픽이 일치하는 보안 프로필입니다.

[0474] 세션 테이블 이벤트(session Table Event)

[0475] 아래와 같은 정보가 세션 테이블 이벤트에 포함될 수 있다.

[0476] - event_name: 'SESSION_USAGE_HIGH'

[0477] - current: 동시 세션 수

[0478] - max: 세션 테이블이 지원할 수있는 최대 세션 수

[0479] - threshold: 이벤트를 트리거하는 임계 값

[0480] - 메시지: '세션 테이블의 수가 임계 값을 초과했습니다.'

[0482] 바이러스 이벤트(Virus Event)

[0483] 아래와 같은 정보가 바이러스 이벤트에 포함될 수 있다.

[0484] - event_Name : 'SEC_EVENT_VIRUS'

[0485] - virus_type : 바이러스 유형 (예 : 트로이 목마, 웜, 매크로) 바이러스 유형, 바이러스 이름

[0486] - dst_ip : 바이러스가 발견 된 패킷의 대상 IP 주소

[0487] - src_ip : 바이러스가 발견 된 패킷의 소스 IP 주소

[0488] - src_port : 바이러스가 발견 된 패킷의 소스 포트

[0489] - dst_port : 바이러스가 발견 된 패킷의 대상 포트

[0490] - src_zone : 바이러스가 발견 된 패킷의 소스 보안 영역

[0491] - dst_zone : 바이러스가 발견 된 패킷의 대상 보안 영역

[0492] - file_type : 바이러스가 숨겨진 파일의 유형

[0493] - file_name : 바이러스가 숨겨진 파일의 이름

[0494] - virus_info : 바이러스의 간단한 소개

[0495] - raw_info : 이벤트를 트리거하는 패킷을 설명하는 정보.

[0496] - rule_id : 트리거되는 규칙의 ID입니다.

[0497] - rule_name : 트리거되는 규칙의 이름

[0498] - 프로필 : 트래픽이 일치하는 보안 프로필입니다.

[0500] 침입 이벤트(Intrusion Event)

[0501] - Intrustion Event에는 다음 정보가 포함되어야합니다.

[0502] - event_name: 이벤트 이름 : 'SEC_EVENT_Intrusion'

[0503] - sub_attack_type: 공격 유형, 예 : 잔인한 힘, 버퍼 오버 플로우

[0504] - src_ip: 패킷의 소스 IP 주소

[0505] - dst_ip: 패킷의 목적지 IP 주소

[0506] - src_port: 패킷의 소스 포트 번호

[0507] - dst_port : 패킷의 목적지 포트 번호

[0508] - src_zone: 패킷의 소스 보안 영역

[0509] - dst_zone: 패킷의 대상 보안 영역

[0510] - 프로토콜: 사용 된 전송 계층 프로토콜, 예를 들어, TCP, UDP

[0511] - app: 채용 된 애플리케이션 계층 프로토콜 (예를 들면: HTTP, FTP)

[0512] - rule_id: 트리거 되는 규칙의 ID입니다.

[0513] - rule_name: 트리거 되는 규칙의 이름

[0514] - 프로필: 트래픽이 일치하는 보안 프로필

[0515] - intrusion_info: 침입에 대한 간단한 설명

[0516] - raw_info: 이벤트를 트리거 하는 패킷을 설명하는 정보.

[0517] 봇넷 이벤트(Botnet Event)

[0518] 아래와 같은 정보는 봇넷 이벤트에 포함될 수 있다.

[0519] - event_name : 이벤트 이름 : 'SEC_EVENT_Botnet'

[0520] - botnet_name : 탐지 된 봇넷의 이름

[0521] - src_ip : 패킷의 소스 IP 주소

[0522] - dst_ip : 패킷의 목적지 IP 주소

[0523] - src_port : 패킷의 소스 포트 번호

[0524] - dst_port : 패킷의 목적지 포트 번호

[0525] - src_zone : 패킷의 소스 보안 영역

[0526] - dst_zone : 패킷의 대상 보안 영역

[0527] - 프로토콜 : 사용 된 전송 계층 프로토콜, 예를 들어, TCP, UDP

[0528] - app : 채용 된 애플리케이션 계층 프로토콜 (예 : HTTP, FTP)

[0529] - 역할 : 봇넷 내 통신 당사자의 역할 :

[0530] 1. 좀비 호스트에서 공격자까지의 패킷

[0532] * 2. 공격자에서 좀비 호스트로 가는 패킷

[0533] 3. IRC / WEB 서버에서 좀비 호스트로 가는 패킷

[0534] 4. 좀비 호스트에서 IRC / WEB 서버로 보내는 패킷

[0535] 5. 공격자에서 IRC / WEB 서버로 보낸 패킷

[0536] 6. IRC / WEB 서버에서 공격자로 가는 패킷

[0537] 7. 좀비 호스트에서 희생자까지의 패킷

[0538] - botnet_info : Botnet에 대한 간단한 설명

[0539] - rule_id : 트리거 되는 규칙의 ID입니다.

[0540] - rule_name : 트리거 되는 규칙의 이름

[0541] - 프로필: 트래픽이 일치하는 보안 프로필

[0542] - raw_info : 이벤트를 트리거 하는 패킷을 설명하는 정보

[0543] 웹 공격 이벤트(Web Attack Event)

[0544] 아래와 같은 정보가 웹 공격 이벤트에 포함될 수 있다.

[0545] - event_name : 이벤트 이름 : 'SEC_EVENT_WebAttack'

[0546] - sub_attack_type : 구체적인 웹 공격 유형 (예 : sql injection, command injection, XSS, CSRF)

[0547] - src_ip : 패킷의 소스 IP 주소

[0548] - dst_ip : 패킷의 목적지 IP 주소

[0549] - src_port : 패킷의 소스 포트 번호

[0550] - dst_port : 패킷의 목적지 포트 번호

[0551] - src_zone : 패킷의 소스 보안 영역

[0552] - dst_zone : 패킷의 대상 보안 영역

[0553] - req_method : 요구 사항의 방법. 예를 들어 HTTP에서 'PUT'또는 'GET'

[0554] - req_url : 요청 된 URL

[0555] - url_category : 일치하는 URL 카테고리

[0556] - filtering_type : 블랙리스트, 허용 목록, 사용자 정의, 미리 정의된, 악의적인 카테고리, 알 수없는 URL 필터링 유형

[0557] - rule_id : 트리거되는 규칙의 ID입니다.

[0558] - rule_name : 트리거되는 규칙의 이름

[0559] - 프로필 : 트래픽이 일치하는 보안 프로필입니다.

[0561] **NSF 로그(NSF Logs)**

[0562] DDoS 로그(DDoS Logs)

[0563] DDoS 경보의 필드 외에도 필드 외에도 아래와 같은 정보가 DDoS 로그에 포 함될 수 있다.

[0564] - 공격 유형 : DDoS

[0565] - attack_ave_rate : 기록 된 시간 내에 공격 트래픽의 평균 pps

[0566] - attack_ave_speed : 기록 된 시간 내에 공격 트래픽의 평균 bps

[0567] - attack_pkt_num : 기록 된 시간 내의 공격 패킷 수

[0568] - attack_src_ip : 공격 트래픽의 소스 IP 주소입니다. 많은 양의 IP 주소가 있는 경우 다른 규칙에 따라 특정 수의 자원을 선택.

[0569] - 액션 : DDoS 공격 (예를 들면: 허용, 경고, 차단, 폐기, 선언, 차단 IP, 차단 서비스)에 대한 작업.

[0571] 바이러스 로그(Virus Logs)

[0572] 바이러스 경보의 필드 외에도 아래와 같은 정보가 바이러스 로그에 포함될 수 있다,

[0573] - 공격 유형 : 바이러스

[0574] - 프로토콜 : 전송 계층 프로토콜

[0575] - app : 응용 프로그램 계층 프로토콜의 이름

[0576] - times : 바이러스 탐지 시간

[0577] - 액션 : 바이러스를 다루는 액션 (예 : 경고, 차단)

[0578] - os : 바이러스가 영향을 미치는 OS (예 : all, android, ios, unix, windows).

[0580] 침입 로그(Intrusion Logs)

[0581] 침입 경보의 필드 외에도 아래와 같은 정보가 침입 로그에 포함도리 수 있다.

[0582] - 공격 유형 : 침입

[0583] - 시간 : 기록된 시간에 침입 시간이 발생했습니다.

[0584] - os : 침입에 영향을 주는 OS입니다 (예 : all, android, ios, unix, windows).

[0585] - 액션 : 침입을 다루는 액션들, 예를 들어 허용, 경고, 차단, 폐기, 선언, 차단 -IP, 차단 - 서비스

[0586] - attack_rate : 공격 트래픽의 pps NUM

[0587] - attack_speed : NUM 공격 트래픽의 bps

[0588] 봇넷 로그(Botnet Logs)

[0589] Botnet Alarm의 필드 외에도 아래와 같은 정보가 봇넷 로그에 포함될 수 있다.

[0590] - attack_type : 봇넷

[0591] - botnet_pkt_num : 탐지된 봇넷으로 보내거나 받은 패킷 수

[0592] - 액션 : 탐지된 패킷을 처리하는 액션 (예 : 허용, 경고, 차단, 폐기, 선언, 차단 IP, 차단 서비스, 기타

[0593] - os : 공격 대상인 모든 OS, 예를 들어, android, ios, unix, windows 등

[0594] DPI 로그(DPI Logs)

[0595] DPI 로그는 업로드 및 다운로드 된 파일 및 데이터, 전송 및 수신 된 전자 메일에 대한 통계를 제공하고 웹 사이트에 기록을 경고하고 차단할 수 있다.

[0596] - 유형 : DPI 작업 유형. 예 : 파일 차단, 데이터 필터링, 애플리케이션 동작 제어

[0597] - file_name : 파일 이름

[0598] - file_type : 파일 형식

[0599] - src_zone : 트래픽 소스 보안 영역

[0600] - dst_zone : 트래픽의 대상 보안 영역

[0601] - src_region : 트래픽 소스 영역

[0602] - dst_region : 트래픽의 대상 영역

[0603] - src_ip : 트래픽 소스 IP 주소

[0604] - src_user : 트래픽을 생성 한 사용자

[0605] - dst_ip : 트래픽의 대상 IP 주소

[0606] - src_port : 트래픽 소스 포트

[0607] - dst_port : 트래픽의 대상 포트

[0608] - 프로토콜 : 트래픽의 프로토콜 유형

[0609] - 앱 : 트래픽의 애플리케이션 유형

[0610] - policy_id : 트래픽이 일치하는 보안 정책 ID

[0611] - policy_name : 트래픽이 일치하는 보안 정책 이름

[0612] - 동작 : 트래픽이 일치하는 파일 차단 규칙, 데이터 필터링 규칙 또는 응용 프로그램 동작 제어 규칙에 정의된 작업이다.

[0614] Vulnerabillity 검색 로그

[0615] 취약점 검색 로그에는 피해 호스트 및 관련 취약점 정보가 기록되어야 한다. 다음 정보가 보고서에 포함 되어야 합니다.

[0616] - victim_ip : 취약성이 있는 희생 된 호스트의 IP 주소

[0617] - 취약점 ID : 취약점 ID

[0618] - vulnerability_level : 취약점 수준. 예 : 높음, 낮음, 낮음

[0619] - 운영 체제 : 대상 호스트의 운영 체제

[0620] - 서비스 : 피해자 호스트에 취약성이 있는 서비스

[0621] - protocol : 프로토콜 유형. 예 : TCP, UDP

[0622] - port : 포트 번호

[0623] - vulnerability_info : 취약점에 대한 정보

[0624] - fix_suggestion : 취약점에 대한 수정 제안.

[0625] - 8.6.7. 웹 공격 로그

[0626] - 웹 공격 경보의 필드 외에도 다음 정보가 웹 공격 보고서에 포함되어야 한다.

[0627] - attack_type : 웹 공격

[0628] - rsp_code : 응답 코드

[0629] - req_clientapp : 클라이언트 응용 프로그램

[0630] - req_cookies : 쿠키

[0631] - req_host : 요청한 호스트의 도메인 이름

[0632] - raw_info : 이벤트를 트리거 하는 패킷을 설명하는 정보.

[0634] **NSF 카운터(NSF Counter)**

[0635] 방화벽 카운터(Firewall Counters)

[0636] 방화벽 카운터는 트래픽 서명, 대역폭 사용 및 구성된 보안 및 대역폭 정책이 어떻게 적용되었는지에 대한 가시성을 제공합니다.

[0637] - src_zone : 트래픽 소스 보안 영역

[0638] - dst_zone : 트래픽의 대상 보안 영역

[0639] - src_region : 트래픽 소스 영역

[0641] *623- dst_region : 트래픽의 대상 영역

[0642] - src_ip : 트래픽 소스 IP 주소

[0643] - src_user : 트래픽을 생성 한 사용자

[0644] - dst_ip : 트래픽의 대상 IP 주소

[0645] - src_port : 트래픽 소스 포트

[0646] - dst_port : 트래픽의 대상 포트

[0647] - 프로토콜 : 트래픽의 프로토콜 유형

[0648] - 앱 : 트래픽의 애플리케이션 유형

[0649] - policy_id : 트래픽이 일치하는 보안 정책 ID

[0650] - policy_name : 트래픽이 일치하는 보안 정책 이름

[0651] - in_interface : 트래픽의 인바운드 인터페이스

[0652] - out_interface : 트래픽의 아웃 바운드 인터페이스

[0653] - total_traffic : 총 트래픽 양

[0654] - in_traffic_ave_rate : 인바운드 트래픽 평균 요금 (pps)

[0655] - in_traffic_peak_rate : 인바운드 트래픽 피크 속도 (pps)

[0656] - in_traffic_ave_speed : 인바운드 트래픽 평균 속도 (bps)

[0657] - in_traffic_peak_speed : 인바운드 트래픽 최고 속도 (bps)

[0658] - out_traffic_ave_rate : 아웃 바운드 트래픽 평균 요금 (pps)

[0659] - out_traffic_peak_rate : 아웃 바운드 트래픽 피크 속도 (pps)

[0660] - out_traffic_ave_speed : 아웃 바운드 트래픽 평균 속도 (bps)

[0662] 정책 방문 횟수 카운터(Policy Hit Counters)

[0663] 정책 적중 카운터는 트래픽이 일치하는 보안 정책과 적중 횟수를 기록합니다. 정책 구성이 올바른지 확인할 수 있습니다.

[0664] - src_zone : 트래픽 소스 보안 영역

[0666] *647- dst_zone : 트래픽의 대상 보안 영역

[0667] - src_region : 트래픽 소스 영역

[0668] - dst_region : 트래픽의 대상 영역

[0669] - src_ip : 트래픽 소스 IP 주소

[0670] - src_user : 트래픽을 생성 한 사용자

[0671] - dst_ip : 트래픽의 대상 IP 주소

[0672] - src_port : 트래픽 소스 포트

[0673] - dst_port : 트래픽의 대상 포트

[0674] - 프로토콜 : 트래픽의 프로토콜 유형

[0675] - 앱 : 트래픽의 애플리케이션 유형

[0676] - policy_id : 트래픽이 일치하는 보안 정책 ID

[0677] - policy_name : 트래픽이 일치하는 보안 정책 이름

[0678] - hit_times: 보안 정책이 지정된 트래픽과 일치하는 횟수.

[0679] 도 20a 내지 도 20j는 본 발명의 일 실시 예에 따른 NSF 모니터링을 위한 데이터 모델을 예시한다.

[0680] 도 20a 내지 도 20j를 참조하면, 앞에서 살펴본 NSF를 모니터링하기 위한 정보 모델을 이용하여 데이터 모델을 설계할 수 있다.

[0681] 도 21a 내지 도 22i는 본 발명의 일 실시 예에 따른 모니터링을 위한 YANG 데이터 모델을 예시한다.

[0682] NSF 모니터링이 포괄적인 방법으로 수행되는 경우, 악의적인 활동, 비정상적인 행동 또는 잠재적 인 서비스 거부 공격의 징후를 적시에 감지 할 수 있다. 이러한 모니터링 기능은 앞에서 살펴본 NSF가 생성한 모니터링 정보를 기반으로 합니다.

[0683] 따라서, 본 발명은 NSF 모니터링을 위한 정보 모델을 지정하는 데이터 모델 구조 트리뿐만 아니라 NSF 모니터링을 위한 해당 YANG 데이터 모델을 설계하는 방법을 제안한다.

[0684] 도 21a 내지 도 22i를 참조하면 앞에서 살펴본 NSF 모니터링을 위한 정보 모델 및 데이터 모델을 이용하여 해당 YANG 데이터 모델을 설계할 수 있다.

[0685] 도 1 내지 도 22i에서 설명한 정보 모델, 데이터 모델 및 YANG 데이터 모델은 선택적으로 조합되어 사용될 수 있다.

[0686] 이상에서 설명된 실시 예들은 본 발명의 구성요소들과 특징들이 소정 형태로 결합된 것들이다. 각 구성요소 또는 특징은 별도의 명시적 언급이 없는 한 선택적인 것으로 고려되어야 한다. 각 구성요소 또는 특징은 다른 구성요소나 특징과 결합되지 않은 형태로 실시될 수 있다. 또한, 일부 구성요소들 및/또는 특징들을 결합하여 본 발명의 실시 예를 구성하는 것도 가능하다. 본 발명의 실시 예들에서 설명되는 동작들의 순서는 변경될 수 있다. 어느 실시예의 일부 구성이나 특징은 다른 실시 예에 포함될 수 있고, 또는 다른 실시예의 대응하는 구성 또는 특징과 교체될 수 있다. 특허청구범위에서 명시적인 인용 관계가 있지 않은 청구항들을 결합하여 실시 예를 구성하거나 출원 후의 보정에 의해 새로운 청구항으로 포함시킬 수 있음은 자명하다.

[0687] 본 발명에 따른 실시 예는 다양한 수단, 예를 들어, 하드웨어, 펌웨어(firmware), 소프트웨어 또는 그것들의 결합 등에 의해 구현될 수 있다. 하드웨어에 의한 구현의 경우, 본 발명의 일 실시 예는 하나 또는 그 이상의 ASICs(application specific integrated circuits), DSPs(digital signal processors), DSPDs(digital signal processing devices), PLDs(programmable logic devices), FPGAs(field programmable gate arrays), 프로세서, 콘트롤러, 마이크로 콘트롤러, 마이크로 프로세서 등에 의해 구현될 수 있다.

[0688] 펌웨어나 소프트웨어에 의한 구현의 경우, 본 발명의 일 실시 예는 이상에서 설명된 기능 또는 동작들을 수행하는 모듈, 절차, 함수 등의 형태로 구현될 수 있다. 소프트웨어 코드는 메모리에 저장되어 프로세서에 의해 구동될 수 있다. 상기 메모리는 상기 프로세서 내부 또는 외부에 위치하여, 이미 공지된 다양한 수단에 의해 상기 프로세서와 데이터를 주고 받을 수 있다.

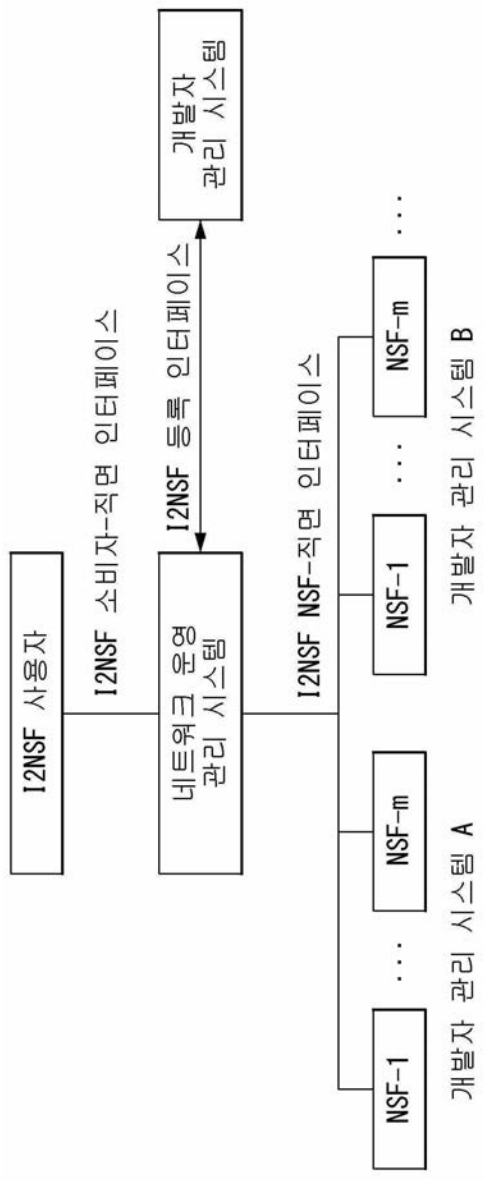
[0689] 본 발명은 본 발명의 필수적 특징을 벗어나지 않는 범위에서 다른 특정한 형태로 구체화될 수 있음은 당업자에게 자명하다. 따라서, 상술한 상세한 설명은 모든 면에서 제한적으로 해석되어서는 아니 되고 예시적인 것으로 고려되어야 한다. 본 발명의 범위는 첨부된 청구항의 합리적 해석에 의해 결정되어야 하고, 본 발명의 등가적 범위 내에서의 모든 변경은 본 발명의 범위에 포함된다.

산업상 이용가능성

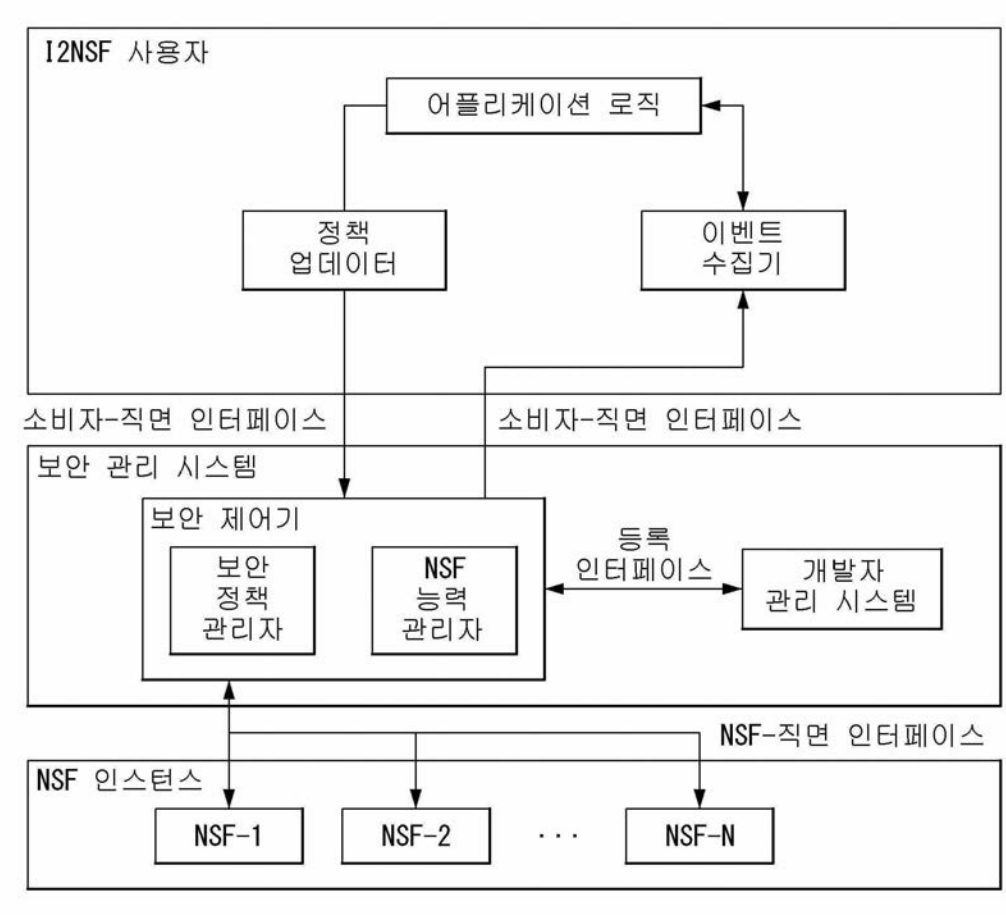
[0690] 본 발명은 다양한 보안 관리 시스템에 적용될 수 있다.

도면

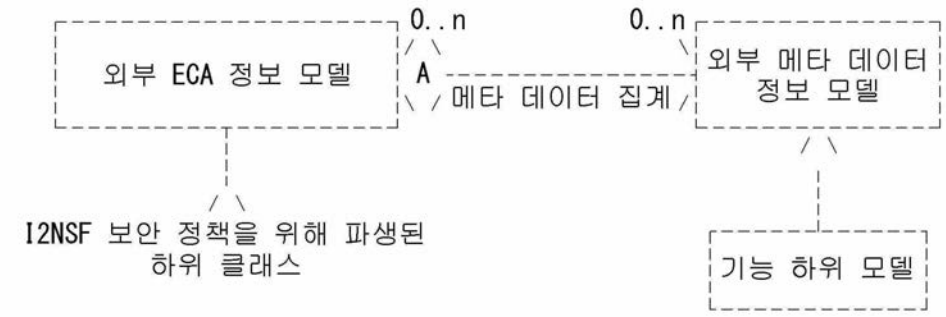
도면1


I2NSF 사용자
I2NSF 소비자-직면 인터페이스
네트워크 운영 관리 시스템
I2NSF 등록 인터페이스
개발자 관리 시스템
I2NSF NSF-직면 인터페이스
NSF-1
NSF-m
개발자 관리 시스템 A
NSF-1
NSF-m
개발자 관리 시스템 B

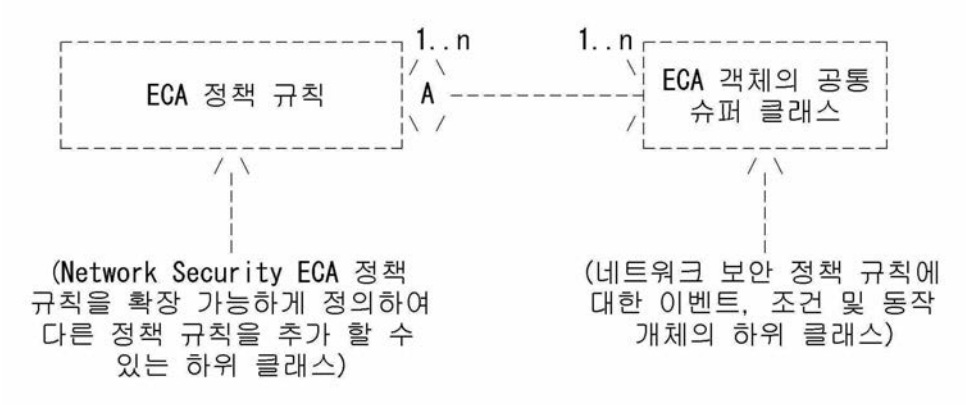
도면2


I2NSF 사용자
어플리케이션 로직
정책
업데이터
이벤트
수집기
소비자-직면 인터페이스
소비자-직면 인터페이스
보안 관리 시스템
보안 제어기
보안
정책
관리자
NSF
능력
관리자
등록
인터페이스
개발자
관리 시스템
NSF-직면 인터페이스
NSF 인스턴스
NSF-1
NSF-2
· · ·
NSF-N

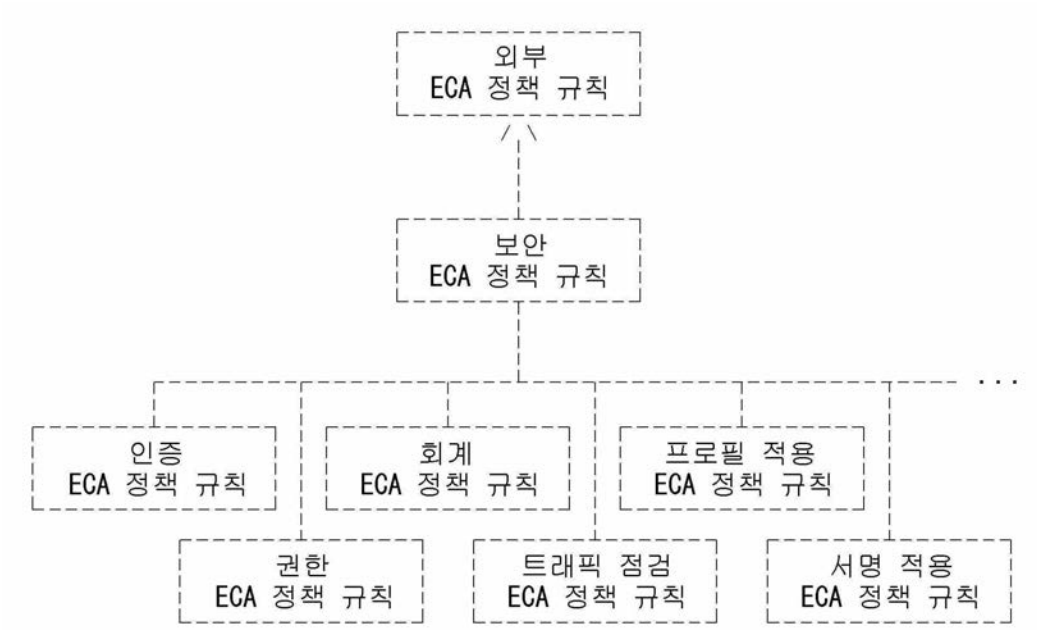
도면3


외부 ECA 정보 모델
0..n
A
메타 데이터 집계
0..n
외부 메타 데이터
정보 모델
I2NSF 보안 정책을 위해 파생된
하위 클래스
기능 하위 모델

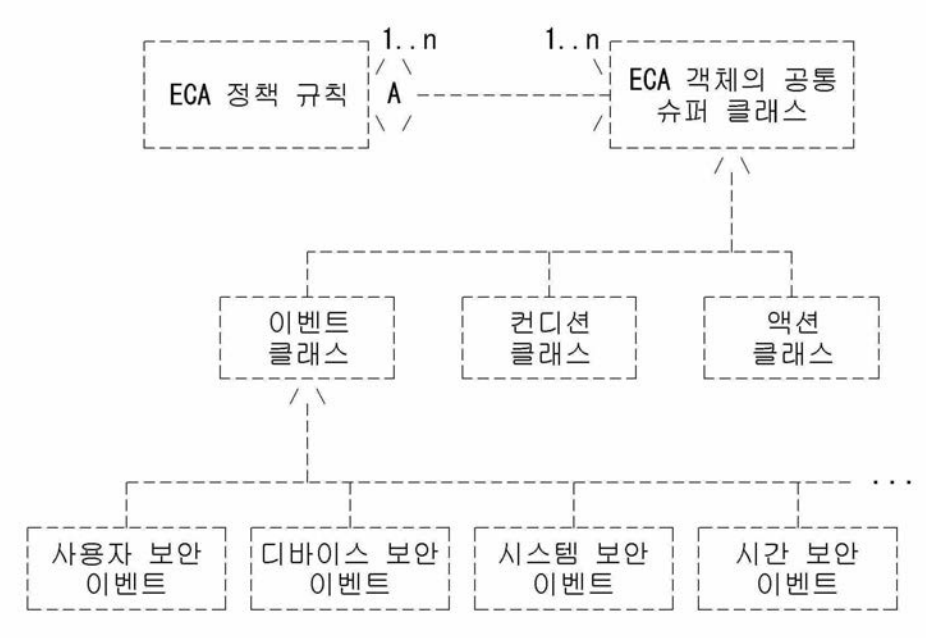
도면4


ECA 정책 규칙
1..n
A
1..n
ECA 객체의 공통 슈퍼 클래스
(Network Security ECA 정책 규칙을 확장 가능하게 정의하여 다른 정책 규칙을 추가 할 수 있는 하위 클래스)
(네트워크 보안 정책 규칙에 대한 이벤트, 조건 및 동작 개체의 하위 클래스)

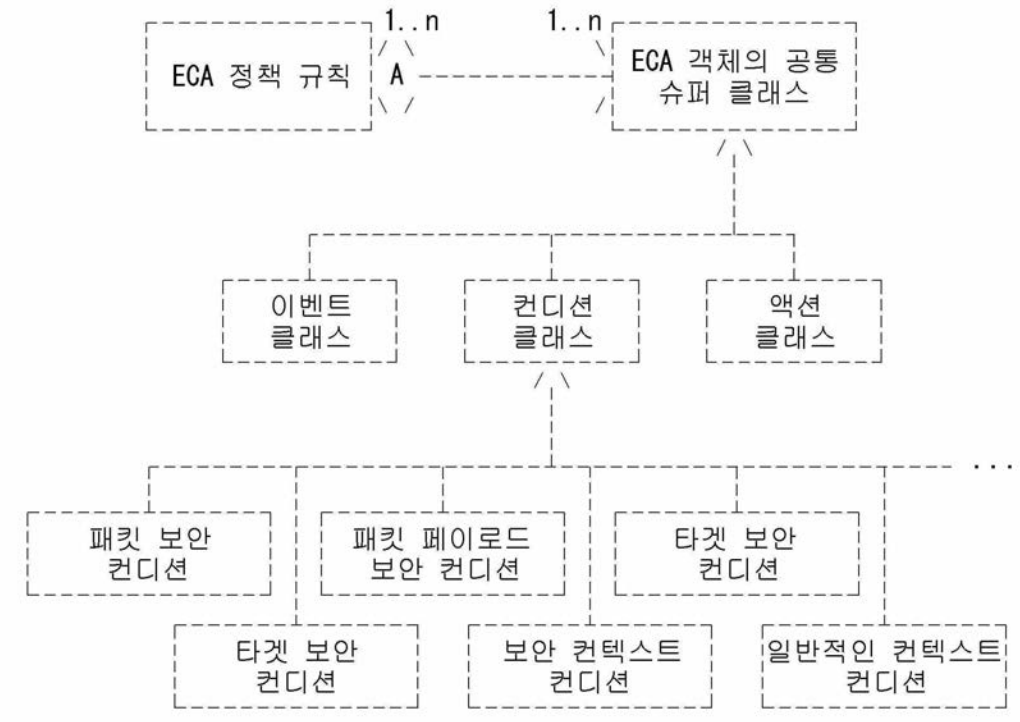
도면5


외부 ECA 정책 규칙
보안 ECA 정책 규칙
...
인증 ECA 정책 규칙
회계 ECA 정책 규칙
프로필 적용 ECA 정책 규칙
권한 ECA 정책 규칙
트래픽 점검 ECA 정책 규칙
서명 적용 ECA 정책 규칙

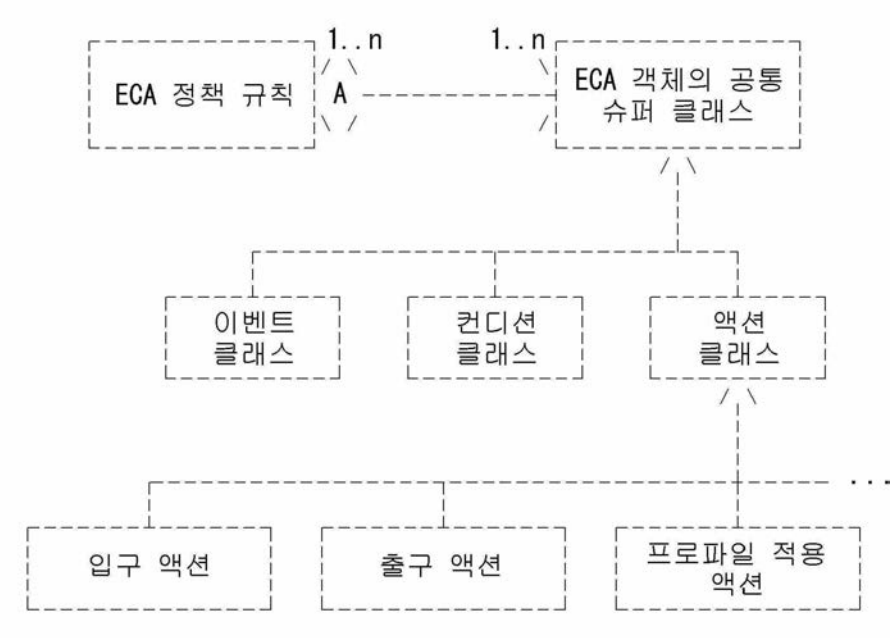
도면6


ECA 정책 규칙
1..n
A
1..n
ECA 객체의 공통
슈퍼 클래스
이벤트
클래스
컨디션
클래스
액션
클래스
사용자 보안
이벤트
디바이스 보안
이벤트
시스템 보안
이벤트
시간 보안
이벤트

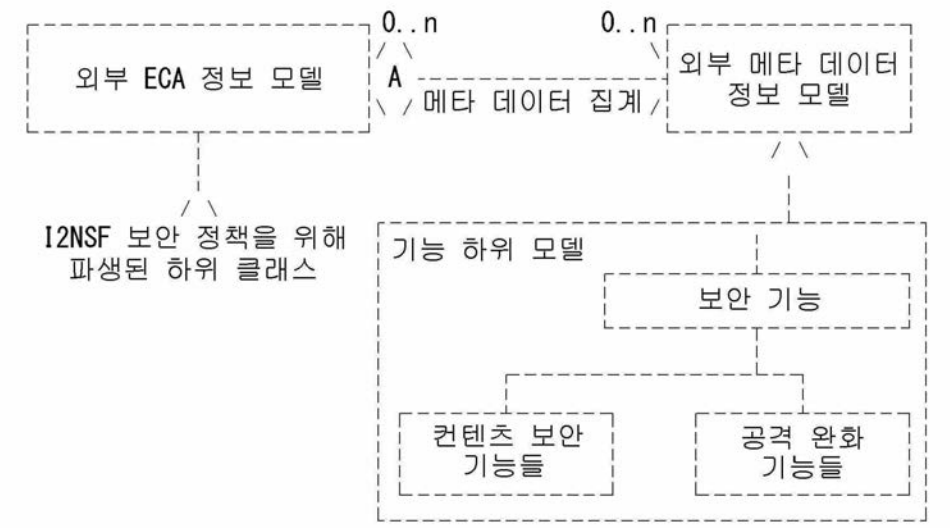
도면7


ECA 정책 규칙
1..n
A
1..n
ECA 객체의 공통
슈퍼 클래스
이벤트
클래스
컨디션
클래스
액션
클래스
패킷 보안
컨디션
패킷 페이로드
보안 컨디션
타겟 보안
컨디션
타겟 보안
컨디션
보안 컨텍스트
컨디션
일반적인 컨텍스트
컨디션

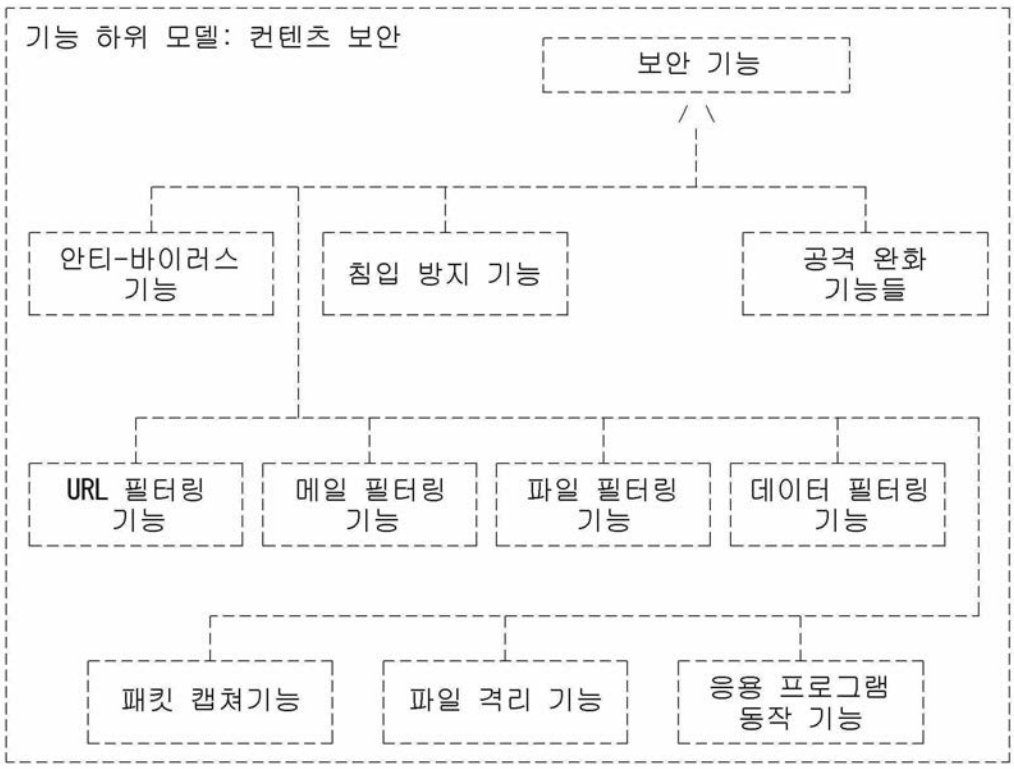
도면8


ECA 정책 규칙
1..n
A
1..n
ECA 객체의 공통 슈퍼 클래스
이벤트 클래스
컨디션 클래스
액션 클래스
...
입구 액션
출구 액션
프로파일 적용 액션

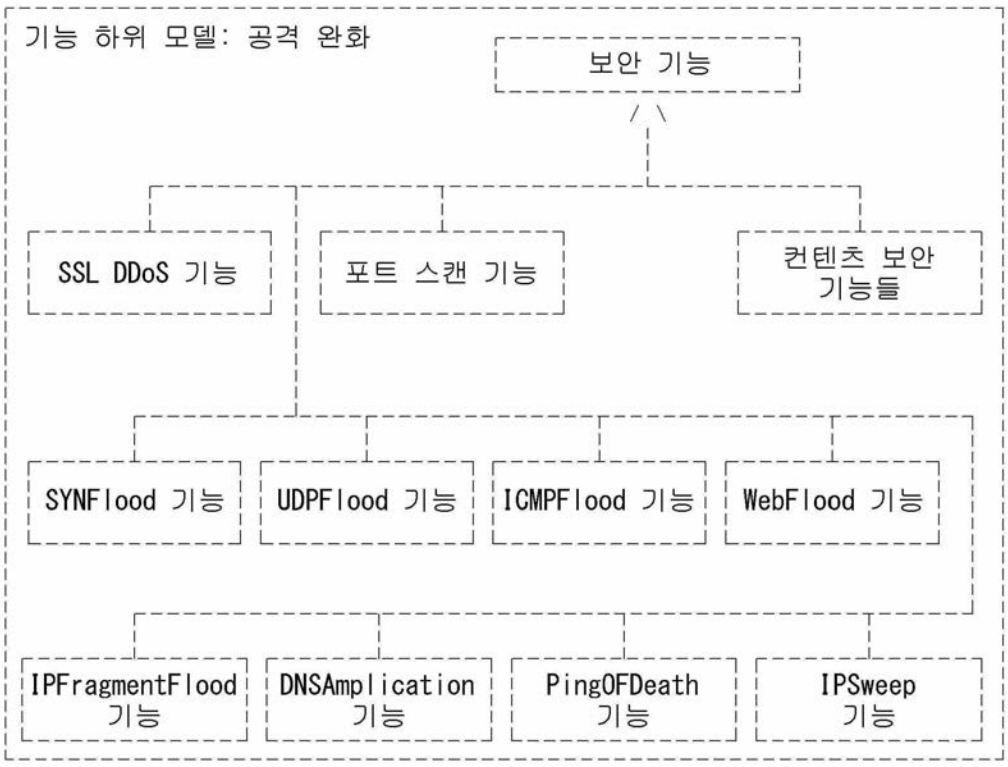
도면9


외부 ECA 정보 모델
0..n
A
메타 데이터 집계
0..n
외부 메타 데이터 정보 모델
I2NSF 보안 정책을 위해 파생된 하위 클래스
기능 하위 모델
보안 기능
컨텐츠 보안 기능들
공격 완화 기능들

도면10


기능 하위 모델: 컨텐츠 보안
보안 기능
안티-바이러스 기능
침입 방지 기능
공격 완화 기능들
URL 필터링 기능
메일 필터링 기능
파일 필터링 기능
데이터 필터링 기능
패킷 캡쳐기능
파일 격리 기능
응용 프로그램 동작 기능

도면11


기능 하위 모델: 공격 완화
보안 기능
SSL DDoS 기능
포트 스캔 기능
컨텐츠 보안 기능들
SYNFlood 기능
UDPFlood 기능
ICMPFlood 기능
WebFlood 기능
IPFragmentFlood 기능
DNSAmplication 기능
PingOFDeath 기능
IPSweep 기능

도면12

```
module: ietf-i2nsf-policy-rule-for-nsf
  +--rw i2nsf-security-policy
  |  +--rw policy-name?          string
  |  +--rw rules* [rule-name]
  |  |  +--rw rule-name                          string
  |  |  +--rw rule-description?                   string
  |  |  +--rw rule-priority?                      uint8
  |  |  +--rw enable?                            boolean
  |  |  +--rw session-aging-time?                 uint16
  |  |  +--rw long-connection
  |  |  |  +--rw enable?   boolean
  |  |  |  +--rw during?   uint16
  |  |  +--rw policy-event-clause-agg-ptr*      instance-identifier
  |  |  +--rw policy-condition-clause-agg-ptr*   instance-identifier
  |  |  +--rw policy-action-clause-agg-ptr*      instance-identifier
  |  |  +--rw time-zone
  |  |     +--rw absolute-time-zone
  |  |     |  +--rw time
  |  |     |  |  +--rw start-time?   yang:date-and-time
  |  |     |  |  +--rw end-time?     yang:date-and-time
  |  |     |  +--rw date
  |  |     |     +--rw absolute-date?   yang:date-and-time
  |  |     +--rw periodic-time-zone
  |  |        +--rw day
  |  |        |  +--rw sunday?      boolean
  |  |        |  +--rw monday?      boolean
  |  |        |  +--rw tuesday?     boolean
  |  |        |  +--rw wednesday?   boolean
  |  |        |  +--rw thursday?    boolean
  |  |        |  +--rw friday?      boolean
  |  |        |  +--rw saturday?    boolean
  |  |        +--rw month
  |  |           +--rw january?     boolean
  |  |           +--rw february?    boolean
  |  |           +--rw march?       boolean
  |  |           +--rw april?       boolean
  |  |           +--rw may?         boolean
  |  |           +--rw june?        boolean
  |  |           +--rw july?        boolean
  |  |           +--rw august?      boolean
  |  |           +--rw september?   boolean
  |  |           +--rw october?     boolean
  |  |           +--rw november?    boolean
  |  |           +--rw december?    boolean
  |  +--rw resolution-strategy
  |  |  +--rw (resolution-strategy-type)?
  |  |     +--:(fmr)
  |  |     |  +--rw first-matching-rule?   boolean
  |  |     +--:(lmr)
  |  |        +--rw last-matching-rule?    boolean
  |  +--rw default-action
  |  |  +--rw default-action-type?   boolean
  |  +--rw rule-group
  |     +--rw groups* [group-name]
  |        +--rw group-name     string
  |        +--rw rule-range
  |        |  +--rw start-rule?   string
  |        |  +--rw end-rule?     string
  |        +--rw enable?        boolean
  |        +--rw description?   string
  +--rw event-clause-container
  |  ...
  +--rw condition-clause-container
  |  ...
  +--rw action-clause-container
     ...
```

도면13

```
module: ietf-i2nsf-policy-rule-for-nsf
+--rw i2nsf-security-policy* [policy-name]
|      ...
|   +--rw eca-policy-rules* [rule-id]
|      ...
|   +--rw resolution-strategy
|      ...
|   +--rw default-action
|      ...
+--rw event-clause-container
|   +--rw event-clause-list* [eca-object-id]
|      +--rw entity-class?         identityref
|      +--rw eca-object-id         string
|      +--rw manual?               string
|      +--rw sec-event-content     string
|      +--rw sec-event-format      sec-event-format
|      +--rw sec-event-type        string
+--rw condition-clause-container
|   ...
+--rw action-clause-container
    ...
```

도면14a

```
module: ietf-i2nsf-policy-rule-for-nsf
+--rw i2nsf-security-policy* [policy-name]
|      ...
|   +--rw eca-policy-rules* [rule-id]
|      ...
|   +--rw resolution-strategy
|      ...
|   +--rw default-action
|      ...
+--rw event-clause-container
```

도면14b

```
|   ...
+--rw condition-clause-container
|   +--rw condition-clause-list* [eca-object-id]
|      +--rw entity-class?                        identityref
|      +--rw eca-object-id                        string
|      +--rw packet-security-condition
|         +--rw packet-manual?                       string
|         +--rw packet-security-mac-condition
|            +--rw pkt-sec-cond-mac-dest*            yang:phys-address
|            +--rw pkt-sec-cond-mac-src*             yang:phys-address
|            +--rw pkt-sec-cond-mac-8021q*           string
|            +--rw pkt-sec-cond-mac-ether-type*      string
|            +--rw pkt-sec-cond-mac-tci*             string
|         +--rw packet-security-ipv4-condition
|            +--rw pkt-sec-cond-ipv4-header-length*     uint8
|            +--rw pkt-sec-cond-ipv4-tos*               uint8
|            +--rw pkt-sec-cond-ipv4-total-length*      uint16
|            +--rw pkt-sec-cond-ipv4-id*                uint8
|            +--rw pkt-sec-cond-ipv4-fragment*          uint8
|            +--rw pkt-sec-cond-ipv4-fragment-offset*   uint16
|            +--rw pkt-sec-cond-ipv4-ttl*               uint8
|            +--rw pkt-sec-cond-ipv4-protocol*          uint8
|            +--rw pkt-sec-cond-ipv4-src*           inet:ipv4-address
|            +--rw pkt-sec-cond-ipv4-dest*          inet:ipv4-address
|            +--rw pkt-sec-cond-ipv4-ipopts?              string
|            +--rw pkt-sec-cond-ipv4-sameip?             boolean
|            +--rw pkt-sec-cond-ipv4-geoip*              string
|         +--rw packet-security-ipv6-condition
|            +--rw pkt-sec-cond-ipv6-dscp*             string
|            +--rw pkt-sec-cond-ipv6-ecn*              string
|            +--rw pkt-sec-cond-ipv6-traffic-class*    uint8
|            +--rw pkt-sec-cond-ipv6-flow-label*       uint32
```

도면14c

```
|       |  |  +--rw pkt-sec-cond-ipv6-payload-length*   uint16
|       |  |  +--rw pkt-sec-cond-ipv6-next-header*      uint8
|       |  |  +--rw pkt-sec-cond-ipv6-hop-limit*        uint8
|       |  |  +--rw pkt-sec-cond-ipv6-src*     inet:ipv6-address
|       |  |  +--rw pkt-sec-cond-ipv6-dest*    inet:ipv6-address
|       |  +--rw packet-security-tcp-condition
|       |  |  +--rw pkt-sec-cond-tcp-src-port*          inet:port-number
|       |  |  +--rw pkt-sec-cond-tcp-dest-port*          inet:port-number
|       |  |  +--rw pkt-sec-cond-tcp-seq-num*       uint32
|       |  |  +--rw pkt-sec-cond-tcp-ack-num*       uint32
|       |  |  +--rw pkt-sec-cond-tcp-window-size*   uint16
|       |  |  +--rw pkt-sec-cond-tcp-flags*         uint8
|       |  +--rw packet-security-udp-condition
|       |  |  +--rw pkt-sec-cond-udp-src-port*          inet:port-number
|       |  |  +--rw pkt-sec-cond-udp-dest-port*          inet:port-number
|       |  |  +--rw pkt-sec-cond-udp-length*    string
```

도면14d

```
|     |  +--rw packet-security-icmp-condition
|     |     +--rw pkt-sec-cond-icmp-type*      uint8
|     |     +--rw pkt-sec-cond-icmp-code*      uint8
|     |     +--rw pkt-sec-cond-icmp-seg-num*   uint32
|     +--rw packet-payload-condition
|     |  +--rw packet-payload-description?   string
|     |  +--rw pkt-payload-content*          string
|     +--rw acl-number?                 uint32
|     +--rw application-condition
|     |  +--rw application-description?   string
|     |  +--rw application-object*        string
|     |  +--rw application-group*         string
|     |  +--rw application-label*         string
|     |  +--rw category
|     |     +--rw application-category* [name application-subcategory]
|     |        +--rw name                      string
|     |        +--rw application-subcategory    string
|     +--rw target-condition
|     |  +--rw target-description?       string
|     |  +--rw device-sec-context-cond
|     |     +--rw pc?                boolean
|     |     +--rw mobile-phone?       boolean
|     |     +--rw voip-volte-phone?   boolean
|     |     +--rw tablet?             boolean
|     |     +--rw iot?                boolean
|     |     +--rw vehicle?            boolean
|     +--rw users-condition
|     |  +--rw users-description?   string
|     |  +--rw user
|     |  |  +--rw (user-name)?
|     |  |     +--:(tenant)
|     |  |     |  +--rw tenant    uint8
|     |  |     +--:(vn-id)
|     |  |        +--rw vn-id     uint8
|     |  +--rw group
|     |  |  +--rw (group-name)?
|     |  |     +--:(tenant)
|     |  |     |  +--rw tenant    uint8
|     |  |     +--:(vn-id)
|     |  |        +--rw vn-id     uint8
|     |  +--rw security-grup        string
|     +--rw url-category-condition
|     |  +--rw pre-defined-category*    string
|     |  +--rw user-defined-category*   string
|     +--rw context-condition
|     |  +--rw context-description?   string
|     +--rw gen-context-condition
|        +--rw gen-context-description?   string
|        +--rw geographic-location
|           +--rw src-geographic-location*    uint32
|           +--rw dest-geographic-location*   uint32
+--rw action-clause-container
   ...
```

도면15

```
module: ietf-i2nsf-policy-rule-for-nsf
+--rw i2nsf-security-policy* [policy-name]
|     ...
|  +--rw eca-policy-rules* [rule-id]
|     ...
|  +--rw resolution-strategy
|     ...
|  +--rw default-action
|     ...
+--rw event-clause-container
|  ...
+--rw condition-clause-container
|  ...
+--rw action-clause-container
   +--rw action-clause-list* [eca-object-id]
      +--rw entity-class?     identityref
      +--rw eca-object-id     string
      +--rw rule-log?         boolean
      +--rw session-log?      boolean
      +--rw ingress-action
      |  +--rw ingress-description?   string
      |  +--rw ingress-action-type?   ingress-action
      +--rw egress-action
      |  +--rw egress-description?   string
      |  +--rw egress-action-type?   egress-action
      +--rw apply-profile
         +--rw profile-description?        string
         +--rw content-security-control
         |  +--rw content-security-control-types
         |     +--rw antivirus?            string
         |     +--rw ips?                  string
         |     +--rw ids?                  string
         |     +--rw url-filtering?        string
         |     +--rw data-filtering?       string
         |     +--rw mail-filtering?       string
         |     +--rw file-blocking?        string
         |     +--rw file-isolate?         string
         |     +--rw pkt-capture?          string
         |     +--rw application-control?  string
         |     +--rw voip-volte?           string
         +--rw attack-mitigation-control
            +--rw ddos-attack
            |  +--rw ddos-attack-type
            |     +--rw network-layer-ddos-attack
            |     |  +--rw network-layer-ddos-attack-type
            |     |     +--rw syn-flood?       string
            |     |     +--rw udp-flood?       string
            |     |     +--rw icmp-flood?      string
            |     |     +--rw ip-frag-flood?   string
            |     |     +--rw ipv6-related?    string
            |     +--rw app-layer-ddos-attack
            |        +--rw app-ddos-attack-types
            |           +--rw http-flood?      string
            |           +--rw https-flood?     string
            |           +--rw dns-flood?       string
            |           +--rw dns-amp-flood?   string
            |           +--rw ssl-ddos?        string
            +--rw single-packet-attack
               +--rw single-packet-attack-type
                  +--rw scan-and-sniff-attack
                  |  +--rw scan-and-sniff-attack-types
                  |     +--rw ip-sweep?        string
                  |     +--rw port-scanning?   string
                  +--rw malformed-packet-attack
                  |  +--rw malformed-packet-attack-types
                  |     +--rw ping-of-death?   string
                  |     +--rw teardrop?        string
                  +--rw special-packet-attack
                     +--rw special-packet-attack-types
                        +--rw oversized-icmp?   string
                        +--rw tracert?          string
```

도면16a

```
<CODE BEGINS> file "ietf-i2nsf-policy-rule-for-nsf@2018-03-05.yang"

module ietf-i2nsf-policy-rule-for-nsf {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-i2nsf-policy-rule-for-nsf";
  prefix
    policy-rule-for-nsf;
```

도면16b

```
import ietf-inet-types{
  prefix inet;
}
import ietf-yang-types{
  prefix yang;
}

organization
  "IETF I2NSF (Interface to Network Security Functions)
   Working Group";
```

도면16c

```
contact
  "WG Web: <http://tools.ietf.org/wg/i2nsf>
   WG List: <mailto:i2nsf@ietf.org>

   WG Chair: Adrian Farrel
   <mailto:Adrain@olddog.co.uk>

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

   Editor: Jingyong Tim Kim
   <mailto:timkim@skku.edu>

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

   Editor: Susan Hares
   <mailto:shares@ndzh.com>";

description
  "This module defines a YANG data module for network security
   functions.";
revision "2018-07-02"{
  description "The fourth revision";
  reference
    "draft-ietf-i2nsf-capability-00";
}

typedef sec-event-format {
    type enumeration {
      enum unknown {
          description
            "If SecEventFormat is unknown";
      }
      enum guid {
          description
            "If SecEventFormat is GUID
```

도면16d

```
grouping i2nsf-event-type {
    description "TBD";
    leaf description {
      type string;
      description
        "This is description for event.
        Vendors can write instructions for event
        that vendor made";
    }

    leaf sec-event-content {
      type string;
      mandatory true;
```

도면16e

```
typedef ingress-action {
    type enumeration {
      enum pass {
          description
            "If ingress action is pass";
      }
      enum drop {
          description
            "If ingress action is drop";
      }
      enum reject {
          description
            "If ingress action is reject";
      }
      enum alert {
          description
            "If ingress action is alert";
      }
      enum mirror {
          description
            "If ingress action is mirror";
```

도면16f

```
      }
    }
    description
      "This is used for ingress action.";
}

typedef egress-action {
    type enumeration {
      enum invoke-signaling {
          description
            "If egress action is invoke signaling";
      }
      enum tunnel-encapsulation {
          description
            "If egress action is tunnel encapsulation";
      }
      enum forwarding {
          description
            "If egress action is forwarding";
      }
      enum redirection {
          description
            "If egress action is redirection";
      }
    }
    description
      "This is used for egress action.";
}
```

도면16g

```
identity ECA-OBJECT-TYPE {
  description "TBD";
}

identity ECA-EVENT-TYPE {
  base ECA-OBJECT-TYPE;
  description "TBD";
}

identity ECA-CONDITION-TYPE {
  base ECA-OBJECT-TYPE;
  description "TBD";
}

identity ECA-ACTION-TYPE {
  base ECA-OBJECT-TYPE;
  description "TBD";
}
```

도면16h

```
 identity EVENT-USER-TYPE {
  base ECA-EVENT-TYPE;
  description "TBD";
}

 identity EVENT-DEV-TYPE {
  base ECA-EVENT-TYPE;
  description "TBD";
}

 identity EVENT-SYS-TYPE {
  base ECA-EVENT-TYPE;
  description "TBD";
}

 identity EVENT-TIME-TYPE {
  base ECA-EVENT-TYPE;
  description "TBD";
}


grouping i2nsf-eca-object-type {
  leaf entity-class {
    type identityref {
      base ECA-OBJECT-TYPE;
    }
    description "TBD";
  }
  leaf eca-object-id {
      type string;
      description "TBD";
  }
```

도면16i

```
grouping i2nsf-event-type {
    description "TBD";
    leaf manual {
      type string;
      description
        "This is manual for event.
        Vendors can write instructions for event
        that vendor made";
    }

    leaf sec-event-content {
      type string;
      mandatory true;
```

도면16j

```
   description
    "This is a mandatory string that contains the content
     of the SecurityEvent. The format of the content
     is specified in the SecEventFormat class
     attribute, and the type of event is defined in the
     SecEventType class attribute. An example of the
     SecEventContent attribute is a string hrAdmin,
     with the SecEventFormat set to 1 (GUID) and the
     SecEventType attribute set to 5 (new logon).";
}

leaf sec-event-format {
  type sec-event-format;
  mandatory true;
  description
   "This is a mandatory uint 8 enumerated integer, which
    is used to specify the data type of the
    SecEventContent attribute. The content is
    specified in the SecEventContent class attribute,
    and the type of event is defined in the
    SecEventType class attribute. An example of the
    SecEventContent attribute is string hrAdmin,
    with the SecEventFormat attribute set to 1 (GUID)
    and the SecEventType attribute set to 5
    (new logon).";
}
```

도면16k

```
    leaf sec-event-type {
      type string;
      mandatory true;
      description
       "This is a mandatory uint 8 enumerated integer, which
        is used to specify the type of event that involves
        this user. The content and format are specified in
        the SecEventContent and SecEventFormat class
        attributes, respectively. An example of the
        SecEventContent attribute is string hrAdmin,
        with the SecEventFormat attribute set to 1 (GUID)
        and the SecEventType attribute set to 5
       (new logon).";
    }

}

container i2nsf-security-policy {
  description
```

도면161

```
"policy is a container
 including a set of security rules according to certain logic,
 i.e., their similarity or mutual relations, etc. The network
 security policy is able to apply over both the unidirectional
 and bidirectional traffic across the NSF.";

leaf policy-name {
  type string;
  description
    "The name of the policy.
     This must be unique.";
}

list rules {
  key "rule-name";
  description
    "This is a rule for network security functions.";

  leaf rule-name {
    type string;
    mandatory true;
    description
      "The id of the rule.
       This must be unique.";
  }
```

도면16m

```
leaf rule-description {
       type string;
       description
         "This description gives more information about
          rules.";
     }

     leaf rule-priority {
       type uint8;
       description
         "The priority keyword comes with a mandatory
          numeric value which can range from 1 till 255.";
     }

     leaf enable {
       type boolean;
       description
         "True is enable.
          False is not enbale.";
     }

     leaf session-aging-time {
       type uint16;
       description
         "This is session aging time.";
     }

     container long-connection {
       description
         "This is long-connection";

       leaf enable {
         type boolean;
         description
           "True is enable.
            False is not enbale.";
       }

       leaf during {
         type uint16;
         description
           "This is during time.";
       }
     }


     leaf-list policy-event-clause-agg-ptr {
        type instance-identifier;
```

도면16n

```
    must 'derived-from-or-self (/event-clause-container/
    event-clause-list/entity-class, "ECA-EVENT-TYPE")';
    description
        "TBD";
}
leaf-list policy-condition-clause-agg-ptr {
    type instance-identifier;
    must 'derived-from-or-self (/condition-clause-container/
    condition-clause-list/entity-class, "ECA-CONDITION-TYPE")';
    description
        "TBD";
}
leaf-list policy-action-clause-agg-ptr {
    type instance-identifier;
    must 'derived-from-or-self (/action-clause-container/
    action-clause-list/entity-class, "ECA-ACTION-TYPE")';
    description
        "TBD";
}
```

도면16o

```
container time-zone {
  description
    "This can be used to apply rules according to time-zone";
  container absolute-time-zone {
    description
    "This can be used to apply rules according to
     absolute-time";
    container time {
      description
        "This can be used to apply rules according to time";
      leaf start-time {
        type yang:date-and-time;
        description
          "This is start time for time zone";
      }
      leaf end-time {
        type yang:date-and-time;
        description
          "This is end time for time zone";
      }
    }
    container date {
      description
        "This can be used to apply rules according to date";
      leaf absolute-date {
        type yang:date-and-time;
        description
          "This is absolute date for time zone";
```

도면16p

```
    }
  }
}
container periodic-time-zone {
  description
   "This can be used to apply rules according to
   periodic-time-zone";
  container day {
    description
     "This can be used to apply rules according
      to periodic day";
    leaf sunday {
      type boolean;
      description
        "This is sunday for periodic day";
    }
    leaf monday {
      type boolean;
      description
        "This is monday for periodic day";
    }
    leaf tuesday {
      type boolean;
      description
        "This is tuesday for periodic day";
    }
    leaf wednesday {
      type boolean;
      description
        "This is wednesday for periodic day";
    }
    leaf thursday {
      type boolean;
      description
        "This is thursday for periodic day";
    }
```

도면16q

```
  }
  leaf friday {
    type boolean;
    description
      "This is friday for periodic day";
  }
  leaf saturday {
    type boolean;
    description
      "This is saturday for periodic day";
  }
}
container month {
```

도면16r

```
description
 "This can be used to apply rules according
  to periodic month";
leaf january {
  type boolean;
  description
    "This is january for periodic month";
}
leaf february {
  type boolean;
  description
    "This is february for periodic month";
}
leaf march {
  type boolean;
  description
    "This is march for periodic month";
}
leaf april {
  type boolean;
  description
    "This is april for periodic month";
}
leaf may {
  type boolean;
  description
    "This is may for periodic month";
}
leaf june {
  type boolean;
  description
    "This is june for periodic month";
}
```

도면16s

```
}
leaf july {
  type boolean;
  description
    "This is july for periodic month";
}
leaf august {
  type boolean;
  description
    "This is august for periodic month";
}
leaf september {
  type boolean;
  description
    "This is september for periodic month";
}
```

도면16t

```
        leaf october {
          type boolean;
          description
            "This is october for periodic month";
        }
        leaf november {
          type boolean;
          description
            "This is november for periodic month";
        }
        leaf december {
          type boolean;
          description
            "This is december for periodic month";
        }
      }
    }
  }
}
```

도면16u

```
container resolution-strategy {
  description
    "The resolution strategies can be used to
    specify how to resolve conflicts that occur between
    the actions of the same or different policy rules that
    are matched and contained in this particular NSF";


  choice resolution-strategy-type {
    description
      "Vendors can use YANG data model to configure rules";

    case fmr {
      leaf first-matching-rule {
        type boolean;
        description
          "If the resolution strategy is first matching rule";
      }
    }
    case lmr {
      leaf last-matching-rule {
        type boolean;
        description
          "If the resolution strategy is last matching rule";
      }
    }

  }
```

도면16v

```
container default-action {
  description
    "This default action can be used to specify a predefined
    action when no other alternative action was matched
    by the currently executing I2NSF Policy Rule. An analogy
    is the use of a default statement in a C switch statement.";

  leaf default-action-type {
    type boolean;
    description
      "True is permit
       False is deny.";
  }
}


container rule-group {
  description
    "This is rule group";

  list groups {
    key "group-name";
    description
      "This is a group for rules";

    leaf group-name {
      type string;
      description
        "This is a group for rules";
    }

    container rule-range {
      description
        "This is a rule range.";

      leaf start-rule {
        type string;
        description
          "This is a start rule";
      }
      leaf end-rule {
        type string;
        description
          "This is a end rule";
      }
    }
    leaf enable {
      type boolean;
      description
        "This is enable
         False is not enable.";
    }
    leaf description {
      type string;
      description
        "This is a desription for rule-group";
    }
  }
}
}
```

도면16w

```
container event-clause-container {
  description "TBD";
  list event-clause-list {
  key eca-object-id;
  uses i2nsf-eca-object-type {
    refine entity-class {
        default ECA-EVENT-TYPE;
    }
  }

  description
    " This is abstract. An event is defined as any important
      occurrence in time of a change in the system being
      managed, and/or in the environment of the system being
      managed. When used in the context of policy rules for
      a flow-based NSF, it is used to determine whether the
      Condition clause of the Policy Rule can be evaluated
      or not. Examples of an I2NSF event include time and
      user actions (e.g., logon, logoff, and actions that
      violate any ACL.).";

   uses i2nsf-event-type;
   }
}
  container condition-clause-container {
  description "TBD";
  list condition-clause-list {
    key eca-object-id;
    uses i2nsf-eca-object-type {
```

도면16x

```
    refine entity-class {
        default ECA-CONDITION-TYPE;
    }
}
description
  " This is abstract.  A condition is defined as a set
  of attributes, features, and/or values that are to be
  compared with a set of known attributes, features,
  and/or values in order to determine whether or not the
  set of Actions in that (imperative) I2NSF Policy Rule
  can be executed or not. Examples of I2NSF Conditions
  include matching attributes of a packet or flow, and
  comparing the internal state of an NSF to a desired
  state.";

  container packet-security-condition {
    description
      "TBD";
    leaf packet-manual {
      type string;
      description
        "This is manual for packet condition.
        Vendors can write instructions for packet condition
        that vendor made";
    }
```

도면16y

```
container packet-security-mac-condition {
  description
   "The purpose of this Class is to represent packet MAC
   packet header information that can be used as part of
   a test to determine if the set of Policy Actions in
   this ECA Policy Rule should be execute or not.";

  leaf-list pkt-sec-cond-mac-dest {
    type yang:phys-address;
    description
      "The MAC destination address (6 octets long).";
  }

  leaf-list pkt-sec-cond-mac-src {
    type yang:phys-address;
    description
      "The MAC source address (6 octets long).";
  }

  leaf-list pkt-sec-cond-mac-8021q {
    type string;
    description
```

도면16z

```
      "This is an optional string attribute, and defines
       The 802.1Q tab value (2 octets long).";
  }

  leaf-list pkt-sec-cond-mac-ether-type {
    type string;
    description
      "The EtherType field (2 octets long). Values up to
       and including 1500 indicate the size of the
       payload in octets; values of 1536 and above
       define which protocol is encapsulated in the
       payload of the frame.";
  }

  leaf-list pkt-sec-cond-mac-tci {
    type string;
    description
     "This is an optional string attribute, and defines
      the Tag Control Information. This consists of a 3
      bit user priority field, a drop eligible indicator
      (1 bit), and a VLAN identifier (12 bits).";
  }
}
```

도면17a

```
container packet-security-ipv4-condition {
  description
    "The purpose of this Class is to represent IPv4
     packet header information that can be used as
     part of a test to determine if the set of Policy
     Actions in this ECA Policy Rule should be executed
     or not.";

  leaf-list pkt-sec-cond-ipv4-header-length {
    type uint8;
    description
      "The IPv4 packet header consists of 14 fields,
       of which 13 are required.";
  }

  leaf-list pkt-sec-cond-ipv4-tos {
    type uint8;
    description
      "The ToS field could specify a datagram's priority
       and request a route for low-delay,
       high-throughput, or highly-reliable service..";
  }

  leaf-list pkt-sec-cond-ipv4-total-length {
```

도면17b

```
    type uint16;
    description
      "This 16-bit field defines the entire packet size,
       including header and data, in bytes.";
  }

  leaf-list pkt-sec-cond-ipv4-id {
    type uint8;
    description
      "This field is an identification field and is
       primarily used for uniquely identifying
       the group of fragments of a single IP datagram.";
  }

  leaf-list pkt-sec-cond-ipv4-fragment {
    type uint8;
    description
      "IP fragmentation is an Internet Protocol (IP)
       process that breaks datagrams into smaller pieces
       (fragments), so that packets may be formed that
       can pass through a link with a smaller maximum
       transmission unit (MTU) than the original
       datagram size.";
  }
```

도면17c

```
leaf-list pkt-sec-cond-ipv4-fragment-offset {
  type uint16;
  description
    "Fragment offset field along with Don't Fragment
     and More Fragment flags in the IP protocol
     header are used for fragmentation and reassembly
     of IP datagrams.";
}

leaf-list pkt-sec-cond-ipv4-ttl {
  type uint8;
  description
    "The ttl keyword is used to check for a specific
     IP time-to-live value in the header of
     a packet.";
}

leaf-list pkt-sec-cond-ipv4-protocol {
  type uint8;
  description
    "Internet Protocol version 4(IPv4) is the fourth
     version of the Internet Protocol (IP).";
}
```

도면17d

```
leaf-list pkt-sec-cond-ipv4-src {
  type inet:ipv4-address;
  description
    "Defines the IPv4 Source Address.";
}

leaf-list pkt-sec-cond-ipv4-dest {
  type inet:ipv4-address;
  description
    "Defines the IPv4 Destination Address.";
}

leaf pkt-sec-cond-ipv4-ipopts {
  type string;
  description
    "With the ipopts keyword you can check if
     a specific ip option is set. Ipopts has
     to be used at the beginning of a rule.";
}

leaf pkt-sec-cond-ipv4-sameip {
  type boolean;
  description
    "Every packet has a source IP-address and
     a destination IP-address. It can be that
     the source IP is the same as
     the destination IP.";
```

도면17e

```
  leaf-list pkt-sec-cond-ipv4-geoip {
    type string;
    description
      "The geoip keyword enables you to match on
       the source, destination or source and destination
       IP addresses of network traffic and to see to
       which country it belongs. To do this, Suricata
       uses GeoIP API with MaxMind database format.";
  }
}

container packet-security-ipv6-condition {
  description
     "The purpose of this Class is to represent packet
     IPv6 packet header information that can be used as
     part of a test to determine if the set of Policy
     Actions in this ECA Policy Rule should be executed
     or not.";
```

도면17f

```
leaf-list pkt-sec-cond-ipv6-dscp {
  type string;
  description
    "Differentiated Services Code Point (DSCP)
     of ipv6.";
}

leaf-list pkt-sec-cond-ipv6-ecn {
  type string;
  description
    "ECN allows end-to-end notification of network
     congestion without dropping packets.";
}

leaf-list pkt-sec-cond-ipv6-traffic-class {
  type uint8;
  description
    "The bits of this field hold two values. The 6
     most-significant bits are used for
     differentiated services, which is used to
     classify packets.";
}

leaf-list pkt-sec-cond-ipv6-flow-label {
  type uint32;
  description
    "The flow label when set to a non-zero value
     serves as a hint to routers and switches
     with multiple outbound paths that these
     packets should stay on the same path so that
     they will not be reordered.";
}
```

도면17g

```
leaf-list pkt-sec-cond-ipv6-payload-length {
  type uint16;
  description
    "The size of the payload in octets,
     including any extension headers.";
}

leaf-list pkt-sec-cond-ipv6-next-header {
  type uint8;
  description
    "Specifies the type of the next header.
     This field usually specifies the transport
     layer protocol used by a packet's payload.";
}
```

도면17h

```
  leaf-list pkt-sec-cond-ipv6-hop-limit {
    type uint8;
    description
      "Replaces the time to live field of IPv4.";
  }

  leaf-list pkt-sec-cond-ipv6-src {
    type inet:ipv6-address;
    description
      "The IPv6 address of the sending node.";
  }

  leaf-list pkt-sec-cond-ipv6-dest {
    type inet:ipv6-address;
    description
      "The IPv6 address of the destination node(s).";
  }
}
```

도면17i

```
container packet-security-tcp-condition {
  description
    "The purpose of this Class is to represent packet
     TCP packet header information that can be used as
     part of a test to determine if the set of Policy
     Actions in this ECA Policy Rule should be executed
     or not.";

  leaf-list pkt-sec-cond-tcp-src-port {
    type inet:port-number;
    description
      "This is a mandatory string attribute, and
      defines the Source Port number (16 bits).";
  }

  leaf-list pkt-sec-cond-tcp-dest-port {
    type inet:port-number;
    description
      "This is a mandatory string attribute, and
      defines the Destination Port number (16 bits).";
  }

  leaf-list pkt-sec-cond-tcp-seq-num {
    type uint32;
    description
      "If the SYN flag is set (1), then this is the
       initial sequence number.";
  }
```

도면17j

```
  leaf-list pkt-sec-cond-tcp-ack-num {
    type uint32;
    description
      "If the ACK flag is set then the value of this
       field is the next sequence number that the sender
       is expecting.";
  }

  leaf-list pkt-sec-cond-tcp-window-size {
    type uint16;
    description
      "The size of the receive window, which specifies
       the number of windows size units
       (by default,bytes) (beyond the segment
       identified by the sequence number in the
       acknowledgment field) that the sender of this
       segment is currently willing to recive.";
  }

  leaf-list pkt-sec-cond-tcp-flags {
    type uint8;
    description
      "This is a mandatory string attribute, and defines
       the nine Control bit flags (9 bits).";
  }
}
```

도면17k

```
container packet-security-udp-condition {
  description
   "The purpose of this Class is to represent packet UDP
    packet header information that can be used as part
    of a test to determine if the set of Policy Actions
    in this ECA Policy Rule should be executed or not.";

  leaf-list pkt-sec-cond-udp-src-port {
    type inet:port-number;
    description
      "This is a mandatory string attribute, and
      defines the UDP Source Port number (16 bits).";
  }

  leaf-list pkt-sec-cond-udp-dest-port {
    type inet:port-number;
    description
      "This is a mandatory string attribute, and
      defines the UDP Destination Port number (16 bits).";
  }
```

도면17l

```
    leaf-list pkt-sec-cond-udp-length {
      type string;
      description
       "This is a mandatory string attribute, and defines
        the length in bytes of the UDP header and data
        (16 bits).";
    }
  }

  container packet-security-icmp-condition {
    description
      "The internet control message protocol condition.";

    leaf-list pkt-sec-cond-icmp-type {
      type uint8;
      description
        "ICMP type, see Control messages.";
    }

    leaf-list pkt-sec-cond-icmp-code {
      type uint8;
      description
        "ICMP subtype, see Control messages.";
    }

    leaf-list pkt-sec-cond-icmp-seg-num {
      type uint32;
      description
        "The icmp Sequence Number.";
    }
  }
}
```

도면17m

```
container packet-payload-condition {
  description
    "TBD";
  leaf packet-payload-description {
    type string;
    description
     "This is description for payload condition.
     Vendors can write instructions for payload condition
     that vendor made";
  }
  leaf-list pkt-payload-content {
    type string;
    description
      "The content keyword is very important in
       signatures. Between the quotation marks you
       can write on what you would like the
       signature to match.";
  }
}

leaf acl-number {
  type uint32;
  description
   "This is acl-number.";
}

container application-condition {
  description
    "TBD";
  leaf application-description {
    type string;
    description
     "This is description for application condition.";
  }
  leaf-list application-object {
    type string;
    description
      "This is application object.";
  }
  leaf-list application-group {
    type string;
    description
      "This is application group.";
  }
  leaf-list application-label {
    type string;
    description
      "This is application label.";
  }
```

도면17n

```
  container category {
    description
      "TBD";
    list application-category {
      key "name application-subcategory";
      description
        "TBD";
      leaf name {
        type string;
        description
         "This is name for application category.";
      }
      leaf application-subcategory {
        type string;
        description
         "This is application subcategory.";
      }
    }
  }
}

container target-condition {
  description
    "TBD";
  leaf target-description {
    type string;
    description
      "This is description for target condition.
      Vendors can write instructions for target condition
      that vendor made";
  }

  container device-sec-context-cond {
    description
      "The device attribute that can identify a device,
       including the device type (i.e., router, switch,
       pc, ios, or android) and the device's owner as
       well.";

    leaf pc {
      type boolean;
      description
        "If type of a device is PC.";
    }

    leaf mobile-phone {
      type boolean;
      description
        "If type of a device is mobile-phone.";
    }
```

도면17o

```
leaf voip-volte-phone {
  type boolean;
  description
    "If type of a device is voip-volte-phone.";
}

leaf tablet {
  type boolean;
  description
    "If type of a device is tablet.";
}


leaf iot {
```

도면17p

```
      type boolean;
      description
        "If type of a device is Internet of Things.";
    }

    leaf vehicle {
      type boolean;
      description
        "If type of a device is vehicle.";
    }
  }
}
container users-condition {
  description
    "TBD";
  leaf users-manual {
    type string;
    description
      "This is manual for user condition.
      Vendors can write instructions for user condition
      that vendor made";
  }
```

도면17q

```
container user{
  description
    "The user (or user group) information with which
     network flow is associated: The user has many
     attributes such as name, id, password, type,
     authentication mode and so on. Name/id is often
     used in the security policy to identify the user.
     Besides, NSF is aware of the IP address of the
     user provided by a unified user management system
     via network. Based on name-address association,
     NSF is able to enforce the security functions
     over the given user (or user group)";

  choice user-name {
    description
      "The name of the user.
       This must be unique.";

    case tenant {
      description
        "Tenant information.";

      leaf tenant {
        type uint8;
        mandatory true;
```

도면17r

```
        description
          "User's tenant information.";
      }
    }

    case vn-id {
      description
        "VN-ID information.";

      leaf vn-id {
        type uint8;
        mandatory true;
        description
          "User's VN-ID information.";
      }
    }
  }
}
```

도면17s

```
container group {
  description
    "The user (or user group) information with which
     network flow is associated: The user has many
     attributes such as name, id, password, type,
     authentication mode and so on. Name/id is often
     used in the security policy to identify the user.
     Besides, NSF is aware of the IP address of the
     user provided by a unified user management system
     via network. Based on name-address association,
     NSF is able to enforce the security functions
     over the given user (or user group)";

  choice group-name {
    description
      "The name of the user.
       This must be unique.";

    case tenant {
      description
        "Tenant information.";

      leaf tenant {
        type uint8;
        mandatory true;
        description
          "User's tenant information.";
      }
    }
```

도면17t

```
      case vn-id {
        description
          "VN-ID information.";

        leaf vn-id {
          type uint8;
          mandatory true;
          description
            "User's VN-ID information.";
        }
      }
    }
  }
  leaf security-grup {
    type string;
    mandatory true;
    description
      "security-grup.";
  }
}

container url-category-condition {
  description
    "TBD";
  leaf url-category-description {
    type string;
    description
      "This is description for url category condition.
      Vendors can write instructions for context condition
      that vendor made";
  }

  leaf-list pre-defined-category {
    type string;
    description
      "This is pre-defined-category.";
  }
  leaf-list user-defined-category {
    type string;
    description
      "This user-defined-category.";
  }
}

container context-condition {
  description
    "TBD";
  leaf context-description {
    type string;
    description
      "This is description for context condition.
      Vendors can write instructions for context condition
      that vendor made";
  }
}
```

도면17u

```
container gen-context-condition {
  description
    "TBD";
  leaf gen-context-manual {
    type string;
    description
      "This is manual for generic context condition.
      Vendors can write instructions for generic context
      condition that vendor made";
  }

  container geographic-location {
    description
      "The location where network traffic is associated
       with. The region can be the geographic location
       such as country, province, and city,
       as well as the logical network location such as
       IP address, network section, and network domain.";

    leaf-list src-geographic-location {
      type uint32;
      description
```

도면17v

```
            "This is mapped to ip address. We can acquire
             source region through ip address stored the
             database.";
        }
        leaf-list dest-geographic-location {
          type uint32;
          description
            "This is mapped to ip address. We can acquire
             destination region through ip address stored
             the database.";
        }
      }
    }
  }
}
container action-clause-container {
  description "TBD";
  list action-clause-list {
  key eca-object-id;
  uses i2nsf-eca-object-type {
    refine entity-class {
      default ECA-ACTION-TYPE;
    }
  }
  description
    "An action is used to control and monitor aspects of
     flow-based NSFs when the event and condition clauses
     are satisfied. NSFs provide security functions by
     executing various Actions. Examples of I2NSF Actions
     include providing intrusion detection and/or protection,
     web and flow filtering, and deep packet inspection
     for packets and flows.";
```

도면17w

```
leaf rule-log {
          type boolean;
          description
            "True is enable
             False is not enable.";
        }
        leaf session-log {
          type boolean;
          description
            "True is enable
             False is not enable.";
        }
        container ingress-action {
          description
            "TBD";
          leaf ingress-description {
            type string;
            description
              "This is description for ingress action.
              Vendors can write instructions for ingress action
              that vendor made";
          }
          leaf ingress-action-type {
            type ingress-action;
            description
              "Ingress action type: permit, deny, and mirror.";
```

도면17x

```
    }
  }
  container egress-action {
    description
      "TBD";
    leaf egress-manual {
      type string;
      description
        "This is manual for egress action.
        Vendors can write instructions for egress action
        that vendor made";
    }
    leaf egress-action-type {
      type egress-action;
      description
        "Egress-action-type: invoke-signaling,
         tunnel-encapsulation, and forwarding.";
    }
  }
  container apply-profile {
    description
      "TBD";
    leaf profile-manual {
      type string;
      description
        "This is manual for apply profile action.
        Vendors can write instructions for apply
        profile action that vendor made";
    }
```

도면17y

```
container content-security-control {
  description
   "Content security control is another category of
   security capabilities applied to application layer.
   Through detecting the contents carried over the
   traffic in application layer, these capabilities
   can realize various security purposes, such as
   defending against intrusion, inspecting virus,
   filtering malicious URL or junk email, and blocking
   illegal web access or data retrieval.";

  container content-security-control-types {
    description
     "Content Security types: Antivirus, IPS, IDS,
      url-filtering, data-filtering, mail-filtering,
      file-blocking, file-isolate, pkt-capture,
      application-control, and voip-volte.";
```

도면17z

```
leaf antivirus {
    type boolean;
    description
      "Additional inspection of antivirus.";
}

leaf ips {
    type boolean;
    description
      "Additional inspection of IPS.";
}

leaf ids {
    type boolean;
    description
      "Additional inspection of IDS.";
}

leaf url-filtering {
    type boolean;
    description
      "Additional inspection of URL filtering.";
}

leaf data-filtering {
   type boolean;
   description
     "Additional inspection of data filtering.";
}
```

도면18a

```
leaf mail-filtering {
  type boolean;
  description
    "Additional inspection of mail filtering.";
}

leaf file-blocking {
  type boolean;
  description
    "Additional inspection of file blocking.";
}

leaf file-isolate {
  type boolean;
  description
    "Additional inspection of file isolate.";
}
```

도면18b

```
    leaf pkt-capture {
      type boolean;
      description
        "Additional inspection of packet capture.";
    }

    leaf application-control {
      type boolean;
      description
        "Additional inspection of app control.";
    }

    leaf voip-volte {
      type boolean;
      description
        "Additional inspection of VoIP/VoLTE.";
    }
  }
}
```

도면18c

```
container attack-mitigation-control {
  description
    "This category of security capabilities is
     specially used to detect and mitigate various
     types of network attacks.";

    container ddos-attack {
      description
        "A distributed-denial-of-service (DDoS) is
         where the attack source is more than one,
         often thousands of unique IP addresses.";

      container ddos-attack-type {
        description
          "DDoS-attack types: Network Layer
          DDoS Attacks and Application Layer
          DDoS Attacks.";

        container network-layer-ddos-attack {
          description
            "Network layer DDoS-attack.";
          container network-layer-ddos-attack-type {
            description
              "Network layer DDoS attack types:
               Syn Flood Attack, UDP Flood Attack,
               ICMP Flood Attack, IP Fragment Flood,
               IPv6 Related Attacks, and etc";
```

도면18d

```
    leaf syn-flood {
      type boolean;
      description
        "Additional Inspection of
         Syn Flood Attack.";
    }

    leaf udp-flood {
      type boolean;
      description
        "Additional Inspection of
         UDP Flood Attack.";
    }

    leaf icmp-flood {
      type boolean;
      description
        "Additional Inspection of
         ICMP Flood Attack.";
    }

    leaf ip-frag-flood {
      type boolean;
      description
        "Additional Inspection of
         IP Fragment Flood.";
    }

    leaf ipv6-related {
      type boolean;
      description
        "Additional Inspection of
         IPv6 Related Attacks.";
    }
  }
}
```

도면18e

```
container app-layer-ddos-attack {
  description
    "Application layer DDoS-attack.";

  container app-ddos-attack-types {
    description
      "Application layer DDoS-attack types:
       Http Flood Attack, Https Flood Attack,
       DNS Flood Attack, and
       DNS Amplification Flood Attack,
       SSL DDoS Attack, and etc.";
```

도면18f

```
        leaf http-flood {
          type boolean;
          description
            "Additional Inspection of
             Http Flood Attack.";
        }

        leaf https-flood {
          type boolean;
          description
            "Additional Inspection of
             Https Flood Attack.";
        }

        leaf dns-flood {
          type boolean;
          description
            "Additional Inspection of
             DNS Flood Attack.";
        }

        leaf dns-amp-flood {
          type boolean;
          description
            "Additional Inspection of
             DNS Amplification Flood Attack.";
        }

        leaf ssl-ddos {
          type boolean;
          description
            "Additional Inspection of
             SSL Flood Attack.";
        }
      }
    }
  }
}
```

도면18g

```
container single-packet-attack {
  description
    "Single Packet Attacks.";
  container single-packet-attack-type {
    description
      "DDoS-attack types: Scanning Attack,
       Sniffing Attack, Malformed Packet Attack,
       Special Packet Attack, and etc.";
```

도면18h

```
container scan-and-sniff-attack {
  description
    "Scanning and Sniffing Attack.";
  container scan-and-sniff-attack-types {
    description
      "Scanning and sniffing attack types:
       IP Sweep attack, Port Scanning,
       and etc.";

    leaf ip-sweep {
      type boolean;
      description
        "Additional Inspection of
         IP Sweep Attack.";
    }

    leaf port-scanning {
      type boolean;
      description
        "Additional Inspection of
         Port Scanning Attack.";
    }
  }
}
```

도면18i

```
container malformed-packet-attack {
  description
    "Malformed Packet Attack.";
  container malformed-packet-attack-types {
    description
      "Malformed packet attack types:
       Ping of Death Attack, Teardrop Attack,
       and etc.";

    leaf ping-of-death {
      type boolean;
      description
        "Additional Inspection of
         Ping of Death Attack.";
    }

    leaf teardrop {
      type boolean;
      description
        "Additional Inspection of
         Teardrop Attack.";
    }
  }
```

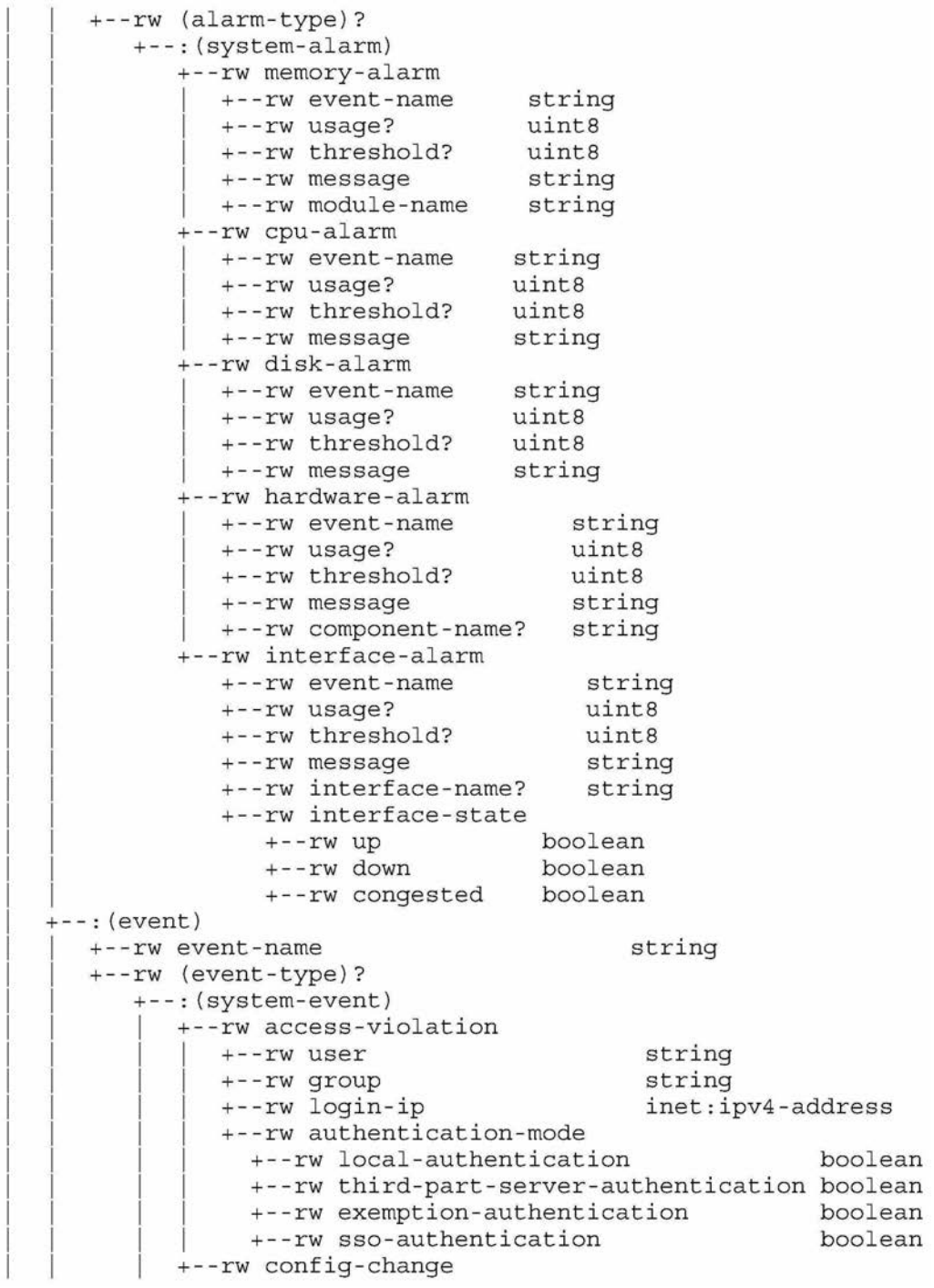
도면18j

```
                    }

                    container special-packet-attack {
                      description
                        "special Packet Attack.";
                      container special-packet-attack-types {
                        description
                          "Special packet attack types:
                           Oversized ICMP Attack, Tracert Attack,
                           and etc.";

                        leaf oversized-icmp {
                          type boolean;
                          description
                            "Additional Inspection of
                             Oversize ICMP Attack.";
                        }

                        leaf tracert {
                          type boolean;
                          description
                            "Additional Inspection of
                             Tracrt Attack.";
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

도면19a

```
module: ietf-i2nsf-nsf-monitoring-dm
  +--rw monitoring-message
    +--rw monitoring-messages* [message-id]
       +--rw message-id                    uint8
       +--rw message-version               uint8
       +--rw (message-type)?
       |  +--:(alarm)
```

도면19b


+--rw (alarm-type)?
+--:(system-alarm)
+--rw memory-alarm
+--rw event-name string
+--rw usage? uint8
+--rw threshold? uint8
+--rw message string
+--rw module-name string
+--rw cpu-alarm
+--rw event-name string
+--rw usage? uint8
+--rw threshold? uint8
+--rw message string
+--rw disk-alarm
+--rw event-name string
+--rw usage? uint8
+--rw threshold? uint8
+--rw message string
+--rw hardware-alarm
+--rw event-name string
+--rw usage? uint8
+--rw threshold? uint8
+--rw message string
+--rw component-name? string
+--rw interface-alarm
+--rw event-name string
+--rw usage? uint8
+--rw threshold? uint8
+--rw message string
+--rw interface-name? string
+--rw interface-state
+--rw up boolean
+--rw down boolean
+--rw congested boolean
+--:(event)
+--rw event-name string
+--rw (event-type)?
+--:(system-event)
+--rw access-violation
+--rw user string
+--rw group string
+--rw login-ip inet:ipv4-address
+--rw authentication-mode
+--rw local-authentication boolean
+--rw third-part-server-authentication boolean
+--rw exemption-authentication boolean
+--rw sso-authentication boolean
+--rw config-change

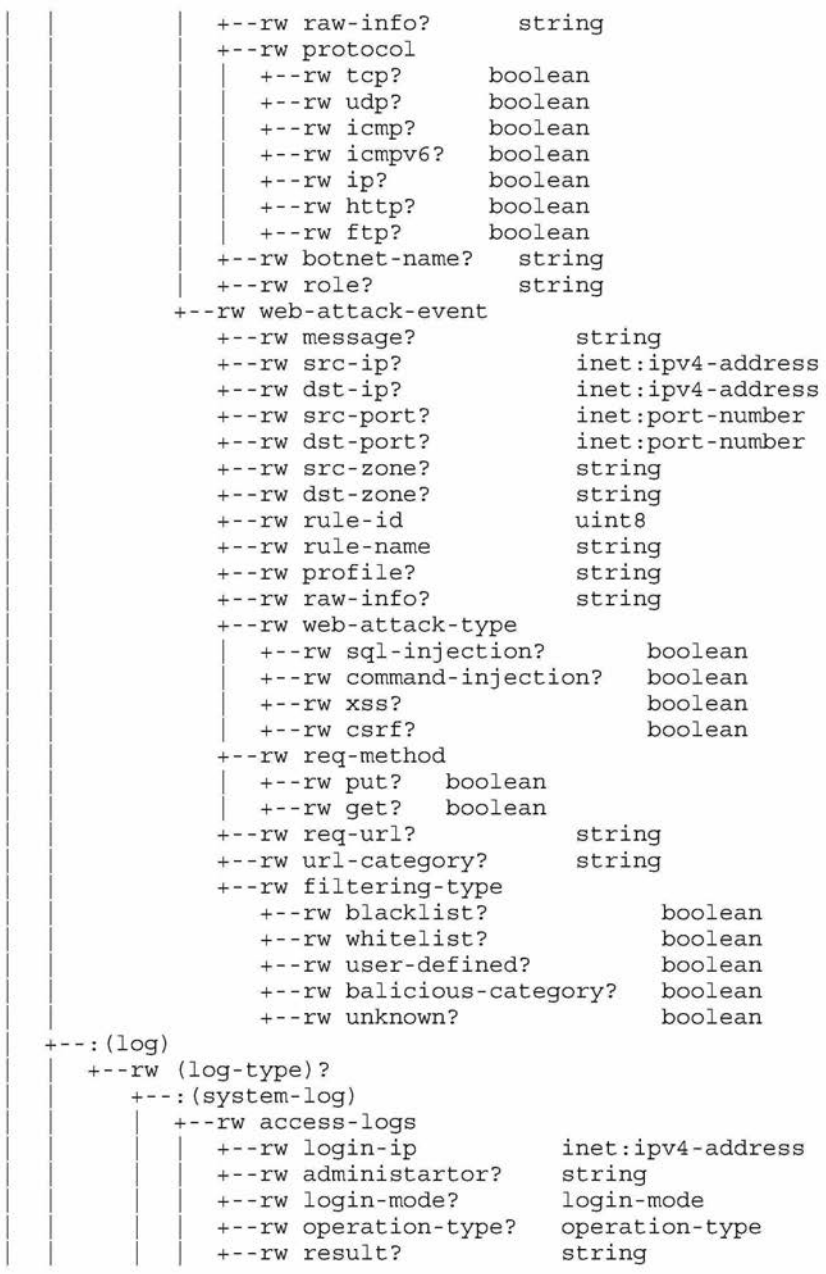
도면19c

```
|   |       |      +--rw user                        string
|   |       |      +--rw group                       string
|   |       |      +--rw login-ip                    inet:ipv4-address
|   |       |      +--rw authentication-mode
|   |       |        +--rw local-authentication                boolean
|   |       |        +--rw third-part-server-authentication    boolean
|   |       |        +--rw exemption-authentication            boolean
|   |       |        +--rw sso-authentication                  boolean
|   |       +--:(nsf-event)
|   |          +--rw ddos-event
|   |             |  +--rw message?              string
|   |             |  +--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 rule-id               uint8
|   |             |  +--rw rule-name             string
|   |             |  +--rw profile?              string
|   |             |  +--rw raw-info?             string
|   |             |  +--rw ddos-attack-type
|   |             |  |  +--rw syn-flood?                boolean
|   |             |  |  +--rw ack-flood?                boolean
|   |             |  |  +--rw syn-ack-flood?            boolean
|   |             |  |  +--rw fin-rst-flood?            boolean
|   |             |  |  +--rw tcp-connection-flood?     boolean
|   |             |  |  +--rw udp-flood?                boolean
|   |             |  |  +--rw icmp-flood?               boolean
|   |             |  |  +--rw https-flood?              boolean
|   |             |  |  +--rw http-flood?               boolean
|   |             |  |  +--rw dns-reply-flood?          boolean
|   |             |  |  +--rw dns-query-flood?          boolean
|   |             |  |  +--rw sip-flood?                boolean
|   |             |  +--rw start-time            yang:date-and-time
|   |             |  +--rw end-time              yang:date-and-time
|   |             |  +--rw attack-rate?          uint32
|   |             |  +--rw attack-speed?         uint32
|   |             +--rw session-table-event
|   |             |  +--rw current-session?   uint8
|   |             |  +--rw maximum-session?   uint8
|   |             |  +--rw threshold?         uint8
|   |             |  +--rw message?           string
|   |             +--rw virus-event
|   |             |  +--rw message?       string
|   |             |  +--rw src-ip?        inet:ipv4-address
|   |             |  +--rw dst-ip?        inet:ipv4-address
|   |             |  +--rw src-port?      inet:port-number
```

도면19d

```
|  |  |  +--rw dst-port?      inet:port-number
|  |  |  +--rw src-zone?      string
|  |  |  +--rw dst-zone?      string
|  |  |  +--rw rule-id        uint8
|  |  |  +--rw rule-name      string
|  |  |  +--rw profile?       string
|  |  |  +--rw raw-info?      string
|  |  |  +--rw virus-type
|  |  |  |  +--rw trajan?   boolean
|  |  |  |  +--rw worm?     boolean
|  |  |  |  +--rw macro?    boolean
|  |  |  +--rw virus-name?   string
|  |  |  +--rw file-type?    string
|  |  |  +--rw file-name?    string
|  |  +--rw intrusion-event
|  |  |  +--rw message?                  string
|  |  |  +--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 rule-id                   uint8
|  |  |  +--rw rule-name                 string
|  |  |  +--rw profile?                  string
|  |  |  +--rw raw-info?                 string
|  |  |  +--rw protocol
|  |  |  |  +--rw tcp?      boolean
|  |  |  |  +--rw udp?      boolean
|  |  |  |  +--rw icmp?     boolean
|  |  |  |  +--rw icmpv6?   boolean
|  |  |  |  +--rw ip?       boolean
|  |  |  |  +--rw http?     boolean
|  |  |  |  +--rw ftp?      boolean
|  |  |  +--rw intrusion-attack-type
|  |  |     +--rw brutal-force?      boolean
|  |  |     +--rw buffer-overflow?   boolean
|  |  +--rw botnet-event
|  |  |  +--rw message?       string
|  |  |  +--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 rule-id        uint8
|  |  |  +--rw rule-name      string
|  |  |  +--rw profile?       string
```

도면19e


+--rw raw-info? string
+--rw protocol
+--rw tcp? boolean
+--rw udp? boolean
+--rw icmp? boolean
+--rw icmpv6? boolean
+--rw ip? boolean
+--rw http? boolean
+--rw ftp? boolean
+--rw botnet-name? string
+--rw role? string
+--rw web-attack-event
+--rw message? string
+--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 rule-id uint8
+--rw rule-name string
+--rw profile? string
+--rw raw-info? string
+--rw web-attack-type
+--rw sql-injection? boolean
+--rw command-injection? boolean
+--rw xss? boolean
+--rw csrf? boolean
+--rw req-method
+--rw put? boolean
+--rw get? boolean
+--rw req-url? string
+--rw url-category? string
+--rw filtering-type
+--rw blacklist? boolean
+--rw whitelist? boolean
+--rw user-defined? boolean
+--rw balicious-category? boolean
+--rw unknown? boolean
+--:(log)
+--rw (log-type)?
+--:(system-log)
+--rw access-logs
+--rw login-ip inet:ipv4-address
+--rw administartor? string
+--rw login-mode? login-mode
+--rw operation-type? operation-type
+--rw result? string

도면19f

```
|  |  |  |  +--rw content?                string
|  |  |  +--rw resource-utiliz-logs
|  |  |  |  +--rw system-status?        string
|  |  |  |  +--rw cpu-usage?            uint8
|  |  |  |  +--rw memory-usage?         uint8
|  |  |  |  +--rw disk-usage?           uint8
|  |  |  |  +--rw disk-left?            uint8
|  |  |  |  +--rw session-num?          uint8
|  |  |  |  +--rw process-num?          uint8
|  |  |  |  +--rw in-traffic-rate?      uint32
|  |  |  |  +--rw out-traffic-rate?     uint32
|  |  |  |  +--rw in-traffic-speed?     uint32
|  |  |  |  +--rw out-traffic-speed?    uint32
|  |  |  +--rw user-activity-logs
|  |  |     +--rw user                      string
|  |  |     +--rw group                     string
|  |  |     +--rw login-ip                  inet:ipv4-address
|  |  |     +--rw authentication-mode
|  |  |     |  +--rw local-authentication             boolean
|  |  |     |  +--rw third-part-server-authentication boolean
|  |  |     |  +--rw exemption-authentication         boolean
|  |  |     |  +--rw sso-authentication               boolean
|  |  |     +--rw access-mode
|  |  |     |  +--rw ppp?     boolean
|  |  |     |  +--rw svn?     boolean
|  |  |     |  +--rw local?   boolean
|  |  |     +--rw online-duration?          string
|  |  |     +--rw logout-duration?          string
|  |  |     +--rw addtional-info?           string
|  |  |     +--rw cause?                    string
|  |  +--:(nsf-log)
|  |     +--rw ddos-logs
|  |     |  +--rw attack-type?          string
|  |     |  +--rw attack-ave-rate?      uint32
|  |     |  +--rw attack-ave-speed?     uint32
|  |     |  +--rw attack-pkt-num?       uint32
|  |     |  +--rw attack-src-ip?        inet:ipv4-address
|  |     |  +--rw action?               all-action
|  |     |  +--rw os?                   string
|  |     +--rw virus-logs
|  |     |  +--rw protocol
|  |     |  |  +--rw tcp?      boolean
|  |     |  |  +--rw udp?      boolean
|  |     |  |  +--rw icmp?     boolean
|  |     |  |  +--rw icmpv6?   boolean
|  |     |  |  +--rw ip?       boolean
|  |     |  |  +--rw http?     boolean
|  |     |  |  +--rw ftp?      boolean
```

도면19g

```
|   |         |   +--rw attack-type?    string
|   |         |   +--rw action?         all-action
|   |         |   +--rw os?             string
|   |         |   +--rw time            yang:date-and-time
|   |         +--rw intrusion-logs
|   |         |   +--rw attack-type?     string
|   |         |   +--rw action?          all-action
|   |         |   +--rw time             yang:date-and-time
|   |         |   +--rw attack-rate?     uint32
|   |         |   +--rw attack-speed?    uint32
|   |         +--rw botnet-logs
|   |         |   +--rw attack-type?       string
|   |         |   +--rw botnet-pkt-num?    uint8
|   |         |   +--rw action?            all-action
|   |         |   +--rw os?                string
|   |         +--rw dpi-logs
|   |         |   +--rw dpi-type?       dpi-type
|   |         |   +--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
|   |         |   |  +--rw tcp?       boolean
|   |         |   |  +--rw udp?       boolean
|   |         |   |  +--rw icmp?      boolean
|   |         |   |  +--rw icmpv6?    boolean
|   |         |   |  +--rw ip?        boolean
|   |         |   |  +--rw http?      boolean
|   |         |   |  +--rw ftp?       boolean
|   |         |   +--rw file-type?      string
|   |         |   +--rw file-name?      string
|   |         +--rw vulnerability-scanning-logs* [vulnerability-id]
|   |         |   +--rw vulnerability-id    uint8
|   |         |   +--rw victim-ip?          inet:ipv4-address
|   |         |   +--rw protocol
|   |         |   |  +--rw tcp?       boolean
|   |         |   |  +--rw udp?       boolean
|   |         |   |  +--rw icmp?      boolean
|   |         |   |  +--rw icmpv6?    boolean
|   |         |   |  +--rw ip?        boolean
|   |         |   |  +--rw http?      boolean
```

도면19h

```
|   |         |   |  +--rw ftp?       boolean
|   |         |   +--rw port-num?           inet:port-number
|   |         |   +--rw level?              severity
|   |         |   +--rw os?                 string
|   |         |   +--rw additional-info?    string
|   |         +--rw web-attack-logs
|   |            +--rw attack-type?     string
|   |            +--rw rsp-code?        string
|   |            +--rw req-clientapp?   string
|   |            +--rw req-cookies?     string
|   |            +--rw req-host?        string
|   |            +--rw raw-info?        string
|   +--:(counters)
|      +--rw (counter-type)?
|         +--:(system-counter)
|         |  +--rw interface-counters
|         |     +--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
|         +--:(nsf-counter)
|         |  +--rw firewall-counters
|         |     +--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
```

도면19i

```
|          |      |  +--rw tcp?       boolean
|          |      |  +--rw udp?       boolean
|          |      |  +--rw icmp?      boolean
|          |      |  +--rw icmpv6?    boolean
|          |      |  +--rw ip?        boolean
|          |      |  +--rw http?      boolean
|          |      |  +--rw ftp?       boolean
|          |      +--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 bound
|          |         +--rw in-interface?    boolean
|          |         +--rw out-interface?   boolean
|          +--:(policy-hit-counters)
|             +--rw policy-hit-counters
|                +--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
|                |  +--rw tcp?       boolean
|                |  +--rw udp?       boolean
|                |  +--rw icmp?      boolean
|                |  +--rw icmpv6?    boolean
|                |  +--rw ip?        boolean
|                |  +--rw http?      boolean
|                |  +--rw ftp?       boolean
|                +--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
```

도면19j

```
|                +--rw out-traffic-peak-speed?   uint32
|                +--rw hit-times?                uint32
+--rw message                          string
+--rw time-stamp                       yang:date-and-time
+--rw vendor-name?                     string
+--rw nsf-name?                        string
+--rw severity                         severity
```

도면20a

```
<CODE BEGINS> file "ietf-i2nsf-nsf-monitoring-dm@2018-03-05.yang"

module ietf-i2nsf-nsf-monitoring-dm {
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-i2nsf-nsf-monitoring-dm";
  prefix
    monitoring-information;
  import ietf-inet-types{
    prefix inet;
  }
  import ietf-yang-types {
    prefix yang;
  }
  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: Dongjin Hong
     <mailto:dong.jin@skku.edu>

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

  description
    "This module defines a YANG data module for monitoring NSFs.";
```

도면20b

```
revision "2017-10-29" {
  description "Initial revision";
  reference
    "draft-zhang-i2nsf-info-model-monitoring-04";
}

typedef severity {
  type enumeration {
    enum high {
      description
        "high-level";
    }
    enum middle {
      description
        "middle-level";
    }
    enum low {
      description
        "low-level";
    }
  }
  description
    "This is used for indicating the severity";
}
typedef all-action {
  type enumeration {
    enum allow {
      description
        "If action is allow";
```

도면20c

```
}
enum alert {
  description
    "If action is alert";
}
enum block {
  description
    "If action is block";
}
enum discard {
  description
    "If action is discard";
}
enum declare {
  description
    "If action is declare";
}
enum block-ip {
  description
```

도면20d

```
        "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";
}
```

도면20e

```
typedef operation-type{
  type enumeration {
    enum login{
      description
        "Login operation";
    }
    enum logout{
      description
        "Logout operation";
    }
    enum configuration{
      description
        "Configuration operation";
    }
  }
  description
    "This is used for operation type";
}
typedef login-mode{
  type enumeration {
```

도면20f

```
    enum root{
      description
        "Root login-mode";
    }
    enum user{
      description
        "User login-mode";
    }
    enum guest{
      description
        "Guest login-mode";
    }
  }
  description
    "This is used for login mode";
}
grouping protocol {
  description
    "A set of protocols";
  container protocol {
    description
    "Protocol types:
     TCP, UDP, ICMP, ICMPv6, IP, HTTP, FTP and etc.";
    leaf tcp {
      type boolean;
      description
        "TCP protocol type.";
    }
    leaf udp {
      type boolean;
      description
        "UDP protocol type.";
    }
    leaf icmp {
      type boolean;
      description
        "ICMP protocol type.";
    }
    leaf icmpv6 {
      type boolean;
      description
        "ICMPv6 protocol type.";
    }
    leaf ip {
      type boolean;
      description
        "IP protocol type.";
    }
```

도면20g

```
    leaf http {
      type boolean;
      description
        "HTTP protocol type.";
    }
    leaf ftp {
      type boolean;
      description
        "ftp protocol type.";
    }
  }
}
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;
```

도면20h

```
    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 authentication-mode{
  description
  "A set of authentication-mode";
   container authentication-mode {
     description
       "User authentication mode. e.g., Local Authentication,
       Third-Party Server Authentication,
       Authentication Exemption, SSO Authentication.";
     leaf local-authentication {
       type boolean;
       mandatory true;
       description
         "Authentication-mode : local authentication.";
     }
     leaf third-part-server-authentication {
       type boolean;
       mandatory true;
       description
         "If authentication-mode is
         third-part-server-authentication";
     }
     leaf exemption-authentication {
       type boolean;
       mandatory true;
       description
         "If authentication-mode is
         exemption-authentication";
     }
     leaf sso-authentication {
       type boolean;
       mandatory true;
       description
         "If authentication-mode is
         sso-authentication";
```

도면20i

```
      }
    }
}
grouping i2nsf-system-alarm-type-content {
  description
    "A set of system alarm type contents";
  leaf event-name {
    type string;
    mandatory true;
    description
      "This is used to distinguish event type";
  }
  leaf usage {
    type uint8;
    description
      "specifies the amount of usage";
  }
  leaf threshold {
    type uint8;
    description
      "The threshold triggering the alarm or the event";
  }
  leaf message {
    type string;
    mandatory true;
    description
      "The usage exceeded the threshold";
  }
}
grouping i2nsf-system-event-type-content {
  description
    "A set of system event type contents";
  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 {
    type inet:ipv4-address;
    mandatory true;
    description
```

도면20j

```
      "Login IP address of a user.";
  }
  uses authentication-mode;
}
grouping i2nsf-nsf-event-type-content {
  description
    "A set of nsf event type contents";
  leaf message {
    type string;
    description
      "The message for nsf events";
  }
  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";
  }
```

도면20k

```
    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-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;
```

도면201

```
    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";
  }
```

도면20m

```
  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;
    mandatory true;
    description
      "The ID of the policy being triggered";
  }
  leaf policy-name {
    type string;
    mandatory true;
    description
      "The name of the policy being triggered";
  }
  leaf src-user{
    type string;
    description
      "User who generates traffic";
  }
  uses protocol;
  uses traffic-rates;
}

container monitoring-message {
  description
    "The message for monitoring information";
  list monitoring-messages {
    key message-id;
    description
      "The messages according to monitoring information";
    leaf message-id {
      type uint8;
      mandatory true;
      description
        "This is message ID
```

도면20n

```
      This is key for monitoring messages";
}
leaf message-version {
  type uint8;
  mandatory true;
  description
    "The version of message";
}
choice message-type {
  description
    "The type of message";
  case alarm {
    description
      "If the message type is alarm";
    choice alarm-type {
      description
        "This is alarm type such as system alarm";
      case system-alarm{
        description
          "If the alarm type is system alarm";
        container memory-alarm {
          description
            "This is memory alarm in
            system alarm";
          uses i2nsf-system-alarm-type-content;
          leaf module-name {
            type string;
            mandatory true;
            description
              "Indicate the NSF module
              responsible for generating
              this alarm";
          }
        }
        container cpu-alarm {
          description
            "This is cpu alarm in system alarm";
          uses i2nsf-system-alarm-type-content;
        }
        container disk-alarm {
          description
            "This is disk alarm in system alarm";
          uses i2nsf-system-alarm-type-content;
        }
        container hardware-alarm {
          description
            "This is hardware alarm
            in system alarm";
```

도면20o

```
        uses i2nsf-system-alarm-type-content;
        leaf component-name {
          type string;
          description
            "Indicate the HW component
            responsible for generating
            this alarm.";
        }
      }
      container interface-alarm {
        description
          "This is interface alarm
          in system alarm";
        uses i2nsf-system-alarm-type-content;
        leaf interface-name {
          type string;
          description
            "The name of interface.";
        }
        container interface-state {
          description
            "This is isnteface state
            in interface-alarm";
          leaf up {
            type boolean;
            mandatory true;
            description
             "The state of interface is up";
          }
          leaf down {
            type boolean;
            mandatory true;
            description
            "The state of interface is down";
          }
          leaf congested {
            type boolean;
            mandatory true;
            description
             "The state of interface is
             congested";
          }
        }
      }
    }
  }
}
case event {
```

도면20p

```
description
  "If the message type is event";
leaf event-name {
  type string;
  mandatory true;
  description
    "The name of the event";
}
choice event-type {
  description
    "This is event type such as system event
    and nsf event.";
  case system-event {
    description
      "If the event type is system event";
    container access-violation {
      description
        "If the system event is
        access violation";
      uses i2nsf-system-event-type-content;
    }
    container config-change {
      description
        "If the system event is
        config change violation";
      uses i2nsf-system-event-type-content;
    }
  }
  case nsf-event {
    description
      "If the event type is nsf event";
    container ddos-event {
      description
        "If the event type is DDoS event";
      uses i2nsf-nsf-event-type-content;
      container ddos-attack-type{
        description
          "Type of DDoS attack";
        leaf syn-flood{
          type boolean;
          description
            "If the DDoS attack is
            syn flood";
        }
        leaf ack-flood{
          type boolean;
          description
            "If the DDoS attack is
```

도면20q

```
    ack flood";
}
leaf syn-ack-flood{
  type boolean;
  description
    "If the DDoS attack is
    syn ack flood";
}
leaf fin-rst-flood{
  type boolean;
  description
    "If the DDoS attack is
    fin rst flood";
}
leaf tcp-connection-flood{
  type boolean;
  description
    "If the DDoS attack is
    tcp connection flood";
}
leaf udp-flood{
  type boolean;
  description
    "If the DDoS attack is
    udp flood";
}
leaf icmp-flood{
  type boolean;
  description
    "If the DDoS attack is
    icmp flood";
}
leaf https-flood{
  type boolean;
  description
    "If the DDoS attack is
    https flood";
}
leaf http-flood{
  type boolean;
  description
    "If the DDoS attack is
    http flood";
}
leaf dns-reply-flood{
  type boolean;
  description
    "If the DDoS attack is
```

도면20r

```
        dns reply flood";
    }
    leaf dns-query-flood{
      type boolean;
      description
        "If the DDoS attack is
        dns query flood";
    }
    leaf sip-flood{
      type boolean;
      description
        "If the DDoS attack is
        sip flood";
    }
  }
  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 of attack traffic";
  }
  leaf attack-speed {
    type uint32;
    description
      "the bps of attack traffic";
  }
}
container session-table-event {
  description
    "If the event type is session
    table event";
    leaf current-session {
      type uint8;
      description
       "The number of concurrent
```

도면20s

```
        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";
    }
    leaf message {
      type string;
      description
       "The number of session table
        exceeded the threshold";
    }
}
container virus-event {
  description
    "If the event type is virus event";
  uses i2nsf-nsf-event-type-content;
  container virus-type {
    description
      "The type of virus";
    leaf trajan {
      type boolean;
      description
       "If the virus type is trajan";
    }
    leaf worm {
      type boolean;
      description
        "If the virus type is worm";
    }
    leaf macro {
      type boolean;
      description
        "If the virus type is macro";
    }
  }
  leaf virus-name {
    type string;
    description
```

도면20t

```
        "The name of virus";
    }
    leaf file-type {
      type string;
      description
        "The type of file";
    }
    leaf file-name {
      type string;
      description
        "The name of file";
    }
  }
  container intrusion-event {
    description
     "If the event type is intrusion event";
    uses i2nsf-nsf-event-type-content;
    uses protocol;
    container intrusion-attack-type {
      description
        "The attack type of intrusion";
      leaf brutal-force {
        type boolean;
        description
          "The intrusion type is
          brutal force";
      }
      leaf buffer-overflow {
        type boolean;
        description
          "The intrusion type is
          buffer overflow";
      }
    }
  }
  container botnet-event {
    description
      "If the event type is botnet event";
    uses i2nsf-nsf-event-type-content;
    uses protocol;
    leaf botnet-name {
      type string;
      description
       "The name of the detected botnet";
    }
    leaf role {
      type string;
      description
```

도면20u

```
      "The role of the communicating
       parties within the botnet";
  }
}
container web-attack-event {
  description
    "If the event type is web
    attack event";
  uses i2nsf-nsf-event-type-content;
  container web-attack-type {
    description
      "To determine the attack
      type";
    leaf sql-injection {
      type boolean;
      description
        "If the web attack type is
        sql injection";
    }
    leaf command-injection {
      type boolean;
      description
        "If the web attack type is
        command injection";
    }
    leaf xss {
      type boolean;
      description
        "If the web attack type is
        xss injection";
    }
    leaf csrf {
      type boolean;
      description
        "If the web attack type is
        csrf injection";
    }
  }
  container req-method {
    description
     "The method of requirement.
      For instance, PUT or GET
      in HTTP";
    leaf put{
      type boolean;
      description
        "If req method is PUT";
    }
```

도면20v

```
  leaf get {
   type boolean;
    description
      "If req method is GET";
  }
}
leaf req-url {
  type string;
  description
    "Requested URL";
}
leaf url-category {
  type string;
  description
    "Matched URL category";
}
container filtering-type {
  description
   "URL filtering type,
    e.g., Blacklist, Whitelist,
    User-Defined, Predefined,
    Malicious Category, Unknown";
  leaf blacklist {
    type boolean;
    description
      "The filtering type is
      blacklist";
  }
  leaf whitelist {
    type boolean;
    description
      "The filtering type is
      whitelist";
  }
  leaf user-defined {
    type boolean;
    description
      "The filtering type is
      user defined";
  }
  leaf balicious-category{
    type boolean;
    description
      "The filtering type is
      balicious category";
  }
  leaf unknown {
    type boolean;
```

도면20

```
            description
              "The filtering type is
              unknown";
          }
        }
      }
    }
  }
}
case log {
  description
    "If the message type is log";
  choice log-type {
    description
      "The type of log";
    case system-log{
      description
        "If the log type is system log";
      container access-logs {
        description
          "If the log is access logs
          in system log";
        leaf login-ip {
          type inet:ipv4-address;
          mandatory true;
          description
            "Login IP address of a user.";
        }
        leaf administartor {
          type string;
          description
            "Administrator that
            operates on the device";
        }
        leaf login-mode {
          type login-mode;
          description
            "Specifies the
            administrator logs in mode";
        }
        leaf operation-type {
          type operation-type;
          description
            "The operation type that
            the administrator execute";
        }
        leaf result {
          type string;
```

도면20x

```
    description
      "Command execution result";
  }
  leaf content {
    type string;
    description
      "Operation performed by
      an administrator after login.";
  }
}
container resource-utiliz-logs {
  description
    "If the log is resource utilize
    logs in system log";
  leaf system-status {
    type string;
    description
      "Running status of
      current system";
  }
  leaf cpu-usage {
    type uint8;
    description
      "specifies the amount of
      cpu usage";
  }
  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
```

도면20y

```
      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";
  }
}
container user-activity-logs {
  description
    "If the log is user activity
    logs in system log";
  leaf user {
    type string;
    mandatory true;
    description
      "Name of a user";
  }
  leaf group {
    type string;
    mandatory true;
    description
      "Group to which a user belongs.";
```

도면20z

```
}
leaf login-ip {
  type inet:ipv4-address;
  mandatory true;
  description
    "Login IP address of a user.";
}
uses authentication-mode;
container access-mode {
  description
    "User access mode. e.g., PPP, SVN, LOCAL";
  leaf ppp{
    type boolean;
    description
      "Access-mode : ppp";
  }
  leaf svn{
    type boolean;
    description
      "Access-mode : svn";
  }
  leaf local{
    type boolean;
    description
      "Access-mode : local";
  }
}
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";
}
leaf cause{
```

도면21a

```
      type string;
      description
         "Cause of a failed user activity";
      }
    }
}
case nsf-log{
  description
    "If the log type is nsf log";
  container ddos-logs {
    description
      "If the log is DDoS logs
      in nsf log";
    leaf attack-type{
      type string;
      description
        "DDoS";
    }
    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 all-action;
      description
```

도면21b

```
      "Action type: allow, alert,
      block, discard, declare,
      block-ip, block-service";
  }
  leaf os  {
    type string;
    description
      "simple os information";
  }
}
container virus-logs {
  description
    "If the log is virus logs
    in nsf log";
  uses protocol;
  leaf attack-type{
    type string;
    description
      "Virus";
  }
  leaf action{
    type all-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";
  }
}
container intrusion-logs {
  description
    "If the log is intrusion logs
    in nsf log";
  leaf attack-type{
    type string;
    description
      "Intrusion";
```

도면21c

```
  }
  leaf action{
    type all-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";
  }
}
container botnet-logs {
  description
    "If the log is botnet logs
    in nsf log";
  leaf attack-type{
    type string;
    description
      "Botnet";
  }
  leaf botnet-pkt-num{
    type uint8;
    description
      "The number of the packets
      sent to or from the
      detected botnet";
  }
  leaf action{
    type all-action;
    description
      "Action type: allow, alert,
      block, discard, declare,
      block-ip, block-service";
```

도면21d

```
  }
  leaf os{
    type string;
    description
      "simple os information";
  }
}
container dpi-logs {
  description
    "If the log is dpi logs
    in nsf log";
  leaf dpi-type{
    type dpi-type;
    description
      "The type of dpi";
  }
  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;
```

도면21e

```
    description
      "Source region of the traffic";
  }
  leaf dst-region{
    type string;
    description
      "Destination region of the traffic";
  }
  leaf policy-id {
    type uint8;
    mandatory true;
    description
      "The ID of the policy being triggered";
  }
  leaf policy-name {
    type string;
    mandatory true;
    description
      "The name of the policy being triggered";
  }
  leaf src-user{
    type string;
    description
      "User who generates traffic";
  }
  uses protocol;
  leaf file-type {
    type string;
    description
      "The type of file";
  }
  leaf file-name {
    type string;
    description
      "The name of file";
  }
}
list vulnerability-scanning-logs {
  key vulnerability-id;
  description
    "If the log is vulnerability
    scanning logs in nsf log";
  leaf vulnerability-id{
    type uint8;
    description
      "The vulnerability id";
  }
  leaf victim-ip {
```

도면21f

```
    type inet:ipv4-address;
    description
      "IP address of the victim
      host which has vulnerabilities";
  }
  uses protocol;
  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 additional-info{
    type string;
    description
      "Additional-info for logs.
      It includes The fix suggestion
      to the vulnerability.";
  }
}
container web-attack-logs {
  description
    "If the log is web attack
    logs in nsf log";
  leaf attack-type{
    type string;
    description
      "Web Attack";
  }
  leaf rsp-code{
    type string;
    description
      "Response code";
  }
  leaf req-clientapp{
    type string;
    description
      "The client application";
  }
```

도면21g

```
        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.";
        }
      }
    }
  }
}
case counters {
  description
    "If the message type is counters";
  choice counter-type {
    description
      "The type of counter";
    case system-counter {
      container interface-counters {
        description
          "The system counter type is
          interface counter";
        uses i2nsf-system-counter-type-content;
      }
    }
    case nsf-counter{
      container firewall-counters {
        description
          "The nsf counter type is
          firewall counter";
        uses i2nsf-nsf-counters-type-content;
        container bound{
          description
          "Inbound or Outbound";
          leaf in-interface {
            type boolean;
            description
```

도면21h

```
                "If the bound is inbound";
            }
            leaf out-interface {
              type boolean;
              description
                "If the bound is outbound";
            }
          }
        }
      }
      container policy-hit-counters {
        description
          "The counters of policy hit";
        uses i2nsf-nsf-counters-type-content;
        leaf hit-times{
          type uint32;
          description
            "The hit times for policy";
        }
      }
    }
  }
}
leaf message {
  type string;
  mandatory true;
  description
    "This is a message for monitoring information";
}
leaf time-stamp {
  type yang:date-and-time;
  mandatory true;
  description
    "Indicate the time when the message is generated";
}
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 severity {
  type severity;
```

도면21i

```
        mandatory true;
        description
          "The severity of the alarm such as
          critical, high, middle, low.";
      }
    }
  }
}
<CODE ENDS>
```

(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2018년06월01일
(11) 등록번호 10-1863236
(24) 등록일자 2018년05월25일

(51) 국제특허분류(Int. Cl.)
H04L 29/06 (2006.01)
(52) CPC특허분류
H04L 63/20 (2013.01)
(21) 출원번호 10-2017-0079120
(22) 출원일자 2017년06월22일
심사청구일자 2017년07월25일
(30) 우선권주장
1020160163114 2016년12월01일 대한민국(KR)
(56) 선행기술조사문헌
Framework for Interface to Network Security Functions fraft-ietf-i2nsf-framework-04(I2NSF internet draft, 2016.10.30.)*
I2NSF Data Model of Consumer-Facing Interface for Security Management draft-jeong-i2nsf-consumer-facing-interface-dm-00(Network Working Group Internet-Draft, 2016.11.13.)*
WO2016000160 A1
KR101669518 B1
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
성균관대학교산학협력단
경기도 수원시 장안구 서부로 2066 (천천동, 성균관대학교내)
(72) 발명자
정재훈
부산광역시 금정구 금강로 225 장전동 717 벽산블루밍장전디자인시티 204동 1501호
김형식
경기도 수원시 장안구 화산로 85 천천동 333 천천푸르지오아파트 132-401
(뒷면에 계속)
(74) 대리인
특허법인로알

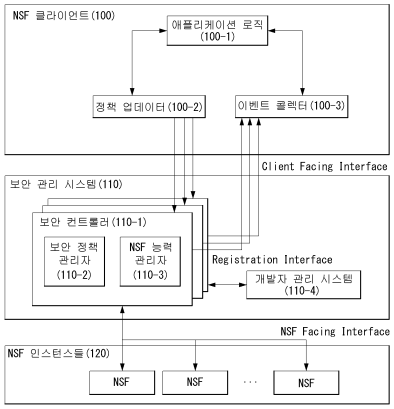
전체 청구항 수 : 총 18 항 심사관 : 남기영

(54) 발명의 명칭 **네트워크 가상화 환경에서 보안 관리를 위한 장치 및 방법**

(57) 요 약

본 발명은 네트워크 가상화 기반의 네트워크 보안 기능(Network Security Function, NSF) 제공에 관한 I2NSF (Interface to Network Security Functions) 프레임워크에서 NSF 클라이언트가 효과적으로 NSF의 보안 관리를 실현하기 위한 구조를 제시한다. 제안하는 구조를 통해 NSF 클라이언트가 직접 상위 수준의 보안 정책을 수립하고 NSF에서 발생하는 이벤트를 피드백 받음으로써 결과적으로 효과적인 보안 관리를 실현할 수 있다. 본 발명에서는 I2NSF 프레임워크에서 NSF 클라이언트의 보안 관리를 위해 추가되는 구성 요소들이 어떤 기능들을 수행하는지를 기술한다.

대 표 도 - 도1



(72) 발명자

오상학

경기도 수원시 장안구 서부로 2066 천천동 300 성균관대학교자연과학캠퍼스 38-22 (율전동) 슬기샘 302호

김은수

경기도 수원시 장안구 서부로 2066 천천동 300 성균관대학교자연과학캠퍼스 26-35, 205호[율전동 439-10]

이 발명을 지원한 국가연구개발사업

과제고유번호 2016-0-00078
부처명 정부)미래창조과학부
연구관리전문기관 정보통신기술진흥센터
연구사업명 정보통신.방송 기술개발사업 및 표준화사업
연구과제명 [EZ-IITP]맞춤형 보안서비스 제공을 위한 클라우드 기반 지능형 보안 기술 개발
기 여 율 1/1
주관기관 한국전자통신연구원
연구기간 2016.04.01 ~ 2016.12.31

공지예외적용 : 있음

명 세 서

청구범위

청구항 1

보안 관리 시스템의 보안 관리 방법에 있어서,

NSF(Network Security Functions) 클라이언트로부터 보안 공격을 차단 또는 완화하기 위한 상위 수준(high-level) 보안 정책을 수신하는 단계;

상기 상위 수준 보안 정책을 상기 보안 관리 시스템에 등록된 NSF 능력(capability)과 관련된 하위 수준(low-level) 보안 정책들에 매핑하는 단계; 및

상기 하위 수준 보안 정책들을 적어도 하나의 NSF에 전달하는 단계; 를 포함하되, 상기 NSF 클라이언트는 애플리케이션 로직, 정책 업데이터 및/또는 이벤트 콜렉터를 포함하는, 보안 관리 방법.

청구항 2

삭제

청구항 3

제 1 항에 있어서,

상기 애플리케이션 로직은 상기 상위 수준 보안 정책을 생성 및 업데이트하여 상기 정책 업데이터로 전송하며,

상기 정책 업데이터는 상기 상위 수준 보안 정책을 클라이언트 지향 인터페이스(Client Facing Interface)를 통해 상기 보안 관리 시스템으로 전달하며,

상기 이벤트 콜렉터는 상기 상위 수준 보안 정책의 생성 또는 업데이트에 기초가 되는 이벤트를 수신하여 상기 애플리케이션 로직으로 전송하는, 보안 관리 방법.

청구항 4

제 3 항에 있어서,

상기 상위 수준 보안 정책은 특정 공격 호스트, 서버 및 네트워크의 IP(Internet Protocol) 주소가 포함된 블랙리스트를 기반으로 생성되는, 보안 관리 방법.

청구항 5

제 4 항에 있어서,

상기 이벤트는 상기 블랙리스트로의 포함 기준을 만족하는 IP 주소에 해당하는, 보안 관리 방법.

청구항 6

제 3 항에 있어서,

상기 상위 수준 보안 정책은 차단 웹 사이트 및 차단 시간이 포함된 블랙리스트를 기반으로 생성되는, 보안 관리 방법.

청구항 7

제 3 항에 있어서,

상기 상위 수준 보안 정책은 특정 SIP(Session Initiation Protocol) 장치의 IP 주소, 소스 포트, 만료 시간, 사용자 에이전트 또는 SIP URI가 포함된 불법 장치 차단 목록을 기반으로 생성되는, 보안 관리 방법.

청구항 8

제 7 항에 있어서,

상기 이벤트는 상기 불법 장치 차단 목록으로의 포함 기준을 만족하는 도메인 정보에 해당하는, 보안 관리 방법.

청구항 9

제 1 항에 있어서,

상기 보안 관리 시스템은 보안 정책 관리자, NSF 능력 관리자 또는 개발자 관리 시스템을 포함하는, 보안 관리 방법.

청구항 10

제 9 항에 있어서,

상기 개발자 관리 시스템은 등록 인터페이스를 통해 NSF 능력을 등록 및 업데이트하며,

상기 NSF 능력 관리자는 상기 개발자 관리 시스템에 등록 및 업데이트된 NSF 능력을 저장하며,

상기 보안 정책 관리자는 상기 상위 수준 보안 정책을 상기 NSF 능력 관리자에 저장된 상기 NSF 능력과 관련된 상기 하위 수준 보안 정책들과 매핑하고, 상기 하위 수준 보안 정책들을 NSF 지향 인터페이스(NSF Facing Interface)를 통해 상기 적어도 하나의 NSF로 전달하는, 보안 관리 방법.

청구항 11

보안 관리 장치에 있어서,

보안 관리 아키텍처를 구현하는 프로세서; 를 포함하되,

상기 프로세서는,

보안 공격을 차단 또는 완화하기 위한 상위 수준(high-level) 보안 정책을 생성하는 NSF(Network Security Functions) 클라이언트;

상기 NSF 클라이언트로부터 상기 상위 수준 보안 정책을 수신하고, 상기 상위 수준 보안 정책을 상기 보안 관리 시스템에 등록된 NSF 능력(capability)과 관련된 하위 수준(low-level) 보안 정책들에 매핑하는, 보안 관리 시스템; 및

상기 보안 관리 시스템으로부터 상기 하위 수준 보안 정책들을 수신하는 적어도 하나의 NSF; 를 구현하되,

상기 NSF 클라이언트는 애플리케이션 로직, 정책 업데이터 및/또는 이벤트 콜렉터를 포함하는, 보안 관리 장치.

청구항 12

삭제

청구항 13

제 11 항에 있어서,

상기 애플리케이션 로직은 상기 상위 수준 보안 정책을 생성 및 업데이트하여 상기 정책 업데이터로 전송하며,

상기 정책 업데이터는 상기 상위 수준 보안 정책을 클라이언트 지향 인터페이스(Client Facing Interface)를 통해 상기 보안 관리 시스템으로 전달하며,

상기 이벤트 콜렉터는 상기 상위 수준 보안 정책의 생성 또는 업데이트에 기초가 되는 이벤트를 수신하여 상기 애플리케이션 로직으로 전송하는, 보안 관리 장치.

청구항 14

제 13 항에 있어서,

상기 상위 수준 보안 정책은 특정 공격 호스트, 서버 및 네트워크의 IP(Internet Protocol) 주소가 포함된 블랙

리스트를 기반으로 생성되는, 보안 관리 장치.

청구항 15

제 14 항에 있어서,

상기 이벤트는 상기 블랙리스트로의 포함 기준을 만족하는 IP 주소에 해당하는, 보안 관리 장치.

청구항 16

제 13 항에 있어서,

상기 상위 수준 보안 정책은 차단 웹 사이트 및 차단 시간이 포함된 블랙리스트를 기반으로 생성되는, 보안 관리 장치.

청구항 17

제 13 항에 있어서,

상기 상위 수준 보안 정책은 특정 SIP(Session Initiation Protocol) 장치의 IP 주소, 소스 포트, 만료 시간, 사용자 에이전트 또는 SIP URI가 포함된 불법 장치 차단 목록을 기반으로 생성되는, 보안 관리 장치.

청구항 18

제 17 항에 있어서,

상기 이벤트는 상기 불법 장치 차단 목록으로의 포함 기준을 만족하는 도메인 정보에 해당하는, 보안 관리 장치.

청구항 19

제 11 항에 있어서,

상기 보안 관리 시스템은 보안 정책 관리자, NSF 능력 관리자 또는 개발자 관리 시스템을 포함하는, 보안 관리 장치.

청구항 20

제 19 항에 있어서,

상기 개발자 관리 시스템은 등록 인터페이스를 통해 NSF 능력을 등록 및 업데이트하며,

상기 NSF 능력 관리자는 상기 개발자 관리 시스템에 등록 및 업데이트된 NSF 능력을 저장하며,

상기 보안 정책 관리자는 상기 상위 수준 보안 정책을 상기 NSF 능력 관리자에 저장된 상기 NSF 능력과 관련된 상기 하위 수준 보안 정책들과 매핑하고, 상기 하위 수준 보안 정책들을 NSF 지향 인터페이스(NSF Facing Interface)를 통해 상기 적어도 하나의 NSF로 전달하는, 보안 관리 장치.

발명의 설명

기 술 분 야

[0001] 본 발명은 네트워크 기능 가상화의 보안 관리 아키텍처에 관한 것이다.

배 경 기 술

[0002] 네트워크 기능 가상화(Network Functions Virtualization; NFV)는 네트워크 산업을 위한 새로운 영역이다. NFV는 네트워크 기능을 전용 하드웨어 기기에서 분리하여 범용 제품 서버에서 실행되는 순수 소프트웨어 인스턴스로 이러한 기능을 구현함으로써 네트워크 배포 및 유지 관리 비용 절감을 보장한다. 방화벽, 침입 탐지 시스템(Intrusion Detection System; IDS) 및 침입 방지 시스템 (Intrusion Protection System; IPS)과 같은 NSF(Network Security Functions)는, 실시간 보안 요구 사항에 따라 자동으로 제공되고 동적으로 이동될 수 있는 가상 네트워크 기능으로서 제공될 수도 있다. 본 명세서에서는 일반 NFV가 아닌 NSF에 초점을 맞춰

기술한다.

[0003] NFV 기반 보안 애플리케이션을 성공적으로 배포하기 위해서는, NSF가 여러 공급 업체에서 개발하거나 다른 네트워크 운영 업체에서 관리하기 때문에 표준화가 중요하다. 최근에는 NSF를 제어하기 위한 몇 가지 기본 표준 인터페이스가 IETF(Internet Engineering Task Force)라는 국제 인터넷 표준화 기구의 일부인 I2NSF (Interface to Network Security Functions) 워킹 그룹에 의해 개발중이다. 따라서, 몇 년 내로, 다양한 NSF가 표준 인터페이스를 통해 NFV 기반 보안 서비스의 중앙 관리 엔티티인 보안 컨트롤러라는 네트워크 엔티티에 의해 원격으로 제어될 수 있다.

[0004] 그러나, 보안 컨트롤러는 네트워크 상의 보안 정책을 생성하고 관리할 수 있는 임의의 NSF 클라이언트(예를 들어, I2NSF 클라이언트)와 통신해야 하기 때문에 NF 기반 보안 애플리케이션에서 표준 개발을 위한 여지가 여전히 남아있다. 본 명세서에서는 보안 컨트롤러로 관리되는 모든 NFV 기반 보안 애플리케이션을 NFV 지원 네트워크에 심리스하게/원활하게 통합하는 계층화된 아키텍처를 제안한다. 이 아키텍처를 통해 애플리케이션 사용자는 사용자 친화적인 방법으로 상위 수준의 보안 정책을 시행할 수 있다.

발명의 내용

해결하려는 과제

[0005] 본 명세서에서는 NFV를 이용한 NSF 기반의 보안 관리를 위한 일반적인 아키텍처 제시를 목적으로 한다.

[0006] 또한, 본 명세서는 제안된 프레임 워크가 위험 도메인의 블랙리스트, 시간별 액세스 제어 정책 및 VoIP(Voice over IP)-VoLTE(Voice over LTE) 서비스에 대한 의심스러운 전화 탐지와 같은 몇 가지 실제 공격 시나리오를 완화할 수 있는 방법 제안을 목적으로 한다.

[0007] 이를 위해, 본 명세서에서는 구체적으로 제안된 아키텍처의 상세한 구현을 설명하기 위한 예제를 중심으로 설명한다. 이에 기초하여, 향후 다양한 네트워크 공격을 완화할 수 있는 가능성을 보여주기 위해 제안된 프레임 워크가 완벽하게 구현될 수 있다.

과제의 해결 수단

[0008] 본 발명의 일 실시예에 따르면, 네트워크 기능 가상화에서의 보안 관리를 위한 아키텍처는 I2NSF 프레임 워크, 애플리케이션 로직, 정책 업데이터 및 이벤트 콜렉터를 포함할 수 있다.

발명의 효과

[0009] 본 발명의 일 실시예에 따른 아키텍처를 통해, 애플리케이션 사용자는 사용자 친화적인 방법으로 상위 수준의 보안 정책을 시행할 수 있다. 보다 상세하게는, 본 발명의 일 실시예에 따른 아키텍처를 통해, 네트워크 리소스 및 프로토콜에 대한 특정 정보가 필요 없는 상위 수준의 보안 인터페이스를 사용자에게 제공함으로써, 사용자에게 친숙한 방식으로 사용자로 하여금 보안 요구 사항을 정의하도록 할 수 있다.

[0010] 또한, 본 발명의 일 실시예에 따른 아키텍처를 통해, NSF 클라이언트가 직접 상위 수준 보안 정책을 수립하고 NSF에서 발생하는 이벤트를 피드백 받음으로써 효율적인 보안 관리를 실현할 수 있다.

도면의 간단한 설명

[0011] 도 1은 NFV에 기반한 보안 관리를 위한 아키텍처를 예시한 도면이다.

도 2는 NSF 클라이언트의 사용자 인터페이스를 예시한 도면이다.

도 3은 Client Facing Interface의 데이터 모델을 예시한 도면이다.

도 4는 I2NSF의 보안 관리 아키텍처를 예시한 도면이다.

도 5는 Malware 도메인 블랙리스트 작성의 보안 관리 아키텍처를 예시한 도면이다.

도 6은 I2NSF 프레임 워크 내의 보안 관리 아키텍처를 예시한 도면이다.

도 7은 Malware 도메인 블랙리스트 작성의 보안 관리 아키텍처를 예시한 도면이다.

도 8은 본 발명의 일 실시예에 따른 보안 관리 시스템의 보안 관리 방법에 관한 순서도이다.

발명을 실시하기 위한 구체적인 내용

[0012] 본 명세서에서 사용되는 용어는 본 명세서에서의 기능을 고려하면서 가능한 현재 널리 사용되는 일반적인 용어를 선택하였으나, 이는 당 분야에 종사하는 기술자의 의도, 관례 또는 새로운 기술의 출현 등에 따라 달라질 수 있다. 또한 특정 경우는 출원인이 임의로 선정한 용어도 있으며, 이 경우 해당되는 실시예의 설명 부분에서 그 의미를 기재할 것이다. 따라서 본 명세서에서 사용되는 용어는, 단순한 용어의 명칭이 아닌 그 용어가 아닌 실질적인 의미와 본 명세서의 전반에 걸친 내용을 토대로 해석되어야 함을 밝혀두고자 한다.

[0013] 더욱이, 이하 첨부 도면들 및 첨부 도면들에 기재된 내용들을 참조하여 실시예를 상세하게 설명하지만, 실시예들에 의해 제한되거나 한정되는 것은 아니다.

[0014] 이하, 첨부한 도면들을 참조하여 본 발명의 바람직한 실시예를 보다 상세하게 설명하고자 한다.

[0016] **[1] 제1 실시예**

[0017] 네트워크 기능 가상화 (NFV)는 네트워크 보안 서비스를 설계하고 배포하는 새로운 방법을 제공하지만, 네트워크 보안 서비스들 사이의 규격/표준화된 네트워크 인터페이스 서비스가 없는 경우, 네트워크 보안 서비스들을 완벽하게 통합하는 실용적인 에코 시스템을 구축하지 못할 수도 있다. 따라서, 본 명세서에서는 NFV를 사용하는 NSF(Network Security Functions) 기반의 보안 관리 서비스를 위한 아키텍처를 제안한다. 제안된 아키텍처는 네트워크 리소스 및 프로토콜에 대한 특정 정보가 필요 없는 상위 수준의 보안 인터페이스를 사용자에게 제공함으로써, 사용자에게 친숙한 방식으로 사용자로 하여금 보안 요구 사항을 정의하도록 할 수 있다.

[0018] 1. 도입

[0019] 제1 실시예에서는 제안된 아키텍처를 위한 기본 구성 요소(예를 들어, 보안 정책 관리자, NSF 능력 관리자, 애플리케이션 논리, 정책 업데이터 및 이벤트 수집기)와 인터페이스 설계 방식을 제안한다. 또한, 제1 실시예에서는 (1) 위험 도메인의 블랙리스트, (2) 시간별 액세스 제어 정책 및 (3) VoIP-VoLTE 서비스에 대한 의심스러운 전화 탐지, 이렇게 세 가지 케이스에 대한 본 발명의 사용예를 중심으로 살펴본다. 또한, 본 명세서에서는 제안된 아키텍처의 구현 방법에 대해 예를 들어 상세히 설명한다. 또한, 본 명세서에서는 제안된 아키텍처를 실제 네트워크 환경에 배치/적용하기 위한 몇 가지 기술적 과제에 대해 살펴본다.

[0020] 이하, 설명의 편의를 위해 주제별로 절을 나누어 설명한다. 2절에서는 NFV 기반의 보안 관리 서비스 아키텍처에 대해 제안한다. 3절에서는 제안된 아키텍처를 사용하는 세 가지 주요 사용예에 대해 설명한다. 4절에서는 실제로 제안된 아키텍처를 사례를 통해 구현하는 방법에 대해 설명한다. 5절에서는 아키텍처 구현에 대한 기술적 문제에 대해 설명한다. 6절에서는 관련 연구에 대한 요약 및 분석 내용을 설명한다. 7절에서는 결론에 대해 살펴본다.

[0021] 2. 아키텍처

[0022] 본 명세서에서는 NFV에 기반한 보안 관리 서비스를 위한 추가 구성 요소를 통합한 계층화된 아키텍처를 제안한다. 본 명세서의 도면들에서 화살표는 기능 구성 요소들 간의 통신을 나타낸다. 특히, 도면에서 양방향 화살표는 양방향으로의 두 구성 요소간 상호 작용을 나타내며, 단방향 화살표는 화살표가 가리키는 방향으로의 두 구성 요소 간 상호 작용을 나타낸다.

[0023] 도 1은 NFV에 기반한 보안 관리를 위한 아키텍처를 예시한 도면이다.

[0024] 도 1을 참조하면, 유연하고 효과적인 보안 정책 시행을 지원하기 위해 제안된 아키텍처는, (1) NSF 클라이언트(100) (2) 보안 관리 시스템(110) 및 (3) NSF 인스턴스들(120), 이렇게 세 가지 계층으로 구성될 수 있다.

[0025] 본 발명의 아키텍처는 유연하고 효과적인 보안 정책 시행을 지원하도록 설계되었다. 본 명세서에서 NSF 클라이언트(100)라는 용어는 NFV 기반의 보안 애플리케이션을 의미한다.

[0026] NSF 클라이언트(100)에서 애플리케이션 로직(100-1)은 상위 수준 보안 정책을 생성할 수 있다. 정책 업데이터(100-2)는 클라이언트 지향 인터페이스(Client Facing interface)를 통해 보안 컨트롤러(110-1)에 정책들을 배포할 수 있다. 보안 컨트롤러(110-1)에서 보안 정책 관리자(110-2)는 상위 수준 정책을 NSF 능력 관리자(110-3)에 등록된 NSF 능력과 관련된 하위 수준 보안 정책에 매핑할 수 있다. 매핑 후, 보안 정책 관리자(110-2)는 NSF 지향 인터페이스(NSF Facing Interface)를 통해 NSF(120)에 해당 정책들을 전달할 수 있다. 이하에서는, 각 네트워크 구성 요소의 동작에 대해 상세히 살펴본다.

[0027] 2.1 보안 정책 관리자(110-2)

[0028] 보안 정책 관리자(110-2)는 Client Facing Interface를 통해 정책 업데이터(100-2)로부터 상위 수준의 정책을 수신하고, 상위 수준의 정책을 NSF 능력 관리자(110-3)에 등록된 특정 NSF 능력과 관련된 여러 하위 수준의 정책들과 매핑하거나 또는 하위 수준의 정책들을 상위 수준의 정책으로 매핑하는 구성 요소이다. 또한, 보안 정책 관리자(110-2)는 이러한 정책들을 NSF Facing Interface를 통해 NSF(들)에게 전달할 수 있다.

[0029] 하위 수준 정책 변경이 필요한 이벤트가 NSF(120)에서 발생하면, NSF(120)는 NSF Facing Interface를 통해 보안 정책 관리자(110-2)에게 이벤트를 전송할 수 있다. 이후, 보안 정책 관리자(110-2)는 Client Facing Interface를 통해 해당 이벤트를 이벤트 콜렉터(100-3)로 전송할 수 있다.

[0030] 2.2 NSF 능력 관리자(110-3)

[0031] NSF 능력 관리자(110-3)는 보안 컨트롤러(110-1)에 통합된 구성일 수 있다. NSF 능력 관리자(110-3)는 등록 인터페이스를 통해 개발자 관리 시스템(110-4)에 등록된 NSF의 능력을 저장하고 이를 보안 정책 관리자(110-2)와 공유하여 보안 정책 관리자(110-2)가 특정 NSF 능력과 관련된 하위 수준 정책을 생성할 수 있도록 할 수 있다. 또한, NSF 능력 관리자(110-3)는 새로운 NSF가 등록될 때마다 등록 인터페이스를 통해 NSF 능력 관리자(110-3)의 관리 테이블에 NSF의 능력을 등록하도록 개발자의 관리 시스템에 요청할 수 있다. 기존의 NSF가 삭제되면 NSF 능력 관리자(110-3)는 관리 테이블에서 NSF의 능력을 제거할 수 있다.

[0032] 2.3 개발자 관리 시스템(110-4)

[0033] 개발자 관리 시스템(110-4)은 등록 인터페이스를 통해 NSF 능력 관리자(110-3)에 새로운 NSF 능력을 등록하는 구성 요소일 수 있다. 등록된 NSF에 업데이트가 있으면, 업데이트된 내용/정보는 개발자 관리 시스템에서 NSF 능력 관리자(110-3)에게 전달될 수 있다.

[0034] 2.4 애플리케이션 로직(100-1)

[0035] 애플리케이션 로직(100-1)은 (보안 관리 아키텍처에서) 보안 공격을 차단 또는 완화하기 위해 상위 수준의 보안 정책을 생성하는 구성일 수 있다. 애플리케이션 로직(100-1)은 이벤트 콜렉터(100-3)로부터 상위 수준의 정책을 업데이트(또는 생성)하는 이벤트를 수신하고, 수집된 이벤트를 기반으로 상위 수준의 정책을 업데이트(또는 생성)할 수 있다. 다음으로, 애플리케이션 로직(100-1)은 최근 업데이트된 정책을 전달하기 위해 상위 수준 정책을 정책 업데이터(100-2)로 전송할 수 있다. 이하의 3절에서는 세 가지 사용예를 통해 애플리케이션 로직(100-1)을 설계하는 방법에 대해 설명한다.

[0036] 2.5 정책 업데이터(100-2)

[0037] 정책 업데이터(100-2)는 애플리케이션 로직(100-1)에 의해 생성된 상위 수준의 보안 정책을 수신하고, 이를 Client Facing Interface를 통해 보안 컨트롤러(110-1)로 배포/전달하는 구성일 수 있다.

[0038] 2.6 이벤트 콜렉터(100-3)

[0039] 이벤트 콜렉터(100-3)는 애플리케이션 로직(100-1)의 상위 수준 정책 업데이트(또는 생성)에 반영되어야 하는 이벤트를 보안 컨트롤러(110-1)로부터 수신할 수 있다. NSF에서 발생하는 이벤트에 따라 하위 수준의 보안 정책이 업데이트될 수 있으므로, NSF에서 이벤트를 수신하는 절차가 필요하다. 이벤트를 수신한 후 이벤트 콜렉터(100-3)는, 이를 애플리케이션 로직(100-1)으로 전달하여 애플리케이션 로직(100-1)이 보안 컨트롤러(110-1)로부터 수신한 이벤트를 기반으로 상위 수준의 보안 정책을 업데이트(또는 생성)하도록 할 수 있다.

[0040] 3. 사용예

[0041] NFV를 기반으로 한 일반적인 아키텍처는 가능한 보안 공격에 대응하도록 설계되었다. 본 절에서는 위험한 도메인의 블랙리스트에 있는 보안 공격 방어, 시간에 따른 액세스 제어 정책 및 VoIP-VoLTE 서비스에 대한 의심스러운 전화의 탐지 절차에 대해 설명한다.

[0042] 3.1 위험한 도메인들의 블랙리스트

[0043] 위험한 도메인(예를 들어, malware 배포에 사용되는 도메인 등) 블랙리스트 작성은 악성 활동이 의심되는 공격 호스트, 서버 및 네트워크의 IP 주소에 대한 블랙리스트를 유지 및 게시하는 것을 의미한다. 위험한 도메인 블랙리스트 작성을 위한 보안 관리 아키텍처의 경우, 위험 도메인 관리자는 보안 관리를 수행하기 위해 애플리케이션 로직의 역할을 담당할 수 있다.

[0044] 위험한 도메인 블랙리스트 작성에 기초하여, 위험한 도메인 리스트는 위험한 도메인 데이터베이스에 저장되며, 애플리케이션 로직 기능을 하는 위험한 도메인 관리자에 의해 수동 또는 자동으로 업데이트될 수 있다. 또한, 위험한 도메인 관리자는 위험한 도메인 데이터 베이스로부터 위험한 도메인 리스트를 주기적으로 로드하고, 새로 추가된 위험한 도메인들과의 패킷 전달을 방지하기 위해 새로운 상위 수준의 보안 정책(예를 들어, IP 주소를 사용하여 위험한 도메인들의 리스트 차단, 블랙리스트 차단)을 생성할 수 있다. 위험한 도메인 관리자는 새로운 상위 수준 보안 정책을 정책 업데이터로 전송할 수 있으며, 정책 업데이트는 수신한 새로운 상위 수준 보안 정책을 보안 컨트롤러에 배포할 수 있다. 보안 컨트롤러는 상위 수준 정책을 하위 수준 정책들에 매핑하고, 하위 수준 보안 정책들을 NSF에 적용할 수 있다.

[0045] NSF가 새로운 위험한 도메인을 탐지하면, 탐지한 도메인에 대응하는 IP 주소를 NSF Facing Interface를 통해 보안 컨트롤러로 전송할 수 있다. 보안 컨트롤러는 이벤트 콜렉터에 해당 IP 주소를 전달할 수 있다. 이벤트 콜렉터는 IP 주소를 위험 도메인 관리자로 전달하면, 이를 기초로 위험 도메인 관리자가 위험 도메인 데이터베이스를 업데이트할 수 있다.

[0046] 3.2 시간별 액세스 제어 정책들

[0047] 시간별 액세스 제어 정책들은 특정 기간 동안 특정 웹 사이트에 대한 사용자의 액세스를 관리할 수 있다. 예를 들어, 회사에서 관리자는 직원이 업무 시간의 집중을 방해하는 Youtube 웹 사이트에 액세스하는 것을 차단할 수 있다.

[0048] NSF 클라이언트는 시간별 접근 제어를 기반으로 애플리케이션 로직에서 차단 웹 사이트 및 차단 시간이 포함된 블랙리스트를 등록할 수 있다. 애플리케이션 로직은 해당 목록을 데이터 베이스에 저장하고 상위 수준의 보안 정책을 생성할 수 있다(예를 들어, 차단 웹 사이트 및 차단 시간을 확인하여 차단 시간 동안의 차단 웹 사이트에 대한 액세스 차단).

[0049] 애플리케이션 로직은 생성한 상위 수준의 보안 정책을 정책 업데이터로 전달하면, 정책 업데이터가 이를 보안 컨트롤러로 전달할 수 있다. 보안 컨트롤러에서 보안 정책 관리자는 상위 수준 정책을 하위 수준 정책들에 매핑한 다음, 이들을 NSF에 전송 및 적용할 수 있다.

[0050] 3.3 VoIP-VoLTE 서비스에 대한 의심스러운 전화 탐지

[0051] VoIP-VoLTE 보안 관리는 불법적인 전화나 인증이 의심되는 SIP(Session Initiation Protocol) 장치의 IP 주소, 소스 포트, 만료 시간, 사용자 에이전트 및 SIP URI가 포함된 불법 장치 차단 목록을 유지 및 게시할 수 있다. 일반 보안 관리 아키텍처에서 VoIP-VoLTE 보안 관리자는 도 1에서의 VoIP-VoLTE 보안 서비스를 위한 애플리케이션 로직 역할을 담당한다.

[0052] VoIP-VoLTE 보안 관리에 기초하여, 불법 장치 정보 목록은 VoIP-VoLTE 데이터 베이스에 저장되며, VoIP-VoLTE 보안 관리자에 의해 수동 또는 자동으로 업데이트될 수 있다. 또한, VoIP-VoLTE 보안 관리자는 주기적으로 VoIP-VoLTE 데이터 베이스로부터 불법 장치 정보 목록을 로드하고, 새로 추가된 VoIP-VoLTE 공격자와의 패킷 전달을 방지하기 위해 새로운 상위 수준의 보안 정책(예를 들어, IP 주소, 소스 포트 등을 사용하는 불법 장치 차단 목록)을 생성할 수 있다. 또한, VoIP-VoLTE 보안 관리자는 생성한 새로운 상위 수준 보안 정책을 정책 업데이터로 전송할 수 있으며, 정책 업데이터는 수신한 상위 수준 보안 정책을 보안 컨트롤러로 배포할 수 있다. 보안 컨트롤러는 상위 수준 정책을 여러 하위 수준 정책들에 매핑하고 하위 수준 보안 정책을 NSF에 적용하게 된다.

[0053] NSF가 도메인으로부터 전달된 비정상적인 메시지 또는 전화를 검출하면, IP 주소, 사용자 에이전트 및 만료 시간 값과 같은 도메인의 정보가 NSF에 의해 NSF Facing Interface를 통해 보안 컨트롤러로 전송될 수 있다. 보안 컨트롤러는 이를 이벤트 콜렉터로 전달할 수 있다. 이벤트 콜렉터는 탐지된 도메인 정보를 VoIP-VoLTE 보안 관리자에게 전달하면, VoIP-VoLTE 보안 관리자는 이에 기초하여 VoIP-VoLTE 데이터베이스를 업데이트할 수 있다.

[0054] 4. 구현

[0055] 본 절에서는 제안된 아키텍처에서 각 구성 요소와 인터페이스를 구현하는 방법에 대해 제안하며, 이때 앞서 상술한 3.3 절의 사용예를 고려한다. 이러한 구현을 통해 착신호에 사기 전화 동작(예를 들어, 비정상적인 시간대에 블랙리스트에 올라있는 위치에서 통화가 이루어지는 동작 등)이 있는지 여부를 확인하여 의심스러운 VoIP-VoLTE 전화의 차단이 가능하다.

[0056] 4.1 NSF 클라이언트

[0057] 관리자에게보다 친숙하고 접근 가능한 관리 서비스를 제공하기 위해, 웹 서버 및 관리자가 상위 수준의 보안 정책을 설정할 수 있는 사용자 인터페이스를 제공하는 몇 개의 웹 페이지가 제공/생성될 수 있다.

[0058] 도 2는 NSF 클라이언트의 사용자 인터페이스를 예시한 도면이다.

[0059] 도 2를 참조하면, 관리자가 보안 정책을 관리하기 위해서는, (1) 정책 업데이터에 대한 정책 설정 페이지와 (2) 이벤트 콜렉터에 대한 로그 메시지 페이지, 이렇게 두 가지 웹 페이지가 고려될 수 있다. YANG은 YANG에서 정의된 데이터로의 HTTP를 통한 접속을 위한 프로그래밍 인터페이스를 제공하는 RESTCONF와 같은 표준 네트워크 프로토콜에 의해 조작된 구성과 상태 데이터를 모델링하는 데 널리 사용되기 때문에, NSF 클라이언트와 보안 관리 시스템 간의 통신을 위한 데이터 모델을 정의하기 위해 YANG이 고려/사용될 수 있다.

[0060] 정책 설정 페이지에서, 특정 시간 동안 블랙리스트에 포함된 국가와 같은 상위 수준의 보안 정책을 정의하기 위한 필드가 생성될 수 있다. 만일, 관리자가 새로운 상위 수준의 보안 정책을 설정하면, NSF 클라이언트의 데이터 모델 파서(parser)가 정책을 해석하고 YANG 데이터 모델에 따라 XML 파일을 생성할 수 있다.

[0061] 로그 메시지 페이지에는, 보안 관리 시스템에서 이벤트 콜렉터로 이벤트가 전달되는 경우, 보안 관리 시스템 및 NSF 인스턴스에서 보안 애플리케이션의 결과 및/또는 기능 구성 요소의 상태를 보고하는 이벤트에 대한 정보가 표시될 수 있다.

[0062] 4.2 Client Facing Interface

[0063] NSF 클라이언트와 보안 관리 시스템 간의 상호 작용을 가능하게 하기 위해 RESTCONF를 기반으로 한 통신 채널이 구현될 수 있다. 또한, NSF 클라이언트는 구현 시 웹 애플리케이션을 기반으로 하기 때문에, 네트워크 구성(NETCONF) 프로토콜 대신 RESTCONF가 선호될 수 있다.

[0064] 또한, Client Facing Interface를 위한 표준화된 데이터 모델이 아직 없기 때문에, 보안 정책 요구 사항에 기반한 데이터 모델이 설계될 필요가 있다.

[0065] 도 3은 Client Facing Interface의 데이터 모델을 예시한 도면이다.

[0066] 도 3에서는 VoIP-VoLTE 서비스에서 의심스러운 전화를 탐지하기 위한 정책 관리와 관련된 데이터 모델 설계의 일부를 보여준다.

[0067] 알려지지 않은 공격 및 조건에 정책을 적용하기 위한 일반적인 데이터 모델이 설계될 수 있다. 이러한 데이터 모델은 (1) 정책 라이프 사이클 관리, (2) 정책 규칙 및 (3) 조치로 구성될 수 있다. (1) 정책 라이프 사이클 관리 필드는 정책 자체의 수명을 결정하기 위해 만료 시간 및/또는 만료 이벤트 세트를 지정할 수 있다. (2) 정책 규칙 필드는 서비스 타입, 조건 및 유효한 시간 간격과 같은 상위 수준 정책에 대한 특정 정보를 나타낼 수 있다. (3) 조치 필드는 어떤 행동을 취해야 하는지를 지정한다. 예를 들어, 허가(permit) 및 미러(mirror) 모두 'true'인 경우, 예외 시간(유효 시간 간격에 포함)에 블랙리스트에 있는 발신자 위치의 통화 트래픽은 차단될 수 있으며, 딥 패킷 검사(Deep Packet Inspection; DPI)를 위해 사전 정의된 호스트로 순차적으로 전달 될 수 있다.

[0068] 4.3 보안 관리 시스템

[0069] 보안 관리 시스템의 주된 역할은 상위 수준의 정책을 하위 수준의 정책 집합으로 변환하는 것이다. 예를 들어, 보안 관리 시스템은 각 국가의 IP 주소를 제공하는 위치 정보 데이터베이스를 사용하여 국가 이름을 일련의 IP 주소들의 세트로 매핑할 수 있다. 상위 수준의 보안 정책을 변환한 후, 보안 관리 시스템은 네트워크 트래픽을 해당 IP 주소 및/또는 해당 IP 주소로 지정하기 위해 하위 수준 보안 정책들을 생성할 수 있다. 데이터 모델 파서는 하위 수준 보안 정책을 위한 XML 파일을 생성하여, 이를 적절한 NSF 인스턴스에 전달할 수 있다. 또한, 보안 관리 시스템은 NSF에 의해 생성된 보안 이벤트를 YANG 데이터 모델의 상위 수준 로그 메시지로 해석하여, 이를 NSF 클라이언트에게 반대 방향으로 전달할 수 있다.

[0070] 4.4 NSF Facing Interface

[0071] Client Facing Interface와 마찬가지로 NSF Facing Interface 역시, 구현 시 RESTCONF 프로토콜과 YANG 데이터 모델을 사용할 수 있다. I2NSF는 최근 NSF Facing Interface에 대한 표준 데이터 모델과 프로토콜을 정의하기 위한 작업을 진행 중에 있다.

[0072] 4.5 NSF 인스턴스들

[0073] 사용예에서는 발신자/착신자의 위치 및 전화 시간을 확인함으로써 VoIP-VoLTE 전화가 의심스러운지 여부를 결정하기 위해, NSF 인스턴스로서 방화벽 애플리케이션이 선택될 수 있다. 전화에 의심스러운 동작 패턴이 있는 경우, 해당 전화의 네트워크 트래픽은 하위 수준 보안 정책에 따라 방화벽 애플리케이션에 의해 효과적으로 차단될 수 있다. 방화벽 애플리케이션의 결과는 RESANGON 프로토콜을 통해 YANG 데이터 모델에서 보안 관리 시스템으로 전달될 수 있다.

[0074] 특정 상황에 따라 다수의 NSF 인스턴스들이 고려될 수 있다. 예를 들어, 의심스러운 발신자의 네트워크 트래픽을 분석하는 데 DPI가 추가로 사용될 수 있다.

[0075] 5. 주요 기술적 과제

[0076] 본 절에서는 본 발명을 구현하고 시스템 성능을 향상시키기 위해 추가적으로 고려해야 할 사항에 대해 살펴보며, 다음과 같다:

[0077] (1) 정책 업데이터가 보안 컨트롤러를 최근의 상위 수준 정책으로 업데이트함에 따라 업데이트 시간이 보안 컨트롤러와 달라질 수 있으며, 이 업데이트 프로세스 중에 상위 레벨 보안 정책의 불일치가 발생할 수 있다. 이러한 상위 수준 보안 정책의 불일치는 SDN 전환에서 공통적으로 볼 수 있는 업데이트 프로세스 중 구성 불일치와 유사하다.

[0078] (2) 보안 컨트롤러가 수신하는 정책 흐름으로 확장할 수 없기 때문에, 하나의 보안 컨트롤러가 증가하는 다수의 NSF 클라이언트들을 모두 처리 할 수 없게 되어 확장성 문제가 발생할 수 있다.

[0079] (3) 네트워크 엔티티들(예를 들어, NSF 클라이언트와 보안 관리 시스템) 사이에 안전하고 인증된 통신 채널이 설정되어야 한다. 이러한 통신 채널을 보장하지 않으면, 부적절한 보안 정책이 공격자에 의해 악의적으로 변경될 수 있다. 따라서, 네트워크 엔티티들에 키를 적절히 분배하기 위해서는 효율적인 키 관리가 필요하다.

[0080] (4) 보안 컨트롤러가 상위 수준 및 하위 수준 정책을 처리할 때, 처리 시퀀스들은 보안 컨트롤러와 NSF들 모두에서 동기화 문제를 일으킬 수 있다. 이 동기화 문제가 발생하지 않도록 보안 컨트롤러에 적절한 스케줄링 모델을 정의해야 한다.

[0081] (5) Client Facing Interface를 통해 전달될 높은 수준의 정책들을 생성하기 위해, 먼저 Client Facing Interface에서 일반적인 정책 데이터 모델이 정의되어야 한다. 이를 위해, 형태 및 내용과 무관하게, 정책을 쉽게 관리 할 수 있는 정책 추상화(Simplified Use of Policy Abstractions; SUPA)의 일반적인 데이터 모델이 사용될 수 있다.

[0082] 6. 관련 연구

[0083] 네트워크 서비스 가상화를 보안 서비스에 사용하는 것에 대한 관심은 네트워킹 커뮤니티에서 꾸준히 증가하고 있다. 그러나, 앞서 상술한 바와 같이, NSF에 대한 표준 인터페이스 및 스펙이 없다면, NSF를 매끄럽게 통합 및 관리하는 것이 불가능하다. 특히, 상위 수준의 보안 정책을 위한 표준 인터페이스가 없기 때문에 NSF 기반 애플리케이션을 실제 환경에 배포하기가 어렵다.

[0084] 이를 해결하기 위해, 본 명세서에서는 제1 실시예로서 클라이언트가 NSF와 관련된 세부 구현없이 NSF 인스턴스를 제어하기 위한 상위 수준의 보안 정책을 구성 및 관리 할 수 있는 아키텍처를 제시하였다.

[0085] 7. 결론

[0086] 이상으로, 제1 실시예로서 NFV를 이용한 NSF 기반의 보안 관리를 위한 일반적인 아키텍처를 제시하였다. 또한, 앞서 제1 실시예로 제안된 프레임 워크가 위험 도메인의 블랙리스트, 시간별 액세스 제어 정책 및 VoIP-VoLTE 서비스에 대한 의심스러운 전화 탐지와 같은 몇 가지 실제 공격 시나리오를 완화시킬 수 있는 방법에 대해 살펴보았다. 또한, 다양한 예제를 도입하여 구체적으로 제안된 아키텍처의 상세한 구현을 설명하였다.

[0088] **[2] 제2 실시예**

[0089] 제2 실시예는 I2NSF(Interface to Network Security Functions) 프레임 워크에서의 보안 관리 아키텍처를 제안한다. 이 보안 관리 아키텍처는 I2NSF 클라이언트, 보안 관리 시스템(즉, 보안 컨트롤러 및 개발자 관리 시스템) 및 I2NSF 프레임 워크의 NSF(Network Security Functions)를 포함할 수 있다. I2NSF 클라이언트는 애플리케이션 로직, 정책 업데이터 및 정책 콜렉터를 포함할 수 있다. 보안 컨트롤러는 보안 정책 관리자와 NSF

능력 관리자를 포함할 수 있다. 각 구성에 관한 설명은 앞서 도 1과 관련하여 상술한 설명이 동일하게 적용될 수 있으며, 중복되는 설명은 생략한다.

[0090] 또한, 제2 실시예는 상술한 구성들의 기능과 상위 수준에서의 보안 관리 처리에 대해 제안한다. 또한, 제2 실시예는 malware 도메인 리스트 보안 관리 및 VoIP-VoLTE 보안 관리와 같은 대표적인 사용 사례에 대해서도 설명한다.

[0091] 이외에, 제2 실시예에는 앞서 상술한 제1 실시예의 설명이 동일/유사하게 적용될 수 있으며, 중복되는 설명은 생략한다. 또한, 주제별로 절을 나누어 설명한다.

[0092] 1. 도입

[0093] I2NSF 프레임 워크[i2nsf-framework]에 사용자의 상위 수준 보안 정책을 적용하기 위해, I2NSF 클라이언트는 Client Facing Interface를 통해 보안 컨트롤러에 이러한 정책을 제공할 수 있다. 제2 실시예에서는 보안 관리를 위한 아키텍처가 I2NSF 프레임 워크의 주어진 상위 수준 정책에 대해 제안될 수 있다. 이 아키텍처는 I2NSF 클라이언트, 보안 관리 시스템(즉, 보안 컨트롤러 및 개발자 관리 시스템) 및 I2NSF 프레임 워크의 NSF를 포함할 수 있다. I2NSF 클라이언트에는 애플리케이션 로직, 정책 업데이터 및 정책 콜렉터가 포함될 수 있다. 보안 컨트롤러는 보안 정책 관리자 및 NSF 능력 관리자를 포함할 수 있다.

[0094] 보안 컨트롤러의 보안 정책 관리자 및 NSF 능력 관리자는 Client Facing Interface를 통해 I2NSF 클라이언트의 정책 업데이터에서 제공하는 업데이트된 보안 정책을 제어할 수 있다. 정책 업데이터는 보안 컨트롤러에 새롭거나 업데이트된 정책을 제공할 수 있다. 반면, NSF가 하위 수준 정책을 변경하는 이벤트가 발생하면, 정책 콜렉터는 이에 상응하여 보안 컨트롤러를 통해 상위 수준 정책을 수신할 수 있다. 그 후, 정책 콜렉터는 애플리케이션 로직의 현재 정책도 이에 따라 업데이트할 수 있다.

[0095] 제2 실시예에서는 보안 관리를 위한 추가 구성 요소를 I2NSF 프레임 워크에 통합하는 보안 관리 아키텍처를 제안한다. 이러한 아키텍처는 유연하고 효과적인 보안 정책을 지원하도록 설계되었다. Application Logic은 상위 수준의 정책을 생성하고, 정책 업데이터는 이를 Client Facing Interface를 통해 보안 정책 관리자로 전송할 수 있다. 보안 정책 관리자는 상위 수준 정책을 보안 컨트롤러의 여러 하위 수준 정책들에 매핑할 수 있다. 하위 정책들에 매핑한 후, 보안 정책 관리자는 이러한 정책들이 NSF에 적용될 수 있도록 NSF로 전송하게 된다.

[0096] 2. 목적

[0097] 제2 실시예는 다음과 같이 보안 관리 아키텍처에 대한 두 가지 주요 목표를 갖는다.

[0098] (1) 높은 수준의 보안 관리: NSF에서의 유연하고 효과적인 보안 정책의 시행을 지원하기 위해 일반적인 보안 관리 아키텍처의 설계를 제안한다.

[0099] (2) 보안 정책들의 자동 업데이트: 새로운 보안 공격에 대한 업데이트된 하위 수준의 보안 정책을 대응하는 상위 수준 보안 정책에 반영한다.

[0100] 3. 보안 관리를 위한 아키텍처

[0101] 본 절에서는 I2NSF의 보안 관리 아키텍처에 대해 설명하고 보안 컨트롤러와 개발자 관리 시스템을 갖춘 보안 관리 시스템에 중점을 두고 설명한다. 또한, 보안 컨트롤러의 기본 동작 및 아키텍처의 각 구성 요소에 대한 세부 정보를 설명한다.

[0102] 도 4는 I2NSF의 보안 관리 아키텍처를 예시한 도면이다.

[0103] 도 4의 보안 관리 아키텍처는 유연하고 효과적인 보안 정책의 시행을 지원하도록 설계되었다. I2NSF 클라이언트의 애플리케이션 로직은 새로운 보안 공격에 따라 상위 수준의 정책을 생성하면, I2NSF 클라이언트의 정책 업데이터가 이러한 정책을 보안 컨트롤러의 보안 정책 관리자에게 전송한다. 보안 정책 관리자는 상위 수준 정책을 NSF 능력 관리자에 등록된 NSF 능력과 관련된 몇 가지 하위 수준의 정책들에 매핑할 수 있다. 이와 같은 낮은 수준의 정책으로의 매핑이 완료된 후, 보안 정책 관리자는 이러한 정책들을 NSF Facing Interface를 통해 NSF에 전달할 수 있다. 이하에서는 각 구성에 대해 후술한다.

[0104] 2.1. 보안 정책 관리자

[0105] 보안 정책 관리자는 Client Facing Interface를 통해 정책 업데이터로부터 상위 수준의 정책을 수신하고, 상위 수준의 정책을 NSF 능력 관리자에 등록된 특정 NSF 능력과 관련된 여러 하위 수준의 정책들과 매핑하거나 또는

하위 수준의 정책들을 상위 수준의 정책으로 매핑하는 구성 요소이다. 또한, 보안 정책 관리자는 이러한 정책들을 NSF Facing Interface를 통해 NSF(들)에게 전달할 수 있다.

[0106] 하위 수준 정책 변경이 필요한 이벤트가 NSF에서 발생하면, NSF는 NSF Facing Interface를 통해 보안 정책 관리자에게 변경된 하위 수준 정책을 전송할 수 있다. 이후, 보안 정책 관리자는 Client Facing Interface를 통해 상기 변경된 하위 수준의 정책을 상위 수준의 정책에 매핑하고, 상기 변경된 하위 수준의 정책 또는 상위 수준의 정책을 정책 콜렉터로 전송할 수 있다.

[0107] 2.2 NSF 능력 관리자

[0108] NSF 능력 관리자는 보안 컨트롤러에 통합된 구성일 수 있다. NSF 능력 관리자는 등록 인터페이스를 통해 개발자 관리 시스템에 등록된 NSF의 능력을 저장하고 이를 보안 정책 관리자와 공유하여 보안 정책 관리자가 특정 NSF 능력과 관련된 하위 수준 정책을 생성할 수 있도록 할 수 있다. 또한, NSF 능력 관리자는 새로운 NSF가 등록될 때마다 등록 인터페이스를 통해 NSF 능력 관리자의 관리 테이블에 NSF의 능력을 등록하도록 개발자의 관리 시스템에 요청할 수 있다. 기존의 NSF가 삭제되면 NSF 능력 관리자는 관리 테이블에서 NSF의 능력을 제거할 수 있다.

[0109] 2.3 개발자 관리 시스템

[0110] 개발자 관리 시스템은 등록 인터페이스를 통해 NSF 능력 관리자에 새로운 NSF 능력을 등록하는 구성 요소일 수 있다. 등록된 NSF에 업데이트가 있으면, 업데이트된 내용/정보는 개발자 관리 시스템에서 NSF 능력 관리자에게 전달될 수 있다.

[0111] 2.4 애플리케이션 로직

[0112] 애플리케이션 로직은 (보안 관리 아키텍처에서) 보안 공격을 차단 또는 완화하기 위해 상위 수준의 보안 정책을 생성하는 구성일 수 있다. 애플리케이션 로직은 생성된 정책들을 정책 업데이터로 전송할 수 있다. 애플리케이션 로직의 보다 상세한 동작에 관해서는 이하의 사용예와 함께 후술한다.

[0113] 2.5 정책 업데이터

[0114] 정책 업데이터는 애플리케이션 로직에 의해 생성된 상위 수준의 보안 정책을 수신하고, 이를 Client Facing Interface를 통해 보안 정책 관리자로 배포/전달하는 구성일 수 있다.

[0115] 2.6 정책 콜렉터

[0116] 정책 콜렉터는 업데이트된 상위 수준의 보안 정책을 Client Facing Interface를 통해 보안 컨트롤러로부터 수신하고, 이를 애플리케이션 로직으로 전달할 수 있다. NSF에서 발생하는 이벤트에 따라 하위 수준의 보안 정책이 업데이트될 수 있으므로, 상기와 같은 업데이트가 필요하다. 이벤트를 수신한 후 정책 콜렉터는, 이를 애플리케이션 로직으로 전달하여 애플리케이션 로직이 보안 컨트롤러로부터 수신한 해당 상위 수준의 보안 정책을 업데이트(또는 생성)하도록 할 수 있다.

[0117] 3. 사용예

[0118] 본 아키텍처는 가능한 보안 공격에 대응하도록 설계되었다. 본 절에서는 malware 도메인 및 VoIP/VoLTE 보안 공격에서 주어진 보안 공격 리스트에 대해 I2NSF 프레임 워크[i2nsf-framework]의 보안 공격 방어를 위한 절차를 예시한다.

[0119] 3.1. Malware 도메인 리스트에 대한 보안 관리

[0120] 도 5는 Malware 도메인 블랙리스트 작성의 보안 관리 아키텍처를 예시한 도면이다.

[0121] Malware 도메인 블랙리스트 작성은 악의적인 활동이 의심되는 가능한 공격 호스트, 서버 및 네트워크들의 IP 주소들을 유지 및 게시하는 것을 말한다.

[0122] Malware 도메인 블랙리스트 작성에 기반하여, Malware 도메인 리스트는 I2NSF 클라이언트의 Malware 도메인 관리자에 의해 수동 또는 자동으로 업데이트될 수 있다. 또한, Malware 도메인 관리자는 새롭게 추가된 Malware 도메인과의 패킷 전달을 방지하고 NSF의 하위 수준 보안 정책을 시행하기 위해, 주기적으로 새로운 상위 수준의 보안 정책을 생성할 수 있다. 또한, Malware 도메인 관리자는 새로운 상위 수준 보안 정책을 Policy Updater로 전송할 수 있으며, Policy Updater는 이를 보안 컨트롤러로 전달할 수 있다.

[0123] 업데이트된 하위 수준 정책은 NSF Facing Interface를 통해 NSF에 의해 보안 컨트롤러로 전송되어, 보안 컨트롤러가 하위 수준 정책과 대응하는 상위 수준 보안 정책을 생성하도록 할 수 있다. 보안 컨트롤러는 정책 콜렉터에 상위 수준 보안 정책을 제공할 수 있다. 정책 콜렉터는 애플리케이션 로직으로서 정책을 Malware 도메인 관리자에 전달할 수 있다.

[0124] 3.2 VoIP-VoLTE를 위한 보안 관리

[0125] VoIP-VoLTE 보안 관리는 불법적인 전화 및 인증이 의심되는 SIP 장치의 IP 주소, 소스 포트, 만료 시간, 사용자 에이전트 및 SIP(Session Initiation Protocol) URI의 블랙리스트를 유지 및 게시할 수 있다. 일반적인 보안 관리 아키텍처에서 VoIP-VoLTE 보안 관리자는 도 4의 VoIP-VoLTE 보안 서비스를 위한 애플리케이션 로직에 해당할 수 있다.

[0126] VoIP-VoLTE 보안 관리를 기반으로, 애플리케이션 로직 기능을 수행하는 VoIP-VoLTE 보안 관리자는 불법 장치 정보 리스트를 수동 또는 자동으로 업데이트할 수 있다. 또한, VoIP-VoLTE 보안 관리자는 새롭게 추가된 VoIP-VoLTE 공격자와의 패킷 전달을 방지하고 NSF의 하위 수준 보안 정책을 시행하기 위해 주기적으로 새로운 상위 수준 보안 정책을 생성할 수 있다. VoIP-VoLTE 보안 관리자는 새로운 상위 수준 보안 정책을 정책 업데이터로 전송할 수 있으며, 정책 업데이터는 이를 보안 컨트롤러로 전달할 수 있다.

[0127] NSF는 VoIP-VoLTE 공격에 대해 업데이트된 하위 수준 정책을 NSF Facing Interface를 통해 보안 컨트롤러로 전송되므로, 보안 컨트롤러가 IP 주소, 사용자 에이전트 및 해당 에이전트와 같은 상기 하위 수준 정책과 대응되는 상위 수준 보안 정책을 생성하고, 보안 컨트롤러에 의해 추가될 필요가 있는 시간 값을 만료시킬 수 있다. 보안 컨트롤러는 정책 수집기에 상위 수준 보안 정책을 제공할 수 있다. 정책 콜렉터는 애플리케이션 로직으로서 VoIP-VoLTE 보안 관리자에 정책을 전달할 수 있다.

[0128] 7. 보안 고려 사항

[0129] 보안 관리 아키텍처는 I2NSF 프레임 워크[i2nsf-framework]로부터 파생되므로, I2NSF 프레임 워크의 보안 고려사항이 존재할 수 있다. 특히, 제안된 아키텍처의 구성 요소들간에 제어 메시지 또는 관리 메시지를 전달하는데 적절한 보안 통신 채널이 사용되어야 한다.

[0131] **[3] 제3 실시예**

[0132] 본 실시예에서는 I2NSF(Interface to Network Security Functions) 프레임 워크의 보안 관리 아키텍처에 대해 설명한다. 이 보안 관리 아키텍처는 I2NSF 사용자, 보안 관리 시스템(즉, 보안 컨트롤러 및 개발자 관리 시스템) 및 I2NSF 프레임 워크의 NSF(Network Security Functions)를 포함할 수 있다. I2NSF 사용자는 애플리케이션 로직, 정책 업데이터 및 이벤트 콜렉터를 포함할 수 있다. 보안 컨트롤러는 보안 정책 관리자와 NSF 능력 관리자를 포함할 수 있다. 이하에서는 앞서 상술한 구성들의 기능과 보안 관리 처리에 대해 설명한다. 또한, 이하에서는 Malware 도메인 리스트 보안 관리, VoIP-VoLTE 보안 관리 및 시간별 액세스 제어와 같은 대표적인 사용 사례에 대해 설명한다. 각 구성에 관한 설명은 앞서 제1 및 제2 실시예와 관련하여 상술한 설명이 동일하게 적용될 수 있으며, 중복되는 설명은 생략한다.

[0133] 1. 도입

[0134] I2NSF 프레임 워크 [i2nsf-framework]에 사용자의 상위 수준 보안 정책을 적용하기 위해, I2NSF 사용자는 소비자 지향 인터페이스(Consumer Facing Interface)를 통해 보안 컨트롤러에 이러한 상위 수준 보안 정책을 제공할 수 있다. 이하에서는 I2NSF 프레임 워크의 주어진 상위 수준 정책을 위한 보안 관리 아키텍처를 제안한다. 이 아키텍처는 I2NSF 사용자, 보안 관리 시스템(즉, 보안 컨트롤러 및 개발자 관리 시스템) 및 I2NSF 프레임 워크의 NSF를 포함할 수 있다. I2NSF 사용자는 애플리케이션 로직, 정책 업데이터 및 이벤트 콜렉터를 포함할 수 있다. 보안 컨트롤러는 보안 정책 관리자 및 NSF 능력 관리자를 포함한다.

[0135] 보안 컨트롤러의 보안 정책 관리자와 NSF 능력 관리자는 I2NSF 사용자의 정책 업데이터가 Consumer Facing Interface를 통해 제공하는 업데이트된 보안 정책을 제어할 수 있다. 새 정책이 생성되었거나 기존 정책이 업데이트되어야 할 필요가 있는 경우, 정책 업데이터가 새 정책 및/또는 기존 정책의 업데이트 내용을 보안 컨트롤러에 제공할 수 있다. 반면, NSF가 하위 수준 정책을 변경해야 하는 이벤트가 발생하면, NSF는 해당 이벤트를 보안 컨트롤러로 전송할 수 있다. 보안 컨트롤러는 해당 이벤트를 이벤트 콜렉터로 전달하면, 이벤트 콜렉터가 이를 애플리케이션 로직으로 전달할 수 있다. 그 후, 애플리케이션 로직은 수신한 이벤트에 따라 현재 정책을 업데이트할 수 있다.

[0136] 제3 실시예에서는 보안 관리를 위한 추가 구성 요소를 I2NSF 프레임 워크에 통합하는 보안 관리 아키텍처를 제안한다. 이러한 아키텍처는 유연하고 효과적인 보안 정책을 지원하도록 설계되었다. 애플리케이션 로직은 상위 수준 정책을 생성하고 정책 업데이터는 이를 Consumer Facing Interface를 통해 보안 정책 관리자로 전송할 수 있다. 보안 정책 관리자는 상위 수준 정책을 보안 컨트롤러의 여러 하위 수준 정책들에 매핑할 수 있다. 매핑 후, 하위 수준의 정책들은 NSF에 배포되어 NSF에 적용될 수 있다.

[0137] 2. 목표

[0138] 제3 실시예에 따른 보안 관리 아키텍처의 두 가지 주 목적은 다음과 같다.

[0139] (1) 높은 수준의 보안 관리: NSF들에서의 유연하고 효과적인 보안 정책의 시행을 지원하기 위한 보안 관리 아키텍처의 설계를 제안한다.

[0140] (2) 보안 정책의 자동 업데이트: NSF들이 새로운 보안 공격에 대해 하위 수준 정책을 변경해야 하는 이벤트가 발생한 경우, 해당 이벤트를 대응하는 상위 수준의 보안 정책들에 반영하기 위함이다.

[0141] 3. 보안 관리 아키텍처

[0142] 본 절에서는 I2NSF의 보안 관리 아키텍처에 대해 설명하고 보안 컨트롤러 및 개발자 관리 시스템이 포함된 보안 관리 시스템에 중점을 두고 설명한다. 또한, 본 절에서는 보안 컨트롤러의 기본 동작에 대해 설명하고 아키텍처에 포함되는 각 구성 요소의 세부 사항을 설명한다.

[0143] 도 6은 I2NSF 프레임 워크 내의 보안 관리 아키텍처를 예시한 도면이다.

[0144] 도 6의 아키텍처는 유연하고 효과적인 보안 정책의 시행을 지원하도록 설계되었다. I2NSF 사용자의 애플리케이션 로직은 새로운 보안 공격에 따라 상위 수준의 정책을 생성하고, I2NSF 사용자의 정책 업데이터는 상기 상위 수준의 정책을 보안 컨트롤러의 보안 정책 관리자에게 보낸다. 보안 정책 관리자는 상위 수준 정책을 NSF 능력 관리자에 등록된 NSF 능력과 관련된 몇 개의 하위 수준 정책들에 매핑할 수 있다. 매핑 후, 보안 정책 관리자는 NSF Facing Interface를 통해 하위 수준 정책들을 NSF(들)로 배포할 수 있다. 다음 절에서는 각 구성 요소의 세부 사항에 대해 설명한다.

[0145] 4.1 보안 정책 관리자

[0146] 보안 정책 관리자는 Consumer Facing Interface를 통해 정책 업데이터로부터 상위 수준 정책을 수신하고, 상위 수준 정책을 NSF 능력 관리자의 특정 NSF 능력과 관련된 몇 개의 하위 수준 정책들로 매핑하는 구성이다. 또한, 보안 정책 관리자는 이러한 하위 수준 정책들을 NSF Facing Interface를 통해 NSF(들)로 전달할 수 있다.

[0147] 한편, 하위 레벨 정책을 변경해야 하는 이벤트가 NSF에서 발생하면, NSF는 NSF Facing Interface를 통해 보안 정책 관리자에게 해당 이벤트를 전송할 수 있다. 이후, 보안 정책 관리자는 Consumer Facing Interface를 통해 해당 이벤트를 이벤트 콜렉터로 전송한다.

[0148] 4.2 NSF 능력 관리자

[0149] NSF 능력 관리자는 보안 컨트롤러에 통합된 구성이다. NSF 능력 관리자는 등록 인터페이스를 통해 개발자 관리 시스템에 등록된 NSF 능력을 저장하고, 이를 보안 정책 관리자와 공유하여 보안 정책 관리자가 NSF 능력과 관련된 하위 수준 정책을 생성하도록 할 수 있다. 또한, 새로운 NSF가 등록될 때마다, NSF 능력 관리자는 등록 인터페이스를 통해 NSF 능력 관리자의 관리 테이블에 NSF 능력을 등록하도록 개발자 관리 시스템에 요청할 수 있다. 한편, 기존의 NSF가 삭제되면, NSF 능력 관리자는 관리 테이블에서 NSF 능력을 제거한다.

[0150] 4.3 개발자 관리 시스템

[0151] 개발자 관리 시스템은 등록 인터페이스를 통해 NSF 능력 관리자에 새로운 NSF 능력을 등록하는 구성이다. 또한, 등록된 NSF에 업데이트가 있는 경우, 해당 업데이트는 개발자 관리 시스템에서 NSF 능력 관리자로 전달된다.

[0152] 4.4. 애플리케이션 로직

[0153] 애플리케이션 로직은 보안 공격을 차단 또는 완화하기 위한 상위 수준 보안 정책을 생성하고, 생성된 정책을 정책 업데이터로 전송하는 구성이다. 이러한 애플리케이션 로직에 대해서는 이하의 사용예에서 자세한 동작을 설명한다.

[0154] 4.5. 정책 업데이터

[0155] 정책 업데이터는 애플리케이션 로직에서 생성된 상위 수준 보안 정책을 수신하고, 이를 Consumer Facing Interface를 통해 보안 정책 관리자에게 전달하는 구성이다.

[0156] 4.6 이벤트 콜렉터

[0157] 이벤트 콜렉터는 애플리케이션 로직의 상위 수준 정책을 업데이트(또는 생성)할 때 반영되어야 하는 이벤트를 보안 컨트롤러로부터 수신한다. NSF에서 발생하는 특정 이벤트에 따라 하위 수준의 보안 정책이 업데이트되므로, NSF에서 이벤트를 수신하는 절차가 필요하다. 이벤트를 수신한 후, 이벤트 콜렉터는 이를 애플리케이션 로직으로 전달하여, 애플리케이션 로직이 보안 컨트롤러로부터 수신한 이벤트를 기반으로 상위 수준의 보안 정책을 업데이트(또는 생성)할 수 있도록 한다.

[0158] 5. 사용예

[0159] 본 아키텍처는 실제 환경에서 발생할 수 있는 보안 공격에 대응하도록 설계된다. 본 절에서는 Malware 도메인에서 주어진 보안 공격 리스트, VoIP/VoLTE 보안 공격 및 시간별 액세스 제어에 대한 I2NSF 프레임 워크[i2nsf-framework]의 보안 공격에 대한 방어 절차를 설명한다.

[0160] 5.1. Malware 도메인 리스트 보안 관리

[0161] Malware 도메인 차단 리스트 작성은 악의적인 활동이 의심되는 가능한 공격 호스트, 서버 및 네트워크의 IP 주소들을 유지 및 게시한다.

[0162] 도 7은 Malware 도메인 블랙리스트 작성의 보안 관리 아키텍처를 예시한 도면이다.

[0163] Malware 도메인 블랙리스트 작성에 기초하여, Malware 도메인들의 리스트는 I2NSF 사용자의 Malware 도메인 관리자에 의해 수동 또는 자동으로 업데이트될 수 있다. 또한, Malware 도메인 관리자는 새로 추가된 Malware 도메인과의 패킷 전달을 방지하고 NSF의 하위 수준 보안 정책들을 시행하기 위해 주기적으로 새로운 상위 수준 보안 정책을 생성할 수 있다. 또한, Malware 도메인 관리자는 새로운 상위 수준 보안 정책을 정책 업데이터로 전송할 수 있으며, 정책 업데이터는 보안 컨트롤러로 이를 전달할 수 있다.

[0164] NSF가 새로운 위험 도메인을 탐지하면, 해당 IP 주소를 NSF Facing Interface를 통해 보안 컨트롤러로 전송할 수 있다. 보안 컨트롤러는 이벤트 콜렉터에 IP 주소를 전달하며, 이벤트 콜렉터는 해당 IP 주소를 위험 도메인 관리자에게 전달할 수 있다. 이에 기초하여, 위험 도메인 관리자는 위험 도메인 데이터 베이스를 업데이트할 수 있다.

[0165] 5.2 VoIP-VoLTE에 대한 보안 관리

[0166] VoIP-VoLTE 보안 관리는 불법적인 전화 및 인증이 의심되는 SIP 장치의 IP 주소, 소스 포트, 만료 시간, 사용자 에이전트 및 SIP(Session Initiation Protocol) URI의 블랙리스트를 유지 관리하고 게시한다. 보안 관리 아키텍처에서 VoIP-VoLTE 보안 관리자는 도 6에서의 VoIP-VoLTE 보안 서비스를 위한 애플리케이션 로직의 역할을 수행한다.

[0167] VoIP-VoLTE 보안 관리에 기초하여, 애플리케이션 로직 기능을 수행하는 VoIP-VoLTE 보안 관리자는 불법 장치 정보의 목록를 수동 또는 자동으로 업데이트할 수 있다. 또한, VoIP-VoLTE 보안 관리자는 새롭게 추가된 VoIP-VoLTE 공격자와의 패킷 전달을 방지하고 NSF의 하위 수준 보안 정책을 시행하기 위해 주기적으로 새로운 상위 수준 보안 정책을 생성할 수 있다. 또한, VoIP-VoLTE 보안 관리자는 새로운 상위 수준 보안 정책을 정책 업데이터로 전송할 수 있으며, 정책 업데이터는 이를 보안 컨트롤러로 전달할 수 있다.

[0168] NSF가 도메인으로부터 전달된 비정상적인 메시지 또는 전화를 검출하는 경우, IP 주소, 사용자 에이전트 및 만료 시간 값과 같은 도메인 정보는 NSF acing 인터페이스를 통해 NSF에 의해 보안 컨트롤러로 전송된다. 보안 컨트롤러는 이벤트 콜렉터에 탐지된 도메인 정보를 전달하고, 이벤트 콜렉터는 탐지된 도메인 정보를 VoIP-VoLTE 보안 관리자로 전달하며, VoIP-VoLTE 보안 관리자는 탐지된 도메인 정보에 기초하여 VoIP-VoLTE 데이터 베이스를 업데이트한다.

[0169] 5.3 시간별 액세스 제어에 대한 보안 관리

[0170] 시간별 액세스 제어 정책은 특정 기간 동안 특정 웹 사이트에 대한 사용자의 액세스를 관리한다. 예를 들어, 회사에서 관리자는 직원이 근무 시간 동안 업무에 방해가 될 수 있는 Youtube에 액세스하는 것을 차단할 수 있다.

[0171] I2NSF 사용자는, 시간별 액세스 제어에 기초하여, 애플리케이션 로직에 차단된 웹 사이트 및 차단 시간의 리스

트를 등록한다. 애플리케이션 로직은 리스트를 데이터 베이스에 저장하고 상위 수준의 보안 정책(예를 들어, 리스트에서 차단된 웹 사이트 및 차단 시간을 확인하여 웹 사이트에 대한 액세스 차단)을 생성한다. 애플리케이션 로직은 이를 정책 업데이터로 전달하면, 정책 업데이터는 이를 보안 컨트롤러로 전달한다. 보안 컨트롤러에서 보안 정책 관리자는 상위 수준 정책을 하위 수준 정책들에 매핑한 다음, NSF가 하위 수준 정책들을 적용할 수 있도록 이들을 NSF로 전송한다.

[0172] 6. 보안 고려 사항

[0173] 보안 관리 아키텍처는 I2NSF 프레임 워크[i2nsf-framework]로부터 파생되므로, I2NSF 프레임 워크의 보안 고려사항이 존재할 수 있다. 특히, 제안된 아키텍처의 구성 요소들간에 제어 메시지 또는 관리 메시지를 전달하는데 적절한 보안 통신 채널이 사용되어야 한다.

[0174] 도 8은 본 발명의 일 실시예에 따른 보안 관리 시스템의 보안 관리 방법에 관한 순서도이다. 본 순서도와 관련하여 앞서 상술한 실시예들이 동일/유사하게 적용될 수 있으며, 중복되는 설명은 생략할 수 있다.

[0175] 우선, 보안 관리 시스템은 NSF 클라이언트로부터 보안 공격을 차단 또는 완화하기 위한 상위 수준 정책을 수신할 수 있다(S810).

[0176] 다음으로, 보안 관리 시스템은 상기 상위 수준 정책을 보안 관리 시스템에 등록된 NSF 능력과 관련된 하위 수준 보안 정책들에 매핑할 수 있다(S820).

[0177] 마지막으로, 보안 관리 시스템은 하위 수준 보안 정책들을 적어도 하나의 NSF에 전달할 수 있다(S830).

[0178] NSF 클라이언트는 애플리케이션 로직, 정책 업데이터 및/또는 이벤트 콜렉터를 포함할 수 있다. 여기서, 애플리케이션 로직은 상위 수준 정책을 생성 및 업데이트하여 정책 업데이터로 전송하며, 정책 업데이터는 상위 수준 정책을 클라이언트 지향 인터페이스를 통해 보안 관리 시스템으로 전달하며, 이벤트 콜렉터는 상위 수준 정책의 생성 또는 업데이트에 기초가 되는 이벤트를 수신하여 애플리케이션 로직으로 전송할 수 있다.

[0179] 상위 수준 정책은 일 실시예로서, 특정 공격 호스트, 서버 및 네트워크의 IP 주소가 포함된 블랙리스트를 기반으로 생성될 수 있다. 이 경우, 이벤트는 상기 블랙리스트로의 포함 기준을 만족하는 IP 주소에 해당할 수 있다.

[0180] 또는, 상위 수준 정책은 다른 실시예로서, 차단 웹 사이트 및 차단 시간이 포함된 블랙리스트를 기반으로 생성될 수 있다.

[0181] 또는, 상위 수준 정책은 다른 실시예로서, 특정 SIP 장치의 IP 주소, 소스 포트, 만료 시간, 사용자 에이전트 및/또는 SIP URI가 포함된 불법 장치 차단 목록을 기반으로 생성될 수 있다. 이 경우, 이벤트는 불법 장치 차단 목록으로의 포함 기준을 만족하는 도메인 정보에 해당할 수 있다.

[0182] 보안 관리 시스템은 보안 정책 관리자, NSF 능력 관리자 및/또는 개발자 관리 시스템을 포함할 수 있다. 개발자 관리 시스템은 등록 인터페이스를 통해 NSF 능력을 등록 및 업데이트하며, NSF 능력 관리자는 개발자 관리 시스템에 등록 및 업데이트된 NSF 능력을 저장할 수 있다. 보안 정책 관리자는 상위 수준 정책을 NSF 능력 관리자에 저장된 NSF 능력과 관련된 하위 수준 보안 정책들과 매핑하고, 하위 수준 보안 정책들을 NSF 지향 인터페이스(NSF Facing Interface)를 통해 적어도 하나의 NSF로 전달할 수 있다.

[0184] 본 발명에 따른 실시예는 다양한 수단, 예를 들어, 하드웨어, 펌웨어(firmware), 소프트웨어 또는 그것들의 결합 등에 의해 구현될 수 있다. 하드웨어에 의한 구현의 경우, 본 발명의 일 실시예는 하나 또는 그 이상의 ASICs(application specific integrated circuits), DSPs(digital signal processors), DSPDs(digital signal processing devices), PLDs(programmable logic devices), FPGAs(field programmable gate arrays), 프로세서, 콘트롤러, 마이크로 콘트롤러, 마이크로 프로세서 등에 의해 구현될 수 있다.

[0185] 또한, 펌웨어나 소프트웨어에 의한 구현의 경우, 본 발명의 일 실시예는 이상에서 설명된 기능 또는 동작들을 수행하는 모듈, 절차, 함수 등의 형태로 구현되어, 다양한 컴퓨터 수단을 통하여 판독 가능한 기록매체에 기록될 수 있다. 여기서, 기록매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 기록매체에 기록되는 프로그램 명령은 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 예컨대 기록매체는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(Magnetic Media), CD-ROM(Compact Disk Read Only Memory), DVD(Digital Video Disk)와 같은 광 기록 매체(Optical Media), 플롭티컬 디스크(Floptical Disk)와 같은 자기-광 매체

(Magneto-Optical Media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치를 포함한다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함할 수 있다. 이러한 하드웨어 장치는 본 발명의 동작을 수행하기 위해 하나 이상의 소프트웨어 모로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0186] 아울러, 본 발명에 따른 장치나 단말은 하나 이상의 프로세서로 하여금 앞서 설명한 기능들과 프로세스를 수행하도록 하는 명령에 의하여 구동될 수 있다. 예를 들어 그러한 명령으로는, 예컨대 JavaScript나 ECMAScript 명령 등의 스크립트 명령과 같은 해석되는 명령이나 실행 가능한 코드 혹은 컴퓨터로 판독 가능한 매체에 저장되는 기타의 명령이 포함될 수 있다. 나아가 본 발명에 따른 장치는 서버 팜(Server Farm)과 같이 네트워크에 걸쳐서 분산형으로 구현될 수 있으며, 혹은 단일의 컴퓨터 장치에서 구현될 수도 있다.

[0187] 또한, 본 발명에 따른 장치에 탑재되고 본 발명에 따른 방법을 실행하는 컴퓨터 프로그램(프로그램, 소프트웨어, 소프트웨어 어플리케이션, 스크립트 혹은 코드로도 알려져 있음)은 컴파일 되거나 해석된 언어나 선험적 혹은 절차적 언어를 포함하는 프로그래밍 언어의 어떠한 형태로도 작성될 수 있으며, 독립형 프로그램이나 모듈, 컴포넌트, 서브루틴 혹은 컴퓨터 환경에서 사용하기에 적합한 다른 유닛을 포함하여 어떠한 형태로도 전개될 수 있다. 컴퓨터 프로그램은 파일 시스템의 파일에 반드시 대응하는 것은 아니다. 프로그램은 요청된 프로그램에 제공되는 단일 파일 내에, 혹은 다중의 상호 작용하는 파일(예컨대, 하나 이상의 모듈, 하위 프로그램 혹은 코드의 일부를 저장하는 파일) 내에, 혹은 다른 프로그램이나 데이터를 보유하는 파일의 일부(예컨대, 마크업 언어 문서 내에 저장되는 하나 이상의 스크립트) 내에 저장될 수 있다. 컴퓨터 프로그램은 하나의 사이트에 위치하거나 복수의 사이트에 걸쳐서 분산되어 통신 네트워크에 의해 상호 접속된 다중 컴퓨터나 하나의 컴퓨터 상에서 실행되도록 전개될 수 있다.

[0188] 설명의 편의를 위하여 각 도면을 나누어 설명하였으나, 각 도면에 서술되어 있는 실시예들을 병합하여 새로운 실시예를 구현하도록 설계하는 것도 가능하다. 또한, 본 발명은 상술한 바와 같이 설명된 실시예들의 구성과 방법이 한정되게 적용될 수 있는 것이 아니라, 상술한 실시예들은 다양한 변형이 이루어질 수 있도록 각 실시 예들의 전부 또는 일부가 선택적으로 조합되어 구성될 수도 있다.

[0189] 또한, 이상에서는 바람직한 실시예에 대하여 도시하고 설명하였지만, 본 명세서는 상술한 특정의 실시예에 한정되지 아니하며, 청구 범위에서 청구하는 요지를 벗어남이 없이 당해 명세서가 속하는 기술분야에서 통상의 지식을 가진 자에 의해 다양한 변형실시가 가능한 것은 물론이고, 이러한 변형 실시들은 본 명세서의 기술적 사상이나 전망으로부터 개별적으로 이해되어서는 안될 것이다.

부호의 설명

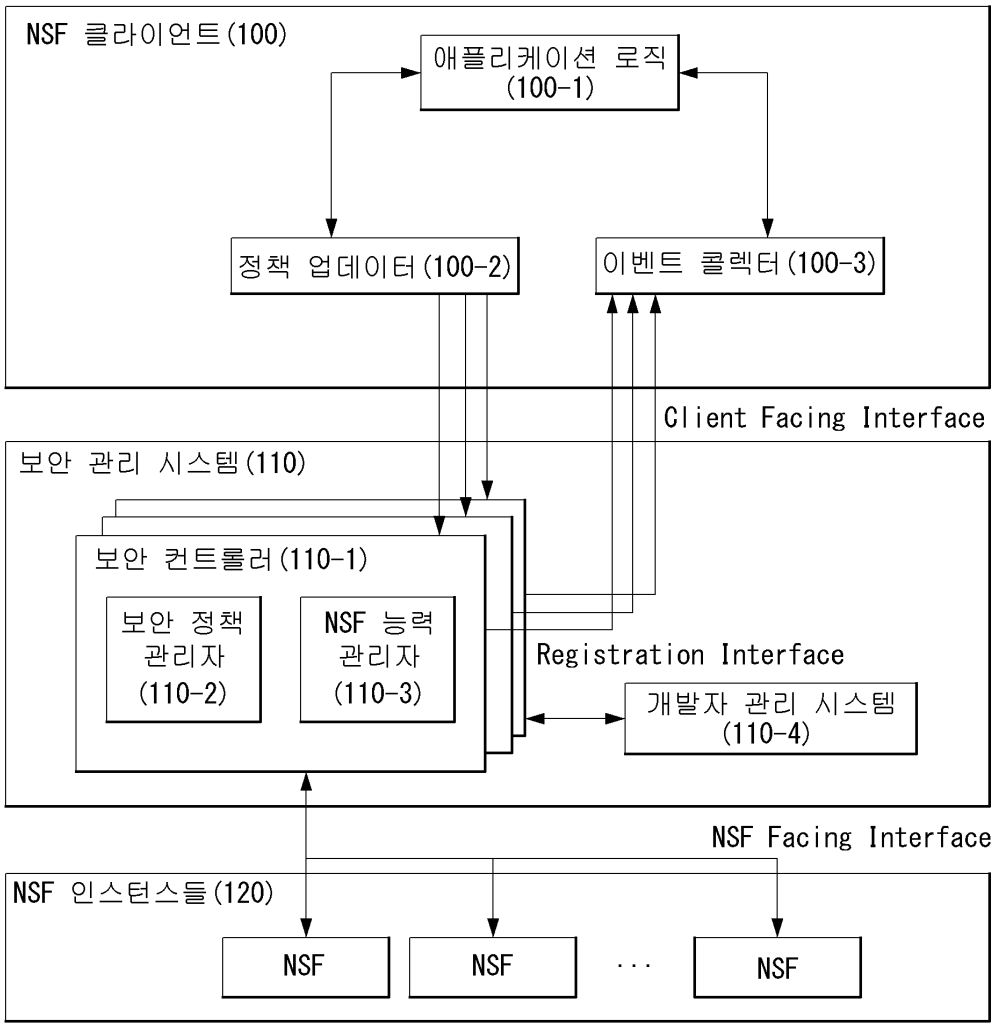
[0190] 100: NSF 클라이언트

110: 보안 관리 시스템

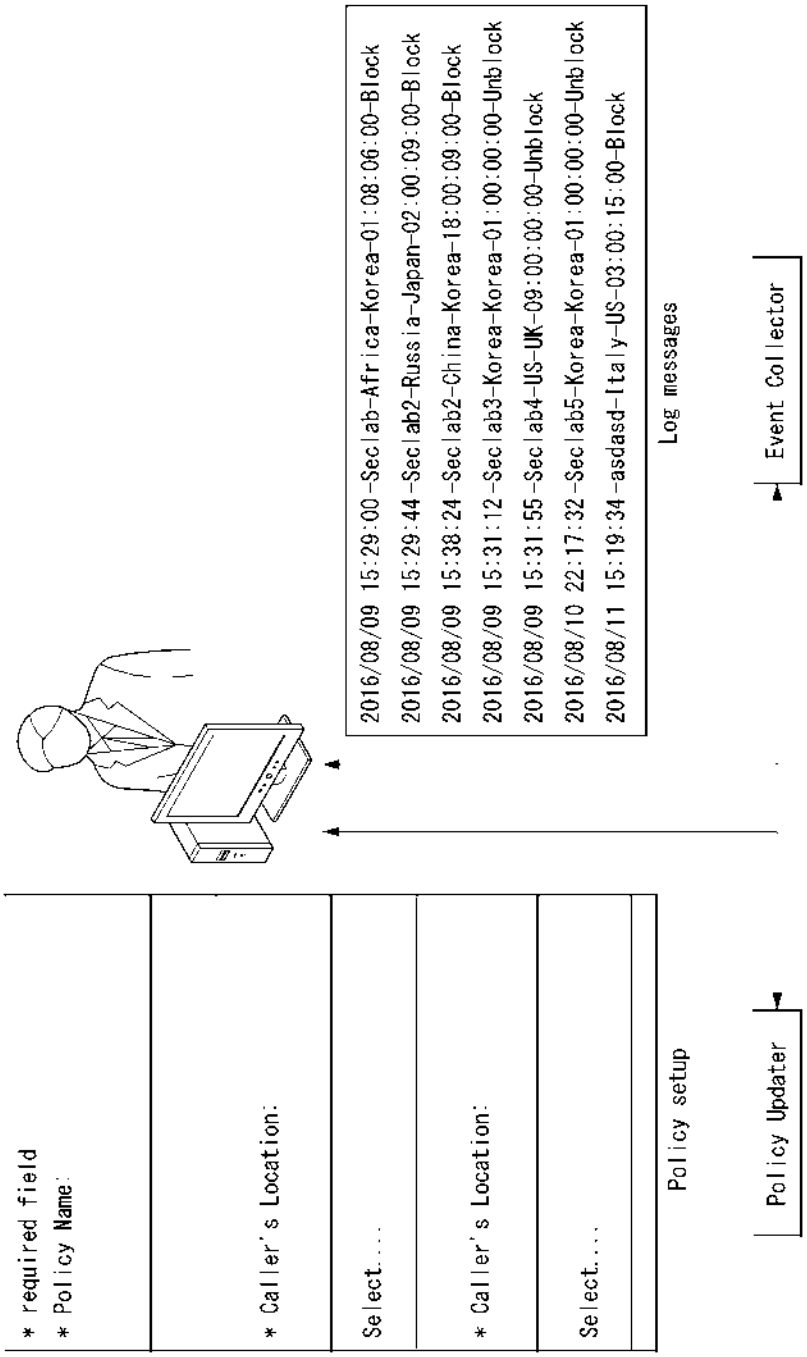
120: NSF 인스턴스들

도면

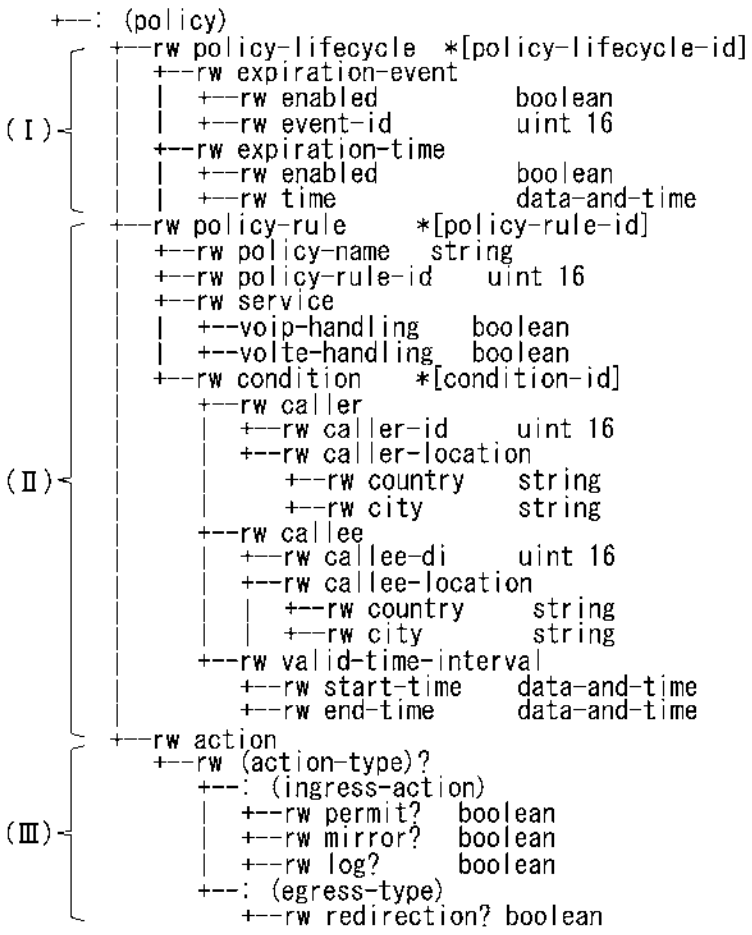
도면1


NSF 클라이언트(100)
애플리케이션 로직
(100-1)
정책 업데이터(100-2)
이벤트 콜렉터(100-3)
Client Facing Interface
보안 관리 시스템(110)
보안 컨트롤러(110-1)
보안 정책
관리자
(110-2)
NSF 능력
관리자
(110-3)
Registration Interface
개발자 관리 시스템
(110-4)
NSF Facing Interface
NSF 인스턴스들(120)
NSF
NSF
···
NSF

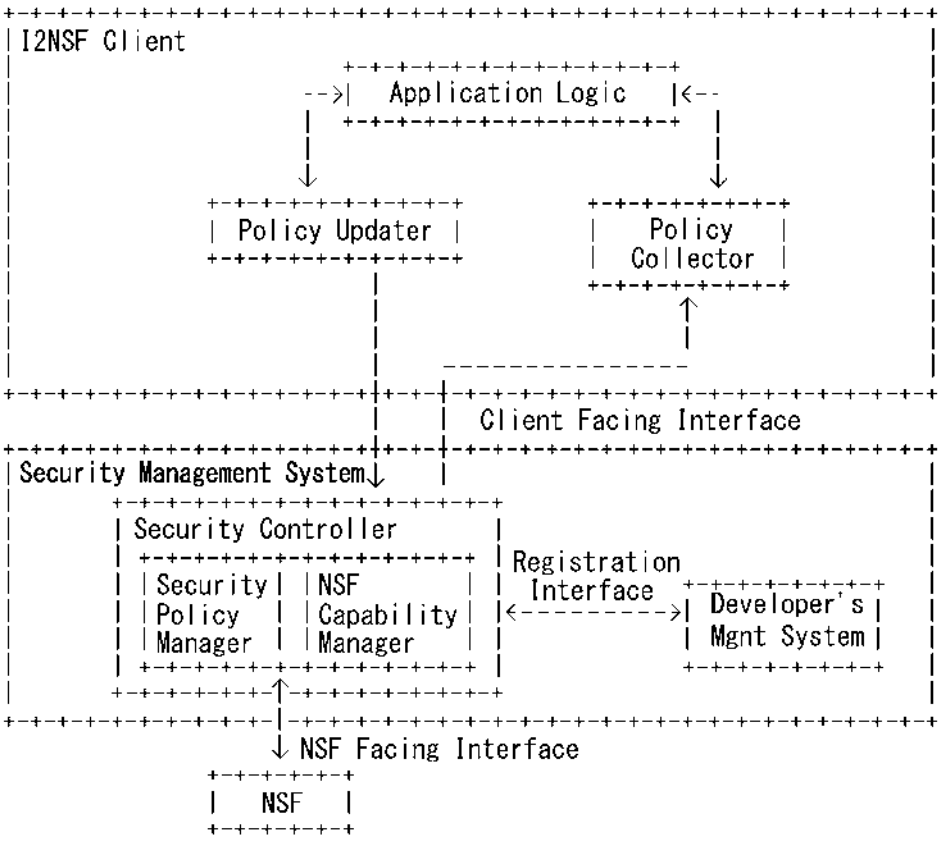
도면2


* required field
* Policy Name:
* Caller's Location:
Select....
* Caller's Location:
Select....
Policy setup
Policy Updater
2016/08/09 15:29:00-Seclab-Africa-Korea-01:08:06:00-Block
2016/08/09 15:29:44-Seclab2-Russia-Japan-02:00:09:00-Block
2016/08/09 15:38:24-Seclab2-China-Korea-18:00:09:00-Block
2016/08/09 15:31:12-Seclab3-Korea-Korea-01:00:00:00-Unblock
2016/08/09 15:31:55-Seclab4-US-UK-09:00:00:00-Unblock
2016/08/10 22:17:32-Seclab5-Korea-Korea-01:00:00:00-Unblock
2016/08/11 15:19:34-asdasd-Italy-US-03:00:15:00-Block
Log messages
Event Collector

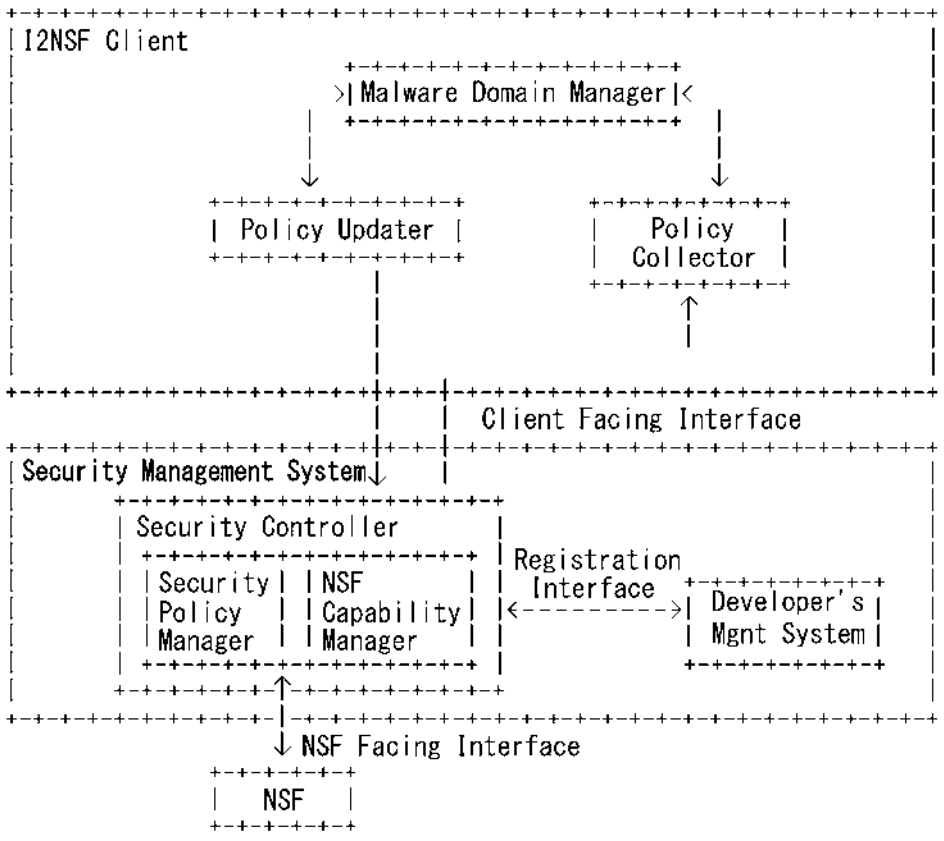
도면3


+--: (policy)
+--rw policy-lifecycle *[policy-lifecycle-id]
+--rw expiration-event
+--rw enabled boolean
+--rw event-id uint 16
+--rw expiration-time
+--rw enabled boolean
+--rw time data-and-time
(Ⅰ)
+--rw policy-rule *[policy-rule-id]
+--rw policy-name string
+--rw policy-rule-id uint 16
+--rw service
+--voip-handling boolean
+--volte-handling boolean
+--rw condition *[condition-id]
+--rw caller
+--rw caller-id uint 16
+--rw caller-location
+--rw country string
+--rw city string
+--rw callee
+--rw callee-di uint 16
+--rw callee-location
+--rw country string
+--rw city string
+--rw valid-time-interval
+--rw start-time data-and-time
+--rw end-time data-and-time
(Ⅱ)
+--rw action
+--rw (action-type)?
+--: (ingress-action)
+--rw permit? boolean
+--rw mirror? boolean
+--rw log? boolean
+--: (egress-type)
+--rw redirection? boolean
(Ⅲ)

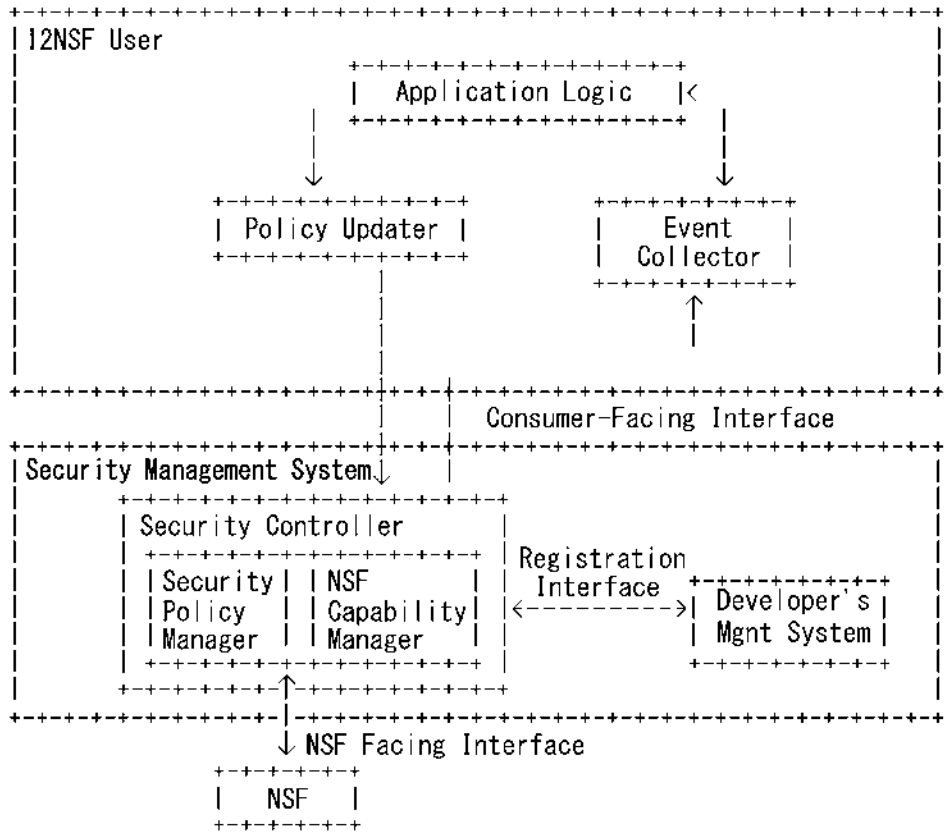
도면4


I2NSF Client
Application Logic
Policy Updater
Policy Collector
Client Facing Interface
Security Management System
Security Controller
Security Policy Manager
NSF Capability Manager
Registration Interface
Developer's Mgnt System
NSF Facing Interface
NSF

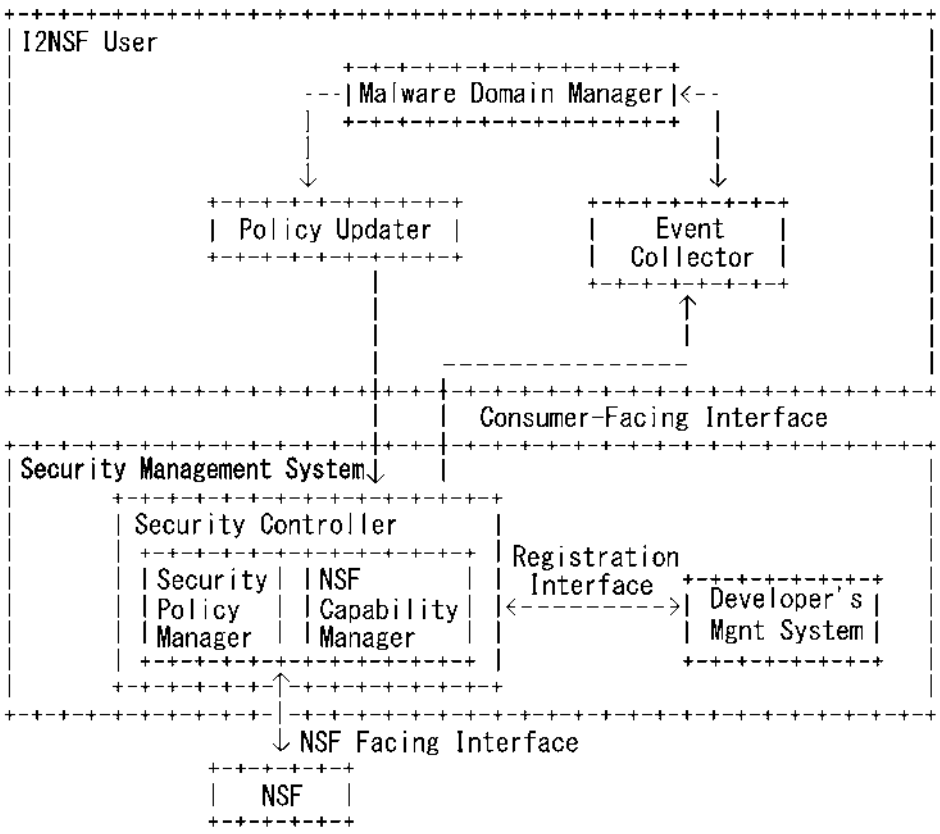
도면5


I2NSF Client
Malware Domain Manager
Policy Updater
Policy Collector
Client Facing Interface
Security Management System
Security Controller
Security Policy Manager
NSF Capability Manager
Registration Interface
Developer's Mgnt System
NSF Facing Interface
NSF

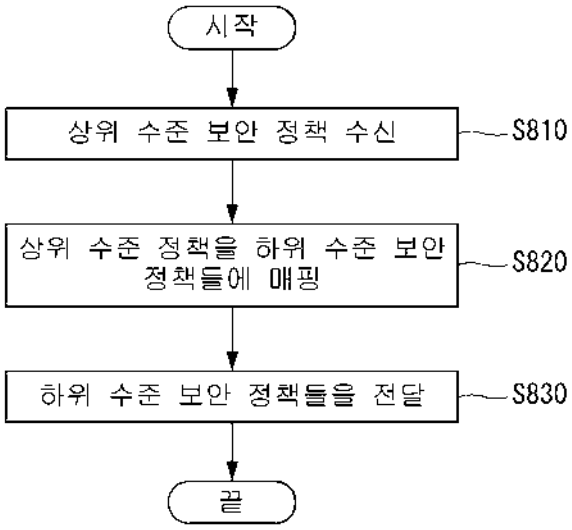
도면6


I2NSF User
Application Logic
Policy Updater
Event Collector
Consumer-Facing Interface
Security Management System
Security Controller
Security Policy Manager
NSF Capability Manager
Registration Interface
Developer's Mgnt System
NSF Facing Interface
NSF

도면7


I2NSF User
Malware Domain Manager
Policy Updater
Event Collector
Consumer-Facing Interface
Security Management System
Security Controller
Security Policy Manager
NSF Capability Manager
Registration Interface
Developer's Mgnt System
NSF Facing Interface
NSF

도면8


시작
상위 수준 보안 정책 수신
S810
상위 수준 정책을 하위 수준 보안 정책들에 매핑
S820
하위 수준 보안 정책들을 전달
S830
끝

(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0045400
(43) 공개일자 2020년05월04일

(51) 국제특허분류(Int. Cl.)
H04L 29/06 (2006.01) *H04L 12/24* (2006.01)
(52) CPC특허분류
H04L 63/20 (2013.01)
H04L 41/145 (2013.01)
(21) 출원번호 10-2019-0125454
(22) 출원일자 2019년10월10일
심사청구일자 2019년10월10일
(30) 우선권주장
1020180126339 2018년10월22일 대한민국(KR)

(71) 출원인
성균관대학교산학협력단
경기도 수원시 장안구 서부로 2066 (천천동, 성균관대학교내)
(72) 발명자
정재훈
부산광역시 금정구 금강로 225, 207동 2203호(장전동, 벽산블루밍디자인시티)
양진혁
경기도 화성시 동탄공원로 21-12, 907동 1302호(능동, 푸른마을 포스코더샵2차)
(74) 대리인
특허법인로얄

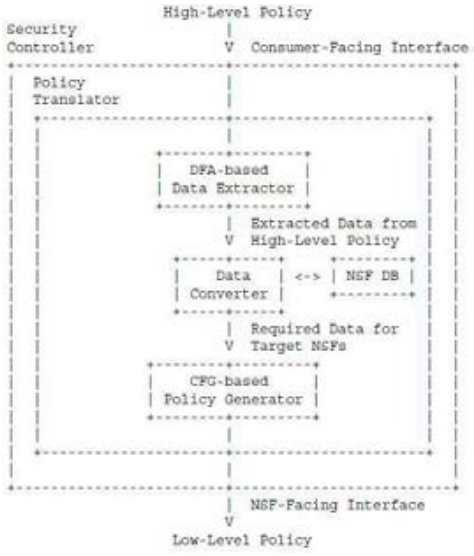
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 **네트워크 보안 기능 인터페이스를 위한 보안 정책 번역**

(57) 요 약

본 명세서는 네트워크 보안 기능 인터페이스를 위한 정책 번역기(Policy Translator)의 보안 정책을 번역하기 위한 방법에 관한 것으로서, I2NSF(Interface to Network Security Functions) 사용자로부터 상위 레벨(High-Level)의 제1 보안 정책을 수신하는 단계; 상기 제1 보안 정책에 근거하여, 상기 제1 보안 정책과 관련된 데이터를 추출하는 단계; 상기 제 1 보안 정책과 관련된 데이터를 NSF(Network Security Function)를 위한 필수 데이터로 변환하는 단계; 및 상기 필수 데이터에 근거하여, 대상(Target) NSF를 검색하고, 상기 대상 NSF와 관련된 하위 레벨(Low-Level)의 제2 보안 정책을 생성하는 단계; 를 포함하되, 상기 정책 번역기와 상기 I2NSF 사용자는 사용자-직면 인터페이스로 연결되며, 상기 정책 번역기와 상기 NSF는 NSF-직면 인터페이스로 연결되는 보안 정책을 번역할 수 있다.

대 표 도 - 도27



이 발명을 지원한 국가연구개발사업
과제고유번호 2016-0-00078
부처명 과학기술정보통신부
연구관리전문기관 정보통신기술진흥센터
연구사업명 정보통신.방송 연구개발사업(정보보호핵심원천기술개발사업)
연구과제명 맞춤형 보안서비스 제공을 위한 클라우드 기반 지능형 보안 기술 개발
기 여 율 1/1
주관기관 성균관대학교 산학협력단
연구기간 2018.01.01 ~ 2018.12.31

명 세 서

청구범위

청구항 1

네트워크 보안 기능 인터페이스를 위한 정책 번역기(Policy Translator)의 보안 정책을 번역하기 위한 방법에 있어서,

I2NSF(Interface to Network Security Functions) 사용자로부터 상위 레벨(High-Level)의 제1 보안 정책을 수신하는 단계;

상기 제1 보안 정책에 근거하여, 상기 제1 보안 정책과 관련된 데이터를 추출하는 단계;

상기 제 1 보안 정책과 관련된 데이터를 NSF(Network Security Function)를 위한 필수 데이터로 변환하는 단계; 및

상기 필수 데이터에 근거하여, 대상(Target) NSF를 검색하고, 상기 대상 NSF와 관련된 하위 레벨(Low-level)의 제2 보안 정책을 생성하는 단계;

를 포함하되,

상기 정책 번역기와 상기 I2NSF 사용자는 사용자-직면 인터페이스로 연결되며, 상기 정책 번역기와 상기 NSF는 NSF-직면 인터페이스로 연결되는 보안 정책을 번역하기 위한 방법.

청구항 2

제1항에 있어서,

상기 제2 보안 정책과 관련된 데이터를 생성하는 단계;

를 더 포함하고,

상기 제2 보안 정책과 관련된 데이터는 상기 대상 NSF를 위한 구조 또는 내용을 포함하며, 태그(Tag)를 통해 그룹화되는 방법.

청구항 3

제2항에 있어서,

상기 제2 보안 정책과 관련된 데이터를 상기 NSF-직면 인터페이스를 통해, 상기 대상 NSF로 전달하는 단계;

를 더 포함하는 방법.

청구항 4

제3항에 있어서,

상기 제2 보안 정책과 관련된 데이터에 근거하여, 상기 대상 NSF를 설정하는 단계;

를 더 포함하는 방법.

청구항 5

제1항에 있어서,

상기 제2 보안 정책은

적용되는 정책 규칙, 및 일반적인 보안 기능을 위한 동작을 나타나내는 기본 동작 정보를 포함하는 방법.

청구항 6

제5항에 있어서,

상기 정책 규칙은

정책 정보 및 규칙 정보를 포함하며,

상기 정책 정보 및 상기 규칙 정보는 시스템의 변경을 나타내는 이벤트 절(Event Clause), 정책 규칙의 적용 조건을 나타내는 조건 절(Condition Clause), 및 상기 이벤트 절 및 상기 조건 절을 만족할 때 수행되는 보안 기능을 나타내는 동작 절(Action Clause)을 포함하는 방법.

청구항 7

제1항에 있어서,

제1 보안 정책과 관련된 데이터를 추출하는 단계는

DFA(Deterministic Finite Automation)를 이용하는, 방법.

청구항 8

제1항에 있어서,

상기 필수 데이터로 변환하는 단계는

NSF 데이터베이스를 이용하여, 상기 NSF 데이터베이스의 데이터와 매핑을 통해 수행되는, 방법.

청구항 9

제1항에 있어서,

상기 대상 NSF는

상기 제1 보안 정책을 모두 커버(cover)할 수 있는 능력(capability)을 갖는, 방법.

청구항 10

제1항에 있어서,

상기 정책 번역기는

상기 NSF의 모든 기능들이 등록된, 방법.

청구항 11

네트워크 보안 기능 인터페이스를 위한 보안 정책을 번역하는 정책 번역기(Policy Translator)에 있어서,

I2NSF(Interface to Network Security Functions) 사용자로부터 수신된, 상위 레벨(High-Level)의 제1 보안 정책에 근거하여, 상기 제1 보안 정책과 관련된 데이터를 추출하는 추출기(Extractor);

상기 제 1 보안 정책과 관련된 데이터를 NSF(Network Security Function)를 위한 필수 데이터로 변환하는 데이터 변환기(Data Converter); 및

상기 필수 데이터에 근거하여, 대상(Target) NSF를 검색하고, 상기 대상 NSF와 관련된 하위 레벨(Low-Level)의 제2 보안 정책을 생성하는 생성기(Generator);

를 포함하되,

상기 정책 번역기와 상기 I2NSF 사용자는 사용자-직면 인터페이스로 연결되며, 상기 정책 번역기와 상기 NSF는 NSF-직면 인터페이스로 연결되는 정책 번역기.

청구항 12

제11항에 있어서,

상기 생성기는

상기 제2 보안 정책과 관련된 데이터를 생성하고,

상기 제2 보안 정책과 관련된 데이터는 상기 대상 NSF를 위한 구조 또는 내용을 포함하며, 태그(Tag)를 통해 그룹화되는 정책 번역기.

청구항 13

제 12 항에 있어서,

상기 제2 보안 정책과 관련된 데이터는

상기 NSF-직면 인터페이스를 통해, 상기 대상 NSF로 전달되는 정책 번역기.

청구항 14

제 13 항에 있어서,

상기 대상 NSF는

상기 제2 보안 정책과 관련된 데이터에 근거하여, 설정되는 정책 번역기.

청구항 15

제 11 항에 있어서,

상기 제2 보안 정책은

적용되는 정책 규칙, 및 일반적인 보안 기능을 위한 동작을 나타나내는 기본 동작 정보를 포함하는 정책 번역기.

청구항 16

제 15 항에 있어서,

상기 정책 규칙은

정책 정보 및 규칙 정보를 포함하며,

상기 정책 정보 및 상기 규칙 정보는 시스템의 변경을 나타내는 이벤트 절(Event Clause), 정책 규칙의 적용 조건을 나타내는 조건 절(Condition Clause), 및 상기 이벤트 절 및 상기 조건 절을 만족할 때 수행되는 보안 기능을 나타내는 동작 절(Action Clause)을 포함하는 정책 번역기.

청구항 17

제11항에 있어서,

상기 추출기는

DFA(Deterministic Finite Automation)를 이용하여, 상기 제1 보안 정책과 관련된 데이터를 추출하는 정책 번역기.

청구항 18

제 11 항에 있어서,

상기 데이터 변환기는

NSF 데이터베이스를 이용하여, 상기 NSF 데이터베이스의 데이터와 매핑을 통해 상기 제 1 보안 정책과 관련된 데이터를 NSF(Network Security Function)를 위한 필수 데이터로 변환하는 정책 변환기.

청구항 19

제11항에 있어서,

상기 대상 NSF는

상기 제1 보안 정책을 모두 커버(cover)할 수 있는 능력(capability)을 갖는, 정책 번역기.

청구항 20

제11항에 있어서,

상기 정책 번역기는

상기 NSF의 모든 기능들이 등록된, 정책 번역기.

발명의 설명

기 술 분 야

[0001] 본 명세서는 보안 정책 번역에 관한 것으로서, 보다 자세하게는 I2NSF(Interface to Network Security Functions) 에서 보안 정책 번역의 모델을 제안한다.

배 경 기 술

[0002] 네트워크를 전세계에 연결하면 지리적 거리에 관계없이 신속하게 정보에 액세스할 수 있다. 인터넷은 본질적으로 서로 다른 레벨들의 계층 구조가 서로 연결된 수많은 네트워크이다.

[0003] 인터넷은 IETF (Internet Engineering Task Force)에서 공표 한 TCP / IP (전송 제어 프로토콜/인터넷 프로토콜)에 따라 운영되며, TCP/IP는 RFC (Request For Comments) 703 및 IETF에서 발행 한 RFC 791에서 찾을 수 있다.

발명의 내용

해결하려는 과제

[0004] 본 명세서의 목적은, I2NSF(Interface to Network Security Functions)에서 보안 정책 번역의 모델을 제안한다.

[0005] 또한, 본 명세서는 추상 데이터가 포함된 고수준 보안 정책을 NSF(Network Security Funtion)가 이용할 수 있는 저수준 보안 정책으로 번역하는 방법을 제안한다.

[0006] 본 명세서에서 이루고자 하는 기술적 과제들은 이상에서 언급한 기술적 과제들로 제한되지 않으며, 언급하지 않은 또 다른 기술적 과제들은 아래의 기재로부터 본 명세서가 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0007] 본 명세서의 일 양상은, 네트워크 보안 기능 인터페이스를 위한 정책 번역기(Policy Translator)의 보안 정책을 번역하기 위한 방법에 있어서, I2NSF(Interface to Network Security Functions) 사용자로부터 상위 레벨(High-Level)의 제1 보안 정책을 수신하는 단계; 상기 제1 보안 정책에 근거하여, 상기 제1 보안 정책과 관련된 데이터를 추출하는 단계; 상기 제 1 보안 정책과 관련된 데이터를 NSF(Network Security Function)를 위한 필수 데이터로 변환하는 단계; 및 상기 필수 데이터에 근거하여, 대상(Target) NSF를 검색하고, 상기 대상 NSF와 관련된 하위 레벨(Low-Level)의 제2 보안 정책을 생성하는 단계; 를 포함하되, 상기 정책 번역기와 상기 I2NSF 사용자는 사용자-직면 인터페이스로 연결되며, 상기 정책 번역기와 상기 NSF는 NSF-직면 인터페이스로 연결되는 보안 정책을 번역할 수 있다.

[0008] 또한, 상기 제2 보안 정책과 관련된 데이터를 생성하는 단계; 를 더 포함하고, 상기 제2 보안 정책과 관련된 데이터는 상기 대상 NSF를 위한 구조 또는 내용을 포함하며, 태그(Tag)를 통해 그룹화될 수 있다.

[0009] 또한, 상기 제2 보안 정책과 관련된 데이터를 상기 NSF-직면 인터페이스를 통해, 상기 대상 NSF로 전달하는 단계; 를 더 포함할 수 있다.

[0010] 또한, 상기 제2 보안 정책과 관련된 데이터에 근거하여, 상기 대상 NSF를 설정하는 단계; 를 더 포함할 수 있다.

[0011] 또한, 상기 제2 보안 정책은 적용되는 정책 규칙, 및 일반적인 보안 기능을 위한 동작을 나타나내는 기본 동작 정보를 포함할 수 있다.

[0012] 또한, 상기 정책 규칙은 정책 정보 및 규칙 정보를 포함하며, 상기 정책 정보 및 상기 규칙 정보는 시스템의 변경을 나타내는 이벤트 절(Event Clause), 정책 규칙의 적용 조건을 나타내는 조건 절(Condition Clause), 및 상기 이벤트 절 및 상기 조건 절을 만족할 때 수행되는 보안 기능을 나타내는 동작 절(Action Clause)을 포함할 수 있다.

[0013] 또한, 제1 보안 정책과 관련된 데이터를 추출하는 단계 DFA(Deterministic Finite Automation)를 이용할 수 있다.

[0014] 또한, 상기 필수 데이터로 변환하는 단계는 NSF 데이터베이스를 이용하여, 상기 NSF 데이터베이스의 데이터와 매핑을 통해 수행될 수 있다.

[0015] 또한, 상기 대상 NSF는 상기 제1 보안 정책을 모두 커버(cover)할 수 있는 능력(capability)을 갖을 수 있다.

[0016] 또한, 상기 정책 번역기는 상기 NSF의 모든 기능들이 등록될 수 있다.

[0017] 본 발명의 또 다른 일 양상은, 네트워크 보안 기능 인터페이스를 위한 보안 정책을 번역하는 정책 번역기(Policy Translator)에 있어서, I2NSF(Interface to Network Security Functions) 사용자로부터 수신된, 상위 레벨(High-Level)의 제1 보안 정책에 근거하여, 상기 제1 보안 정책과 관련된 데이터를 추출하는 추출기(Extractor); 상기 제 1 보안 정책과 관련된 데이터를 NSF(Network Security Function)를 위한 필수 데이터로 변환하는 데이터 변환기(Data Converter); 및 상기 필수 데이터에 근거하여, 대상(Target) NSF를 검색하고, 상기 대상 NSF와 관련된 하위 레벨(Low-Level)의 제2 보안 정책을 생성하는 생성기(Generator); 를 포함하되, 상기 정책 번역기와 상기 I2NSF 사용자는 사용자-직면 인터페이스로 연결되며, 상기 정책 번역기와 상기 NSF는 NSF-직면 인터페이스로 연결될 수 있다.

발명의 효과

[0018] 본 명세서의 실시 예에 따르면, I2NSF(Interface to Network Security Functions)에서 보안 정책 번역을 위한 모델을 설계할 수 있다.

[0019] 또한, 본 명세서의 실시 예에 따르면, 본 명세서는 고수준 보안 정책을 NSF(Network Security Funtion)가 이용할 수 있는 저수준 보안 정책으로 번역 할 수 있다.

[0020] 본 명세서에서 얻을 수 있는 효과는 이상에서 언급한 효과들로 제한되지 않으며, 언급하지 않은 또 다른 효과들은 아래의 기재로부터 본 명세서가 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

[0021] 본 명세서에 관한 이해를 돕기 위해 상세한 설명의 일부로 포함되는, 첨부 도면은 본 명세서에 대한 실시 예를 제공하고, 상세한 설명과 함께 본 명세서의 기술적 특징을 설명한다.

도 1은 본 명세서의 일 실시 예에 따른 I2NSF(Interface to Network Security Functions) 시스템을 예시한다.

도 2는 본 명세서의 일 실시 예에 따른 I2NSF 시스템의 아키텍처를 예시한다.

도 3은 본 명세서가 적용될 수 있는 전체 I2NSF 정보 모델 디자인의 일 예를 나타낸다.

도 4는 본 명세서가 적용될 수 있는 네트워크 보안 정보 하위 모델 개요의 일 예를 나타낸다.

도 5은 본 명세서가 적용될 수 있는 네트워크 보안 정보 하위 모델의 확장의 일 예를 나타낸다.

도 6는 본 명세서가 적용될 수 있는 네트워크 보안 정보 하위 모델 이벤트 클래스의 확장의 일 예를 나타낸다.

도 7는 본 명세서가 적용될 수 있는 네트워크 보안 정보 하위 모델 컨디션 클래스의 확장의 일 예를 나타낸다.

도 8은 본 명세서가 적용될 수 있는 네트워크 보안 정보 하위 모델 액션의 확장의 일 예를 나타낸다.

도 9은 본 명세서가 적용될 수 있는 I2NSF 보안 기능의 상위 레벨 모델의 일 예를 나타낸다.

도 10은 본 명세서가 적용될 수 있는 네트워크 보안 기능 정보 모델의 일 예를 나타낸다.

도 11는 본 명세서가 적용될 수 있는 공격 완화 기능 정보 모델의 일 예를 나타낸다.

도 12는 본 명세서의 일 실시 예에 따른 네트워크 보안 정책 식별을 위한 데이터 모델 구조를 예시한다.

도 13은 본 명세서의 일 실시 예에 따른 이벤트 규칙을 위한 데이터 모델 구조를 예시한다.

도 14a 내지 도 14d는 본 명세서의 일 실시 예에 따른 컨디션 규칙을 위한 데이터 모델 구조를 예시한다.

도 15는 본 명세서의 일 실시 예에 따른 액션 규칙을 위한 데이터 모델 구조를 예시한다.

도 16a 내지 도 18j는 본 명세서의 일 실시 예에 따른 I2NSF NSF-Facing Interface의 YANG 데이터 모듈을 예시한다.

도 19a 내지 도 19j는 본 명세서의 일 실시 예에 따른 NSF 모니터링을 위한 데이터 모델을 예시한다.

도 20a 내지 도 21i는 본 명세서의 일 실시 예에 따른 모니터링을 위한 YANG 데이터 모델을 예시한다.

도 22는 본 명세서가 작용될 수 있는 소비자-직면 인터페이스를 위한 고수준 추출의 예시이다.

도 22a 내지 도 22d는 본 명세서의 일 실시 예에 따른 소비자-직면 인터페이스를 위한 보안 정책의 데이터 모델의 예시이다.

도 23은 본 명세서의 일 실시예에 따른 소비자-직면 인터페이스의 보안 정책을 위한 YANG 데이터 모델의 예시이다.

도 24 및 도 25는 본 명세서가 적용될 수 있는 보안 정책의 예시이다.

도 26는 본 명세서가 적용될 수 있는 정책 적용을 위한 절차의 예시이다.

도 27는 본 명세서가 적용될 수 있는 보안 정책 번역기 모델의 예시이다.

도 28은 본 명세서가 적용될 수 있는 보안 정책 번역기의 보안 정책 번역을 위한 순서도이다.

도 29은 본 명세서가 적용될 수 있는 추출기 모델의 예시이다.

도 30은 본 명세서에 적용될 수 있는 데이터 변환기 모델의 예시이다.

도 31는 본 명세서가 적용될 수 있는 정책 프로비저닝의 예시이다.

도 32은 본 명세서가 적용될 수 있는 NSF-직면 인터페이스 YANG 데이터 모델에 근거한 트리구조의 예시이다.

발명을 실시하기 위한 구체적인 내용

[0022] 이하, 본 명세서에 따른 바람직한 실시 형태를 첨부된 도면을 참조하여 상세하게 설명한다. 첨부된 도면과 함께 이하에 개시될 상세한 설명은 본 명세서의 예시적인 실시형태를 설명하고자 하는 것이며, 본 명세서가 실시될 수 있는 유일한 실시형태를 나타내고자 하는 것이 아니다. 이하의 상세한 설명은 본 명세서의 완전한 이해를 제공하기 위해서 구체적 세부사항을 포함한다. 그러나, 당업자는 본 명세서가 이러한 구체적 세부사항 없이도 실시될 수 있음을 안다.

[0023] 몇몇 경우, 본 명세서의 개념이 모호해지는 것을 피하기 위하여 공지의 구조 및 장치는 생략되거나, 각 구조 및 장치의 핵심기능을 중심으로 한 블록도 형식으로 도시될 수 있다.

[0024] 이하의 설명에서 사용되는 특정 용어들은 본 명세서의 이해를 돕기 위해서 제공된 것이며, 이러한 특정 용어의 사용은 본 명세서의 기술적 사상을 벗어나지 않는 범위에서 다른 형태로 변경될 수 있다.

[0026] 최근에는, Network Functions Virtualization(NFV)-based Security Function을 위한 기본 표준 인터페이스가 I2NSF(Interface to Network Security Functions) 워킹 그룹에 의해 개발되고 있다. 이는 인터넷 엔지니어링 태스크 포스(IETF: Internet Engineering Task Force)로 불리는 국제 인터넷 표준 기구의 일부이다.

[0027] I2NSF의 목적은 다수의 보안 솔루션 벤더(security solution vendor)들에 의해 제공되는 이종의(heterogeneous) 네트워크 보안 기능(들)(NSF: Network Security Function)을 위한 표준화된 인터페이스를 정의하기 위함이다.

[0028] I2NSF 아키텍처(architecture)에서, NSF(들)의 관리에 대하여 상세히 고려할 필요 없이(NSF의 관리는 결국 보안 정책의 시행(enforce)을 요구한다), 사용자는 사용자의 네트워크 시스템 내 네트워크 자원을 보호하기 위한 보호 정책을 정의할 수 있다. 또한, 다수의 벤더(Vendors)들로부터 NSF(들)로의 표준화된 인터페이스는 이종의 NSF(들)에 대한 태스크(task)의 설정 및 관리를 단순화할 수 있다.

[0029] 도 1은 본 명세서의 일 실시 예에 따른 I2NSF(Interface to Network Security Functions) 시스템을 예시한다.

[0030] 도 1을 참조하면, I2NSF 시스템은 I2NSF 사용자(I2NSF User), 네트워크 운영 관리 시스템(Network Operator Management System), 개발자 관리 시스템(DMS : Developer's Management System) 및/또는 적어도 하나의 NSF(Network Security Function)을 포함한다.

[0031] I2NSF 사용자는 I2NSF 소비자-직면 인터페이스(I2NSF Consumer-Facing Interface)를 통해 네트워크 운영 관리 시스템과 통신한다. 네트워크 운영 관리 시스템은 I2NSF NSF-직면 인터페이스(I2NSF NSF-Facing Interface)를 통해 NFC(들)과 통신한다. 개발자 관리 시스템은 I2NSF 등록 인터페이스(I2NSF Registration Interface)를 통해 네트워크 운영 관리 시스템과 통신한다. 이하에서는 I2NSF 시스템의 각 컴포넌트(I2NSF 컴포넌트) 및 각 인터페이스(I2NSF 인터페이스)에 설명한다.

[0033] **I2NSF 사용자**

[0034] I2NSF 사용자는 다른 I2NSF 컴포넌트(예컨대, 네트워크 운영 관리 시스템)에서 정보를 요청하거나 및/또는 다른 I2NSF 컴포넌트(예컨대, 개발자 관리 시스템)에 의해 제공되는 서비스(예컨대, 네트워크 보안 서비스)를 사용하는 I2NSF 컴포넌트이다. 예를 들면, I2NSF 사용자는 오버레이 네트워크 관리 시스템, 기업 네트워크 관리자 시스템, 다른 네트워크 도메인 관리자 등일 수 있다.

[0035] 이러한 I2NSF 사용자 컴포넌트에 할당된 역할을 수행하는 대상은 I2NSF 소비자로 지칭될 수 있다. I2NSF 소비자의 예로는, 일정 기간(time span) 동안 패킷의 특정 필드에 기초하여 흐름을 허용, 속도-제한(rate-limit), 또는 거부하기 위해 언더레이 네트워크(Underlay Network)에 동적으로 알릴 필요가 있는 화상 회의 네트워크 관리

자(Video-conference network manager), 특정 흐름에 대한 특정 I2NSF 정책을 시행(enforce)하기 위해 제공자 네트워크를 요청할 필요가 있는 기업 네트워크 관리자(Enterprise network administrators) 및 관리 시스템(Management Systems), 특정 조건의 세트와 일치하는 흐름을 차단하기 위해 언더레이 네트워크에 요청을 전송하는 IoT 관리 시스템(IoT management system)가 포함될 수 있다.

[0036] I2NSF 사용자는 고수준(High-level) 보안 정책(Security Policy)을 생성 및 배포할 수 있다. 구체적으로 설명하면, I2NSF 사용자는 다양한 악의적인(malicious) 공격으로부터 네트워크 트래픽(network traffic)을 보호하기 위하여 네트워크 보안 서비스(network security service)를 이용할 필요가 있다. 이 보안 서비스를 요청하기 위하여, I2NSF 사용자는 자신이 원하는 보안 서비스에 대한 고수준 보안 정책을 생성하고 네트워크 운영 관리 시스템에게 이를 알릴 수 있다.

[0037] 한편, 고수준 보안 정책을 준비하는 과정에서, I2NSF 사용자는 각 NSF(들)를 위한 보안 서비스 또는 보안 정책 규칙 구성(security policy rule configuration)을 실현하기 위하여 요구되는 NSF(들)의 타입에 대하여 고려하지 않을 수 있다.

[0038] 또한, I2NSF 사용자는 네트워크 운영 관리 시스템에 의해 기본적인(underlying) NSF(들) 내에서 발생되는 보안 이벤트(들)(security event)를 통지받을 수 있다. 이들의 보안 이벤트(들)을 분석함으로써, I2NSF 사용자는 새로운 공격을 식별하고, 새로운 공격에 대처하기 위한 고수준 보안 정책을 업데이트(또는 생성)할 수 있다. 이와 같이, I2NSF 사용자는 보안 정책을 정의, 관리 및 모니터링할 수 있다.

[0040] **네트워크 운영 관리 시스템**

[0041] 네트워크 운영 관리 시스템은 보안 제공, 모니터링 및 기타 동작을 위한 수집(collection) 및 배포(distribution) 지점(point)의 역할을 수행하는 컴포넌트이다. 예를 들면, 네트워크 운영 관리 시스템은 보안 제어기(Security Controller)일 수 있다. 이러한 네트워크 운영 관리 시스템은 네트워크 보안 관리자에 의해 관리될 수 있고, I2NSF 관리 시스템으로 지칭될 수도 있다.

[0042] 네트워크 운영 관리 시스템(또는 보안 제어기)의 주요한 역할 중 하나는 I2NSF 사용자로부터의 고수준 보안 정책(또는 정책 규칙)을 특정 NSF(들)을 위한 저수준(Low-level) 보안 정책 규칙으로 번역(translate)하는 것이다. 네트워크 운영 관리 시스템(또는 보안 제어기)은 고수준 보안 정책을 I2NSF 사용자로부터 수신한 후, 우선 I2NSF 사용자에 의해 요구되는 정책을 시행하기 위하여 요구되는 NSF(들)의 타입을 결정할 수 있다. 그리고, 네트워크 운영 관리 시스템(또는 보안 제어기)은 요구되는 각 NSF(들)을 위한 저수준(Low-level) 보안 정책을 생성할 수 있다. 결국, 네트워크 운영 관리 시스템(또는 보안 제어기)은 생성된 저수준 보안 정책을 각 NSF(들)에게 설정할 수 있다.

[0043] 또한, 네트워크 운영 관리 시스템(또는 보안 제어기)은 시스템 내 구동 중인 NSF(들)을 모니터링하고, 각 NSF(들)에 대한 다양한 정보(예를 들어, 네트워크 액세스(access) 정보 및 작업로드(workload) 상태 등)를 유지할 수 있다. 또한, 네트워크 운영 관리 시스템(또는 보안 제어기)은 개발자 관리 시스템의 도움을 받아 NSF 인스턴스의 동적인 수명시간(life-cycle) 관리를 통해 NSF 인스턴스(instance)의 풀(pool)을 동적으로 관리할 수 있다.

[0045] **NSF**

[0046] NSF는 보안 관련 서비스를 제공하는 논리적 엔티티(logical entity) 또는 소프트웨어 컴포넌트이다. 예를 들면, NFC는 저수준 보안정책을 수신하고, 이에 기초하여 악의적인 네트워크 트래픽을 감지하고, 이를 차단하거나 완화할 수 있다. 이를 통해, 네트워크 통신 스트림의 무결성(integrity) 및 기밀성(confidentiality)이 보장될 수 있다.

[0048] **개발자 관리 시스템**

[0049] 개발자 관리 시스템은 다른 I2NSF 컴포넌트(예컨대, I2NSF 사용자, 네트워크 운영 관리 시스템)으로 정보를 보내거나, 및/또는 서비스(예컨대, 네트워크 보안 서비스)를 제공하는 I2NSF 컴포넌트이다. 개발자 관리 시스템은 벤더 관리 시스템(Vendor's Management System)으로 지칭될 수도 있다. 이러한 개발자 관리 시스템에 할당된 역

할을 수행하는 대상은 I2NSF 생산자(producer)로 지칭될 수 있다.

[0050] 개발자 관리 시스템은 네트워크 운영 관리 시스템에게 NSF(들)을 제공하는 제3자(third-party) 보안 벤더에 의해 관리될 수 있다. 다양한 보안 벤더의 다수의 개발자 관리 시스템(들)이 존재할 수 있다.

[0052] I2NSF 소비자-직면 인터페이스(간단히, 소비자-직면 인터페이스(CFI))

[0053] CFI는 I2NSF 사용자와 네트워크 운영 관리 시스템 사이에 위치하는, 사용자의 I2NSF 시스템으로의 인터페이스이다. 이렇게 설계됨으로써, 하위(underlying) NSF(들)의 상세한 내용을 숨기고, 사용자에게 NSF(들)의 추상적인 시각(abstract view)만을 제공한다.

[0054] 이 CFI는 주어진 I2NSF 시스템의 상이한 사용자가 관리 도메인 내의 특정 흐름(flow)에 대한 보안 정책을 정의, 관리 및 모니터링할 수 있게 하기 위해 사용될 수 있다. I2NSF 사용자에 의해 생성된 고수준 보안 정책(또는 정책 규칙)은 이 CFI를 통해 네트워크 운영 관리 시스템으로 전달될 수 있다.

[0056] I2NSF NSF-직면 인터페이스(간단히, NSF-직면 인터페이스(NFI))

[0057] NFI는 네트워크 운영 관리 시스템(또는 보안 제어기)과 NSF(들) 사이에 위치하는 인터페이스이다.

[0059] NFI는 하나 이상의 NSF에 의해 시행되는 흐름-기반(flow-based) 보안 정책을 지정하고 모니터링하기 위해 사용될 수 있다. 예를 들면, I2NSF 시스템은 흐름-기반 NSF를 사용할 수 있다. 여기서, 흐름-기반 NSF는 보안 특성을 강화하기 위해 정책의 세트에 따라 네트워크 흐름을 검사하는 NSF이다. 이러한 흐름-기반 NSF에 의한 흐름-기반 보안은 수신된 순서대로 패킷들이 검사되고, 검사 프로세스에 따라 패킷에 대한 수정이 없는 것을 의미한다. 흐름-기반 NSF에 대한 인터페이스는 다음과 같이 분류될 수 있다:

[0060] - NSF 운영 및 관리 인터페이스(NSF Operational and Administrative Interface): NSF의 운영 상태를 프로그래밍하기 위해 I2NSF 관리 시스템에 의해 사용되는 인터페이스 그룹; 이 인터페이스 그룹은 또한 관리 제어 기능을 포함한다. I2NSF 정책 규칙은 일관된 방식으로 이 인터페이스 그룹을 변경하는 한가지 방법을 나타낸다. 어플리케이션 및 I2NSF 컴포넌트가 그들이 송신 및 수신하는 트래픽의 동작을 동적으로 제어할 필요가 있기 때문에, I2NSF 노력(effort)의 대부분이 이 인터페이스 그룹에 집중된다.

[0061] - 모니터링 인터페이스(Monitoring Interface): 하나 이상의 선택된 NSF로부터의 모니터링 정보를 획득하기 위해 I2NSF 관리 시스템에 의해 사용되는 인터페이스 그룹; 이 인터페이스 그룹의 각 인터페이스는 쿼리 또는 리포트 기반 인터페이스일 수 있다. 둘 사이의 차이점은 쿼리 기반 인터페이스는 정보를 획득하기 위해 I2NSF 관리 시스템에 의해 사용되고, 이에 반하여 리포트 기반 인터페이스는 정보를 제공하기 위해 NSF에 의해 사용된다는 것이다. 이 인터페이스 그룹의 기능은 또한 SYSLOG 및 DOTS와 같은 다른 프로토콜에 의해 정의될 수 있다. I2NSF 관리 시스템은 정보의 수신에 기초하여 하나 이상의 동작(action)을 취할 수 있다. 이는 I2NSF 정책 규칙에 의해 지정되어야 한다. 이 인터페이스 그룹은 NSF의 운영 상태를 변경하지 않는다.

[0062] 이와 같이, NFI는 흐름-기반 패러다임을 사용하여 개발될 수 있다. 흐름-기반 NSF의 공동 특성(common trait)은 수신된 패킷의 콘텐츠(예컨대, 헤더/페이로드) 및/또는 컨텍스트(예컨대, 세션 상태 및 인증 상태)에 기초하여 패킷을 처리하는 것이다. 이 특징은 I2NSF 시스템의 동작을 정의하기 위한 요구사항(requirement) 중 하나이다.

[0063] 한편, I2NSF 관리 시스템은 주어진 NSF의 모든 기능들을 사용할 필요가 없으며, 모든 사용 가능한 NSF들을 사용할 필요도 없다. 따라서, 이 추상화(abstraction)는 NSF 특징(feature)을 NSF 시스템에 의해 빌딩 블록(building block)으로 취급될 수 있게 해준다. 그러므로, 개발자는 벤더 및 기술에 독립적인 NSF에 의해 정의되는 보안 기능을 자유롭게 사용할 수 있게 된다.

[0065] **I2NSF 등록 인터페이스(간단히, 등록 인터페이스(RI))**

[0066] RI는 네트워크 운영 관리 시스템 및 개발자 관리 시스템 사이에 위치하는 인터페이스이다. 상이한 벤더에 의해 제공되는 NSF는 상이한 기능(capability)을 가질 수 있다. 따라서, 상이한 벤더에 의해 제공되는 여러 유형의 보안 기능을 이용하는 프로세스를 자동화하기 위해, 벤더가 그들의 NSF의 기능을 정의하기 위한 전용 인터페이

스를 가질 필요가 있다. 이러한 전용 인터페이스는 I2NSF 등록 인터페이스(RI)로 지칭될 수 있다.

[0067] NSF의 기능은 미리 구성되거나 또는 I2NSF 등록 인터페이스를 통해 동적으로 검색될 수 있다. 만일 소비자에게 노출되는 새로운 기능이 NSF에 추가된다면, 관심 있는(interested) 관리 및 제어 엔티티가 그것들을 알 수 있도록, 그 새로운 기능의 capability가 I2NSF 등록 인터페이스를 통해 I2NSF 레지스트리(registry)에 등록될 필요가 있다.

[0069] 도 2는 본 명세서의 일 실시예에 따른 I2NSF 시스템의 아키텍처를 예시한다. 도 2의 I2NSF 시스템은 도 1의 I2NSF 시스템에 비하여 I2NSF 사용자 및 네트워크 운영 관리 시스템의 구성을 더 구체적으로 나타낸다. 도 2에서는 도 1에서 상술한 설명과 중복된 설명은 생략한다.

[0070] 도 2를 참조하면, I2NSF 시스템은 I2NSF 사용자, 보안 관리 시스템(Security Management System), 및 NSF 인스턴스(instances) 계층을 포함한다. I2NSF 사용자 계층은 어플리케이션 로직(Application Logic), 정책 업데이터(Policy Updater), 및 이벤트 수집기(Event Collector)을 컴포넌트로서 포함한다. 보안 관리 시스템 계층은 보안 제어기 및 개발자 관리 시스템을 포함한다. 보안 관리 시스템 계층의 보안 제어기는 보안 정책 관리자(Security policy manager) 및 NSF 기능 관리자(NSF capability manager)를 컴포넌트로서 포함한다.

[0071] I2NSF 사용자 계층은 소비자-직면 인터페이스를 통해 보안 관리 시스템 계층과 통신한다. 예를 들면, I2NSF 사용자 계층의 정책 업데이터 및 이벤트 수집기는 소비자-직면 인터페이스를 통해 보안 관리 시스템 계층의 보안 제어기와 통신한다. 또한, 보안 관리 시스템 계층은 NSF-직면 인터페이스를 통해 NSF 인스턴스 계층과 통신한다. 예를 들면, 보안 관리 시스템 계층의 보안 제어기는 NSF-직면 인터페이스를 통해 NSF 인스턴스 계층의 NSF 인스턴스(들)과 통신한다. 또한, 보안 관리 시스템 계층의 개발자 관리 시스템은 등록 인터페이스를 통해 보안 관리 시스템 계층의 보안 제어기와 통신한다.

[0072] 도 2의 I2NSF 사용자 계층, 보안 관리 시스템 계층의 보안 제어기 컴포넌트, 보안 관리 시스템 계층의 개발자 관리 시스템 컴포넌트 및 NSF 인스턴스 계층은 각각 도 1의 I2NSF 사용자 컴포넌트, 네트워크 운영 관리 시스템 컴포넌트, 개발자 관리 시스템 컴포넌트 및 NSF 컴포넌트에 대응된다. 또한, 도 2의 소비자-직면 인터페이스, NSF-직면 인터페이스 및 등록 인터페이스는 도 1의 소비자-직면 인터페이스, NSF-직면 인터페이스 및 등록 인터페이스에 대응된다. 이하에서는, 각 계층에 포함된 새로 정의된 컴포넌트들에 대하여 설명한다.

[0074] **I2NSF 사용자**

[0075] 상술한 것처럼, I2NSF 사용자 계층은 다음 3 개의 컴포넌트를 포함한다: 어플리케이션 로직(Application Logic), 정책 업데이터(Policy Updater), 및 이벤트 수집기(Event Collector). 각각의 역할 및 동작을 설명하면 다음과 같다.

[0076] 어플리케이션 로직은 고수준 보안 정책을 생성하는 컴포넌트이다. 이를 위해, 어플리케이션 로직은 이벤트 수집기로부터 고수준 정책을 업데이트(또는 생성)하기 위한 이벤트를 수신하고, 수집된 이벤트에 기초하여 고수준 정책을 업데이트(또는 생성)한다. 그 이후에, 고수준 정책은 보안 제어기로 배포하기 위해 정책 업데이터로 보내진다. 고수준 정책을 업데이트(또는 생성)하기 위해, 이벤트 수집기는 보안 수집기에 의해 보내진 이벤트를 수신하고, 그들을 어플리케이션 로직으로 보낸다. 이 피드백에 기초하여, 어플리케이션 로직은 고수준 보안 정책을 업데이트(또는 생성)할 수 있다.

[0077] 도 2에서는, 어플리케이션 로직, 정책 업데이터 및 이벤트 수집기를 각각 별도의 구성으로 도시하고 있으나, 본 명세서의 이에 한정되지 않는다. 다시 말해, 각각은 논리적인 컴포넌트로서, I2NSF 시스템에서 하나 또는 2 개의 컴포넌트로 구현될 수도 있다.

[0079] **보안 관리 시스템**

[0080] 상술한 것처럼, 보안 관리 시스템 계층의 보안 제어기는 보안 정책 관리자(Security policy manager) 및 NSF 기능 관리자(NSF capability manager)와 같은 2개의 컴포넌트를 포함한다

[0081] 보안 정책 관리자는 CFI를 통해 정책 업데이터로부터 고수준 정책을 수신하고, 이 정책을 여러 저수준 정책으로

맵핑할 수 있다. 이 저수준 정책은 NSF 기능 관리자에 등록된 주어진 NSF 기능과 관련된다. 또한, 보안 정책 관리자는 이 정책을 NFI를 통해 NSF(들)로 전달할 수 있다.

[0082] NSF 기능 관리자는 주어진 NSF 기능과 관련된 저수준 정책을 생성하기 위해, 개발자 관리 시스템에 의해 등록된 NSF의 기능을 지정하고, 그것을 보안 정책 관리자와 공유할 수 있다. 새로운 NSF가 등록될 때마다, NSF 기능 관리자는 등록 인터페이스를 통해 NSF 기능 관리자의 관리 테이블에 NSF의 기능을 등록하도록 개발자 관리 시스템에 요청할 수 있다. 개발자 관리 시스템은 새로운 NSF의 기능을 NSF 기능 관리자로 등록하기 위한 보안 관리 시스템의 다른 부분에 해당한다.

[0083] 도 2에서는, 보안 정책 관리자 및 NSF 기능 관리자를 각각 별도의 구성으로 도시하고 있으나, 본 명세서의 이에 한정되지 않는다. 다시 말해, 각각은 논리적인 컴포넌트로서, I2NSF 시스템에서 하나의 컴포넌트로 구현될 수도 있다.

[0085] **NSF 인스턴스(NSF Instances)**

[0086] 도 2에 도시된 것처럼, NSF 인스턴스 계층은 NSF들을 포함한다. 이때, 모든 NSF들은 이 NSF 인스턴스 계층에 위치된다. 한편, 고수준 정책을 저수준 정책에 맵핑한 후에, 보안 정책 관리자는 NFI를 통해 정책을 NSF(들)로 전달한다. 이 경우, NSF는 수신된 저수준 보안 정책에 기초하여 악의적인 네트워크 트래픽을 감지하고, 이를 차단하거나 완화할 수 있다.

[0087] 가상화 시스템의 신속한 개발을 위해서는 다양한 시나리오에서 고급 보안 기능이 필요하다(예를 들면, 엔터프라이즈 네트워크의 네트워크 장치, 모바일 네트워크의 사용자 장비, 인터넷의 장치 또는 거주자 액세스 사용자 등).

[0088] 여러 보안 업체에서 생산한 NSF는 고객에게 다양한 보안 기능을 제공할 수 있다. 즉, NSF는 물리적 또는 가상 기능으로 구현되었는지 여부와 관계없이 여러 NSF가 함께 결합되어 주어진 네트워크 트래픽에 대한 보안 서비스를 제공할 수 있다.

[0089] 보안 기능은 보안 정책 시행 목적으로 사용할 수 있는 일련의 네트워크의 보안과 관련된 기능을 말한다. 보안 기능은 실제 구현되는 보안 제어 메커니즘과는 독립적이며, 모든 NSF는 NSF에서 제공할 수 있는 기능들의 세트가 등록되어 있다.

[0090] 보안 기능(security capability)은 특정 NSF가 제공하는 보안 기능을 모호하지 않게 설명함으로써 맞춤형 보안 보호를 정의할 수 있는 기능 명세(capability specification)를 제공한다. 또한, 보안 기능을 통해 보안 기능의 공급 업체의 중립적인 방식으로 설명할 수 있다.

[0091] 즉, 네트워크를 설계할 때 특정 제품을 언급할 필요가 없으며, 기능별로 특징이 고려될 수 있다.

[0092] 앞에서 살펴본 바와 같이 보안 정책 제공에 사용될 수 있는 I2NSF 인터페이스는 아래와 같이 두 가지 유형이 존재할 수 있다.

[0094] I2NSF 사용자와 응용 프로그램 간의 인터페이스 및 보안 컨트롤러 (Consumer-Facing Interface): NSF 데이터 및 서비스 사용자와 네트워크 운영 관리시스템(또는 보안 제어기) 사이에 통신 채널을 제공하는 소비자 지향 인터페이스.

[0095] I2NSF Consumer-Facing Interface는 보안 정보가 다양한 애플리케이션(예를 들면: OpenStack 또는 다양한 BSS / OSS 구성 요소)과 보안 컨트롤러 간의 교환에 사용될 수 있다. Consumer-Facing Interface의 설계 목표는 보안 서비스의 스펙을 구현과 분리하는데 있다.

[0096] - NSF 간의 인터페이스(예를 들면: 방화벽, 침입 방지 또는 안티 바이러스) 및 보안 컨트롤러 (NSF-Facing Interface): NSF-Facing Interface는 보안 관리 체계를 NSF 집합과 여러 가지 구현에서 분리하는 데 사용되며 NSF가 구현되는 방식(예를 들면: 가상 머신 또는 실제 appliances 등)에서 독립적이다.

[0097] 이하, 연관된 I2NSF 정책 객체와 함께 네트워크 보안, 콘텐츠 보안 및 공격 완화 기능에 대한 객체 지향 정보 모델에 대해 살펴보도록 한다.

[0098] 본 명세서에서 정보 모델에 사용되는 용어는 다음과 같이 정의될 수 있다

[0099] AAA: Access control, Authorization, Authentication

[0100] ACL: Access Control List

[0101] (D)DoD: (Distributed) Denial of Service (attack)

[0102] ECA: Event-Condition-Action

[0103] FMR: First Matching Rule (resolution strategy)

[0104] FW: Firewall

[0105] GNSF: Generic Network Security Function

[0106] HTTP: HyperText Transfer Protocol

[0107] I2NSF: Interface to Network Security Functions

[0108] IPS: Intrusion Prevention System

[0109] LMR: Last Matching Rule (resolution strategy)

[0110] MIME: Multipurpose Internet Mail Extensions

[0111] NAT: Network Address Translation

[0112] NSF: Network Security Function

[0113] RPC: Remote Procedure Call

[0114] SMA: String Matching Algorithm

[0115] URL: Uniform Resource Locator

[0116] VPN: Virtual Private Network

[0118] **정보 모델 설계**

[0119] 기능 정보 모델(Capability Information Model)의 설계의 출발점은 보안 기능의 유형을 분류하는 것이다. 예를 들어, "IPS", "안티 바이러스" 및 "VPN 집중 장치"와 같은 보안 기능의 유형을 분류하는 것이다.

[0120] 또는, "패킷 필터"는 다양한 조건(예를 들면: 발신 및 수신 IP 주소, 발신 및 수신 포트 및 IP 프로토콜 유형 필드 등)에 따라 패킷 전달을 허용하거나 거부 할 수 있는 저장 장치로 분류될 수 있다.

[0122] 그러나, 상태 기반 방화벽이나 응용 프로그램 계층 필터와 같은 다른 장치의 경우 더 많은 정보가 필요하다. 이러한 장치는 패킷이나 통신을 필터링하지만 패킷과 통신들을 카테고리화하고 유지하는 상태에서 차이가 있다.

[0123] 아날로그적 고려사항은 채널 보호 프로토콜들에서 고려될 수 있다. 여기서 채널 보호 프로토콜들은 비대칭 암호로 협상될 수 있는 대칭 알고리즘을 통해 패킷을 보호할 수 있으며, 서로 다른 계층에서 작동하고 서로 다른 알고리즘과 프로토콜을 지원할 수 있다.

[0124] 안전한 보호를 위해 이러한 프로토콜은 무결성, 선택적으로 기밀성, anti-reply 보호 및 피어 인증이 적용되어야 한다.

[0126] **기능 정보 모델 오버뷰(Capability Information Model Overview)**

[0127] 기능 정보 모델은 NSF의 자동 관리를 위한 토대를 제공하는 보안 기능 모델을 정의한다. 기능 정보 모델은 보안 컨트롤러가 NSF를 적절하게 식별 및 관리할 수 있도록 하고, NSF가 기능들을 올바른 방법으로 사용할 수 있도록 적절하게 선언하는 것을 허용하는 것도 포함한다.

[0129] 보안을 위한 몇 가지 기본 설계 원칙 및 이를 관리해야 하는 시스템은 다음과 같다.

[0131] - 독립성(Independence): 각 보안 기능은 다른 기능에 최소한의 중첩 또는 종속성을 갖는 독립적인 기능이어야 한다. 이를 통해 각 보안 기능을 자유롭게 사용 및 조합할 수 있다. 더 중요한 것은, 하나의 기능으로의 변경이 다른 기능에 영향을 미치지 않는다는 것이다.

[0132] 이것은 Single Responsibility Principle [Martin] [OODSRP]을 따른다.

[0133] - 추상성(Abstraction): 각 기능은 벤더 독립적인 방식으로 정의되어야 하며 잘 알려진 인터페이스와 연결되어 처리 결과를 기술하고 보고할 수 있는 표준화된 기능을 제공해야 한다. 따라서, 다중 공급 벤더와의 상호 운용성이 향상될 수 있다.

[0134] - 자동화(Automation): 시스템은 보안 기능(즉, 사용자 개입없이)을 자동 검색, 자동 협상 및 자동 업데이트 할 수 있어야 한다. 이러한 자동화 기능은 다수의 NSF를 관리하는 데 특히 유용하다.

[0135] 채택된 보안 체계에 대한 스마트 서비스(예를 들면: 분석, 정제, 기능 추론 및 최적화)를 추가하는 것은 필수적이다. 이러한 기능은 Observer Pattern [OODOP], Mediator Pattern [OODMP] 및 Message Exchange Patterns [Hohpe]와 같은 많은 디자인 패턴에서 지원된다.

[0136] - 확장성: 관리 시스템에는 scale up/down 또는 scale in/out 기능이 있어야 한다. 따라서, 이러한 확장성으로 인하여 변경 가능한 네트워크 트래픽 또는 서비스 요청에서 파생된 다양한 성능 요구 사항을 충족할 수 있다. 또한, 확장성의 영향을 받는 보안 기능은 보안 컨트롤러에 통계보고를 지원해야 스케일링을 호출해야 하는지 여부를 결정하는 데 도움이 될 수 있다.

[0138] 위의 원칙에 따라 표준 인터페이스를 갖춘 추상 및 벤더 중립 기능 집합이 정의될 수 있다. 이것은 주어진 시간에 필요한 NSF 세트를 사용할 수 있게 해주는 Capability 모델과 사용된 NSF 세트에 의해 제공되는 보안의 모호하지 않도록 정의를 제공한다.

[0139] 보안 컨트롤러는 사용자 및 응용 프로그램의 요구 사항을 현재 사용할 수 있는 기능 집합과 비교하여 해당 요구 사항을 충족하는데 필요한 NSF를 선택한다.

[0140] 또한, NSF에 의해 알려지지 않은 위협(예를 들어, zero-day exploits 및 unknown malware)이 보고될 때, 새로운 기능이 생성될 수 있고 및/또는 기존의 기능이 업데이트 될 수 있다(예를 들어, 그의 서명 및 알고리즘을 업데이트함으로써). 그 결과 새로운 위협에 대처하기 위해 기존의 NSF를 강화(및/또는 새로운 NSF를 생성)하게 된다.

[0141] 새로운 기능은 중앙 리포지토리(Repository)에 전송되어 저장되거나 벤더의 로컬 리포지토리에 개별적으로 저장될 수 있다. 두 경우 모두 표준 인터페이스가 업데이트 프로세스가 용이하게 수행되도록 한다.

[0142] ECA 정책 모델 오버뷰(ECA Policy Model Overview)

[0143] "Event-Condition-Action"(ECA) 정책 모델은 I2NSF 정책 규칙의 설계를 위한 기초로 사용된다. 이때, I2NSF 정책과 관련된 용어는 아래와 같이 정의될 수 있다([I-D.draft-ietf-i2nsf-terminology] 참조):

[0145] - 이벤트: 이벤트는 관리되는 시스템이 변경될 때 및/또는 관리되는 시스템의 환경에서 중요한 시점에 발생한다. 이벤트는 I2NSF 정책 규칙의 컨텍스트에서 사용될 때 I2NSF 정책 규칙의 조건 절을 평가할 수 있는지 여부를 결정하는 데 사용될 수 있다. I2NSF 이벤트의 예로는 시간 및 사용자 동작(예를 들면: 로그온, 로그 오프 및 ACL을 위반하는 동작)이 있을 수 있다.

[0146] - 조건(Condition): 조건은 알려진 속성, 특징 및/또는 값의 세트와 비교될 속성, 기능 및/또는 값의 집합으로 정의되어 그(명령형) I2NSF 정책 규칙을 실행하거나 실행하지 않을 수 있다. I2NSF 조건의 예에는 패킷 또는 흐름의 일치하는 속성과 NSF의 내부 상태를 원하는 상태와 비교하는 것이 포함될 수 있다.

[0147] - 동작(Action): 동작은 이벤트 및 조건 절이 충족될 때 흐름 기반 NSF의 측면을 제어하고 모니터링하는데 사용

된다. NSF는 다양한 액션을 실행하여 보안 기능을 제공한다. I2NSF 작업의 예에는 침입 탐지 및 / 또는 보호, 웹 및 플로우 필터링, 패킷 및 플로우에 대한 심층 패킷 검사 제공이 포함될 수 있다.

[0149] I2NSF 정책 규칙은 Event 절, Condition 절 및 Action 절의 세 가지 Boolean 절로 구성된다.

[0150] Boolean 절은 TRUE 또는 FALSE로 평가되는 논리문을 의미하며, 하나 이상의 용어로 구성될 수 있습니다.

[0151] 두 개 이상의 용어가 있는 경우 Boolean 절은 논리 연결 요소(즉, AND, OR 및 NOT)를 사용하여 용어를 연결한다. 이때, 논리적 연결 요소는 아래의 표 1과 같은 의미를 가질 수 있다.

표 1

[0153]

```
IF <event-clause> is TRUE
        IF <condition-clause> is TRUE
            THEN execute <action-clause>
        END-IF
    END-IF
```

[0154] 기술적으로 "정책 규칙"은 실제로 메타 데이터뿐 아니라 앞에서 설명한 “이벤트”, “동작” 및 “조건”을 집계하는 컨테이너 역할을 수행할 수 있다.

[0155] 앞에서 설명한 ECA 정책 모델은 매우 일반적이며 쉽게 확장 할 수 있으며 일반 보안 기능 구현을 제한 할 수 있는 잠재적 제약을 피할 수 있다.

[0157] **외부 정보 모델과의 관계**

[0158] 도 3은 본 명세서가 적용될 수 있는 전체 I2NSF 정보 모델 디자인의 일 예를 나타낸다.

[0159] I2NSF NSF-Facing Interface는 NSF의 기능을 사용하여 NSF를 선택 및 관리하며, 이는 아래와 같은 접근법을 이용하여 수행된다.

[0160] 1) 각 NSF는 "참여"할 때 관리 시스템에 기능을 등록하므로 관리 시스템에서 해당 기능을 사용할 수 있다.

[0161] 2) 보안 컨트롤러는 관리하는 모든 사용 가능한 NSF에서 보안 서비스의 요구 사항을 충족시키는 데 필요한 기능 집합을 선택한다.

[0162] 3) 보안 컨트롤러는 Capability 정보 모델을 사용하여 선택한 기능을 공급 업체와 독립적인 NSF로 일치시킨다.

[0163] 4) 보안 컨트롤러는 위의 정보를 가져 와서 기능 정보 모델의 하나 이상의 데이터 모델을 생성 또는 사용하여 NSF를 관리합니다.

[0164] 5) 제어 및 모니터링을 시작할 수 있습니다.

[0165] 이러한 접근법은 외부 정보 모델이 ECA 정책 규칙 및 그 구성 요소(예를 들면: 이벤트, 조건 및 조치 객체 등)의 개념을 정의하는 데 사용된다고 가정할 수 있다. 이를 통해 외부 정보 모델로부터 I2NSF 정책 규칙을 하위 클래스로 분류 할 수 있다(I-D.draft-ietf-i2nsf-terminology 참조).

[0166] 본 명세서에서 데이터 모델은 데이터의 저장소, 데이터 정의 언어, 쿼리 언어, 구현 언어 및 프로토콜에 의존하는 형식으로 환경에 대한 관심의 컨셉을 나타낸 것이다.

[0168] 또한, 정보 모델은 데이터 저장소, 데이터 정의 언어, 쿼리 언어, 구현 언어 및 프로토콜과 독립적인 형태로 환경에 대한 관심 컨셉을 나타낸 것이다.

[0169] 기능은 클래스(예를 들면: 공통된 특성 및 행동 집합을 나타내는 객체의 집합)로 정의될 수 있다(I-D.draft-ietf-supa-generic-policy-info-model 참조).

[0170] 각 기능은 시스템의 다른 모든 객체와 구별되는 하나 이상의 모델 요소(예를 들면: 속성, 메소드 또는 관계)로 구성될 수 있다. 기능은 일반적으로 일종의 메타 데이터(즉, 객체의 행동을 설명 및 / 또는 처방하는 정보)이다.

[0171] 따라서, 각 기능은 외부 정보 모델이 메타 데이터를 정의하는데 사용될 수 있다(클래스 계층 구조의 형태가 바람직함). 따라서, 기능들은 외부 메타 데이터 모델에서 하위 클래스로 분류될 수 있다.

[0172] 기능 하위 모델은 NSF가 포함된 장치의 유형 및 공급 업체와 독립적인 특정 보안 기능 세트를 광고, 생성, 선택 및 관리하는데 사용된다.

[0173] 즉, NSF-Facing Interface의 사용자는 NFC가 가상화 되거나 호스팅되는지, NSF 공급 업체가 누구인지, NSF가 통신하는 엔티티 세트(예를 들면, 방화벽 또는 IPS)를 고려하지 않는다.

[0174] 대신 사용자는 NSF가 가지고 있는 패킷 필터링이나 딥 패킷 검사와 같은 기능 세트만을 고려한다.

[0175] 이러한 전체 ISNSF 정보 모델의 설계는 도 3과 같다.

[0176] 도 3에 도시된 외부 모델은 모두 SUPA 정보 모델을 기반으로 할 수 있다(I-D.draft-ietf-supa-generic-policy-info-model 참조). 기능 하위 모델의 클래스는 외부 메타 데이터 정보 모델에서 메타 데이터 집계(AggregatesMetadata)의 집합을 이어받는다.

[0177] 도 3에 도시된 외부 ECA 정보 모델은 일반 ECA 정책 규칙을 나타내는 최소한의 클래스 집합과 일반 ECA 정책 규칙에 의해 집계 될 수 있는 이벤트, 조건 및 동작을 나타내는 클래스 집합을 제공한다.

[0178] 이를 통해, I2NSF는 이러한 일반 모델을 다른 목적으로 재사용 할 수 있을 뿐만 아니라 I2NSF 관련 개념을 표현하기 위해 새로운 하위 클래스를 생성하거나 속성 및 관계를 추가 할 수 있다.

[0179] 본 명세서에서 외부 ECA 정보 모델은 메타 데이터를 수집하는 기능을 가지고 있다고 가정한다. 기능들은 외부 메타 데이터 정보 모델의 적절한 클래스에서 하위 클래스로 분류될 수 있다.

[0180] 이는 ECA 개체가 메타 데이터와 기존의 집계를 사용하여 메타 데이터를 적절한 ECA 개체에 추가할 수 있게 한다.

[0181] 이하 정보 모델의 각 부분에 대해서 살펴보도록 한다.

[0183] **I2NSF 기능 정보 모델: 운영 이론(I2NSF Capability Information Model: Theory of Operation)**

[0184] 기능은 일반적으로 호출할 수 있는 NSF 함수를 나타내는 데 사용된다. 기능은 객체이므로 I2NSF ECA 정책 규칙의 이벤트, 조건 및/또는 액션을 설명하는 절에서 사용할 수 있다.

[0185] I2NSF 기능 정보 모델은 사전 정의된 메타 데이터 모델을 구체화한다. I2NSF 기능의 적용은 기능 집합을 사용, 관리 또는 조작하는 방법을 정의하는 사전 정의된 ECA 정책 규칙 정보 모델을 수정함으로써 수행될 수 있다. 이러한 접근법에서 I2NSF 정책 규칙은 이벤트 절, 조건 절 및 작업 절의 세 가지 절로 구성된 컨테이너 역할을 수행할 수 있다.

[0186] I2NSF 정책 엔진이 일련의 이벤트를 수신하면 해당 이벤트를 활성 ECA 정책 규칙의 이벤트와 일치시킨다. 이벤트가 일치하면 일치하는 I2NSF 정책 규칙의 조건절의 평가를 트리거한다. 조건 절이 평가되고, 이것이 일치하는 경우, 일치하는 I2NSF 정책 규칙에 있는 일련의 행동이 실행될 수 있다.

[0188] **초기 NSFs 기능 카테고리(Initial NSFs Capability Categories)**

[0189] 이하, 네트워크 보안, 콘텐츠 보안 및 공격 완화의 세 가지 일반적인 기능에 대해서 살펴본다. 본 명세서에서 살펴보는 특정 카테고리 내의 카테고리 수와 기능 유형은 모두 확장될 수 있다.

[0191] **네트워크 보안 기능(Network Security Capabilities)**

[0192] 네트워크 보안은 미리 정의 된 보안 정책을 사용하여 네트워크 트래픽을 검사하고 처리하는 방법을 설명하기 위한 카테고리이다.

[0193] 검사 부분은 직접적으로 또는 패킷이 연관된 흐름의 맥락에서 네트워크를 통과하는 패킷을 검사하는 패킷 처리 엔진일 수 있다. 패킷 처리의 관점에서 볼 때 구현할 수 있는 패킷 헤더 및/또는 페이로드의 내용, 유지할 수 있는 다양한 흐름 및 컨텍스트 상태, 패킷 또는 흐름에 적용 할 수 있는 동작이 구현에 따라 달라질 수 있다.

[0195] **콘텐츠 보안 기능(Content Security Capabilities)**

[0196] 콘텐츠 보안은 응용 프로그램 계층에 적용되는 보안 기능의 또 다른 카테고리이다. 예를 들어, 응용 프로그램 계층에서 전달되는 트래픽 내용을 분석하여 콘텐츠 보안 기능을 사용함으로써 필요한 다양한 보안 기능을 식별할 수 있다.

[0197] 여기에는 침입에 대한 방어, 바이러스 검사, 악의적 인 URL 또는 정크 메일 필터링, 불법적 인 웹 액세스 차단 또는 악의적인 데이터 검색 방지가 포함될 수 있다.

[0198] 일반적으로 콘텐츠 보안의 각 위협 유형에는 고유한 특성 집합이 있으며 해당 유형의 콘텐츠에 고유한 메서드 집합을 사용하여 처리해야 한다. 따라서 이러한 기능은 고유한 콘텐츠 별 보안 기능을 특징으로 한다.

[0200] **공격 완화 기능(Attack Mitigation Capabilities)**

[0201] 공격 완화 기능은 다양한 유형의 네트워크 공격을 탐지하고 완화하는데 사용된다. 오늘날 일반적인 네트워크 공격은 아래와 같이 정의될 수 있다.

[0202] - DDoS 공격:

[0203] 네트워크 계층 DDoS 공격: SYN flood, UDP flood, ICMP flood, IP fragment flood, IPv6 routing header attack 및 IPv6 duplicate address detection 공격을 예로 들 수 있다.

[0204] 응용 프로그램 계층 DDoS 공격: 예를 들어 HTTP flood, https flood, 캐시 우회 HTTP floods, WordPress XML RPC floods 및 SSL DDoS가 있습니다.

[0205] - 단일 패킷 공격:

[0206] 스캐닝(scanning) 및 스니핑(sniffing) 공격: IP 스윕(sweep), 포트 스캐닝 등

[0207] 잘못된 패킷 공격: Ping of Death, Teardrop 등

[0208] 특별 패킷 공격: 특대 ICMP, Tracert, IP 타임 스탬프 옵션 패킷 등

[0209] 각 유형의 네트워크 공격에는 고유한 네트워크 동작 및 패킷/흐름 특성이 있다. 따라서, 각 유형의 공격에는 탐지 및 완화를 위해 기능 집합으로 알리는 특수 보안 기능이 필요하다. 이러한 보안 범주의 구현 및 관리 공격 완화 제어 기능은 콘텐츠 보안 제어 범주와 매우 유사할 수 있다.

[0210] 네트워크 보안 기능을 위한 정보 하부 모델(Information Sub-Model for Network Security Capabilities)

[0211] 기능 정보 하위 모델의 목적은 기능의 개념을 정의하고 기능들을 적절한 객체에 집계할 수 있게 하는 것이다. 이하, 네트워크 보안, 콘텐츠 보안 및 공격 완화 기능 하위 모델에 대해 설명하도록 한다.

[0213] **네트워크 보안을 위한 정보 하위 모델(Information Sub-Model for Network Security)**

[0214] 도 4는 본 명세서가 적용될 수 있는 네트워크 보안 정보 하위 모델 개요의 일 예를 나타낸다.

[0215] 네트워크 보안 정보 하위 모델의 목적은 네트워크 트래픽을 정의하는 방법을 정의하고 하나 이상의 네트워크 보안 기능을 트래픽에 적용해야 하는지 여부를 결정하기 위한 것이다.

[0216] 도 4에서 ECA정책규칙은 이벤트, 조건 및 동작 객체와 함께 외부 ECA 정보 모델에 정의되어 있다. 네트워크 보안 하위 모델은 보안 관련 ECA 정책 규칙 및 (일반)이벤트, 조건 및 조치 개체에 대한 확장을 정의하기 위해 이러한 모든 개체를 확장할 수 있다.

[0217] I2NSF 정책 규칙은 이벤트 조건 동작 (ECA) 형식의 특수한 유형의 정책 규칙이다. 정책 규칙, 정책 규칙의 구성 요소(예를 들면: 이벤트, 조건, 작업 및 해결 정책, 기본 작업 및 외부 데이터와 같은 일부 확장자) 및 선택적으로 메타 데이터로 구성될 수 있으며, NSF를 통한 단방향 및 양방향 트래픽에 모두 적용될 수 있다.

[0219] **네트워크 보안 정책 규칙 확장(Network Security Policy Rule Extensions)**

[0220] 도 5은 본 명세서가 적용될 수 있는 네트워크 보안 정보 하위 모델의 확장의 일 예를 나타낸다.

[0221] 도 5는 네트워크 보안 정보 하위 모델에 포함된 ECA 정책 규칙 하위 클래스의 보다 자세한 디자인의 일 예를 나타낸다. 이는 보다 구체적인 네트워크 보안 정책들이 SecurityECAPolicyRule 클래스에서 이전되고 확장되는 방법을 보여준다.

[0222] 다음과 같은 패턴의 클래스 설계를 따르면 새로운 종류의 특정 네트워크 보안 정책을 생성 할 수 있다.

[0223] SecurityECAPolicyRule은 I2NSF ECA 정책 규칙 계층의 맨 위에 위치한다. 이 규칙은 (외부) 일반 ECA 정책 규칙에서 이전되며 보안 관련 ECA 정책 규칙을 추가하기 위한 이러한 일반 ECA 정책 규칙의 특수화를 나타낸다.

[0224] SecurityECAPolicyRule은 슈퍼 클래스에 정의된 모든 속성, 메소드 및 관계를 포함하며 네트워크 보안에 필요한 추가 개념을 추가한다.

[0225] 6 개의 SecurityECAPolicyRule 서브 클래스는 SecurityECAPolicyRule 클래스를 확장하여 6 가지 유형의 Network Security ECA Policy Rules를 나타낸다. (외부) 일반 ECAPolicyRule 클래스는 설명 및 기타 필요한 정보뿐만 아니라 고유한 객체 ID와 같은 속성의 형태로 기본 정보를 정의할 수 있다.

[0226] 네트워크 보안 정책 규칙 동작(Network Security Policy Rule Operation)

[0227] 네트워크 보안 정책은 위에서 설명한 정보 모델로 구성된 하나 이상의 ECA 정책 규칙으로 구성된다. 이벤트 및 조건 절이 변경되지 않은 간단한 경우에는 한 정책 규칙의 작업이 다른 정책 규칙에서 추가 네트워크 보안 작업을 호출 할 수 있다. 네트워크 보안 정책은 다음과 같이 트래픽을 검사하고 기본 처리를 수행한다.

[0228] 1. NSF는 주어진 SecurityECAPolicyRule의 이벤트 절을 평가한다(도 3에 도시된 바와 같이 보안에 일반적이거나 특정 일 수 있음). 보안 이벤트 객체를 사용하여 아래 설명할 평가의 전부 또는 일부를 수행 할 수 있다.

[0229] Event 절이 TRUE로 평가되면 이 SecurityECAPolicyRule의 조건 절이 평가된다. 그렇지 않으면 SecurityECAPolicyRule의 실행이 중지되고 다음 SecurityECAPolicyRule 이 평가될 수 있다.

[0230] 2. 이후, 조건 절이 평가될 수 있다. 보안 요구 사항 객체를 사용하여 아래에서 설명할 평가의 전부 또는 일부가 수행될 수 있다. 조건 절이 TRUE로 평가되면 SecurityECAPolicyRule과 "일치"하는 것으로 정의된다. 그렇지 않으면 SecurityECAPolicyRule의 실행이 중지되고 다음 SecurityECAPolicyRule이 평가될 수 있다.

[0231] 3. 실행될 일련의 작업이 검색되고, 해결 전략이 실행 순서를 정의하는 데 사용된다. Step 3)에서 프로세스에는 SecurityECAPolicyRule과 관련된 선택적 외부 데이터 사용이 포함될 수 있다.

[0232] 4. 실행은 다음 세 가지 형식 중 하나를 취합니다.

[0233] a. 하나 이상의 행동이 선택되면, NSF는 해결 전략에 의해 정의된 행동을 수행 할 수 있다. 예를 들어, 해결 전략은 단일 액션 (예를 들면: FMR 또는 LMR) 만 실행되도록 허용하거나 모든 액션이 실행되도록 허용 할 수 있다(선택적으로 또는 특정 순서로).

[0234] 이러한 경우와 다른 경우 NSF 기능은 실행 방법을 명확하게 정의해야 한다.

[0235] 보안 액션 객체를 사용하여 아래에서 설명하는 실행의 전부 또는 일부를 수행 할 수 있습니다. 기본 액션이 허가 또는 미러인 경우 NSF는 먼저 해당 기능을 수행 한 다음 특정 보안 기능이 규칙에서 참조되는지 여부를 확인한다. 만약 “Yes” 인 경우, Step 5로 이동한다. No인 경우, 트래픽이 허용된다.

[0236] b. 선택된 동작이 없고 기본 동작이 있는 경우, 기본 동작이 수행될 수 있다. 그렇지 않으면 아무 작업도 수행

되지 않는다.

[0237] c. 그렇지 않으면 트래픽이 거부됩니다.

[0239] 5. SecurityECAPolicyRule의 동작 집합에서 다른 보안 기능(예를 들면: 바이러스 백신 또는 IPS 프로파일 NSF가 암시하는 조건 및 / 또는 동작)이 참조되는 경우 NSF는 참조 된 보안 기능을 사용하도록 구성 할 수 있다 (예를 들면: check 조건 또는 행동 집행).

[0240] 이후, 실행이 종료될 수 있다.

[0242] **네트워크 보안 이벤트 하위 서브 모델(Network Security Event Sub-Model)**

[0243] 도 6는 본 명세서가 적용될 수 있는 네트워크 보안 정보 하위 모델 이벤트 클래스의 확장의 일 예를 나타낸다.

[0244] 도 6은 네트워크 보안 정보 하위모델에 포함된 이벤트 하위 클래스의 디자인의 일 예를 나타낸다.

[0245] 도 6의 네 가지 Event 클래스는 (외부) 일반 Event 클래스를 확장하여 네트워크 보안에서 중요한 이벤트를 나타낸다. (외부) 일반 Event 클래스는 고유 이벤트 ID, 설명 및 이벤트가 발생한 날짜 및 시간과 같은 속성 양식의 기본 이벤트 정보를 정의한다고 가정할 수 있다.

[0246] 네트워크 보안 조건 하위 서브 모델(Network Security Condition Sub-Model)

[0247] 도 7는 본 명세서가 적용될 수 있는 네트워크 보안 정보 하위 모델 조건 클래스의 확장의 일 예를 나타낸다.

[0248] 도 7은 네트워크 보안 정보 하위 모델에 포함된 조건 하위 클래스의 보다 상세한 디자인을 나타낸다.

[0249] 도 7에 표시된 여섯 가지 조건 클래스는 (외부) 일반 조건 클래스를 확장하여 네트워크 보안과 관련된 조건을 나타낸다. (외부) 일반 조건 클래스는 추상적이므로 데이터 모델 최적화가 정의 될 수 있다고 가정한다.

[0250] 일반 조건 클래스는 고유 한 객체 ID, 설명 및 0 개 이상의 메타 데이터 객체를 연결하는 메커니즘과 같은 속성의 형태로 기본 조건 정보를 정의한다고 가정한다.

[0252] **네트워크 보안 동작 서브 모델(Network Security Action Sub-Model)**

[0253] 도 8은 본 명세서가 적용될 수 있는 네트워크 보안 정보 하위 모델 액션의 확장의 일 예를 나타낸다.

[0254] 도 8은 네트워크 보안 정보 하부 모델에 포함 된 조치 서브 클래스의 보다 자세한 설계를 나타낸다. 도 8의 네 가지 동작 클래스는 (외부)일반 동작 클래스를 확장하여 네트워크 보안 제어 기능을 수행하는 작업을 나타낸다.

[0255] 도 8의 세 가지 동작 클래스는 (외부) 일반 동작 클래스를 확장하여 네트워크 보안과 관련된 작업을 나타낸다. (외부) Generic Action 클래스는 추상적이므로 데이터 모델 최적화가 정의 될 수 있다.

[0256] 일반적인 동작 클래스는 고유 한 객체 ID, 설명 및 0 개 이상의 메타 데이터 객체를 첨부하는 메커니즘과 같은 속성 형식의 기본 동작 정보를 정의한다고 가정한다.

[0258] **I2NSF 기능을 위한 정보 모델(Information Model for I2NSF Capabilities)**

[0259] 도 9은 본 명세서가 적용될 수 있는 I2NSF 보안 기능의 상위 레벨 모델의 일 예를 나타낸다.

[0260] 도 9에 도시된 바와 같이 I2NSF 기능 모델은 다양한 콘텐츠 보안 및 공격 완화 기능을 나타내는 많은 기능으로 구성된다. 각 기능은 응용 프로그램 계층에서 특정 유형의 위협으로부터 보호한다.

[0261] 도 9는 SecurityCapability라고 하는 일반적인 I2NSF 보안 기능 클래스를 도시한다. 이를 통해 외부 메타 데이터 정보 모델의 디자인에 영향을 주지 않으면 서 이 클래스에 공통 속성, 관계 및 동작을 추가 할 수 있다. 모든 I2NSF 보안 기능은 SecuritCapability 클래스에서 서브 클래싱된다.

[0263] **컨텐츠 보안 기능을 위한 정보 모델(Information Model for Content Security Capabilities)**

[0264] 도 10은 본 명세서가 적용될 수 있는 네트워크 보안 기능 정보 모델의 일 예를 나타낸다.

[0265] 도 10은 콘텐츠 보안 GNSF(Generic Network Security Function)의 예시적인 유형들을 도시한다.

[0266] 도 10에 도시된 바와 같이 콘텐츠 보안은 여러 가지 고유 한 보안 기능으로 구성될 수 있다. 이러한 각 기능은 응용 프로그램 계층에서 특정 유형의 위협으로부터 콘텐츠를 보호할 수 있다.

[0267] 콘텐츠 보안은 도 10에 도시된 바와 같이 GNSF (Generic Network Security Function) 유형일 수 있다.

[0269] **공격완화 기능을 위한 정보 모델(Information Model for Attack Mitigation Capabilities)**

[0270] 도 11는 본 명세서가 적용될 수 있는 공격 완화 기능 정보 모델의 일 예를 나타낸다.

[0271] 도 11에 도시된 바와 같이 공격 완화는 여러 GNSF로 구성될 수 있다. 각각은 특정 유형의 네트워크 공격으로부터 컨텐츠를 보호합니다. 공격 완화 (Acknowledge mitigation) 보안은 잘 정의 된 보안 기능을 요약 한 GNSF 유형이다.

[0273] **I2NSF 보안 정책의 구조와 목적**

[0274] 1. I2NSF 보안 정책 규칙(I2NSF Security Policy Rule)

[0275] I2NSF 보안 정책 규칙은 일반 네트워크 보안 기능에 대한 정책 규칙을 나타낸다. 정책 규칙의 객체는 정책 정보 및 규칙 정보로 정의될 수 있다. 여기에는 Event Clause Objects, Condition Clause Objects, Action Clause Objects, Resolution Strategy 및 Default Action과 같은 ECA 정책 규칙이 포함될 수 있다.

[0276] 2. Event Clause

[0277] 이벤트는 앞에서 살펴본 바와 같이 관리되는 시스템이 변경 될 때 및/또는 관리되는 시스템의 환경에서 중요한 시점에 발생할 수 있다.

[0278] Event Clause Objects는 I2NSF 정책 규칙의 컨텍스트에서 사용될 때 I2NSF 정책 규칙의 조건 절을 평가할 수 있는지 여부를 결정하는 데 사용될 수 있다. 이벤트 절의 대상은 사용자 보안 이벤트, 장치 보안 이벤트, 시스템 보안 이벤트 및 시간 보안 이벤트로 정의될 수 있다. 이벤트 조항의 대상은 특정 공급 업체 이벤트 기능에 따라 확장될 수 있다.

[0279] 3. Condition Clause

[0280] 조건은 앞에서 살펴본 바와 같이 알려진 속성, 특징 및/또는 값의 세트와 비교 될 속성, 기능 및/또는 값의 집합으로 정의되어 그 (명령형) I2NSF 정책 규칙을 실행하거나 실행하지 않을 수 있다.

[0281] 이러한 object는 패킷 보안 조건, 패킷 페이로드 보안 조건, 대상 보안 조건, 사용자 보안 조건, 컨텍스트 조건 및 일반 컨텍스트 조건으로 정의될 수 있다.

[0282] Action 조항의 오브젝트는 특정 공급 업체 조건 기능에 따라 확장될 수 있다.

[0283] 4. Action Clause

[0284] 동작은 이벤트 및 조건 절이 충족될 때 흐름 기반 NSF의 측면을 제어하고 모니터링 하는데 사용된다. NSF는 다양한 액션을 실행하여 보안 기능을 제공한다. 동작 절의 오브젝트는 입력 동작, 송신 동작 및 적용 프로파일 동작으로 정의될 수 있으며, 동작 절의 오브젝트는 특정 벤더 조치 기능에 따라 확장될 수 있다.

[0286] 데이터 모델 구조

[0287] 이하, 본 명세서에서 제안하는 데이터 모델에 대해 살펴보도록 한다.

[0288] 본 명세서에서 제안하는 데이터 모델의 구조는 아래와 같은 사항이 고려되었다.

[0289] - Event, Condition, Action 절 집계에 의한 ECA 정책 모델의 고찰

[0290] - 기능 대수의 고려.

[0291] - NSF 기능 카테고리 (예 : 네트워크 보안, 컨텐츠 보안 및 공격 완화 기능) 고려.

[0292] - 네트워크 보안 이벤트 클래스, 네트워크 보안 조건 클래스 및 네트워크 보안 작업 클래스에 대한 정의.

[0294] 도 12는 본 명세서의 일 실시 예에 따른 네트워크 보안 정책 식별을 위한 데이터 모델 구조를 예시한다.

[0295] 네트워크 보안 정책을 식별하기 위한 데이터 모델은 도 12에 도시된 바와 같은 구조로 구성될 수 있다

[0296] 네트워크 보안 정책을 식별하기 위한 데이터 모델은 보안 정책, 이벤트 절 컨테이너, 조건 절 컨테이너 및 동작 절 컨테이너로 구성될 수 있다.

[0297] 보안 정책의 데이터 필드는 정책 이름, 규칙들, 해결 전략, 고정 동작 및 규칙(rule) 그룹으로 구성될 수 있다.

[0298] 규칙들은 규칙들을 식별하기 위한 이름, 규칙을 설명하기 위한 description, priority, enable, session-aging-time, long-connection, policy-event-clause-agg-ptr*, policy-condition-clause-agg-ptr*, policy-action-clause-agg-ptr* 및 time-zone으로 구성될 수 있다.

[0299] long-connection은 규칙이 적용될 수 있는 지속시간을 설정할 수 있도록 enable 및 during을 포함할 수 있다.

[0300] 또한, time-zone은 적용되는 룰의 절대적인 시간 외에 주기적인 시간을 설정할 수 있도록 absolute-time-zone 및 periodic-time-zone을 포함할 수 있다.

[0301] absolute-time-zone는 룰이 적용되는 절대적인 시간 또는 날짜를 설정하기 위해서 시작 시간 및 종료 시간을 설정하기 위한 start-time?, end-time? 및 날짜를 설정하기 위한 absolute-date*를 포함할 수 있다.

[0302] periodic-time-zone은 룰이 적용되는 주기적인 시간을 설정하기 위한 day 및 month를 포함할 수 있다.

[0303] resolution-strategy은 룰을 위한 해결 전략을 설정하기 위해서 전략의 타입을 설정하기 위한 (resolution-strategy-type)?, 첫 번째로 매칭되는 룰을 설정하기 위한 first-matching-rule? 및 마지막으로 매칭되는 룰을 설정하기 위한 last-matching-rule?을 포함할 수 있다.

[0304] default-action은 선택된 동작이 없는 경우 수행될 수 있는 동작을 설정하기 위한 필드로써 동작의 타입을 설정할 수 있다.

[0305] rule-group은 규칙들이 그룹화되어 관리될 수 있는 그룹들로 구성되며, 각 그룹에 대한 데이터 필드는 group-name, rule-range, enable, description을 포함한다.

[0306] event-clause-container, condition-clause-container 및 action-clause-container는 정책 규칙이 “이벤트”, “동작” 및 “조건”을 집계하기 위해서 사용될 수 있다.

[0308] 도 13은 본 명세서의 일 실시 예에 따른 이벤트 규칙을 위한 데이터 모델 구조를 예시한다.

[0309] 이벤트는 앞에서 살펴본 바와 같이 관리되는 시스템이 변경될 때 및/또는 관리되는 시스템의 환경에서 중요한 시점에 발생하는 사건을 의미한다.

[0310] 도 13에 도시된 이벤트 절을 위한 오브젝트들은 사용자 보안 이벤트, 장치 보안 이벤트, 시스템 보안 이벤트 및 시간 보안 이벤트로 정의될 수 있다. 이러한 개체는 특정 공급 업체 이벤트 기능에 따라 확장될 수 있으며, 보다 일반적인 네트워크 보안 기능을 위한 추가 이벤트 객체가 추가될 수 있다.

[0312] 도 14a 내지 도 14d는 본 명세서의 일 실시 예에 따른 컨디션 규칙을 위한 데이터 모델 구조를 예시한다.

[0313] 조건은 앞에서 살펴본 바와 같이 알려진 속성, 특징 및/또는 값의 세트와 비교될 속성, 기능 및/또는 값의 집합으로 정의되어 그 (명령형) I2NSF 정책 규칙을 실행하거나 실행하지 않을 수 있다.

[0314] 컨디션 규칙을 위한 객체는 패킷 보안 조건, 패킷 페이로드 보안 조건, 대상 보안 조건, 사용자 보안 조건, 컨

텍스트 조건 및 일반 컨텍스트 조건으로 정의될 수 있다.

[0315] 이러한 컨디션 규칙을 위한 개체는 특정 공급 업체 조건 기능에 따라 확장 될 수 있으며, 보다 일반적인 네트워크 보안 기능을 위한 조건 개체를 추가 할 수 있다.

[0316] 또한, 도 14c에 도시된 바와 같이 컨디션 규칙을 위한 데이터 모델 구조는 pkt-sec-cond-tcp-src-port*, pkt-sec-cond-tcp-dest-port*, pkt-sec-cond-udp-src-port* 및 pkt-sec-cond-udp-dest-port*를 통해 포트 번호와 관련된 룰을 설정할 수 있다.

[0317] 도 14d에서는 규칙이 적용될 수 있는 어플리케이션의 상태를 관리하기 위해, application-condition의 데이터 필드는 application-description?, application-object*, application-group*, application-label*, category를 포함한다.

[0318] 또한, 규칙들은 URL(Uniform Resource Locator)에 따라 적용여부가 설정될 수 있으며, 이를 위해 url-category-condition의 데이터 필드는 pre-defined-category*, user-defined-category*를 포함한다.

[0319] 도 15는 본 명세서의 일 실시 예에 따른 액션 규칙을 위한 데이터 모델 구조를 예시한다.

[0321] 동작은 이벤트 및 조건 절이 충족될 때 흐름 기반 NSF의 측면을 제어하고 모니터링 하는데 사용된다.

[0322] 이러한 개체는 수신 동작, 송신 동작 및 적용 프로필 동작으로 정의될 수 있다. 이러한 개체는 특정 공급 업체 작업 기능에 따라 확장될 수 있으며, 보다 일반적인 네트워크 보안 기능을 위한 액션 객체를 추가 할 수 있다.

[0324] 도 14a 내지 도 15에 도시된 컨디션 규칙 및 액션 규칙을 위한 데이터 모델의 구조는 컨테이너 구조가 사용되기 때문에 다중의 컨디션을 적용할 수 있다.

[0326] 도 16a 내지 도 19j는 본 명세서의 일 실시 예에 따른 I2NSF NSF-Facing Interface의 YANG 데이터 모듈을 예시한다.

[0327] 도 16a 내지 도 19j를 참조하면, 도 12a 내지 도 15b에서 설명한 데이터 모델을 이용하여 네트워크 보안 기능들의 정보 모델을 위한 YANG 데이터 모델을 설정할 수 있다.

[0329] 도 16a 내지 19j에 도시된 모듈은 네트워크 보안 기능들을 위한 양 데이터 모듈로 정의될 수 있다.

[0330] 이하, NSF 모니터링을 위한 정보 모델에 대해 살펴보도록 한다.

[0331] 보안 기능을 구성하기 위해 관리 엔티티(예를 들면: NMS, 보안 컨트롤러)에 NSF(예를 들면: FW, IPS, Anti-DDOS 또는 Anti-Virus 기능)가 제공하는 인터페이스 NSF에서 모니터링하고 NSF를 모니터링하는 것을 "I2NSF NSF-Facing Interface"라고 한다(ID.ietf-i2nsf-terminology 참조).

[0332] 모니터링 부분은 NSF에 관한 중요한 정보를 획득하는 것을 의미한다. 알림, 이벤트, 레코드, 카운터. 시의 적절하고 포괄적인 방식으로 수행되면 NSF 모니터링은 전반적인 보안 프레임 워크에서 매우 중요한 역할을 한다. NSF에 의해 생성된 모니터링 정보는 악의적인 활동 또는 비정상적인 행동 또는 서비스 거부 공격의 잠재적 징후의 조기 표시일 수 있다.

[0333] NSF 모니터링 데이터는 아래와 같은 상황에서 사용될 수 있다.

[0334] 위에서 설명한 바와 같이 모니터링은 전반적인 보안 프레임 워크에서 매우 중요한 역할을 한다. NSF를 모니터링하면 규정된 보안 상태를 유지하는 데 있어 보안 컨트롤러에 매우 중요한 정보가 제공된다. 이 외에도 아래와 같이 NSF를 모니터링 할 수 있는 다른 이유가 있다.

[0335] - 보안 관리자는 NSF 또는 네트워크에서 발생한 특정 이벤트에서 트리거 되는 정책을 구성할 수 있다. 보안 컨트롤러는 지정된 이벤트를 모니터링하고 이벤트가 발생하면 정책에 따라 추가 보안 기능을 구성한다.

[0336] - 보안 정책 위반의 결과로 NSF에 의해 촉발된 사건은 의심스러운 활동을 탐지하기 위해 SIEM에 의해 사용될 수

있다.

[0337] - NSF의 이벤트 및 활동 로그를 사용하여 동작 및 예측과 같은 고급 분석을 구축하여 보안 상태를 개선할 수 있다.

[0338] - 보안 컨트롤러는 고 가용성을 달성하기 위해 NSF의 이벤트를 사용할 수 있다. 실패한 NSF 재시작, NSF 수평 확장 등의 수정 조치를 취할 수 있다.

[0339] - NSF의 이벤트 및 활동 로그는 운영 문제의 디버깅 및 근본 원인 분석에 도움이 될 수 있다.

[0340] - NSF의 활동 기록은 운영 및 비즈니스상의 이유로 기록 데이터를 작성하는 데 사용될 수 있다.

[0341] NSF 모니터링 데이터의 분류

[0342] 강력한 보안 상태를 유지하려면 NSF 보안 정책을 구성 할뿐만 아니라 관찰 가능한 정보를 소비하여 NSF를 지속적으로 모니터링 해야 한다. 이를 통해 보안 관리자는 적시에 네트워크에서 어떤 일이 일어나고 있는지 평가할 수 있다.

[0343] 정적 보안 상태에 기반하여 모든 내부 및 외부 위협을 차단하는 것은 불가능하다. 이 목표를 달성하려면 일정한 가시성을 가진 매우 역동적인 자세가 필요하다. 본 명세서는 NSF에서 얻을 수 있고 모니터링 정보로 사용될 수 있는 일련의 정보 요소(및 그 범위)를 정의할 수 있다.

[0344] 본질적으로 이러한 유형의 모니터링 정보는 여러 수준의 세밀성에 대한 지속적인 가시성을 지원하기 위해 활용 될 수 있으며 해당 기능에 의해 소비 될 수 있다

[0345] 이하, 모든 모니터링 데이터를 위한 기본적인 정보 모델에 대해 살펴보도록 한다.

[0346] 모든 모니터링 데이터를 위한 기본적인 정보 모델(Basic Information Model for All Monitoring Data)

[0347] - message_version: 데이터 형식의 버전을 나타내며 01에서 시작하는 2 자리 10 진수.

[0348] - message_type : 이벤트, 경고, 알람, 로그, 카운터 등

[0349] - time_stamp: 메시지가 생성 된 시간을 나타냄.

[0350] - vendor_name: NSF 공급 업체의 이름.

[0351] - NSF_name: 메시지를 생성하는 NSF의 이름 (또는 IP).

[0352] - Module_name: 메시지를 출력하는 모듈 이름

[0353] - Severity: 로그의 레벨을 나타냄. 총 8 개의 레벨 (0에서 7까지)이 존재하며, 숫자가 작을수록 심각도가 높다.

[0354] 모니터링 데이터를위한 확장 정보 모델(Extended Information Model for Monitoring Data)

[0355] 확장 정보 모델은 알람과 같은 구조화 된 데이터에만 사용됩니다. 구조화되지 않은 데이터는 기본 정보 모델로만 지정된다.

[0356] 시스템 알람(System Alarm)

[0357] 메모리 알람(Memory Alarm)

[0358] 다음 정보가 메모리 알람에 포함되어야 한다.

[0359] - event_name: 'MEM_USAGE_ALARM'

[0360] - module_name:알람 생성을 담당하는 NSF 모듈을 나타냄.

[0361] - usage: 사용 된 메모리 양을 지정함.

[0362] - 임계 값: 경보를 트리거 하는 임계 값

[0363] - 심각도: 위험 수준 (예를 들면: 위험 수준, 높음, 보통, 낮음)

[0364] - 메시지: '메모리 사용량이 임계 값을 초과했습니다.'와 같은 메시지를 출력함.

[0366] CPU 알람(CPU Alarm)

[0367] 다음과 같은 정보가 CPU 알람에 포함될 수 있다.

[0368] - event_name: 'CPU_USAGE_ALARM'

[0369] - usage: 사용 된 CPU의 양을 지정합니다.

[0370] - threshold: 이벤트를 트리거 하는 임계 값

[0371] - 심각도: 위험 수준 (예를 들면: 위험 수준, 높음, 보통, 낮음)

[0372] - 메시지: 'CPU 사용량이 임계 값을 초과했습니다.' 와 같은 메시지를 출력함.

[0374] 디스크 알람(Disk Alarm)

[0375] 다음과 같은 정보가 디스크 알람에 포함될 수 있다.

[0376] - event_name: 'DISK_USAGE_ALARM'

[0377] - usage: 사용 된 디스크 공간의 양을 지정합니다.

[0378] - threshold: 이벤트를 트리거 하는 임계 값

[0379] - 심각도: 위험 수준 (예를 들면: 위험 수준, 높음, 보통, 낮음)

[0380] - 메시지: '디스크 사용량이 임계 값을 초과했습니다.' 와 같은 메시지를 출력함.

[0382] 하드웨어 알람(Hardware Alarm)

[0383] 다음과 같은 정보가 하드웨어 알람에 포함될 수 있다.

[0384] - event_name: 'HW_FAILURE_ALARM'

[0385] - component_name:이 알람을 생성하는 HW 구성 요소를 나타냅니다.

[0386] - 임계 값: 경보를 트리거 하는 임계 값

[0387] - 심각도: 위험 수준 (예를 들면: 위험 수준, 높음, 보통, 낮음)

[0388] - 메시지: '하드웨어 구성 요소가 고장 났거나 성능이 저하되었습니다.'와 같은 메시지를 출력함.

[0390] 인터페이스 알람(Interface Alarm)

[0391] 다음과 같은 정보가 인터페이스 알람에 포함될 수 있다.

[0392] - event_name: 'IFNET_STATE_ALARM'

[0393] - interface_Name: 인터페이스 이름

[0394] - interface_state: 'UP', 'DOWN', 'CONGESTED'

[0395] - threshold: 이벤트를 트리거 하는 임계 값

[0396] - 심각도: 위험 수준 (예를 들면: 위험 수준, 높음, 보통, 낮음)

[0397] - 메시지: '현재 인터페이스 상태'를 출력함.

[0399] 시스템 이벤트(System Events)

[0400] 액세스 위반(Access Violation)

[0401] 다음과 같은 정보가 이벤트에 포함될 수 있다.

[0402] - event_name: 'ACCESS_DENIED'

[0403] - user: 사용자 이름

[0404] - group: 사용자가 속한 그룹

[0405] - login_ip_address: 사용자의 로그인 IP 주소

[0406] - authentication_mode: 사용자 인증 모드. 예를 들면: 로컬 인증, 제 3 자 서버 인증, 인증 면제, SSO 인증

[0408] - 메시지: '액세스가 거부되었습니다.' 와 같은 메시지를 출력함.

[0410] 구성 변경(Configuration Change)

[0411] 다음과 같은 정보가 이벤트에 포함될 수 있다.

[0412] - event_name: 'CONFIG_CHANGE'

[0413] - user: 사용자 이름

[0414] - group: 사용자가 속한 그룹

[0415] - login_ip_address: 사용자의 로그인 IP 주소

[0416] - authentication_mode: 사용자 인증 모드. 예를 들면: 로컬 인증, 제 3 자 서버 인증, 인증 면제, SSO 인증

[0417] - 메시지: '구성이 수정되었습니다' 와 같은 메시지를 출력함.

[0419] 시스템 로그(System Log)

[0420] 접속 로그(Access Logs)

[0421] 액세스 로그는 관리자의 로그인, 로그 아웃 및 장치 작동을 기록하고, 이를 분석하여 보안 취약성을 식별 할 수 있다. 운영 보고서에는 아래와 같은 정보가 포함될 수 있다.

[0422] - 관리자: 장치에서 작동하는 관리자

[0423] - login_ip_address : 관리자가 로그인 할 때 사용하는 IP 주소

[0424] - login_mode : 관리자 로그인 모드를 지정합니다.(예를 들면: 뿌리, 사용자)

[0425] - operation_type : 관리자가 수행하는 조작 유형(예를 들면: 로그인, 로그 아웃, 구성 등)

[0426] - 결과: 명령 실행 결과

[0427] - content: 로그인 후 관리자가 수행 한 작업.

[0428] 자원 사용률 로그(Resource Utilization Logs)

[0430] 실행중인 보고서는 장치 시스템의 실행 상태를 기록하며 이는 장치 모니터링에 유용하다. 실행 보고서는 다음과 같은 정보를 포함할 수 있다.

[0431] - system_status: 현재 시스템의 실행 상태

[0432] - CPU_usage: CPU 사용량을 지정합니다.

[0433] - memory_usage: 메모리 사용량을 지정합니다.

[0434] - disk_usage: 디스크 사용량을 지정합니다.

[0435] - disk_left: 사용 가능한 디스크 공간을 지정합니다.

[0436] - session_number: 총 동시 세션 수를 지정합니다.

[0437] - process_number: 총 시스템 프로세스 수를 지정합니다.

[0438] - in_traffic_rate: 총 인바운드 트래픽 속도 (pps)

[0439] - out_traffic_rate: 총 아웃 바운드 트래픽 속도 (pps)

[0440] - in_traffic_speed: 총 인바운드 트래픽 속도 (bps)

[0441] - out_traffic_speed: 총 아웃 바운드 트래픽 속도 (bps)

[0442] 사용자 활동 로그(User Activity Logs)

[0443] 사용자 활동 기록은 사용자의 온라인 기록 (로그인 시간, 온라인 / 잠금 기간 및 로그인 IP 주소)과 사용자가 수행하는 작업에 대한 가시성을 제공한다. 사용자 활동 보고서는 사용자 로그인 및 네트워크 액세스 활동 중 예외를 식별하는 데 유용하다.

[0444] - group: 사용자가 속한 그룹

[0445] - login_ip_address: 사용자의 로그인 IP 주소

[0446] - authentication_mode: 사용자 인증 모드. 예를 들면: 로컬 인증, 제 3 자 서버 인증, 인증 면제, SSO 인증

[0447] - access_mode: 사용자 액세스 모드. 예를 들면: PPP, SVN, LOCAL

[0448] - online_duration: 온라인 기간

[0449] - lockout_duration: 잠금 기간

[0450] - 유형: 사용자 활동. 성공한 사용자 로그인, 실패한 로그인 시도, 사용자 로그 아웃, 성공한 사용자 비밀번호 변경, 실패한 사용자 비밀번호 변경, 사용자 잠금, 사용자 잠금 해제, 알 수 없음

[0451] - 원인: 사용자 작업에 실패했습니다.

[0452] 시스템 카운터(System Counter)

[0453] 인터페이스 카운터(Interface counters)

[0454] 인터페이스 카운터는 NSF로 들어오고 나가는 트래픽, 대역폭 사용에 대한 가시성을 제공한다.

[0455] - interface_name : NSF에서 구성된 네트워크 인터페이스 이름

[0456] - in_total_traffic_pkts : 전체 인바운드 패킷

[0457] - out_total_traffic_pkts : 총 아웃 바운드 패킷

[0458] - in_total_traffic_bytes : 총 인바운드 바이트

[0459] - out_total_traffic_bytes : 총 아웃 바운드 바이트

[0460] - in_drop_traffic_pkts : 총 인바운드 드롭 패킷

[0461] - out_drop_traffic_pkts : 총 아웃 바운드 드롭 패킷

[0462] - in_drop_traffic_bytes : 총 인바운드 드롭 바이트

[0463] - out_drop_traffic_bytes : 총 아웃 바운드 삭제 바이트

[0464] - in_traffic_ave_rate : 인바운드 트래픽 평균 요금 (pps)

[0465] - in_traffic_peak_rate : 인바운드 트래픽 피크 속도 (pps)

[0466] - in_traffic_ave_speed : 인바운드 트래픽 평균 속도 (bps)

[0467] - in_traffic_peak_speed : 인바운드 트래픽 최고 속도 (bps)

[0468] - out_traffic_ave_rate : 아웃 바운드 트래픽 평균 요금 (pps)

[0469] - out_traffic_peak_rate : 아웃 바운드 트래픽 피크 속도 (pps)

[0470] - out_traffic_ave_speed : 아웃 바운드 트래픽 평균 속도 (bps)

[0471] - out_traffic_peak_speed : 아웃 바운드 트래픽 최고 속도 (bps)

[0472] NSF 이벤트(NSF Events)

[0473] DDos 이벤트는 다음과 같은 정보를 포함할 수 있다.

[0474] - event_name : 'SEC_EVENT_DDoS'

[0475] - sub_attack_type : Syn flood, ACK flood, SYN-ACK flood, FIN / RST flood, TCP 연결 flood, UDP flood, Icmp flood, HTTPS flood, HTTP flood, DNS query flood, DNS reply flood, SIP flood 등

[0476] - dst_ip: 공격 받고있는 victum의 IP 주소

[0477] - dst_port: 트래픽을 목표로 삼고있는 포트 번호.

[0478] - start_time: 공격이 시작된 시간을 나타내는 타임 스탬프

[0479] - end_time: 공격이 종료 된 시간을 나타내는 타임 스탬프. 경보를 전송할 때 공격이 계속 발생하면이 필드는 비어있을 수 있습니다.

[0480] - attack_rate: 공격 트래픽의 PPS

[0481] - attack_speed: 공격 트래픽의 bps

[0482] - rule_id: 트리거되는 규칙의 ID입니다.

[0483] - rule_name: 트리거되는 규칙의 이름

[0484] - 프로필: 트래픽이 일치하는 보안 프로필입니다.

[0485] 세션 테이블 이벤트(session Table Event)

[0486] 아래와 같은 정보가 세션 테이블 이벤트에 포함될 수 있다.

[0487] - event_name: 'SESSION_USAGE_HIGH'

[0488] - current: 동시 세션 수

[0489] - max: 세션 테이블이 지원할 수있는 최대 세션 수

[0490] - threshold: 이벤트를 트리거하는 임계 값

[0491] - 메시지: '세션 테이블의 수가 임계 값을 초과했습니다.'

[0493] 바이러스 이벤트(Virus Event)

[0494] 아래와 같은 정보가 바이러스 이벤트에 포함될 수 있다.

[0495] - event_Name : 'SEC_EVENT_VIRUS'

[0496] - virus_type : 바이러스 유형 (예 : 트로이 목마, 웜, 매크로) 바이러스 유형, 바이러스 이름

[0498] - dst_ip : 바이러스가 발견 된 패킷의 대상 IP 주소

[0499] - src_ip : 바이러스가 발견 된 패킷의 소스 IP 주소

[0500] - src_port : 바이러스가 발견 된 패킷의 소스 포트

[0501] - dst_port : 바이러스가 발견 된 패킷의 대상 포트

[0502] - src_zone : 바이러스가 발견 된 패킷의 소스 보안 영역

[0503] - dst_zone : 바이러스가 발견 된 패킷의 대상 보안 영역

[0504] - file_type : 바이러스가 숨겨진 파일의 유형

[0505] - file_name : 바이러스가 숨겨진 파일의 이름

[0506] - virus_info : 바이러스의 간단한 소개

[0507] - raw_info : 이벤트를 트리거하는 패킷을 설명하는 정보.

[0508] - rule_id : 트리거되는 규칙의 ID입니다.

[0509] - rule_name : 트리거되는 규칙의 이름

[0510] - 프로필 : 트래픽이 일치하는 보안 프로필입니다.

[0512] 침입 이벤트(Intrusion Event)

[0513] - Intrustion Event에는 다음 정보가 포함되어야합니다.

[0514] - event_name: 이벤트 이름 : 'SEC_EVENT_Intrusion'

[0515] - sub_attack_type: 공격 유형, 예 : 잔인한 힘, 버퍼 오버 플로우

[0516] - src_ip: 패킷의 소스 IP 주소

[0517] - dst_ip: 패킷의 목적지 IP 주소

[0518] - src_port: 패킷의 소스 포트 번호

[0519] - dst_port : 패킷의 목적지 포트 번호

[0520] - src_zone: 패킷의 소스 보안 영역

[0521] - dst_zone: 패킷의 대상 보안 영역

[0522] - 프로토콜: 사용 된 전송 계층 프로토콜, 예를 들어, TCP, UDP

[0523] - app: 채용 된 애플리케이션 계층 프로토콜 (예를 들면: HTTP, FTP)

[0524] - rule_id: 트리거 되는 규칙의 ID입니다.

[0525] - rule_name: 트리거 되는 규칙의 이름

[0526] - 프로필: 트래픽이 일치하는 보안 프로필

[0527] - intrusion_info: 침입에 대한 간단한 설명

[0528] - raw_info: 이벤트를 트리거 하는 패킷을 설명하는 정보.

[0529] 봇넷 이벤트(Botnet Event)

[0530] 아래와 같은 정보는 봇넷 이벤트에 포함될 수 있다.

[0531] - event_name : 이벤트 이름 : 'SEC_EVENT_Botnet'

[0532] - botnet_name : 탐지 된 봇넷의 이름

[0533] - src_ip : 패킷의 소스 IP 주소

[0534] - dst_ip : 패킷의 목적지 IP 주소

[0535] - src_port : 패킷의 소스 포트 번호

[0536] - dst_port : 패킷의 목적지 포트 번호

[0537] - src_zone : 패킷의 소스 보안 영역

[0538] - dst_zone : 패킷의 대상 보안 영역

[0539] - 프로토콜 : 사용 된 전송 계층 프로토콜, 예를 들어, TCP, UDP

[0540] - app : 채용 된 애플리케이션 계층 프로토콜 (예 : HTTP, FTP)

[0541] - 역할 : 봇넷 내 통신 당사자의 역할 :

[0542] 1. 좀비 호스트에서 공격자까지의 패킷

[0543] 2. 공격자에서 좀비 호스트로 가는 패킷

[0544] 3. IRC / WEB 서버에서 좀비 호스트로 가는 패킷

[0545] 4. 좀비 호스트에서 IRC / WEB 서버로 보내는 패킷

[0546] 5. 공격자에서 IRC / WEB 서버로 보낸 패킷

[0547] 6. IRC / WEB 서버에서 공격자로 가는 패킷

[0548] 7. 좀비 호스트에서 희생자까지의 패킷

[0549] - botnet_info : Botnet에 대한 간단한 설명

[0550] - rule_id : 트리거 되는 규칙의 ID입니다.

[0551] - rule_name : 트리거 되는 규칙의 이름

[0552] - 프로필: 트래픽이 일치하는 보안 프로필

[0553] - raw_info : 이벤트를 트리거 하는 패킷을 설명하는 정보

[0554] 웹 공격 이벤트(Web Attack Event)

[0555] 아래와 같은 정보가 웹 공격 이벤트에 포함될 수 있다.

[0556] - event_name : 이벤트 이름 : 'SEC_EVENT_WebAttack'

[0557] - sub_attack_type : 구체적인 웹 공격 유형 (예 : sql injection, command injection, XSS, CSRF)

[0558] - src_ip : 패킷의 소스 IP 주소

[0559] - dst_ip : 패킷의 목적지 IP 주소

[0560] - src_port : 패킷의 소스 포트 번호

[0561] - dst_port : 패킷의 목적지 포트 번호

[0562] - src_zone : 패킷의 소스 보안 영역

[0563] - dst_zone : 패킷의 대상 보안 영역

[0564] - req_method : 요구 사항의 방법. 예를 들어 HTTP에서 'PUT'또는 'GET'

[0565] - req_url : 요청 된 URL

[0566] - url_category : 일치하는 URL 카테고리

[0567] - filtering_type : 블랙리스트, 허용 목록, 사용자 정의, 미리 정의된, 악의적인 카테고리, 알 수없는 URL 필터링 유형

[0568] - rule_id : 트리거되는 규칙의 ID입니다.

[0569] - rule_name : 트리거되는 규칙의 이름

[0570] - 프로필 : 트래픽이 일치하는 보안 프로필입니다.

[0572] NSF 로그(NSF Logs)

[0573] DDoS 로그(DDoS Logs)

[0574] DDoS 경보의 필드 외에도 필드 외에도 아래와 같은 정보가 DDoS 로그에 포 함될 수 있다.

[0575] - 공격 유형 : DDoS

[0576] - attack_ave_rate : 기록 된 시간 내에 공격 트래픽의 평균 pps

[0577] - attack_ave_speed : 기록 된 시간 내에 공격 트래픽의 평균 bps

[0578] - attack_pkt_num : 기록 된 시간 내의 공격 패킷 수

[0579] - attack_src_ip : 공격 트래픽의 소스 IP 주소입니다. 많은 양의 IP 주소가 있는 경우 다른 규칙에 따라 특정 수의 자원을 선택.

[0580] - 액션 : DDoS 공격 (예를 들면: 허용, 경고, 차단, 폐기, 선언, 차단 IP, 차단 서비스)에 대한 작업.

[0582] 바이러스 로그(Virus Logs)

[0583] 바이러스 경보의 필드 외에도 아래와 같은 정보가 바이러스 로그에 포함될 수 있다,

[0584] - 공격 유형 : 바이러스

[0585] - 프로토콜 : 전송 계층 프로토콜

[0586] - app : 응용 프로그램 계층 프로토콜의 이름

[0587] - times : 바이러스 탐지 시간

[0588] - 액션 : 바이러스를 다루는 액션 (예 : 경고, 차단)

[0589] - os : 바이러스가 영향을 미치는 OS (예 : all, android, ios, unix, windows).

[0591] 침입 로그(Intrusion Logs)

[0592] 침입 경보의 필드 외에도 아래와 같은 정보가 침입 로그에 포함도리 수 있다.

[0593] - 공격 유형 : 침입

[0594] - 시간 : 기록된 시간에 침입 시간이 발생했습니다.

[0595] - os : 침입에 영향을 주는 OS입니다 (예 : all, android, ios, unix, windows).

[0596] - 액션 : 침입을 다루는 액션들, 예를 들어 허용, 경고, 차단, 폐기, 선언, 차단 -IP, 차단 - 서비스

[0597] - attack_rate : 공격 트래픽의 pps NUM

[0598] - attack_speed : NUM 공격 트래픽의 bps

[0599] 봇넷 로그(Botnet Logs)

[0600] Botnet Alarm의 필드 외에도 아래와 같은 정보가 봇넷 로그에 포함될 수 있다.

[0601] - attack_type : 봇넷

[0602] - botnet_pkt_num : 탐지된 봇넷으로 보내거나 받은 패킷 수

[0603] - 액션 : 탐지된 패킷을 처리하는 액션 (예 : 허용, 경고, 차단, 폐기, 선언, 차단 IP, 차단 서비스, 기타

[0604] - os : 공격 대상인 모든 OS, 예를 들어, android, ios, unix, windows 등

[0605] DPI 로그(DPI Logs)

[0606] DPI 로그는 업로드 및 다운로드 된 파일 및 데이터, 전송 및 수신 된 전자 메일에 대한 통계를 제공하고 웹 사

이트에 기록을 경고하고 차단할 수 있다.

[0607] - 유형 : DPI 작업 유형. 예 : 파일 차단, 데이터 필터링, 애플리케이션 동작 제어

[0608] - file_name : 파일 이름

[0609] - file_type : 파일 형식

[0610] - src_zone : 트래픽 소스 보안 영역

[0611] - dst_zone : 트래픽의 대상 보안 영역

[0612] - src_region : 트래픽 소스 영역

[0613] - dst_region : 트래픽의 대상 영역

[0614] - src_ip : 트래픽 소스 IP 주소

[0615] - src_user : 트래픽을 생성 한 사용자

[0616] - dst_ip : 트래픽의 대상 IP 주소

[0617] - src_port : 트래픽 소스 포트

[0618] - dst_port : 트래픽의 대상 포트

[0619] - 프로토콜 : 트래픽의 프로토콜 유형

[0620] - 앱 : 트래픽의 애플리케이션 유형

[0621] - policy_id : 트래픽이 일치하는 보안 정책 ID

[0622] - policy_name : 트래픽이 일치하는 보안 정책 이름

[0623] - 동작 : 트래픽이 일치하는 파일 차단 규칙, 데이터 필터링 규칙 또는 응용 프로그램 동작 제어 규칙에 정의된 작업이다.

[0625] Vulnerabillity 검색 로그

[0626] 취약점 검색 로그에는 피해 호스트 및 관련 취약점 정보가 기록되어야 한다. 다음 정보가 보고서에 포함 되어야 합니다.

[0627] - victim_ip : 취약성이 있는 희생 된 호스트의 IP 주소

[0628] - 취약점 ID : 취약점 ID

[0629] - vulnerability_level : 취약점 수준. 예 : 높음, 낮음, 낮음

[0630] - 운영 체제 : 대상 호스트의 운영 체제

[0631] - 서비스 : 피해자 호스트에 취약성이 있는 서비스

[0632] - protocol : 프로토콜 유형. 예 : TCP, UDP

[0633] - port : 포트 번호

[0634] - vulnerability_info : 취약점에 대한 정보

[0635] - fix_suggestion : 취약점에 대한 수정 제안.

[0636] - 8.6.7. 웹 공격 로그

[0637] - 웹 공격 경보의 필드 외에도 다음 정보가 웹 공격 보고서에 포함되어야 한다.

[0638] - attack_type : 웹 공격

[0639] - rsp_code : 응답 코드

[0640] - req_clientapp : 클라이언트 응용 프로그램

[0641] - req_cookies : 쿠키

[0642] - req_host : 요청한 호스트의 도메인 이름

[0643] - raw_info : 이벤트를 트리거 하는 패킷을 설명하는 정보.

[0645] NSF 카운터(NSF Counter)

[0646] 방화벽 카운터(Firewall Counters)

[0647] 방화벽 카운터는 트래픽 서명, 대역폭 사용 및 구성된 보안 및 대역폭 정책이 어떻게 적용되었는지에 대한 가시성을 제공합니다.

[0648] - src_zone : 트래픽 소스 보안 영역

[0649] - dst_zone : 트래픽의 대상 보안 영역

[0650] - src_region : 트래픽 소스 영역

[0652] - dst_region : 트래픽의 대상 영역

[0653] - src_ip : 트래픽 소스 IP 주소

[0654] - src_user : 트래픽을 생성 한 사용자

[0655] - dst_ip : 트래픽의 대상 IP 주소

[0656] - src_port : 트래픽 소스 포트

[0657] - dst_port : 트래픽의 대상 포트

[0658] - 프로토콜 : 트래픽의 프로토콜 유형

[0659] - 앱 : 트래픽의 애플리케이션 유형

[0660] - policy_id : 트래픽이 일치하는 보안 정책 ID

[0661] - policy_name : 트래픽이 일치하는 보안 정책 이름

[0662] - in_interface : 트래픽의 인바운드 인터페이스

[0663] - out_interface : 트래픽의 아웃 바운드 인터페이스

[0664] - total_traffic : 총 트래픽 양

[0665] - in_traffic_ave_rate : 인바운드 트래픽 평균 요금 (pps)

[0666] - in_traffic_peak_rate : 인바운드 트래픽 피크 속도 (pps)

[0667] - in_traffic_ave_speed : 인바운드 트래픽 평균 속도 (bps)

[0668] - in_traffic_peak_speed : 인바운드 트래픽 최고 속도 (bps)

[0669] - out_traffic_ave_rate : 아웃 바운드 트래픽 평균 요금 (pps)

[0670] - out_traffic_peak_rate : 아웃 바운드 트래픽 피크 속도 (pps)

[0671] - out_traffic_ave_speed : 아웃 바운드 트래픽 평균 속도 (bps)

[0673] 정책 방문 횟수 카운터(Policy Hit Counters)

[0674] 정책 적중 카운터는 트래픽이 일치하는 보안 정책과 적중 횟수를 기록합니다. 정책 구성이 올바른지 확인할 수

있습니다.

[0675] - src_zone : 트래픽 소스 보안 영역

[0677] - dst_zone : 트래픽의 대상 보안 영역

[0678] - src_region : 트래픽 소스 영역

[0679] - dst_region : 트래픽의 대상 영역

[0680] - src_ip : 트래픽 소스 IP 주소

[0681] - src_user : 트래픽을 생성 한 사용자

[0682] - dst_ip : 트래픽의 대상 IP 주소

[0683] - src_port : 트래픽 소스 포트

[0684] - dst_port : 트래픽의 대상 포트

[0685] - 프로토콜 : 트래픽의 프로토콜 유형

[0686] - 앱 : 트래픽의 애플리케이션 유형

[0687] - policy_id : 트래픽이 일치하는 보안 정책 ID

[0688] - policy_name : 트래픽이 일치하는 보안 정책 이름

[0689] - hit_times: 보안 정책이 지정된 트래픽과 일치하는 횟수.

[0690] 도 20a 내지 도 20j는 본 명세서의 일 실시 예에 따른 NSF 모니터링을 위한 데이터 모델을 예시한다.

[0691] 도 20a 내지 도 20j를 참조하면, 앞에서 살펴본 NSF를 모니터링하기 위한 정보 모델을 이용하여 데이터 모델을 설계할 수 있다.

[0692] 도 21a 내지 도 22i는 본 명세서의 일 실시 예에 따른 모니터링을 위한 YANG 데이터 모델을 예시한다.

[0693] NSF 모니터링이 포괄적인 방법으로 수행되는 경우, 악의적인 활동, 비정상적인 행동 또는 잠재적 인 서비스 거부 공격의 징후를 적시에 감지 할 수 있다. 이러한 모니터링 기능은 앞에서 살펴본 NSF가 생성한 모니터링 정보를 기반으로 합니다.

[0694] 따라서, 본 명세서는 NSF 모니터링을 위한 정보 모델을 지정하는 데이터 모델 구조 트리뿐만 아니라 NSF 모니터링을 위한 해당 YANG 데이터 모델을 설계하는 방법을 제안한다.

[0695] 도 21a 내지 도 22i를 참조하면 앞에서 살펴본 NSF 모니터링을 위한 정보 모델 및 데이터 모델을 이용하여 해당 YANG 데이터 모델을 설계할 수 있다.

[0697] **소비자-직면 인터페이스(Consumer-facing interface)를 위한 보안 정책의 데이터 모델**

[0698] 소비자-직면 인터페이스를 위한 보안 정책의 데이터 모델의 목적은 I2NSF 사용자의 고수준 보안 정책에 대한 I2NSF 사용자와 보안 제어기 사이의 소비자-직면 인터페이스를 통해, 제어 및 관리 메시지를 전달하는데 사용할 수 있는 YANG 데이터 모델로 정보 모델(client-facing-inf-im)을 변환하는 것이다.

[0699] 따라서 이러한 데이터 모델은 소비자-직면 인터페이스의 정보 모델과 일치해야 한다. YANG 데이터 모델은 제어 또는 관리 메시지의 효율적인 전달을 위해. 상기 정보 모델의 변환을 실시할 수 있다.

[0700] 상기 데이터 모델은 보안요구에 따라 확장될 수 있는 I2NSF 시스템을 지원하도록 설계되었다. 즉, 상기 데이터 모델의 설계는 특정 정책들, 구현방식에 독립적이다.

[0702] 도 22는 본 명세서가 작용될 수 있는 소비자-직면 인터페이스를 위한 고수준 추출의 예시이다.

[0703] 도 22를 참조하면, 멀티-테넌시(Multi-tenancy)는 어플리케이션 리소스를 관리하기 위한 여러 관리 도메인을 허

용할 수 있다. Enterprise group은 HR, 재무 및 법률과 같은 여러 테넌트 또는 부서를 포함할 수 있다. 따라서, 자체적인 보안 정책을 관리하고자 하는 조직의 사용자에게 할당될 수 있는 권한(permission)들의 집합을 정의하기 위한 객체가 요구된다. 이는 정책의 사용자들에게 조직 내의 작업 기능 또는 권한들의 집합을 할당하는 작업을 의미할 수 있다. 정책-역할(role)의 객체는 보안 정책 관리 권한에 관한 권한을 부여하거나, 거부하기 위해 이름, 날짜 및 엑세스 프로파일을 포함한다.

[0705] 도 22a 내지 도 22d는 본 명세서의 일 실시 예에 따른 소비자-직면 인터페이스를 위한 보안 정책의 데이터 모델을 예시한다. 도 32a 내지 도 32d를 참조하면, 전술한 소비자-직면 인터페이스를 위한 보안 정책의 데이터 모델을 설계할 수 있다.

[0707] 도 23은 본 명세서의 일 실시예에 따른 소비자-직면 인터페이스의 보안 정책을 위한 YANG 데이터 모델을 예시한다. 도 23를 참조하면, 전술한 소비자-직면 인터페이스의 보안 정책을 위한 YANG 데이터 모델을 설계할 수 있다.

[0709] 또한, 본 명세서는 I2NSF(Interface to Network Security Functions) 시스템에서 보안 정책 변환기의 설계를 제안한다.

[0710] I2NSF 사용자는 NSF(Network Security Functions)에 대한 전반적인 지식이 없어도 NSF를 사용할 수 있어야 한다. 일반적으로 I2NSF 사용자로부터 생성되는 정책은 사용자가 정책을 생성할 때, NSF의 속성을 고려하지 않기 때문에, 추상적인 데이터를 포함하고 있다. 따라서 I2NSF 시스템은 보안 컨트롤러는 사용자로부터 보안 정책을 수신하는 경우, 정책에 필요한 NSF를 자동으로 찾아서 선택한 NSF에 대해 번역하는 번역기를 필요로 한다. 본 명세서에서는 오토마타(Automata) 이론을 통해 번역기를 모듈화하는 모델을 제안한다. 보다 자세하게 본 명세서는 I2NSF를 통해 사용자는 고수준 보안 정책을 생성하여, 보안 컨트롤러에 전달하고, 보안 컨트롤러의 보안 정책 번역기는 대상 NSF를 검색하며, 검색한 NSF 각각에 대응하는 하위수준의 정책으로 변환함으로써, I2NSF에 관련된 모든 작업 기능을 설정하는 모델을 제안한다.

[0711] 이를 통해, 본 명세서는 I2NSF 사용자와 시스템 관리자 모두에게 네트워크 보안 기능 이용의 편리함을 제공할 수 있으며, I2NSF 사용자는 NSF의 기능을 알고 있을 필요가 없으며, 시스템 관리자는 정책 변환 프로세스를 유연하게 관리할 수 있다.

[0713] **정책 전달자(Translator)의 필요성**

[0714] 도 24 및 도 25는 본 명세서가 적용될 수 있는 보안 정책의 예시이다.

[0715] 예를 들어, 특정 악성 웹사이트를 차단하기 위한 정책으로서,

[0716] - 아들의 컴퓨터를 악성 웹사이트로부터 차단(도 24)

[0717] - IP 주소 10.0.0.1 및 10.0.0.3에서 www.malicious.com 및 www.illegal.com으로 패킷을 삭제(도 25)

[0718] 라는 정책이 있는 경우, 상기 2가지 정책은 동일한 동작을 요구하지만, 웹 사이트 차단 기능을 지원하는 웹 필터 NSF의 동작을 위해서는 최소한 소스-IP 주소와 웹 사이트 주소가 필요하므로, NSF는 첫번째 정책에 적합한 동작이 불가능하다.

[0719] 반면에, 일반 사용자가 두번째 정책을 생성하기 위해서는 정책 설정을 위한 웹 필터 NSF를 사용하고, NSF에 소스-IP 주소와 웹 사이트 주소가 전달되어야 한다는 것을 알고 있는 경우에만 가능하다. 따라서, 일반 사용자는 NSF를 전문적으로 이해하고 있어야만 한다.

[0720] 따라서, 일반 사용자는 첫번째 정책을 생성하는 반면, NSF는 두번째 정책을 필요로 할 것이다. 본 명세서에서는 첫번째 정책과 같이, 추상적인 데이터를 포함하며, NSF의 특별한 지식이 요구되지 않는 정책을 고수준 보안 정책(또는 고수준 정책)이라 한다. 이와 달리, NSF에서 요구되는 상세한 데이터를 포함하며, 네트워크 운영에 필요한 모든 기능을 설정할 수 있는 정책을 저수준 보안 정책(저수준 정책)이라 한다.

[0721] 따라서, I2NSF 시스템은 고수준 보안 정책을 저수준 보안 정책으로 변환할 수 있어야 하며, 또한 I2NSF 시스템은 고수준 보안 정책을 적용하기 위해, 적절한 NSF를 자동으로 찾을 수 있어야 한다.

[0723] **정책 적용 절차(Process of Applying Policy)**

[0724] 도 26은 본 명세서가 적용될 수 있는 정책 적용을 위한 절차의 예시이다.

[0725] 개발자 관리 시스템은 I2NSF 시스템에 제공되는 NSF의 모든 기능들을 보안 제어기(Security controller)에 등록한다(S2610).

[0726] I2NSF 사용자는 고수준 보안 정책을 생성하고, I2NSF 사용자는 소비자-직면 인터페이스를 이용하여, 보안 제어기에 고수준 보안 정책을 전달한다(S2620).

[0727] 보안 제어기는 등록된 NSF의 능력을 비교하여,고수준 보안 정책을 커버할 수 있는 대상(Target) NSF를 검색한다(S2630).

[0728] 보안 제어기는 고수준 보안 정책을 검색된 대상 NSF를 위한 저수준 보안 정책으로 변환하고, NSF-직면 인터페이스를 통해 대상 NSF로 전달한다(S2640). 보다 자세하게는 저수준 보안 정책과 관련된 데이터를 생성하고, 이를 대상 NSF로 전달할 수 있다.

[0729] 보안 제어기는 대상 NSF를 수신한 저수준 보안 정책에 따라 설정한다(S2650). 보다 자세하게는 저수준 보안 정책과 관련된 데이터에 근거하여, 대상 NSF를 설정할 수 있다.

[0730] 상기 보안 제어기의 동작은 사용자의 설정 또는 설계에 따라, 후술할 보안 정책 번역기의 동작에 의해 수행될 수 있다.

[0732] **보안 정책 번역기**

[0733] 도 27은 본 명세서가 적용될 수 있는 보안 정책 번역기 모델의 예시이며, 도 28은 본 명세서가 적용될 수 있는 보안 정책 번역기의 보안 정책 번역을 위한 순서도이다.

[0734] 도 27을 참조하면, 보안 정책 번역기는 추출기, 데이터 변환기 및 생성기를 포함한다. 또한, 보안 정책 번역기는 보안 제어기에 포함될 수 있다.

[0735] I2NSF 사용자가 고수준 정책을 생성하여, 보안 제어기에 전달하는 경우, 보안 제어기는 대상 NSF를 검색하고, 대상 NSF에 대한 저수준 정책으로 변환할 수 있다. 도 28을 참조하여, 보다 자세하게 보안 정책 번역을 위한 방법을 설명하면 다음과 같다.

[0736] I2NSF 사용자로부터 고수준 정책을 수신하면, 정책 번역기의 추출기(Extractor)는 DFA(Deterministic Finite Automation)를 통해 고수준 정책의 데이터를 추출한다(S2810).

[0737] 정책 번역기의 데이터 변환기(Data Converter)는 추출된 데이터를 NSF 필수 데이터(NSF Required Data)로 변환한다(S2820). NSF 필수 데이터로의 변환은 NSF database의 데이터와의 비교를 통해 수행될 수 있다. 보다 자세하게는 추출된 추상 데이터는 NSF database내의 데이터와 매핑과정을 통해, NSF 필수 데이터로 변환될 수 있다. NSF database는 NSF가 기능하기 위한, 필수 데이터를 포함하며, NSF 가 추가 또는 삭제되는 경우, NSF database는 갱신될 수 있다. 이를 통해, I2NSF에서 고수준 정책은 적합한 NSF 를 위한 필수 데이터로 변환될 수 있고, 다양한 NSF들이 추가되거나 삭제되더라도, NSF database를 통해, 동일한 알고리즘으로 고수준 정책의 번역이 수행될 수 있다.

[0738] 정책 번역기의 생성기(Generator)는 NSF 필수 데이터를 이용하여, 대상 NSF를 검색하고, 대상 NSF에 대응되는 저수준 정책을 생성한다(S2830). 보다 자세하게는 저수준 정책을 위한 데이터를 생성할 수 있으며, 저수준 정책을 위한 데이터는 대상 NSF를 위한 구조 또는 내용을 포함할 수 있고, 태그(Tag)를 통해, 그룹화될 수 있다.

[0740] **추출기(Extractor)**

[0741] 도 29는 본 명세서가 적용될 수 있는 추출기 모델의 예시이다. 보다 자세하게는 도 29는 DFA에 기초한 추출기

모델의 예시이다.

[0742] DFA는 유한 상태 머신(Finite State Machine)으로 문자열을 수신하고, 상태전환을 위한 작업을 생성할 수 있다. 보안 제어기가 고수준 정책을 수신한 경우, 추출기는 XML(Extensible Markup Language) 태그에 근거한 상태 전환에 의해 데이터를 추출할 수 있다. 고수준 정책의 상태 전환이 이루어진 경우, 모든 데이터는 고수준 정책으로부터 자동적으로 추출될 수 있다.

[0743] 추출기의 DFA 구조는 소비자-직면 데이터 모델의 계층을 따르므로, I2NSF 관리 시스템은 소비자-직면 인터페이스의 데이터 모델을 참조하여 DFA를 쉽게 생성할 수 있다. 만일, 소비자-직면 인터페이스의 데이터 모델이 수정될 경우, 관리 시스템은 추출기의 DFA만을 변경함으로써 수정된 데이터 모델을 적용할 수 있다.

[0745] **데이터 변환기(Data Converter)**

[0746] 도 30은 본 명세서에 적용될 수 있는 데이터 변환기 모델의 예시이다. 데이터 변환기는 NSF 능력(capabilities)과 호환되도록 데이터를 지정할 수 있다. 사용자가 지정되지 않은 데이터가 있는 정책을 NSF에 입력하는 경우, NSF는 데이터를 정상적으로 인식할 수 없다. 예를 들어, "아들 컴퓨터"의 데이터가 NSF에 전송되는 경우, 이는 IP 주소 또는 이와 유사한 것으로 지정되지 않은 데이터이므로 NSF는 이를 인식할 수 없다. NSF가 이해하기 위해서는 일반적으로, 추상 데이터는 NSF 능력에 적합한 지정된 데이터로 변환되어야 한다. 이를 위해, 데이터 변환기는 NSF 능력이 포함된 데이터베이스의 데이터와 비교를 통해, 이와 동등한 데이터로 변환할 필요가 있다. 도 30은 이러한 데이터베이스의 데이터와의 비교에 근거한, 데이터 변환 과정을 예시한다. 도 30을 참조하면, 추상 데이터인 'Son'은 사용자 데이터 베이스의 IP 목록과 비교를 통해, 특정 IP 주소[10.0.0.1, 10.0.0.3]로 매핑될 수 있다. 데이터 변환기를 통해, 고수준 정책의 모든 데이터는 NSF 능력과 호환되는 동등한 지정된 데이터로 변환될 수 있다.

[0748] **생성기(Generator)**

[0749] 생성기는 대상 NSF를 자동으로 검색할 수 있고, 각 대상 NSF을 위한 저수준 정책을 생성할 수 있다.

[0750] 먼저, 생성기는 고수준 정책의 모든 기능을 커버할 수 있는 NSF를 검색할 수 있다. 생성기는 개발자 관리 시스템에 등록된 NSF 기능을 비교하여 대상 NSF를 검색한다. 이러한 프로세스는 생성기가 오직 정책만을 이용하여, 적절한 NSF를 찾기 때문에, 프로비저닝(provisioning)된 정책에 의해 호출될 수 있다. 만일, 사용자 정책에 포함되지 않은 다른 데이터를 이용하여 대상 NSF를 찾을 경우, 이는 사용자가 I2SNF 시스템의 NSF에 대한 지식을 알고 있음을 의미할 수 있다. 도 31은 본 명세서가 적용될 수 있는 정책 프로비저닝의 예시이다. 도 31을 참조하면, 생성기는 정책의 기능들을 커버하기 위해, 방화벽(Firewall) NSF 및 웹-필터 NSF를 선택한다.

[0751] 다음으로, 생성기는 추출된 데이터를 사용하여 각 대상 NSF에 대해 저수준 정책을 만들 수 있다. 생성기는 Context-free 문법(Grammar)을 이용하여, 구성될 수 있다. Context-free 문법은 주어진 형식 언어로 가능한 모든 문자열을 설명할 수 있는 프로덕션(production) 규칙의 집합을 의미한다. 저수준의 정책은 또한 NSF-직면 인터페이스의 YANG 데이터 모델에 기반한 자체 언어를 가질 수 있다. 따라서 이러한 프로덕션은 YANG 데이터 모델을 기반으로 구성될 수 있다. 상기 프로덕션은 "컨텐츠 프로덕션"과 "구조 프로덕션"을 포함한다. “컨텐츠 프로덕션”은 적절한 XML 태그에 데이터를 포함시키기 위한 것이다. “구조 프로덕션”은 다른 태그를 그룹화하기 위한 것이다. 도 25를 참조하면, 저수준의 정책이 크게 두 가지 유형의 태그로 구성되어 있음을 알 수 있다. 데이터가 들어 있는 태그는 컨텐츠 프로덕션으로 만들 수 있고, 다른 태그를 묶는 태그는 구조 프로덕션으로 만들 수 있다.

[0752] 컨텐츠 프로덕션은 예를 들어 표 2과 같이 표현될 수 있다.

표 2

$$[content] \rightarrow [content][content] \quad (1)$$

$$[content] \rightarrow \langle content - tag \rangle [data] \langle /content - tag \rangle \quad (2)$$

$$[data] \rightarrow data : 1 | data : 2 | ... | data : n \quad (3)$$

[0753]

[0754] 표 2를 참조하면, 대괄호는 상태를 의미한다. 보다 자세하게는 대괄호가 없는 경우, 문자열은 완전히 생성됨을 의미한다. 태그 복제가 허용되는 경우, 프로덕션 규칙에 첫 번째 프로덕션(1)을 추가할 수 있다. 이것은 선택적인 프로덕션이므로, 복제를 허용할 필요가 없다면, 생략될 수 있다. 프로덕션(2)은 "콘텐츠 프로덕션"의 주 프로덕션이다. 데이터가 들어 있는 태그는 프로덕션(2)을 통해 생성할 수 있다. 프로덕션(3)은 태그에 데이터를 주입하기 위한 것이다. NSF에 대한 데이터가 변경되면, I2NSF 관리 시스템에게는 각 NSF에 대한 데이터 매핑을 위해, 프로덕션(3)만의 변경이 요구된다. 예를 들어, 소스-IP 주소에 대한 저수준 정책을 도 25에서 표현하고자 할 경우, 다음의 표 3과 같이 "컨텐츠 프로덕션"을 구성할 수 있다.

표 3

$$[ip - content] \rightarrow [ip - content][ip - content] \quad (4)$$

$$[ip - content] \rightarrow \langle ipv4 \rangle [data] \langle /ipv4 \rangle \quad (5)$$

$$[data] \rightarrow 10.0.0.1 | 10.0.0.3 \quad (6)$$

[0755]

[0756] "구조 프로덕션"은 예를 들어 다음의 표 4와 같이 표시될 수 있다.

표 4

$$[struct] \rightarrow \langle start - tag \rangle [state : 1] ... [state : n] \langle /end - tag \rangle \quad (7)$$

$$[state : x] \, can \, be \, [struct] \, or \, [content] \quad (8)$$

[0757]

[0758] 프로덕션 (7)은 다른 태그를 자체 태그 이름으로 그룹화하는 것을 의미한다. 그리고 프로덕션 (8)은 프로덕션 (7)에 묶인 여러 상태를 구조 상태 또는 내용 상태로 이전하는 것을 나타낸다. 예를 들어, 도 25에서 I2NSF 태그에 대한 저수준의 정책을 표현하려면 표 5와 같이 "구조 프로덕션"을 구성할 수 있다.

표 5

$$[i2nsf] \rightarrow \langle I2NSF \rangle [rule - name][rules] \langle /I2NSF \rangle \quad (9)$$

$$[rule - name] \, is \, for \, Content \, Production \quad (10)$$

$$[rules] \, is \, for \, Structure \, Production \quad (11)$$

[0759]

[0760] 도 32는 본 명세서가 적용될 수 있는 NSF-직면 인터페이스 YANG 데이터 모델에 근거한 트리구조의 예시이다.

I2NSF 관리 시스템이 Context-free 문법을 기반으로 생성기를 구성하는 경우, 각 대상 NSF는 대상 NSF에 필요한 모든 데이터를 포함하는 저수준 정책을 수신할 수 있다.

[0762] 도 1 내지 도 32에서 설명한 정보 모델, 데이터 모델 및 YANG 데이터 모델은 선택적으로 조합되어 사용될 수 있다.

[0764] 이상에서 설명된 실시 예들은 본 명세서의 구성요소들과 특징들이 소정 형태로 결합된 것들이다. 각 구성요소 또는 특징은 별도의 명시적 언급이 없는 한 선택적인 것으로 고려되어야 한다. 각 구성요소 또는 특징은 다른 구성요소나 특징과 결합되지 않은 형태로 실시될 수 있다. 또한, 일부 구성요소들 및/또는 특징들을 결합하여 본 명세서의 실시 예를 구성하는 것도 가능하다. 본 명세서의 실시 예들에서 설명되는 동작들의 순서는 변경될 수 있다. 어느 실시예의 일부 구성이나 특징은 다른 실시 예에 포함될 수 있고, 또는 다른 실시예의 대응하는 구성 또는 특징과 교체될 수 있다. 특허청구범위에서 명시적인 인용 관계가 있지 않은 청구항들을 결합하여 실시 예를 구성하거나 출원 후의 보정에 의해 새로운 청구항으로 포함시킬 수 있음은 자명하다.

[0765] 본 명세서에 따른 실시 예는 다양한 수단, 예를 들어, 하드웨어, 펌웨어(firmware), 소프트웨어 또는 그것들의 결합 등에 의해 구현될 수 있다. 하드웨어에 의한 구현의 경우, 본 명세서의 일 실시 예는 하나 또는 그 이상의 ASICs(application specific integrated circuits), DSPs(digital signal processors), DSPDs(digital signal processing devices), PLDs(programmable logic devices), FPGAs(field programmable gate arrays), 프로세서, 콘트롤러, 마이크로 콘트롤러, 마이크로 프로세서 등에 의해 구현될 수 있다.

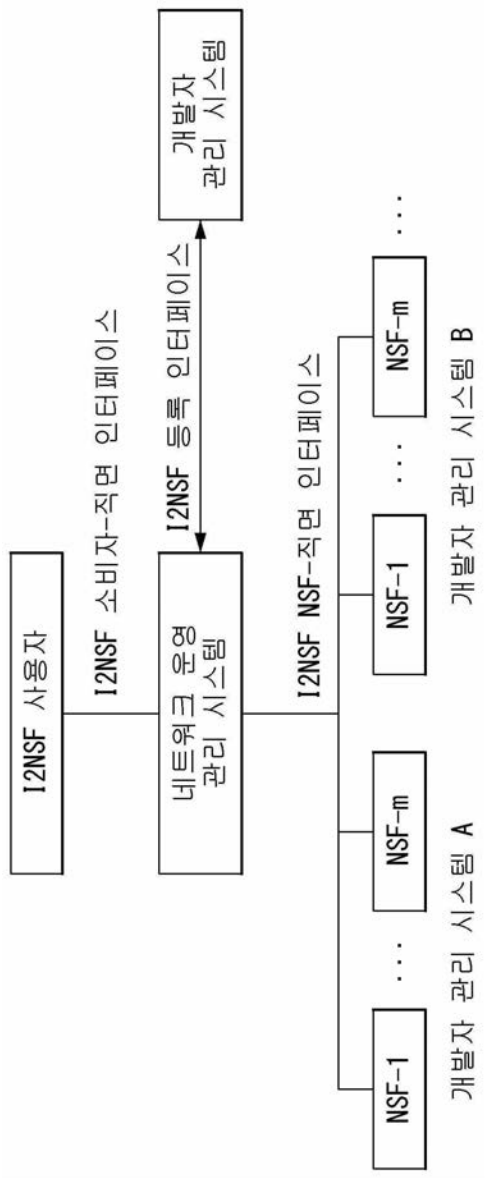
[0766] 펌웨어나 소프트웨어에 의한 구현의 경우, 본 명세서의 일 실시 예는 이상에서 설명된 기능 또는 동작들을 수행하는 모듈, 절차, 함수 등의 형태로 구현될 수 있다. 소프트웨어 코드는 메모리에 저장되어 프로세서에 의해 구동될 수 있다. 상기 메모리는 상기 프로세서 내부 또는 외부에 위치하여, 이미 공지된 다양한 수단에 의해 상기 프로세서와 데이터를 주고 받을 수 있다.

[0767] 본 명세서는 본 명세서의 필수적 특징을 벗어나지 않는 범위에서 다른 특정한 형태로 구체화될 수 있음은 당업자에게 자명하다. 따라서, 상술한 상세한 설명은 모든 면에서 제한적으로 해석되어서는 아니 되고 예시적인 것으로 고려되어야 한다. 본 명세서의 범위는 첨부된 청구항의 합리적 해석에 의해 결정되어야 하고, 본 명세서의 등가적 범위 내에서의 모든 변경은 본 명세서의 범위에 포함된다.

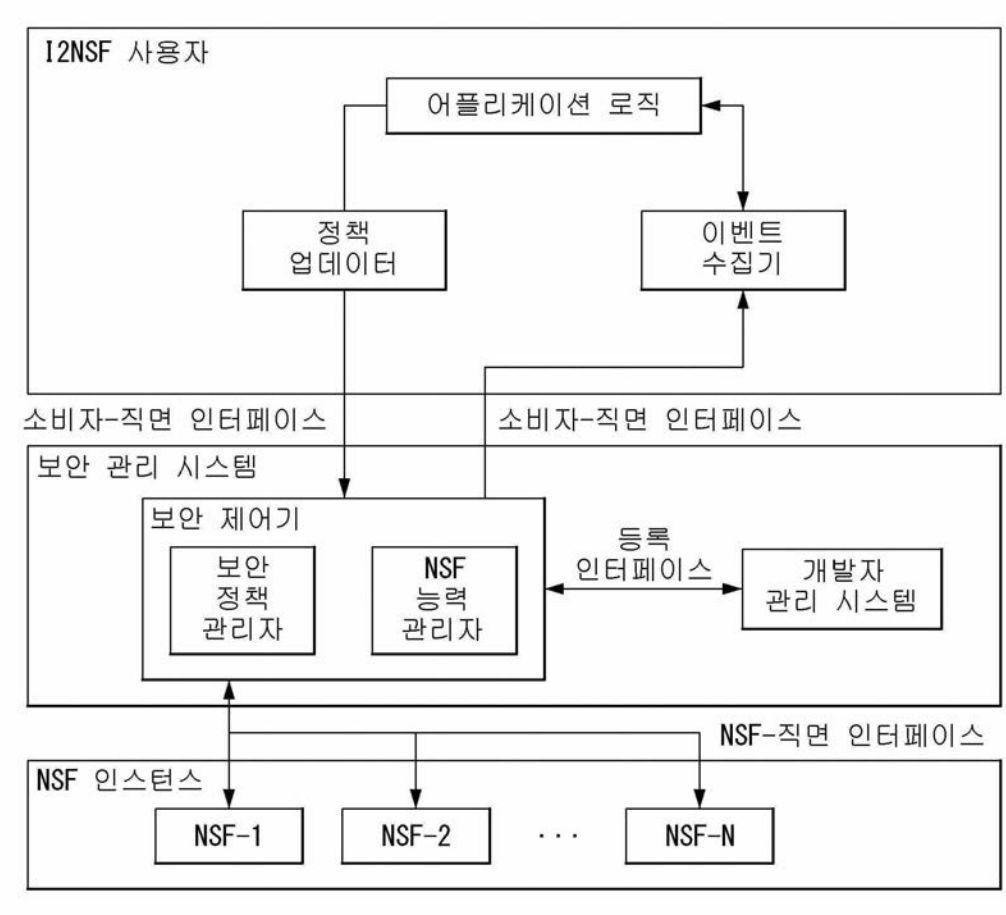
[0768] 본 명세서는 다양한 보안 관리 시스템에 적용될 수 있다.

도면

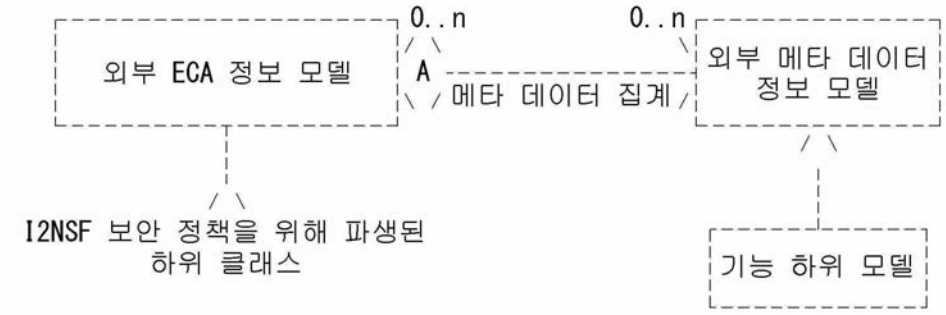
도면1


I2NSF 사용자
I2NSF 소비자-직면 인터페이스
네트워크 운영
관리 시스템
I2NSF 등록 인터페이스
개발자
관리 시스템
I2NSF NSF-직면 인터페이스
NSF-1
NSF-m
개발자 관리 시스템 A
NSF-1
NSF-m
개발자 관리 시스템 B

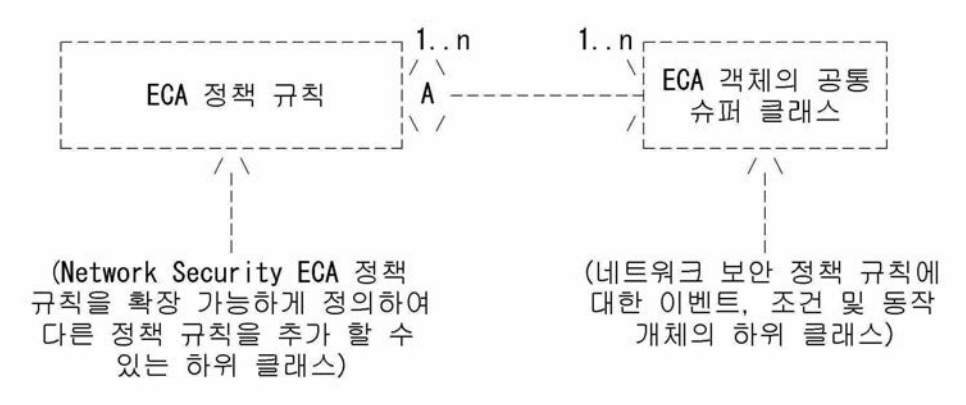
도면2


I2NSF 사용자
어플리케이션 로직
정책
업데이터
이벤트
수집기
소비자-직면 인터페이스
소비자-직면 인터페이스
보안 관리 시스템
보안 제어기
보안
정책
관리자
NSF
능력
관리자
등록
인터페이스
개발자
관리 시스템
NSF-직면 인터페이스
NSF 인스턴스
NSF-1
NSF-2
· · ·
NSF-N

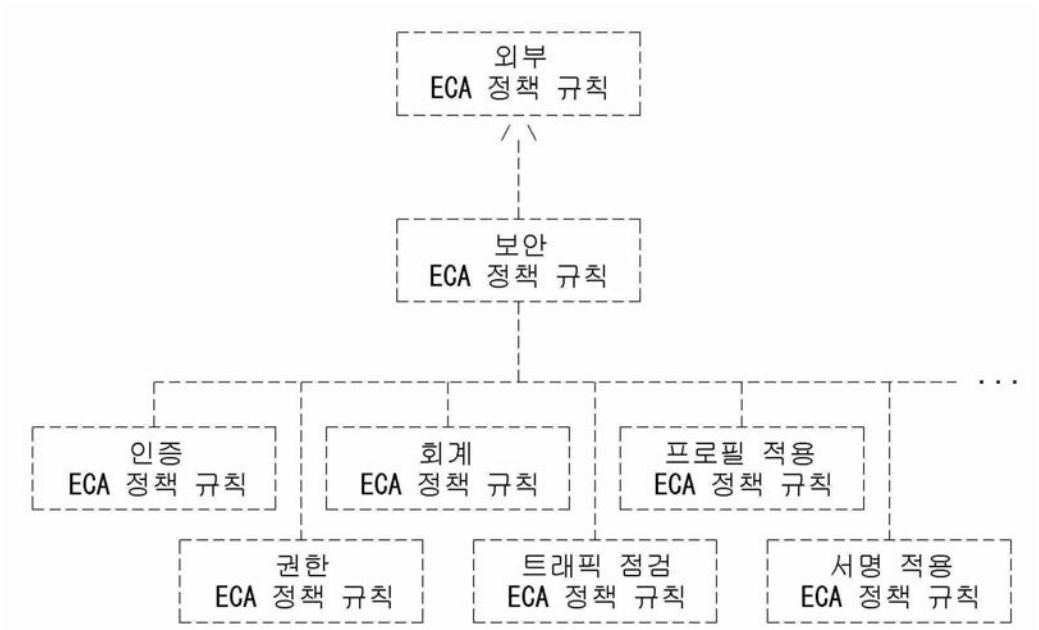
도면3


외부 ECA 정보 모델
0..n
A
메타 데이터 집계
0..n
외부 메타 데이터
정보 모델
I2NSF 보안 정책을 위해 파생된
하위 클래스
기능 하위 모델

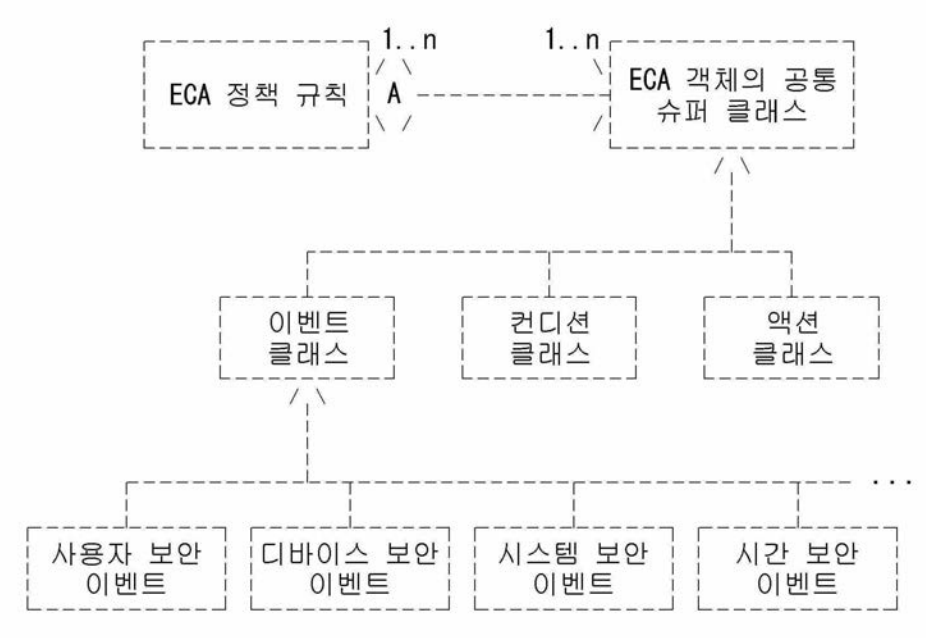
도면4


ECA 정책 규칙
1..n
A
1..n
ECA 객체의 공통 슈퍼 클래스
(Network Security ECA 정책 규칙을 확장 가능하게 정의하여 다른 정책 규칙을 추가 할 수 있는 하위 클래스)
(네트워크 보안 정책 규칙에 대한 이벤트, 조건 및 동작 개체의 하위 클래스)

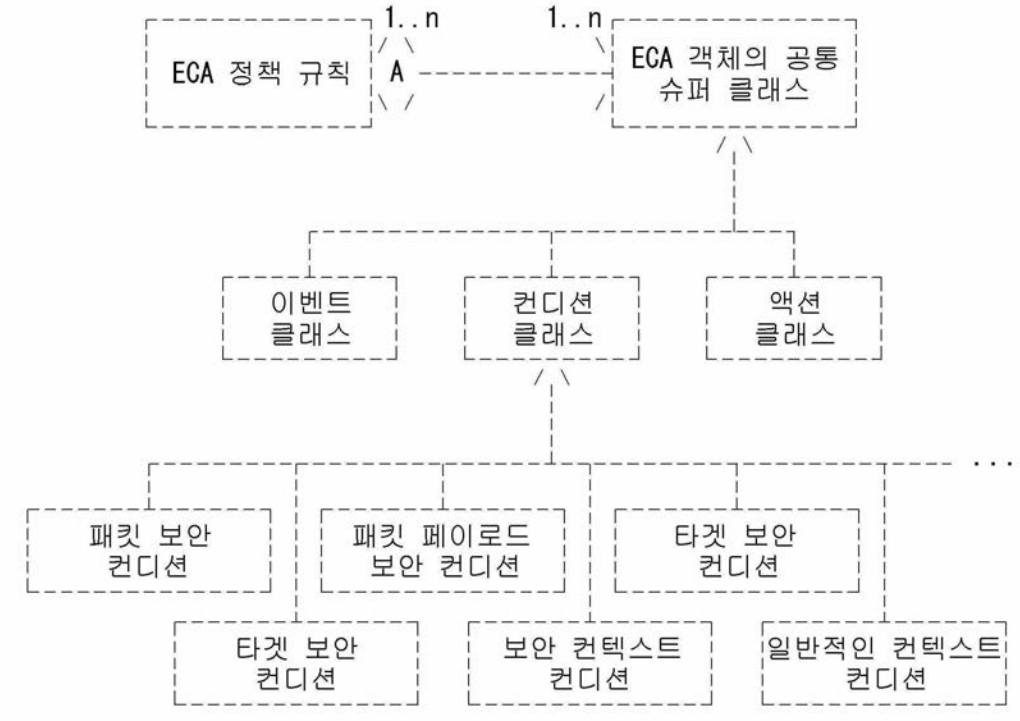
도면5


외부 ECA 정책 규칙
보안 ECA 정책 규칙
...
인증 ECA 정책 규칙
회계 ECA 정책 규칙
프로필 적용 ECA 정책 규칙
권한 ECA 정책 규칙
트래픽 점검 ECA 정책 규칙
서명 적용 ECA 정책 규칙

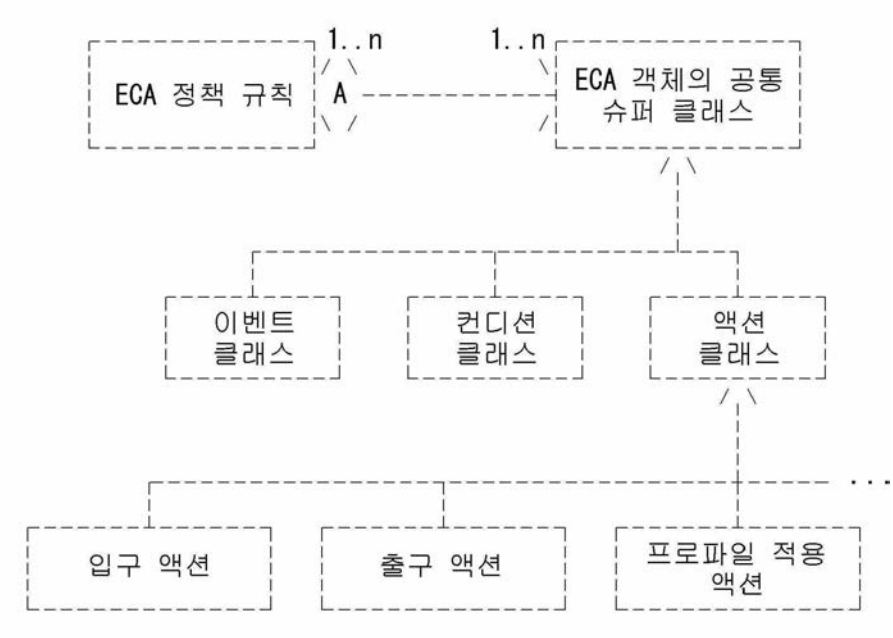
도면6


ECA 정책 규칙
1..n
A
1..n
ECA 객체의 공통
슈퍼 클래스
이벤트
클래스
컨디션
클래스
액션
클래스
사용자 보안
이벤트
디바이스 보안
이벤트
시스템 보안
이벤트
시간 보안
이벤트

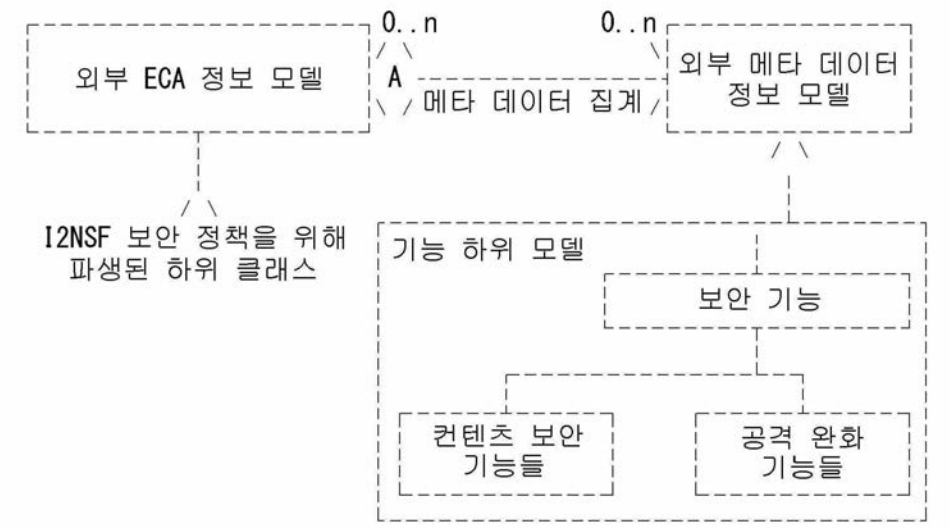
도면7


ECA 정책 규칙
1..n
A
1..n
ECA 객체의 공통
슈퍼 클래스
이벤트
클래스
컨디션
클래스
액션
클래스
패킷 보안
컨디션
패킷 페이로드
보안 컨디션
타겟 보안
컨디션
타겟 보안
컨디션
보안 컨텍스트
컨디션
일반적인 컨텍스트
컨디션

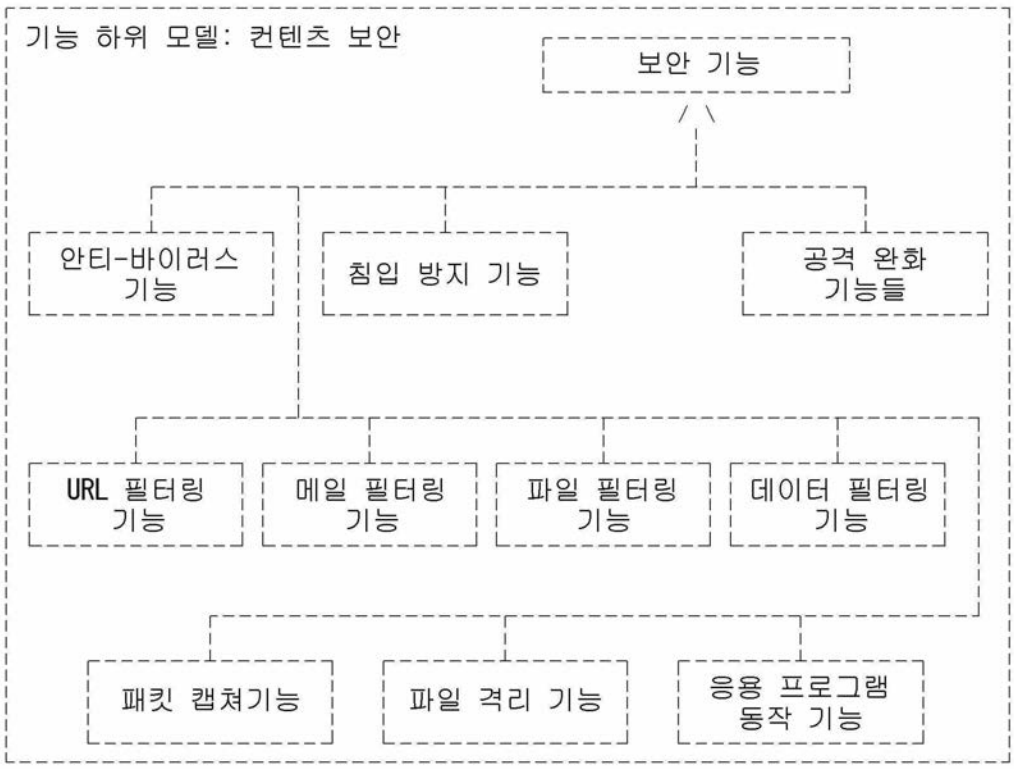
도면8


ECA 정책 규칙
1..n
A
1..n
ECA 객체의 공통 슈퍼 클래스
이벤트 클래스
컨디션 클래스
액션 클래스
...
입구 액션
출구 액션
프로파일 적용 액션

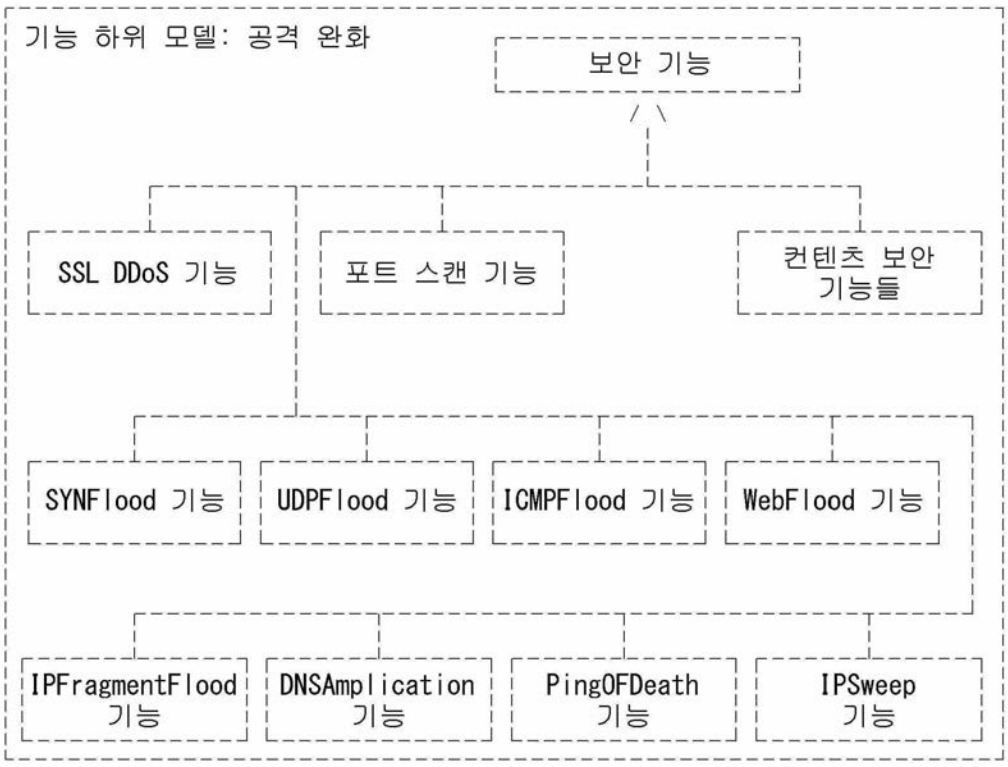
도면9


외부 ECA 정보 모델
0..n
A
메타 데이터 집계
0..n
외부 메타 데이터 정보 모델
I2NSF 보안 정책을 위해 파생된 하위 클래스
기능 하위 모델
보안 기능
컨텐츠 보안 기능들
공격 완화 기능들

도면10


기능 하위 모델: 컨텐츠 보안
보안 기능
안티-바이러스 기능
침입 방지 기능
공격 완화 기능들
URL 필터링 기능
메일 필터링 기능
파일 필터링 기능
데이터 필터링 기능
패킷 캡쳐기능
파일 격리 기능
응용 프로그램 동작 기능

도면11


기능 하위 모델: 공격 완화
보안 기능
SSL DDoS 기능
포트 스캔 기능
컨텐츠 보안 기능들
SYNFlood 기능
UDPFlood 기능
ICMPFlood 기능
WebFlood 기능
IPFragmentFlood 기능
DNSAmplication 기능
PingOFDeath 기능
IPSweep 기능

도면12a

```
module: ietf-i2nsf-policy-rule-for-nsf
+--rw i2nsf-security-policy* [policy-name]
|  +--rw policy-name                string
|  +--rw eca-policy-rules* [rule-id]
|  |  +--rw rule-id                 uint8
|  |  +--rw rule-description?       string
|  |  +--rw rule-rev?               uint8
|  |  +--rw rule-priority?          uint8
|  |  +--rw policy-event-clause-agg-ptr*        instance-identifier
|  |  +--rw policy-condition-clause-agg-ptr*    instance-identifier
```

도면12b

```
|  |  +--rw policy-action-clause-agg-ptr*      instance-identifier
|  |  +--rw time-zone
|  |     +--rw absolute-time-zone
|  |     |  +--rw time
|  |     |  |  +--rw start-time?   yang:date-and-time
|  |     |  |  +--rw end-time?     yang:date-and-time
|  |     |  +--rw date
|  |     |  |  +--rw absolute-date*   yang:date-and-time
|  |     +--rw periodic-time-zone
|  |        +--rw day
|  |        |  +--rw sunday?    boolean
|  |        |  +--rw monday?    boolean
|  |        |  +--rw tuesday?   boolean
|  |        |  +--rw wednesday?   boolean
|  |        |  +--rw thursday?   boolean
|  |        |  +--rw friday?   boolean
|  |        |  +--rw saturday?   boolean
|  |        +--rw month
|  |           +--rw january?   boolean
|  |           +--rw february?   boolean
|  |           +--rw march?   boolean
|  |           +--rw april?   boolean
|  |           +--rw may?   boolean
|  |           +--rw june?   boolean
|  |           +--rw july?   boolean
|  |           +--rw august?   boolean
|  |           +--rw september?   boolean
|  |           +--rw october?   boolean
|  |           +--rw november?   boolean
|  |           +--rw december?   boolean
|  +--rw resolution-strategy
|  |  +--rw (resolution-strategy-type)?
|  |     +--:(fmr)
|  |     |  +--rw first-matching-rule?   boolean
|  |     +--:(lmr)
|  |        +--rw last-matching-rule?    boolean
|  +--rw default-action
|     +--rw default-action-type?   ingress-action
+--rw event-clause-container
|  ...
+--rw condition-clause-container
|  ...
+--rw action-clause-container
   ...
```

도면13

```
module: ietf-i2nsf-policy-rule-for-nsf
+--rw i2nsf-security-policy* [policy-name]
|      ...
|   +--rw eca-policy-rules* [rule-id]
|      ...
|   +--rw resolution-strategy
|      ...
|   +--rw default-action
|      ...
+--rw event-clause-container
|   +--rw event-clause-list* [eca-object-id]
|      +--rw entity-class?         identityref
|      +--rw eca-object-id         string
|      +--rw manual?               string
|      +--rw sec-event-content     string
|      +--rw sec-event-format      sec-event-format
|      +--rw sec-event-type        string
+--rw condition-clause-container
|   ...
+--rw action-clause-container
    ...
```

도면14a

```
module: ietf-i2nsf-policy-rule-for-nsf
+--rw i2nsf-security-policy* [policy-name]
|      ...
|   +--rw eca-policy-rules* [rule-id]
|      ...
|   +--rw resolution-strategy
|      ...
|   +--rw default-action
|      ...
+--rw event-clause-container
```

도면14b

```
|   ...
+--rw condition-clause-container
|   +--rw condition-clause-list* [eca-object-id]
|      +--rw entity-class?                        identityref
|      +--rw eca-object-id                        string
|      +--rw packet-security-condition
|         +--rw packet-manual?                       string
|         +--rw packet-security-mac-condition
|            +--rw pkt-sec-cond-mac-dest*            yang:phys-address
|            +--rw pkt-sec-cond-mac-src*             yang:phys-address
|            +--rw pkt-sec-cond-mac-8021q*           string
|            +--rw pkt-sec-cond-mac-ether-type*      string
|            +--rw pkt-sec-cond-mac-tci*             string
|         +--rw packet-security-ipv4-condition
|            +--rw pkt-sec-cond-ipv4-header-length*     uint8
|            +--rw pkt-sec-cond-ipv4-tos*               uint8
|            +--rw pkt-sec-cond-ipv4-total-length*      uint16
|            +--rw pkt-sec-cond-ipv4-id*                uint8
|            +--rw pkt-sec-cond-ipv4-fragment*          uint8
|            +--rw pkt-sec-cond-ipv4-fragment-offset*   uint16
|            +--rw pkt-sec-cond-ipv4-ttl*               uint8
|            +--rw pkt-sec-cond-ipv4-protocol*          uint8
|            +--rw pkt-sec-cond-ipv4-src*           inet:ipv4-address
|            +--rw pkt-sec-cond-ipv4-dest*          inet:ipv4-address
|            +--rw pkt-sec-cond-ipv4-ipopts?              string
|            +--rw pkt-sec-cond-ipv4-sameip?             boolean
|            +--rw pkt-sec-cond-ipv4-geoip*              string
|         +--rw packet-security-ipv6-condition
|            +--rw pkt-sec-cond-ipv6-dscp*             string
|            +--rw pkt-sec-cond-ipv6-ecn*              string
|            +--rw pkt-sec-cond-ipv6-traffic-class*    uint8
|            +--rw pkt-sec-cond-ipv6-flow-label*       uint32
```

도면14c

```
|      |  |  +--rw pkt-sec-cond-ipv6-payload-length*   uint16
|      |  |  +--rw pkt-sec-cond-ipv6-next-header*      uint8
|      |  |  +--rw pkt-sec-cond-ipv6-hop-limit*        uint8
|      |  |  +--rw pkt-sec-cond-ipv6-src*    inet:ipv6-address
|      |  |  +--rw pkt-sec-cond-ipv6-dest*   inet:ipv6-address
|      |  +--rw packet-security-tcp-condition
|      |  |  +--rw pkt-sec-cond-tcp-src-port*         inet:port-number
|      |  |  +--rw pkt-sec-cond-tcp-dest-port*         inet:port-number
|      |  |  +--rw pkt-sec-cond-tcp-seq-num*       uint32
|      |  |  +--rw pkt-sec-cond-tcp-ack-num*       uint32
|      |  |  +--rw pkt-sec-cond-tcp-window-size*   uint16
|      |  |  +--rw pkt-sec-cond-tcp-flags*         uint8
|      |  +--rw packet-security-udp-condition
|      |  |  +--rw pkt-sec-cond-udp-src-port*         inet:port-number
|      |  |  +--rw pkt-sec-cond-udp-dest-port*         inet:port-number
|      |  |  +--rw pkt-sec-cond-udp-length*   string
```

도면14d

```
|      |  +--rw packet-security-icmp-condition
|      |     +--rw pkt-sec-cond-icmp-type*      uint8
|      |     +--rw pkt-sec-cond-icmp-code*      uint8
|      |     +--rw pkt-sec-cond-icmp-seg-num*   uint32
|      +--rw packet-payload-condition
|      |  +--rw packet-payload-manual?            string
|      |  +--rw pkt-payload-content*              string
|      +--rw target-condition
|      |  +--rw target-manual?                    string
|      |  +--rw device-sec-context-cond
|      |     +--rw pc?                 boolean
|      |     +--rw mobile-phone?       boolean
|      |     +--rw voip-volte-phone?   boolean
|      |     +--rw tablet?             boolean
|      |     +--rw iot?                boolean
|      |     +--rw vehicle?            boolean
|      +--rw users-condition
|      |  +--rw users-manual?                     string
|      |  +--rw user
|      |  |  +--rw (user-name)?
|      |  |     +--:(tenant)
|      |  |     |  +--rw tenant    uint8
|      |  |     +--:(vn-id)
|      |  |        +--rw vn-id     uint8
|      |  +--rw group
|      |     +--rw (group-name)?
|      |        +--:(tenant)
|      |        |  +--rw tenant    uint8
|      |        +--:(vn-id)
|      |           +--rw vn-id     uint8
|      +--rw context-condition
|      |  +--rw context-manual?                   string
|      +--rw gen-context-condition
|         +--rw gen-context-manual?                string
|         +--rw geographic-location
|            +--rw src-geographic-location*    uint32
|            +--rw dest-geographic-location*   uint32
+--rw action-clause-container
   ...
```

도면15a

```
module: ietf-i2nsf-policy-rule-for-nsf
+--rw i2nsf-security-policy* [policy-name]
|     ...
|  +--rw eca-policy-rules* [rule-id]
|     ...
|  +--rw resolution-strategy
|     ...
|  +--rw default-action
|     ...
+--rw event-clause-container
|  ...
+--rw condition-clause-container
|  ...
+--rw action-clause-container
   +--rw action-clause-list* [eca-object-id]
      +--rw entity-class?                    identityref
      +--rw eca-object-id                    string
         +--rw ingress-action
         |  +--rw ingress-manual?                  string
         |  +--rw ingress-action-type?             ingress-action
         +--rw egress-action
         |  +--rw egress-manual?                   string
         |  +--rw egress-action-type?              egress-action
         +--rw apply-profile
            +--rw profile-manual?                  string
               +--rw content-security-control
               |  +--rw content-security-control-types
               |     +--rw antivirus?             boolean
               |     +--rw ips?                   boolean
               |     +--rw ids?                   boolean
               |     +--rw url-filtering?         boolean
               |     +--rw data-filtering?        boolean
               |     +--rw mail-filtering?        boolean
               |     +--rw file-blocking?         boolean
               |     +--rw file-isolate?          boolean
               |     +--rw pkt-capture?           boolean
               |     +--rw application-control?   boolean
               |     +--rw voip-volte?            boolean
               +--rw attack-mitigation-control
                     +--rw ddos-attack
                        +--rw ddos-attack-type
                           +--rw network-layer-ddos-attack
                              +--rw network-layer-ddos-attack-type
                                 +--rw syn-flood?       boolean
```

도면15b

```
|       |     +--rw udp-flood?        boolean
|       |     +--rw icmp-flood?       boolean
|       |     +--rw ip-frag-flood?    boolean
|       |     +--rw ipv6-related?     boolean
|       +--rw app-layer-ddos-attack
|          +--rw app-ddos-attack-types
|             +--rw http-flood?       boolean
|             +--rw https-flood?      boolean
|             +--rw dns-flood?        boolean
|             +--rw dns-amp-flood?    boolean
|             +--rw ssl-ddos?         boolean
+--rw single-packet-attack
   +--rw single-packet-attack-type
      +--rw scan-and-sniff-attack
      |  +--rw scan-and-sniff-attack-types
      |     +--rw ip-sweep?          boolean
      |     +--rw port-scanning?     boolean
      +--rw malformed-packet-attack
      |  +--rw malformed-packet-attack-types
      |     +--rw ping-of-death?     boolean
      |     +--rw teardrop?          boolean
      +--rw special-packet-attack
         +--rw special-packet-attack-types
            +--rw oversized-icmp?    boolean
            +--rw tracert?           boolean
```

도면16a

```
<CODE BEGINS> file "ietf-i2nsf-policy-rule-for-nsf@2018-03-05.yang"

module ietf-i2nsf-policy-rule-for-nsf {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-i2nsf-policy-rule-for-nsf";
  prefix
    policy-rule-for-nsf;
```

도면16b

```
import ietf-inet-types{
  prefix inet;
}
import ietf-yang-types{
  prefix yang;
}

organization
  "IETF I2NSF (Interface to Network Security Functions)
   Working Group";
```

도면16c

```
contact
  "WG Web: <http://tools.ietf.org/wg/i2nsf>
   WG List: <mailto:i2nsf@ietf.org>

   WG Chair: Adrian Farrel
   <mailto:Adrain@olddog.co.uk>

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

   Editor: Jingyong Tim Kim
   <mailto:timkim@skku.edu>

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

   Editor: Susan Hares
   <mailto:shares@ndzh.com>";

description
  "This module defines a YANG data module for network security
   functions.";
revision "2018-03-05"{
  description "The fourth revision";
  reference
    "draft-ietf-i2nsf-capability-00";
}

typedef sec-event-format {
    type enumeration {
      enum unknown {
          description
            "If SecEventFormat is unknown";
      }
      enum guid {
          description
            "If SecEventFormat is GUID
```

도면16d

```
            (Generic Unique IDentifier)";
      }
      enum uuid {
          description
            "If SecEventFormat is UUID
            (Universal Unique IDentifier)";
      }
      enum uri {
          description
            "If SecEventFormat is URI
            (Uniform Resource Identifier)";
      }
      enum fqdn {
          description
            "If SecEventFormat is FQDN
            (Fully Qualified Domain Name)";
      }
      enum fqpn {
          description
            "If SecEventFormat is FQPN
            (Fully Qualified Path Name)";
      }
    }
    description
      "This is used for SecEventFormat.";
}
```

도면16e

```
typedef ingress-action {
    type enumeration {
      enum pass {
          description
            "If ingress action is pass";
      }
      enum drop {
          description
            "If ingress action is drop";
      }
      enum reject {
          description
            "If ingress action is reject";
      }
      enum alert {
          description
            "If ingress action is alert";
      }
      enum mirror {
          description
            "If ingress action is mirror";
```

도면16f

```
      }
    }
    description
      "This is used for ingress action.";
}

typedef egress-action {
    type enumeration {
      enum invoke-signaling {
          description
            "If egress action is invoke signaling";
      }
      enum tunnel-encapsulation {
          description
            "If egress action is tunnel encapsulation";
      }
      enum forwarding {
          description
            "If egress action is forwarding";
      }
      enum redirection {
          description
            "If egress action is redirection";
      }
    }
    description
      "This is used for egress action.";
}
```

도면16g

```
identity ECA-OBJECT-TYPE {
  description "TBD";
}

identity ECA-EVENT-TYPE {
  base ECA-OBJECT-TYPE;
  description "TBD";
}

identity ECA-CONDITION-TYPE {
  base ECA-OBJECT-TYPE;
  description "TBD";
}

identity ECA-ACTION-TYPE {
  base ECA-OBJECT-TYPE;
  description "TBD";
}
```

도면16h

```
 identity EVENT-USER-TYPE {
  base ECA-EVENT-TYPE;
  description "TBD";
}

 identity EVENT-DEV-TYPE {
  base ECA-EVENT-TYPE;
  description "TBD";
}

 identity EVENT-SYS-TYPE {
  base ECA-EVENT-TYPE;
  description "TBD";
}

 identity EVENT-TIME-TYPE {
  base ECA-EVENT-TYPE;
  description "TBD";
}


grouping i2nsf-eca-object-type {
  leaf entity-class {
    type identityref {
      base ECA-OBJECT-TYPE;
    }
    description "TBD";
  }
  leaf eca-object-id {
      type string;
      description "TBD";
  }
```

도면16i

```
grouping i2nsf-event-type {
    description "TBD";
    leaf manual {
      type string;
      description
        "This is manual for event.
        Vendors can write instructions for event
        that vendor made";
    }

    leaf sec-event-content {
      type string;
      mandatory true;
```

도면16j

```
  description
   "This is a mandatory string that contains the content
    of the SecurityEvent. The format of the content
    is specified in the SecEventFormat class
    attribute, and the type of event is defined in the
    SecEventType class attribute. An example of the
    SecEventContent attribute is a string hrAdmin,
    with the SecEventFormat set to 1 (GUID) and the
    SecEventType attribute set to 5 (new logon).";
}

leaf sec-event-format {
  type sec-event-format;
  mandatory true;
  description
   "This is a mandatory uint 8 enumerated integer, which
    is used to specify the data type of the
    SecEventContent attribute. The content is
    specified in the SecEventContent class attribute,
    and the type of event is defined in the
    SecEventType class attribute. An example of the
    SecEventContent attribute is string hrAdmin,
    with the SecEventFormat attribute set to 1 (GUID)
    and the SecEventType attribute set to 5
    (new logon).";
}
```

도면16k

```
    leaf sec-event-type {
      type string;
      mandatory true;
      description
       "This is a mandatory uint 8 enumerated integer, which
        is used to specify the type of event that involves
        this user. The content and format are specified in
        the SecEventContent and SecEventFormat class
        attributes, respectively. An example of the
        SecEventContent attribute is string hrAdmin,
        with the SecEventFormat attribute set to 1 (GUID)
        and the SecEventType attribute set to 5
       (new logon).";
    }

}


 list i2nsf-security-policy {
   key "policy-name";
   description
```

도면161

```
"policy is a list
 including a set of security rules according to certain logic,
 i.e., their similarity or mutual relations, etc. The network
 security policy is able to apply over both the unidirectional
 and bidirectional traffic across the NSF.";

leaf policy-name {
  type string;
  mandatory true;
  description
    "The name of the policy.
     This must be unique.";
}

list eca-policy-rules {
  key "rule-id";
  description
    "This is a rule for network security functions.";

  leaf rule-id {
    type uint8;
    mandatory true;
    description
      "The id of the rule.
       This must be unique.";
  }
```

도면16m

```
leaf rule-description {
  type string;
  description
    "This description gives more information about
     rules.";
}

leaf rule-rev {
  type uint8;
  description
    "This shows rule version.";
}

leaf rule-priority {
  type uint8;
  description
    "The priority keyword comes with a mandatory
     numeric value which can range from 1 till 255.";
}
leaf-list policy-event-clause-agg-ptr {
    type instance-identifier;
```

도면16n

```
    must 'derived-from-or-self (/event-clause-container/
    event-clause-list/entity-class, "ECA-EVENT-TYPE")';
    description
        "TBD";
}
leaf-list policy-condition-clause-agg-ptr {
    type instance-identifier;
    must 'derived-from-or-self (/condition-clause-container/
    condition-clause-list/entity-class, "ECA-CONDITION-TYPE")';
    description
        "TBD";
}
leaf-list policy-action-clause-agg-ptr {
    type instance-identifier;
    must 'derived-from-or-self (/action-clause-container/
    action-clause-list/entity-class, "ECA-ACTION-TYPE")';
    description
        "TBD";
}
```

도면16o

```
container time-zone {
  description
    "This can be used to apply rules according to time-zone";
  container absolute-time-zone {
    description
    "This can be used to apply rules according to
     absolute-time";
    container time {
      description
        "This can be used to apply rules according to time";
      leaf start-time {
        type yang:date-and-time;
        description
          "This is start time for time zone";
      }
      leaf end-time {
        type yang:date-and-time;
        description
          "This is end time for time zone";
      }
    }
    container date {
      description
        "This can be used to apply rules according to date";
      leaf absolute-date {
        type yang:date-and-time;
        description
          "This is absolute date for time zone";
```

도면16p

```
    }
  }
}
container periodic-time-zone {
  description
   "This can be used to apply rules according to
   periodic-time-zone";
  container day {
    description
     "This can be used to apply rules according
      to periodic day";
    leaf sunday {
      type boolean;
      description
        "This is sunday for periodic day";
    }
    leaf monday {
      type boolean;
      description
        "This is monday for periodic day";
    }
    leaf tuesday {
      type boolean;
      description
        "This is tuesday for periodic day";
    }
    leaf wednesday {
      type boolean;
      description
        "This is wednesday for periodic day";
    }
    leaf thursday {
      type boolean;
      description
        "This is thursday for periodic day";
    }
```

도면16q

```
  }
  leaf friday {
    type boolean;
    description
      "This is friday for periodic day";
  }
  leaf saturday {
    type boolean;
    description
      "This is saturday for periodic day";
  }
}
container month {
```

도면16r

```
description
 "This can be used to apply rules according
  to periodic month";
leaf january {
  type boolean;
  description
    "This is january for periodic month";
}
leaf february {
  type boolean;
  description
    "This is february for periodic month";
}
leaf march {
  type boolean;
  description
    "This is march for periodic month";
}
leaf april {
  type boolean;
  description
    "This is april for periodic month";
}
leaf may {
  type boolean;
  description
    "This is may for periodic month";
}
leaf june {
  type boolean;
  description
    "This is june for periodic month";
}
```

도면16s

```
}
leaf july {
  type boolean;
  description
    "This is july for periodic month";
}
leaf august {
  type boolean;
  description
    "This is august for periodic month";
}
leaf september {
  type boolean;
  description
    "This is september for periodic month";
}
```

도면16t

```
        leaf october {
          type boolean;
          description
            "This is october for periodic month";
        }
        leaf november {
          type boolean;
          description
            "This is november for periodic month";
        }
        leaf december {
          type boolean;
          description
            "This is december for periodic month";
        }
      }
    }
  }
}
```

도면16u

```
container resolution-strategy {
  description
    "The resolution strategies can be used to
    specify how to resolve conflicts that occur between
    the actions of the same or different policy rules that
    are matched and contained in this particular NSF";


  choice resolution-strategy-type {
    description
      "Vendors can use YANG data model to configure rules";

    case fmr {
      leaf first-matching-rule {
        type boolean;
        description
          "If the resolution strategy is first matching rule";
      }
    }
    case lmr {
      leaf last-matching-rule {
        type boolean;
        description
          "If the resolution strategy is last matching rule";
      }
    }

  }
```

도면16v

```
  }

  container default-action {
    description
      "This default action can be used to specify a predefined
      action when no other alternative action was matched
      by the currently executing I2NSF Policy Rule. An analogy
      is the use of a default statement in a C switch statement.";

    leaf default-action-type {
      type ingress-action;
      description
        "Ingress action type: permit, deny, and mirror.";
    }
  }
}
```

도면16w

```
container event-clause-container {
  description "TBD";
  list event-clause-list {
  key eca-object-id;
  uses i2nsf-eca-object-type {
    refine entity-class {
        default ECA-EVENT-TYPE;
    }
  }

  description
    " This is abstract. An event is defined as any important
      occurrence in time of a change in the system being
      managed, and/or in the environment of the system being
      managed. When used in the context of policy rules for
      a flow-based NSF, it is used to determine whether the
      Condition clause of the Policy Rule can be evaluated
      or not. Examples of an I2NSF event include time and
      user actions (e.g., logon, logoff, and actions that
      violate any ACL.).";

   uses i2nsf-event-type;
   }
}
  container condition-clause-container {
  description "TBD";
  list condition-clause-list {
    key eca-object-id;
    uses i2nsf-eca-object-type {
```

도면16x

```
    refine entity-class {
        default ECA-CONDITION-TYPE;
    }
}
description
  " This is abstract.  A condition is defined as a set
  of attributes, features, and/or values that are to be
  compared with a set of known attributes, features,
  and/or values in order to determine whether or not the
  set of Actions in that (imperative) I2NSF Policy Rule
  can be executed or not. Examples of I2NSF Conditions
  include matching attributes of a packet or flow, and
  comparing the internal state of an NSF to a desired
  state.";

  container packet-security-condition {
    description
      "TBD";
    leaf packet-manual {
      type string;
      description
        "This is manual for packet condition.
        Vendors can write instructions for packet condition
        that vendor made";
    }
```

도면16y

```
container packet-security-mac-condition {
  description
   "The purpose of this Class is to represent packet MAC
   packet header information that can be used as part of
   a test to determine if the set of Policy Actions in
   this ECA Policy Rule should be execute or not.";

  leaf-list pkt-sec-cond-mac-dest {
    type yang:phys-address;
    description
      "The MAC destination address (6 octets long).";
  }

  leaf-list pkt-sec-cond-mac-src {
    type yang:phys-address;
    description
      "The MAC source address (6 octets long).";
  }

  leaf-list pkt-sec-cond-mac-8021q {
    type string;
    description
```

도면16z

```
      "This is an optional string attribute, and defines
       The 802.1Q tab value (2 octets long).";
  }

  leaf-list pkt-sec-cond-mac-ether-type {
    type string;
    description
      "The EtherType field (2 octets long). Values up to
       and including 1500 indicate the size of the
       payload in octets; values of 1536 and above
       define which protocol is encapsulated in the
       payload of the frame.";
  }

  leaf-list pkt-sec-cond-mac-tci {
    type string;
    description
     "This is an optional string attribute, and defines
      the Tag Control Information. This consists of a 3
      bit user priority field, a drop eligible indicator
      (1 bit), and a VLAN identifier (12 bits).";
  }
}
```

도면17a

```
container packet-security-ipv4-condition {
  description
    "The purpose of this Class is to represent IPv4
     packet header information that can be used as
     part of a test to determine if the set of Policy
     Actions in this ECA Policy Rule should be executed
     or not.";

  leaf-list pkt-sec-cond-ipv4-header-length {
    type uint8;
    description
      "The IPv4 packet header consists of 14 fields,
       of which 13 are required.";
  }

  leaf-list pkt-sec-cond-ipv4-tos {
    type uint8;
    description
      "The ToS field could specify a datagram's priority
       and request a route for low-delay,
       high-throughput, or highly-reliable service..";
  }

  leaf-list pkt-sec-cond-ipv4-total-length {
```

도면17b

```
  type uint16;
  description
    "This 16-bit field defines the entire packet size,
     including header and data, in bytes.";
}

leaf-list pkt-sec-cond-ipv4-id {
  type uint8;
  description
    "This field is an identification field and is
     primarily used for uniquely identifying
     the group of fragments of a single IP datagram.";
}

leaf-list pkt-sec-cond-ipv4-fragment {
  type uint8;
  description
    "IP fragmentation is an Internet Protocol (IP)
     process that breaks datagrams into smaller pieces
     (fragments), so that packets may be formed that
     can pass through a link with a smaller maximum
     transmission unit (MTU) than the original
     datagram size.";
}
```

도면17c

```
leaf-list pkt-sec-cond-ipv4-fragment-offset {
  type uint16;
  description
    "Fragment offset field along with Don't Fragment
     and More Fragment flags in the IP protocol
     header are used for fragmentation and reassembly
     of IP datagrams.";
}

leaf-list pkt-sec-cond-ipv4-ttl {
  type uint8;
  description
    "The ttl keyword is used to check for a specific
     IP time-to-live value in the header of
     a packet.";
}

leaf-list pkt-sec-cond-ipv4-protocol {
  type uint8;
  description
    "Internet Protocol version 4(IPv4) is the fourth
     version of the Internet Protocol (IP).";
}
```

도면17d

```
leaf-list pkt-sec-cond-ipv4-src {
  type inet:ipv4-address;
  description
    "Defines the IPv4 Source Address.";
}

leaf-list pkt-sec-cond-ipv4-dest {
  type inet:ipv4-address;
  description
    "Defines the IPv4 Destination Address.";
}

leaf pkt-sec-cond-ipv4-ipopts {
  type string;
  description
    "With the ipopts keyword you can check if
     a specific ip option is set. Ipopts has
     to be used at the beginning of a rule.";
}

leaf pkt-sec-cond-ipv4-sameip {
  type boolean;
  description
    "Every packet has a source IP-address and
     a destination IP-address. It can be that
     the source IP is the same as
     the destination IP.";
```

도면17e

```
  leaf-list pkt-sec-cond-ipv4-geoip {
    type string;
    description
      "The geoip keyword enables you to match on
       the source, destination or source and destination
       IP addresses of network traffic and to see to
       which country it belongs. To do this, Suricata
       uses GeoIP API with MaxMind database format.";
  }
}

container packet-security-ipv6-condition {
  description
     "The purpose of this Class is to represent packet
     IPv6 packet header information that can be used as
     part of a test to determine if the set of Policy
     Actions in this ECA Policy Rule should be executed
     or not.";
```

도면17f

```
leaf-list pkt-sec-cond-ipv6-dscp {
  type string;
  description
    "Differentiated Services Code Point (DSCP)
     of ipv6.";
}

leaf-list pkt-sec-cond-ipv6-ecn {
  type string;
  description
    "ECN allows end-to-end notification of network
     congestion without dropping packets.";
}

leaf-list pkt-sec-cond-ipv6-traffic-class {
  type uint8;
  description
    "The bits of this field hold two values. The 6
     most-significant bits are used for
     differentiated services, which is used to
     classify packets.";
}

leaf-list pkt-sec-cond-ipv6-flow-label {
  type uint32;
  description
    "The flow label when set to a non-zero value
     serves as a hint to routers and switches
     with multiple outbound paths that these
     packets should stay on the same path so that
     they will not be reordered.";
}
```

도면17g

```
leaf-list pkt-sec-cond-ipv6-payload-length {
  type uint16;
  description
    "The size of the payload in octets,
     including any extension headers.";
}

leaf-list pkt-sec-cond-ipv6-next-header {
  type uint8;
  description
    "Specifies the type of the next header.
     This field usually specifies the transport
     layer protocol used by a packet's payload.";
}
```

도면17h

```
  leaf-list pkt-sec-cond-ipv6-hop-limit {
    type uint8;
    description
      "Replaces the time to live field of IPv4.";
  }

  leaf-list pkt-sec-cond-ipv6-src {
    type inet:ipv6-address;
    description
      "The IPv6 address of the sending node.";
  }

  leaf-list pkt-sec-cond-ipv6-dest {
    type inet:ipv6-address;
    description
      "The IPv6 address of the destination node(s).";
  }
}
```

도면17i

```
container packet-security-tcp-condition {
  description
    "The purpose of this Class is to represent packet
     TCP packet header information that can be used as
     part of a test to determine if the set of Policy
     Actions in this ECA Policy Rule should be executed
     or not.";

  leaf-list pkt-sec-cond-tcp-src-port {
    type inet:port-number;
    description
      "This is a mandatory string attribute, and
      defines the Source Port number (16 bits).";
  }

  leaf-list pkt-sec-cond-tcp-dest-port {
    type inet:port-number;
    description
      "This is a mandatory string attribute, and
      defines the Destination Port number (16 bits).";
  }

  leaf-list pkt-sec-cond-tcp-seq-num {
    type uint32;
    description
      "If the SYN flag is set (1), then this is the
       initial sequence number.";
  }
```

도면17j

```
  leaf-list pkt-sec-cond-tcp-ack-num {
    type uint32;
    description
      "If the ACK flag is set then the value of this
       field is the next sequence number that the sender
       is expecting.";
  }

  leaf-list pkt-sec-cond-tcp-window-size {
    type uint16;
    description
      "The size of the receive window, which specifies
       the number of windows size units
       (by default,bytes) (beyond the segment
       identified by the sequence number in the
       acknowledgment field) that the sender of this
       segment is currently willing to recive.";
  }

  leaf-list pkt-sec-cond-tcp-flags {
    type uint8;
    description
      "This is a mandatory string attribute, and defines
       the nine Control bit flags (9 bits).";
  }
}
```

도면17k

```
container packet-security-udp-condition {
  description
   "The purpose of this Class is to represent packet UDP
    packet header information that can be used as part
    of a test to determine if the set of Policy Actions
    in this ECA Policy Rule should be executed or not.";

  leaf-list pkt-sec-cond-udp-src-port {
    type inet:port-number;
    description
      "This is a mandatory string attribute, and
      defines the UDP Source Port number (16 bits).";
  }

  leaf-list pkt-sec-cond-udp-dest-port {
    type inet:port-number;
    description
      "This is a mandatory string attribute, and
      defines the UDP Destination Port number (16 bits).";
  }
```

도면17l

```
    leaf-list pkt-sec-cond-udp-length {
      type string;
      description
       "This is a mandatory string attribute, and defines
        the length in bytes of the UDP header and data
        (16 bits).";
    }
  }

  container packet-security-icmp-condition {
    description
      "The internet control message protocol condition.";

    leaf-list pkt-sec-cond-icmp-type {
      type uint8;
      description
        "ICMP type, see Control messages.";
    }

    leaf-list pkt-sec-cond-icmp-code {
      type uint8;
      description
        "ICMP subtype, see Control messages.";
    }

    leaf-list pkt-sec-cond-icmp-seg-num {
      type uint32;
      description
        "The icmp Sequence Number.";
    }
  }
}
```

도면17m

```
container packet-payload-condition {
  description
    "TBD";
  leaf packet-payload-manual {
    type string;
    description
     "This is manual for payload condition.
     Vendors can write instructions for payload condition
     that vendor made";
  }
  leaf-list pkt-payload-content {
    type string;
    description
      "The content keyword is very important in
       signatures. Between the quotation marks you
```

도면17n

```
        can write on what you would like the
        signature to match.";
    }
}
container target-condition {
  description
    "TBD";
  leaf target-manual {
    type string;
    description
      "This is manual for target condition.
      Vendors can write instructions for target condition
      that vendor made";
  }

  container device-sec-context-cond {
    description
      "The device attribute that can identify a device,
       including the device type (i.e., router, switch,
       pc, ios, or android) and the device's owner as
       well.";

    leaf pc {
      type boolean;
      description
        "If type of a device is PC.";
    }

    leaf mobile-phone {
      type boolean;
      description
        "If type of a device is mobile-phone.";
```

도면17o

```
leaf voip-volte-phone {
  type boolean;
  description
    "If type of a device is voip-volte-phone.";
}

leaf tablet {
  type boolean;
  description
    "If type of a device is tablet.";
}


leaf iot {
```

도면17p

```
      type boolean;
      description
        "If type of a device is Internet of Things.";
    }

    leaf vehicle {
      type boolean;
      description
        "If type of a device is vehicle.";
    }
  }
}
container users-condition {
  description
    "TBD";
  leaf users-manual {
    type string;
    description
      "This is manual for user condition.
      Vendors can write instructions for user condition
      that vendor made";
  }
```

도면17q

```
container user{
  description
    "The user (or user group) information with which
     network flow is associated: The user has many
     attributes such as name, id, password, type,
     authentication mode and so on. Name/id is often
     used in the security policy to identify the user.
     Besides, NSF is aware of the IP address of the
     user provided by a unified user management system
     via network. Based on name-address association,
     NSF is able to enforce the security functions
     over the given user (or user group)";

  choice user-name {
    description
      "The name of the user.
       This must be unique.";

    case tenant {
      description
        "Tenant information.";

      leaf tenant {
        type uint8;
        mandatory true;
```

도면17r

```
        description
          "User's tenant information.";
      }
    }

    case vn-id {
      description
        "VN-ID information.";

      leaf vn-id {
        type uint8;
        mandatory true;
        description
          "User's VN-ID information.";
      }
    }
  }
}
```

도면17s

```
container group {
  description
    "The user (or user group) information with which
     network flow is associated: The user has many
     attributes such as name, id, password, type,
     authentication mode and so on. Name/id is often
     used in the security policy to identify the user.
     Besides, NSF is aware of the IP address of the
     user provided by a unified user management system
     via network. Based on name-address association,
     NSF is able to enforce the security functions
     over the given user (or user group)";

  choice group-name {
    description
      "The name of the user.
       This must be unique.";

    case tenant {
      description
        "Tenant information.";

      leaf tenant {
        type uint8;
        mandatory true;
        description
          "User's tenant information.";
      }
    }
```

도면17t

```
      case vn-id {
        description
          "VN-ID information.";

        leaf vn-id {
          type uint8;
          mandatory true;
          description
            "User's VN-ID information.";
        }
      }
    }
  }

}
container context-condition {
  description
    "TBD";
  leaf context-manual {
    type string;
    description
      "This is manual for context condition.
      Vendors can write instructions for context condition
      that vendor made";
  }
}
```

도면17u

```
container gen-context-condition {
  description
    "TBD";
  leaf gen-context-manual {
    type string;
    description
      "This is manual for generic context condition.
      Vendors can write instructions for generic context
      condition that vendor made";
  }

  container geographic-location {
    description
      "The location where network traffic is associated
       with. The region can be the geographic location
       such as country, province, and city,
       as well as the logical network location such as
       IP address, network section, and network domain.";

    leaf-list src-geographic-location {
      type uint32;
      description
```

도면17v

```
            "This is mapped to ip address. We can acquire
             source region through ip address stored the
             database.";
        }
        leaf-list dest-geographic-location {
          type uint32;
          description
            "This is mapped to ip address. We can acquire
             destination region through ip address stored
             the database.";
        }
      }
    }
  }
}
container action-clause-container {
  description "TBD";
  list action-clause-list {
  key eca-object-id;
  uses i2nsf-eca-object-type {
    refine entity-class {
      default ECA-ACTION-TYPE;
    }
  }
  description
    "An action is used to control and monitor aspects of
     flow-based NSFs when the event and condition clauses
     are satisfied. NSFs provide security functions by
     executing various Actions. Examples of I2NSF Actions
     include providing intrusion detection and/or protection,
     web and flow filtering, and deep packet inspection
     for packets and flows.";
```

도면17w

```
container ingress-action {
  description
    "TBD";
  leaf ingress-manual {
    type string;
    description
      "This is manual for ingress action.
      Vendors can write instructions for ingress action
      that vendor made";
  }
  leaf ingress-action-type {
    type ingress-action;
    description
      "Ingress action type: permit, deny, and mirror.";
```

도면17x

```
  }
}
container egress-action {
  description
    "TBD";
  leaf egress-manual {
    type string;
    description
      "This is manual for egress action.
      Vendors can write instructions for egress action
      that vendor made";
  }
  leaf egress-action-type {
    type egress-action;
    description
      "Egress-action-type: invoke-signaling,
       tunnel-encapsulation, and forwarding.";
  }
}
container apply-profile {
  description
    "TBD";
  leaf profile-manual {
    type string;
    description
      "This is manual for apply profile action.
      Vendors can write instructions for apply
      profile action that vendor made";
  }
```

도면17y

```
container content-security-control {
  description
   "Content security control is another category of
   security capabilities applied to application layer.
   Through detecting the contents carried over the
   traffic in application layer, these capabilities
   can realize various security purposes, such as
   defending against intrusion, inspecting virus,
   filtering malicious URL or junk email, and blocking
   illegal web access or data retrieval.";

  container content-security-control-types {
    description
     "Content Security types: Antivirus, IPS, IDS,
      url-filtering, data-filtering, mail-filtering,
      file-blocking, file-isolate, pkt-capture,
      application-control, and voip-volte.";
```

도면17z

```
leaf antivirus {
    type boolean;
    description
      "Additional inspection of antivirus.";
}

leaf ips {
    type boolean;
    description
      "Additional inspection of IPS.";
}

leaf ids {
    type boolean;
    description
      "Additional inspection of IDS.";
}

leaf url-filtering {
    type boolean;
    description
      "Additional inspection of URL filtering.";
}

leaf data-filtering {
   type boolean;
   description
     "Additional inspection of data filtering.";
}
```

도면18a

```
leaf mail-filtering {
  type boolean;
  description
    "Additional inspection of mail filtering.";
}

leaf file-blocking {
  type boolean;
  description
    "Additional inspection of file blocking.";
}

leaf file-isolate {
  type boolean;
  description
    "Additional inspection of file isolate.";
}
```

도면18b

```
    leaf pkt-capture {
      type boolean;
      description
        "Additional inspection of packet capture.";
    }

    leaf application-control {
      type boolean;
      description
        "Additional inspection of app control.";
    }

    leaf voip-volte {
      type boolean;
      description
        "Additional inspection of VoIP/VoLTE.";
    }
  }
}
```

도면18c

```
container attack-mitigation-control {
  description
    "This category of security capabilities is
     specially used to detect and mitigate various
     types of network attacks.";

    container ddos-attack {
      description
        "A distributed-denial-of-service (DDoS) is
         where the attack source is more than one,
         often thousands of unique IP addresses.";

      container ddos-attack-type {
        description
          "DDoS-attack types: Network Layer
          DDoS Attacks and Application Layer
          DDoS Attacks.";

        container network-layer-ddos-attack {
          description
            "Network layer DDoS-attack.";
          container network-layer-ddos-attack-type {
            description
              "Network layer DDoS attack types:
               Syn Flood Attack, UDP Flood Attack,
               ICMP Flood Attack, IP Fragment Flood,
               IPv6 Related Attacks, and etc";
```

도면18d

```
    leaf syn-flood {
      type boolean;
      description
        "Additional Inspection of
         Syn Flood Attack.";
    }

    leaf udp-flood {
      type boolean;
      description
        "Additional Inspection of
         UDP Flood Attack.";
    }

    leaf icmp-flood {
      type boolean;
      description
        "Additional Inspection of
         ICMP Flood Attack.";
    }

    leaf ip-frag-flood {
      type boolean;
      description
        "Additional Inspection of
         IP Fragment Flood.";
    }

    leaf ipv6-related {
      type boolean;
      description
        "Additional Inspection of
         IPv6 Related Attacks.";
    }
  }
}
```

도면18e

```
container app-layer-ddos-attack {
  description
    "Application layer DDoS-attack.";

  container app-ddos-attack-types {
    description
      "Application layer DDoS-attack types:
       Http Flood Attack, Https Flood Attack,
       DNS Flood Attack, and
       DNS Amplification Flood Attack,
       SSL DDoS Attack, and etc.";
```

도면18f

```
        leaf http-flood {
          type boolean;
          description
            "Additional Inspection of
             Http Flood Attack.";
        }

        leaf https-flood {
          type boolean;
          description
            "Additional Inspection of
             Https Flood Attack.";
        }

        leaf dns-flood {
          type boolean;
          description
            "Additional Inspection of
             DNS Flood Attack.";
        }

        leaf dns-amp-flood {
          type boolean;
          description
            "Additional Inspection of
             DNS Amplification Flood Attack.";
        }

        leaf ssl-ddos {
          type boolean;
          description
            "Additional Inspection of
             SSL Flood Attack.";
        }
      }
    }
  }
}
```

도면18g

```
container single-packet-attack {
  description
    "Single Packet Attacks.";
  container single-packet-attack-type {
    description
      "DDoS-attack types: Scanning Attack,
       Sniffing Attack, Malformed Packet Attack,
       Special Packet Attack, and etc.";
```

도면18h

```
container scan-and-sniff-attack {
  description
    "Scanning and Sniffing Attack.";
  container scan-and-sniff-attack-types {
    description
      "Scanning and sniffing attack types:
       IP Sweep attack, Port Scanning,
       and etc.";

    leaf ip-sweep {
      type boolean;
      description
        "Additional Inspection of
         IP Sweep Attack.";
    }

    leaf port-scanning {
      type boolean;
      description
        "Additional Inspection of
         Port Scanning Attack.";
    }
  }
}
```

도면18i

```
container malformed-packet-attack {
  description
    "Malformed Packet Attack.";
  container malformed-packet-attack-types {
    description
      "Malformed packet attack types:
       Ping of Death Attack, Teardrop Attack,
       and etc.";

    leaf ping-of-death {
      type boolean;
      description
        "Additional Inspection of
         Ping of Death Attack.";
    }

    leaf teardrop {
      type boolean;
      description
        "Additional Inspection of
         Teardrop Attack.";
    }
  }
```

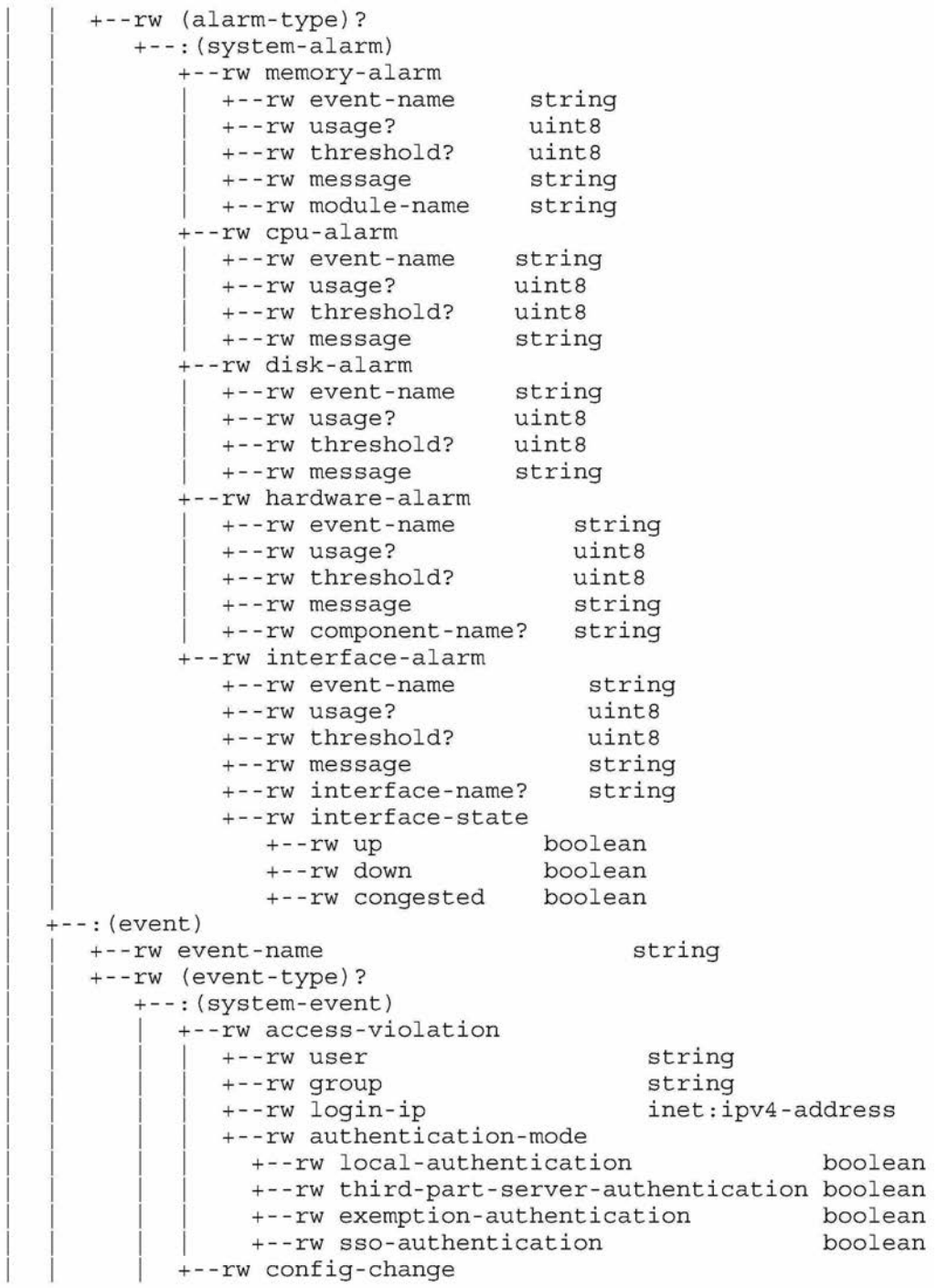
도면18j

```
                    }

                    container special-packet-attack {
                      description
                        "special Packet Attack.";
                      container special-packet-attack-types {
                        description
                          "Special packet attack types:
                           Oversized ICMP Attack, Tracert Attack,
                           and etc.";

                        leaf oversized-icmp {
                          type boolean;
                          description
                            "Additional Inspection of
                             Oversize ICMP Attack.";
                        }

                        leaf tracert {
                          type boolean;
                          description
                            "Additional Inspection of
                             Tracrt Attack.";
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

도면19a

```
module: ietf-i2nsf-nsf-monitoring-dm
  +--rw monitoring-message
    +--rw monitoring-messages* [message-id]
       +--rw message-id                    uint8
       +--rw message-version               uint8
       +--rw (message-type)?
       |  +--:(alarm)
```

도면19b


+--rw (alarm-type)?
+--:(system-alarm)
+--rw memory-alarm
+--rw event-name string
+--rw usage? uint8
+--rw threshold? uint8
+--rw message string
+--rw module-name string
+--rw cpu-alarm
+--rw event-name string
+--rw usage? uint8
+--rw threshold? uint8
+--rw message string
+--rw disk-alarm
+--rw event-name string
+--rw usage? uint8
+--rw threshold? uint8
+--rw message string
+--rw hardware-alarm
+--rw event-name string
+--rw usage? uint8
+--rw threshold? uint8
+--rw message string
+--rw component-name? string
+--rw interface-alarm
+--rw event-name string
+--rw usage? uint8
+--rw threshold? uint8
+--rw message string
+--rw interface-name? string
+--rw interface-state
+--rw up boolean
+--rw down boolean
+--rw congested boolean
+--:(event)
+--rw event-name string
+--rw (event-type)?
+--:(system-event)
+--rw access-violation
+--rw user string
+--rw group string
+--rw login-ip inet:ipv4-address
+--rw authentication-mode
+--rw local-authentication boolean
+--rw third-part-server-authentication boolean
+--rw exemption-authentication boolean
+--rw sso-authentication boolean
+--rw config-change

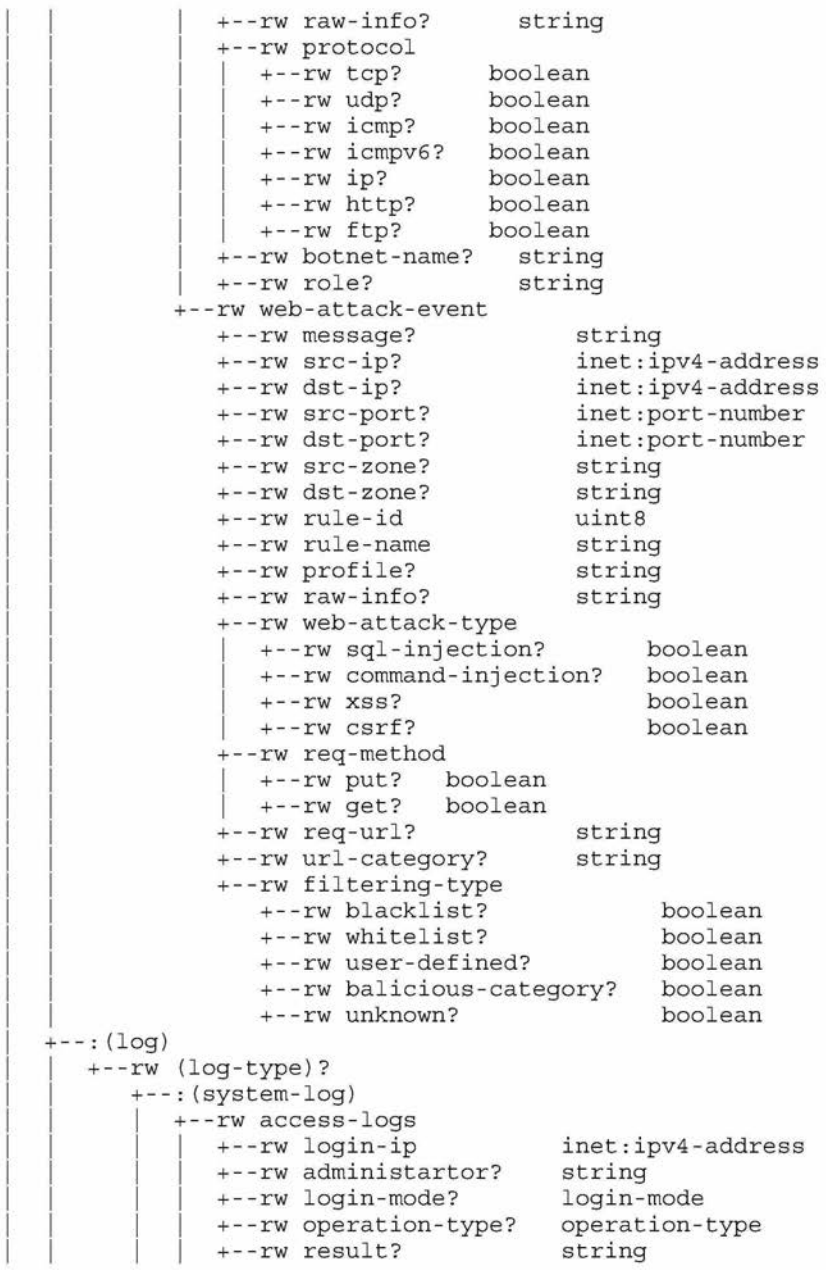
도면19c

```
|  |  |  |     +--rw user                        string
|  |  |  |     +--rw group                       string
|  |  |  |     +--rw login-ip                    inet:ipv4-address
|  |  |  |     +--rw authentication-mode
|  |  |  |       +--rw local-authentication                boolean
|  |  |  |       +--rw third-part-server-authentication boolean
|  |  |  |       +--rw exemption-authentication            boolean
|  |  |  |       +--rw sso-authentication                  boolean
|  |  |  +--:(nsf-event)
|  |  |     +--rw ddos-event
|  |  |     |  +--rw message?              string
|  |  |     |  +--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 rule-id               uint8
|  |  |     |  +--rw rule-name             string
|  |  |     |  +--rw profile?              string
|  |  |     |  +--rw raw-info?             string
|  |  |     |  +--rw ddos-attack-type
|  |  |     |  |  +--rw syn-flood?                boolean
|  |  |     |  |  +--rw ack-flood?                boolean
|  |  |     |  |  +--rw syn-ack-flood?            boolean
|  |  |     |  |  +--rw fin-rst-flood?            boolean
|  |  |     |  |  +--rw tcp-connection-flood?     boolean
|  |  |     |  |  +--rw udp-flood?                boolean
|  |  |     |  |  +--rw icmp-flood?               boolean
|  |  |     |  |  +--rw https-flood?              boolean
|  |  |     |  |  +--rw http-flood?               boolean
|  |  |     |  |  +--rw dns-reply-flood?          boolean
|  |  |     |  |  +--rw dns-query-flood?          boolean
|  |  |     |  |  +--rw sip-flood?                boolean
|  |  |     |  +--rw start-time            yang:date-and-time
|  |  |     |  +--rw end-time              yang:date-and-time
|  |  |     |  +--rw attack-rate?          uint32
|  |  |     |  +--rw attack-speed?         uint32
|  |  |     +--rw session-table-event
|  |  |     |  +--rw current-session?   uint8
|  |  |     |  +--rw maximum-session?   uint8
|  |  |     |  +--rw threshold?         uint8
|  |  |     |  +--rw message?           string
|  |  |     +--rw virus-event
|  |  |     |  +--rw message?       string
|  |  |     |  +--rw src-ip?        inet:ipv4-address
|  |  |     |  +--rw dst-ip?        inet:ipv4-address
|  |  |     |  +--rw src-port?      inet:port-number
```

도면19d

```
|  |  |  +--rw dst-port?      inet:port-number
|  |  |  +--rw src-zone?      string
|  |  |  +--rw dst-zone?      string
|  |  |  +--rw rule-id        uint8
|  |  |  +--rw rule-name      string
|  |  |  +--rw profile?       string
|  |  |  +--rw raw-info?      string
|  |  |  +--rw virus-type
|  |  |  |  +--rw trajan?   boolean
|  |  |  |  +--rw worm?     boolean
|  |  |  |  +--rw macro?    boolean
|  |  |  +--rw virus-name?   string
|  |  |  +--rw file-type?    string
|  |  |  +--rw file-name?    string
|  |  +--rw intrusion-event
|  |  |  +--rw message?                  string
|  |  |  +--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 rule-id                   uint8
|  |  |  +--rw rule-name                 string
|  |  |  +--rw profile?                  string
|  |  |  +--rw raw-info?                 string
|  |  |  +--rw protocol
|  |  |  |  +--rw tcp?      boolean
|  |  |  |  +--rw udp?      boolean
|  |  |  |  +--rw icmp?     boolean
|  |  |  |  +--rw icmpv6?   boolean
|  |  |  |  +--rw ip?       boolean
|  |  |  |  +--rw http?     boolean
|  |  |  |  +--rw ftp?      boolean
|  |  |  +--rw intrusion-attack-type
|  |  |     +--rw brutal-force?      boolean
|  |  |     +--rw buffer-overflow?   boolean
|  |  +--rw botnet-event
|  |  |  +--rw message?       string
|  |  |  +--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 rule-id        uint8
|  |  |  +--rw rule-name      string
|  |  |  +--rw profile?       string
```

도면19e


+--rw raw-info? string
+--rw protocol
+--rw tcp? boolean
+--rw udp? boolean
+--rw icmp? boolean
+--rw icmpv6? boolean
+--rw ip? boolean
+--rw http? boolean
+--rw ftp? boolean
+--rw botnet-name? string
+--rw role? string
+--rw web-attack-event
+--rw message? string
+--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 rule-id uint8
+--rw rule-name string
+--rw profile? string
+--rw raw-info? string
+--rw web-attack-type
+--rw sql-injection? boolean
+--rw command-injection? boolean
+--rw xss? boolean
+--rw csrf? boolean
+--rw req-method
+--rw put? boolean
+--rw get? boolean
+--rw req-url? string
+--rw url-category? string
+--rw filtering-type
+--rw blacklist? boolean
+--rw whitelist? boolean
+--rw user-defined? boolean
+--rw balicious-category? boolean
+--rw unknown? boolean
+--:(log)
+--rw (log-type)?
+--:(system-log)
+--rw access-logs
+--rw login-ip inet:ipv4-address
+--rw administartor? string
+--rw login-mode? login-mode
+--rw operation-type? operation-type
+--rw result? string

도면19f

```
|  |  |  |  +--rw content?                string
|  |  |  +--rw resource-utiliz-logs
|  |  |  |  +--rw system-status?        string
|  |  |  |  +--rw cpu-usage?            uint8
|  |  |  |  +--rw memory-usage?         uint8
|  |  |  |  +--rw disk-usage?           uint8
|  |  |  |  +--rw disk-left?            uint8
|  |  |  |  +--rw session-num?          uint8
|  |  |  |  +--rw process-num?          uint8
|  |  |  |  +--rw in-traffic-rate?      uint32
|  |  |  |  +--rw out-traffic-rate?     uint32
|  |  |  |  +--rw in-traffic-speed?     uint32
|  |  |  |  +--rw out-traffic-speed?    uint32
|  |  |  +--rw user-activity-logs
|  |  |     +--rw user                      string
|  |  |     +--rw group                     string
|  |  |     +--rw login-ip                  inet:ipv4-address
|  |  |     +--rw authentication-mode
|  |  |     |  +--rw local-authentication              boolean
|  |  |     |  +--rw third-part-server-authentication boolean
|  |  |     |  +--rw exemption-authentication          boolean
|  |  |     |  +--rw sso-authentication                boolean
|  |  |     +--rw access-mode
|  |  |     |  +--rw ppp?      boolean
|  |  |     |  +--rw svn?      boolean
|  |  |     |  +--rw local?    boolean
|  |  |     +--rw online-duration?          string
|  |  |     +--rw logout-duration?          string
|  |  |     +--rw addtional-info?           string
|  |  |     +--rw cause?                    string
|  |  +--:(nsf-log)
|  |     +--rw ddos-logs
|  |     |  +--rw attack-type?          string
|  |     |  +--rw attack-ave-rate?      uint32
|  |     |  +--rw attack-ave-speed?     uint32
|  |     |  +--rw attack-pkt-num?       uint32
|  |     |  +--rw attack-src-ip?        inet:ipv4-address
|  |     |  +--rw action?               all-action
|  |     |  +--rw os?                   string
|  |     +--rw virus-logs
|  |     |  +--rw protocol
|  |     |  |  +--rw tcp?       boolean
|  |     |  |  +--rw udp?       boolean
|  |     |  |  +--rw icmp?      boolean
|  |     |  |  +--rw icmpv6?    boolean
|  |     |  |  +--rw ip?        boolean
|  |     |  |  +--rw http?      boolean
|  |     |  |  +--rw ftp?       boolean
```

도면19g

```
|  |  |  +--rw attack-type?    string
|  |  |  +--rw action?         all-action
|  |  |  +--rw os?             string
|  |  |  +--rw time            yang:date-and-time
|  |  +--rw intrusion-logs
|  |  |  +--rw attack-type?     string
|  |  |  +--rw action?          all-action
|  |  |  +--rw time             yang:date-and-time
|  |  |  +--rw attack-rate?     uint32
|  |  |  +--rw attack-speed?    uint32
|  |  +--rw botnet-logs
|  |  |  +--rw attack-type?       string
|  |  |  +--rw botnet-pkt-num?    uint8
|  |  |  +--rw action?            all-action
|  |  |  +--rw os?                string
|  |  +--rw dpi-logs
|  |  |  +--rw dpi-type?       dpi-type
|  |  |  +--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
|  |  |  |  +--rw tcp?       boolean
|  |  |  |  +--rw udp?       boolean
|  |  |  |  +--rw icmp?      boolean
|  |  |  |  +--rw icmpv6?    boolean
|  |  |  |  +--rw ip?        boolean
|  |  |  |  +--rw http?      boolean
|  |  |  |  +--rw ftp?       boolean
|  |  |  +--rw file-type?      string
|  |  |  +--rw file-name?      string
|  |  +--rw vulnerability-scanning-logs* [vulnerability-id]
|  |  |  +--rw vulnerability-id    uint8
|  |  |  +--rw victim-ip?          inet:ipv4-address
|  |  |  +--rw protocol
|  |  |  |  +--rw tcp?       boolean
|  |  |  |  +--rw udp?       boolean
|  |  |  |  +--rw icmp?      boolean
|  |  |  |  +--rw icmpv6?    boolean
|  |  |  |  +--rw ip?        boolean
|  |  |  |  +--rw http?      boolean
```

도면19h

```
|   |         |   |  +--rw ftp?       boolean
|   |         |   +--rw port-num?           inet:port-number
|   |         |   +--rw level?              severity
|   |         |   +--rw os?                 string
|   |         |   +--rw additional-info?    string
|   |         +--rw web-attack-logs
|   |            +--rw attack-type?     string
|   |            +--rw rsp-code?        string
|   |            +--rw req-clientapp?   string
|   |            +--rw req-cookies?     string
|   |            +--rw req-host?        string
|   |            +--rw raw-info?        string
|   +--:(counters)
|      +--rw (counter-type)?
|         +--:(system-counter)
|         |  +--rw interface-counters
|         |     +--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
|         +--:(nsf-counter)
|         |  +--rw firewall-counters
|         |     +--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
```

도면19i

```
|          |      |  +--rw tcp?       boolean
|          |      |  +--rw udp?       boolean
|          |      |  +--rw icmp?      boolean
|          |      |  +--rw icmpv6?    boolean
|          |      |  +--rw ip?        boolean
|          |      |  +--rw http?      boolean
|          |      |  +--rw ftp?       boolean
|          |      +--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 bound
|          |         +--rw in-interface?    boolean
|          |         +--rw out-interface?   boolean
|          +--:(policy-hit-counters)
|             +--rw policy-hit-counters
|                +--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
|                |  +--rw tcp?       boolean
|                |  +--rw udp?       boolean
|                |  +--rw icmp?      boolean
|                |  +--rw icmpv6?    boolean
|                |  +--rw ip?        boolean
|                |  +--rw http?      boolean
|                |  +--rw ftp?       boolean
|                +--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
```

도면19j

```
|                +--rw out-traffic-peak-speed?   uint32
|                +--rw hit-times?                uint32
+--rw message                          string
+--rw time-stamp                       yang:date-and-time
+--rw vendor-name?                     string
+--rw nsf-name?                        string
+--rw severity                         severity
```

도면20a

```
<CODE BEGINS> file "ietf-i2nsf-nsf-monitoring-dm@2018-03-05.yang"

module ietf-i2nsf-nsf-monitoring-dm {
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-i2nsf-nsf-monitoring-dm";
  prefix
    monitoring-information;
  import ietf-inet-types{
    prefix inet;
  }
  import ietf-yang-types {
    prefix yang;
  }
  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: Dongjin Hong
     <mailto:dong.jin@skku.edu>

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

  description
    "This module defines a YANG data module for monitoring NSFs.";
```

도면20b

```
revision "2017-10-29" {
  description "Initial revision";
  reference
    "draft-zhang-i2nsf-info-model-monitoring-04";
}

typedef severity {
  type enumeration {
    enum high {
      description
        "high-level";
    }
    enum middle {
      description
        "middle-level";
    }
    enum low {
      description
        "low-level";
    }
  }
  description
    "This is used for indicating the severity";
}
typedef all-action {
  type enumeration {
    enum allow {
      description
        "If action is allow";
```

도면20c

```
}
enum alert {
  description
    "If action is alert";
}
enum block {
  description
    "If action is block";
}
enum discard {
  description
    "If action is discard";
}
enum declare {
  description
    "If action is declare";
}
enum block-ip {
  description
```

도면20d

```
        "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";
}
```

도면20e

```
typedef operation-type{
  type enumeration {
    enum login{
      description
        "Login operation";
    }
    enum logout{
      description
        "Logout operation";
    }
    enum configuration{
      description
        "Configuration operation";
    }
  }
  description
    "This is used for operation type";
}
typedef login-mode{
  type enumeration {
```

도면20f

```
    enum root{
      description
        "Root login-mode";
    }
    enum user{
      description
        "User login-mode";
    }
    enum guest{
      description
        "Guest login-mode";
    }
  }
  description
    "This is used for login mode";
}
grouping protocol {
  description
    "A set of protocols";
  container protocol {
    description
    "Protocol types:
     TCP, UDP, ICMP, ICMPv6, IP, HTTP, FTP and etc.";
    leaf tcp {
      type boolean;
      description
        "TCP protocol type.";
    }
    leaf udp {
      type boolean;
      description
        "UDP protocol type.";
    }
    leaf icmp {
      type boolean;
      description
        "ICMP protocol type.";
    }
    leaf icmpv6 {
      type boolean;
      description
        "ICMPv6 protocol type.";
    }
    leaf ip {
      type boolean;
      description
        "IP protocol type.";
    }
```

도면20g

```
    leaf http {
      type boolean;
      description
        "HTTP protocol type.";
    }
    leaf ftp {
      type boolean;
      description
        "ftp protocol type.";
    }
  }
}
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;
```

도면20h

```
    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 authentication-mode{
  description
  "A set of authentication-mode";
   container authentication-mode {
     description
       "User authentication mode. e.g., Local Authentication,
       Third-Party Server Authentication,
       Authentication Exemption, SSO Authentication.";
     leaf local-authentication {
       type boolean;
       mandatory true;
       description
         "Authentication-mode : local authentication.";
     }
     leaf third-part-server-authentication {
       type boolean;
       mandatory true;
       description
         "If authentication-mode is
         third-part-server-authentication";
     }
     leaf exemption-authentication {
       type boolean;
       mandatory true;
       description
         "If authentication-mode is
         exemption-authentication";
     }
     leaf sso-authentication {
       type boolean;
       mandatory true;
       description
         "If authentication-mode is
         sso-authentication";
```

도면20i

```
      }
    }
}
grouping i2nsf-system-alarm-type-content {
  description
    "A set of system alarm type contents";
  leaf event-name {
    type string;
    mandatory true;
    description
      "This is used to distinguish event type";
  }
  leaf usage {
    type uint8;
    description
      "specifies the amount of usage";
  }
  leaf threshold {
    type uint8;
    description
      "The threshold triggering the alarm or the event";
  }
  leaf message {
    type string;
    mandatory true;
    description
      "The usage exceeded the threshold";
  }
}
grouping i2nsf-system-event-type-content {
  description
    "A set of system event type contents";
  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 {
    type inet:ipv4-address;
    mandatory true;
    description
```

도면20j

```
      "Login IP address of a user.";
  }
  uses authentication-mode;
}
grouping i2nsf-nsf-event-type-content {
  description
    "A set of nsf event type contents";
  leaf message {
    type string;
    description
      "The message for nsf events";
  }
  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";
  }
```

도면20k

```
    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-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;
```

도면201

```
    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";
  }
```

도면20

```
    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;
      mandatory true;
      description
        "The ID of the policy being triggered";
    }
    leaf policy-name {
      type string;
      mandatory true;
      description
        "The name of the policy being triggered";
    }
    leaf src-user{
      type string;
      description
        "User who generates traffic";
    }
    uses protocol;
    uses traffic-rates;
  }

  container monitoring-message {
    description
      "The message for monitoring information";
    list monitoring-messages {
      key message-id;
      description
        "The messages according to monitoring information";
      leaf message-id {
        type uint8;
        mandatory true;
        description
          "This is message ID
```

도면20n

```
      This is key for monitoring messages";
}
leaf message-version {
  type uint8;
  mandatory true;
  description
    "The version of message";
}
choice message-type {
  description
    "The type of message";
  case alarm {
    description
      "If the message type is alarm";
    choice alarm-type {
      description
        "This is alarm type such as system alarm";
      case system-alarm{
        description
          "If the alarm type is system alarm";
        container memory-alarm {
          description
            "This is memory alarm in
            system alarm";
          uses i2nsf-system-alarm-type-content;
          leaf module-name {
            type string;
            mandatory true;
            description
              "Indicate the NSF module
              responsible for generating
              this alarm";
          }
        }
        container cpu-alarm {
          description
            "This is cpu alarm in system alarm";
          uses i2nsf-system-alarm-type-content;
        }
        container disk-alarm {
          description
            "This is disk alarm in system alarm";
          uses i2nsf-system-alarm-type-content;
        }
        container hardware-alarm {
          description
            "This is hardware alarm
            in system alarm";
```

도면20o

```
        uses i2nsf-system-alarm-type-content;
        leaf component-name {
          type string;
          description
            "Indicate the HW component
            responsible for generating
            this alarm.";
        }
      }
      container interface-alarm {
        description
          "This is interface alarm
          in system alarm";
        uses i2nsf-system-alarm-type-content;
        leaf interface-name {
          type string;
          description
            "The name of interface.";
        }
        container interface-state {
          description
            "This is isnteface state
            in interface-alarm";
          leaf up {
            type boolean;
            mandatory true;
            description
             "The state of interface is up";
          }
          leaf down {
            type boolean;
            mandatory true;
            description
            "The state of interface is down";
          }
          leaf congested {
            type boolean;
            mandatory true;
            description
             "The state of interface is
             congested";
          }
        }
      }
    }
  }
}
case event {
```

도면20p

```
description
  "If the message type is event";
leaf event-name {
  type string;
  mandatory true;
  description
    "The name of the event";
}
choice event-type {
  description
    "This is event type such as system event
    and nsf event.";
  case system-event {
    description
      "If the event type is system event";
    container access-violation {
      description
        "If the system event is
        access violation";
      uses i2nsf-system-event-type-content;
    }
    container config-change {
      description
        "If the system event is
        config change violation";
      uses i2nsf-system-event-type-content;
    }
  }
  case nsf-event {
    description
      "If the event type is nsf event";
    container ddos-event {
      description
        "If the event type is DDoS event";
      uses i2nsf-nsf-event-type-content;
      container ddos-attack-type{
        description
          "Type of DDoS attack";
        leaf syn-flood{
          type boolean;
          description
            "If the DDoS attack is
            syn flood";
        }
        leaf ack-flood{
          type boolean;
          description
            "If the DDoS attack is
```

도면20q

```
    ack flood";
}
leaf syn-ack-flood{
  type boolean;
  description
    "If the DDoS attack is
    syn ack flood";
}
leaf fin-rst-flood{
  type boolean;
  description
    "If the DDoS attack is
    fin rst flood";
}
leaf tcp-connection-flood{
  type boolean;
  description
    "If the DDoS attack is
    tcp connection flood";
}
leaf udp-flood{
  type boolean;
  description
    "If the DDoS attack is
    udp flood";
}
leaf icmp-flood{
  type boolean;
  description
    "If the DDoS attack is
    icmp flood";
}
leaf https-flood{
  type boolean;
  description
    "If the DDoS attack is
    https flood";
}
leaf http-flood{
  type boolean;
  description
    "If the DDoS attack is
    http flood";
}
leaf dns-reply-flood{
  type boolean;
  description
    "If the DDoS attack is
```

도면20r

```
        dns reply flood";
    }
    leaf dns-query-flood{
      type boolean;
      description
        "If the DDoS attack is
        dns query flood";
    }
    leaf sip-flood{
      type boolean;
      description
        "If the DDoS attack is
        sip flood";
    }
  }
  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 of attack traffic";
  }
  leaf attack-speed {
    type uint32;
    description
      "the bps of attack traffic";
  }
}
container session-table-event {
  description
    "If the event type is session
    table event";
    leaf current-session {
      type uint8;
      description
       "The number of concurrent
```

도면20s

```
        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";
    }
    leaf message {
      type string;
      description
       "The number of session table
        exceeded the threshold";
    }
}
container virus-event {
  description
    "If the event type is virus event";
  uses i2nsf-nsf-event-type-content;
  container virus-type {
    description
      "The type of virus";
    leaf trajan {
      type boolean;
      description
       "If the virus type is trajan";
    }
    leaf worm {
      type boolean;
      description
        "If the virus type is worm";
    }
    leaf macro {
      type boolean;
      description
        "If the virus type is macro";
    }
  }
  leaf virus-name {
    type string;
    description
```

도면20t

```
        "The name of virus";
    }
    leaf file-type {
      type string;
      description
        "The type of file";
    }
    leaf file-name {
      type string;
      description
        "The name of file";
    }
  }
  container intrusion-event {
    description
     "If the event type is intrusion event";
    uses i2nsf-nsf-event-type-content;
    uses protocol;
    container intrusion-attack-type {
      description
        "The attack type of intrusion";
      leaf brutal-force {
        type boolean;
        description
          "The intrusion type is
          brutal force";
      }
      leaf buffer-overflow {
        type boolean;
        description
          "The intrusion type is
          buffer overflow";
      }
    }
  }
  container botnet-event {
    description
      "If the event type is botnet event";
    uses i2nsf-nsf-event-type-content;
    uses protocol;
    leaf botnet-name {
      type string;
      description
       "The name of the detected botnet";
    }
    leaf role {
      type string;
      description
```

도면20u

```
      "The role of the communicating
       parties within the botnet";
  }
}
container web-attack-event {
  description
    "If the event type is web
    attack event";
  uses i2nsf-nsf-event-type-content;
  container web-attack-type {
    description
      "To determine the attack
      type";
    leaf sql-injection {
      type boolean;
      description
        "If the web attack type is
        sql injection";
    }
    leaf command-injection {
      type boolean;
      description
        "If the web attack type is
        command injection";
    }
    leaf xss {
      type boolean;
      description
        "If the web attack type is
        xss injection";
    }
    leaf csrf {
      type boolean;
      description
        "If the web attack type is
        csrf injection";
    }
  }
  container req-method {
    description
     "The method of requirement.
      For instance, PUT or GET
      in HTTP";
    leaf put{
      type boolean;
      description
        "If req method is PUT";
    }
```

도면20v

```
  leaf get {
   type boolean;
    description
      "If req method is GET";
  }
}
leaf req-url {
  type string;
  description
    "Requested URL";
}
leaf url-category {
  type string;
  description
    "Matched URL category";
}
container filtering-type {
  description
   "URL filtering type,
    e.g., Blacklist, Whitelist,
    User-Defined, Predefined,
    Malicious Category, Unknown";
  leaf blacklist {
    type boolean;
    description
      "The filtering type is
      blacklist";
  }
  leaf whitelist {
    type boolean;
    description
      "The filtering type is
      whitelist";
  }
  leaf user-defined {
    type boolean;
    description
      "The filtering type is
      user defined";
  }
  leaf balicious-category{
    type boolean;
    description
      "The filtering type is
      balicious category";
  }
  leaf unknown {
    type boolean;
```

도면20

```
            description
              "The filtering type is
              unknown";
          }
        }
      }
    }
  }
}
case log {
  description
    "If the message type is log";
  choice log-type {
    description
      "The type of log";
    case system-log{
      description
        "If the log type is system log";
      container access-logs {
        description
          "If the log is access logs
          in system log";
        leaf login-ip {
          type inet:ipv4-address;
          mandatory true;
          description
            "Login IP address of a user.";
        }
        leaf administartor {
          type string;
          description
            "Administrator that
            operates on the device";
        }
        leaf login-mode {
          type login-mode;
          description
            "Specifies the
            administrator logs in mode";
        }
        leaf operation-type {
          type operation-type;
          description
            "The operation type that
            the administrator execute";
        }
        leaf result {
          type string;
```

도면20x

```
    description
      "Command execution result";
  }
  leaf content {
    type string;
    description
      "Operation performed by
      an administrator after login.";
  }
}
container resource-utiliz-logs {
  description
    "If the log is resource utilize
    logs in system log";
  leaf system-status {
    type string;
    description
      "Running status of
      current system";
  }
  leaf cpu-usage {
    type uint8;
    description
      "specifies the amount of
      cpu usage";
  }
  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
```

도면20y

```
      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";
  }
}
container user-activity-logs {
  description
    "If the log is user activity
    logs in system log";
  leaf user {
    type string;
    mandatory true;
    description
      "Name of a user";
  }
  leaf group {
    type string;
    mandatory true;
    description
      "Group to which a user belongs.";
```

도면20z

```
}
leaf login-ip {
  type inet:ipv4-address;
  mandatory true;
  description
    "Login IP address of a user.";
}
uses authentication-mode;
container access-mode {
  description
    "User access mode. e.g., PPP, SVN, LOCAL";
  leaf ppp{
    type boolean;
    description
      "Access-mode : ppp";
  }
  leaf svn{
    type boolean;
    description
      "Access-mode : svn";
  }
  leaf local{
    type boolean;
    description
      "Access-mode : local";
  }
}
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";
}
leaf cause{
```

도면21a

```
      type string;
      description
         "Cause of a failed user activity";
      }
    }
}
case nsf-log{
  description
    "If the log type is nsf log";
  container ddos-logs {
    description
      "If the log is DDoS logs
      in nsf log";
    leaf attack-type{
      type string;
      description
        "DDoS";
    }
    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 all-action;
      description
```

도면21b

```
      "Action type: allow, alert,
      block, discard, declare,
      block-ip, block-service";
  }
  leaf os  {
    type string;
    description
      "simple os information";
  }
}
container virus-logs {
  description
    "If the log is virus logs
    in nsf log";
  uses protocol;
  leaf attack-type{
    type string;
    description
      "Virus";
  }
  leaf action{
    type all-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";
  }
}
container intrusion-logs {
  description
    "If the log is intrusion logs
    in nsf log";
  leaf attack-type{
    type string;
    description
      "Intrusion";
```

도면21c

```
  }
  leaf action{
    type all-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";
  }
}
container botnet-logs {
  description
    "If the log is botnet logs
    in nsf log";
  leaf attack-type{
    type string;
    description
      "Botnet";
  }
  leaf botnet-pkt-num{
    type uint8;
    description
      "The number of the packets
      sent to or from the
      detected botnet";
  }
  leaf action{
    type all-action;
    description
      "Action type: allow, alert,
      block, discard, declare,
      block-ip, block-service";
```

도면21d

```
  }
  leaf os{
    type string;
    description
      "simple os information";
  }
}
container dpi-logs {
  description
    "If the log is dpi logs
    in nsf log";
  leaf dpi-type{
    type dpi-type;
    description
      "The type of dpi";
  }
  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;
```

도면21e

```
    description
      "Source region of the traffic";
  }
  leaf dst-region{
    type string;
    description
      "Destination region of the traffic";
  }
  leaf policy-id {
    type uint8;
    mandatory true;
    description
      "The ID of the policy being triggered";
  }
  leaf policy-name {
    type string;
    mandatory true;
    description
      "The name of the policy being triggered";
  }
  leaf src-user{
    type string;
    description
      "User who generates traffic";
  }
  uses protocol;
  leaf file-type {
    type string;
    description
      "The type of file";
  }
  leaf file-name {
    type string;
    description
      "The name of file";
  }
}
list vulnerability-scanning-logs {
  key vulnerability-id;
  description
    "If the log is vulnerability
    scanning logs in nsf log";
  leaf vulnerability-id{
    type uint8;
    description
      "The vulnerability id";
  }
  leaf victim-ip {
```

도면21f

```
    type inet:ipv4-address;
    description
      "IP address of the victim
      host which has vulnerabilities";
  }
  uses protocol;
  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 additional-info{
    type string;
    description
      "Additional-info for logs.
      It includes The fix suggestion
      to the vulnerability.";
  }
}
container web-attack-logs {
  description
    "If the log is web attack
    logs in nsf log";
  leaf attack-type{
    type string;
    description
      "Web Attack";
  }
  leaf rsp-code{
    type string;
    description
      "Response code";
  }
  leaf req-clientapp{
    type string;
    description
      "The client application";
  }
```

도면21g

```
        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.";
        }
      }
    }
  }
}
case counters {
  description
    "If the message type is counters";
  choice counter-type {
    description
      "The type of counter";
    case system-counter {
      container interface-counters {
        description
          "The system counter type is
          interface counter";
        uses i2nsf-system-counter-type-content;
      }
    }
    case nsf-counter{
      container firewall-counters {
        description
          "The nsf counter type is
          firewall counter";
        uses i2nsf-nsf-counters-type-content;
        container bound{
          description
          "Inbound or Outbound";
          leaf in-interface {
            type boolean;
            description
```

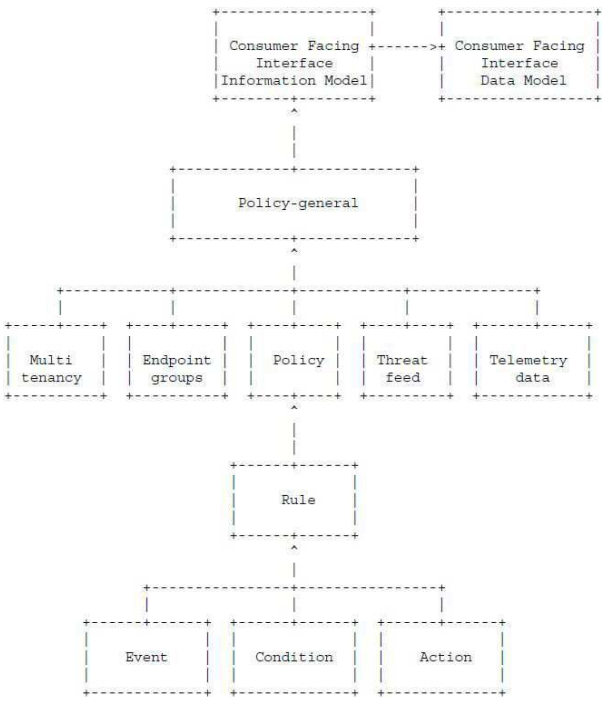
도면21h

```
                "If the bound is inbound";
            }
            leaf out-interface {
              type boolean;
              description
                "If the bound is outbound";
            }
          }
        }
      }
      container policy-hit-counters {
        description
          "The counters of policy hit";
        uses i2nsf-nsf-counters-type-content;
        leaf hit-times{
          type uint32;
          description
            "The hit times for policy";
        }
      }
    }
  }
}
leaf message {
  type string;
  mandatory true;
  description
    "This is a message for monitoring information";
}
leaf time-stamp {
  type yang:date-and-time;
  mandatory true;
  description
    "Indicate the time when the message is generated";
}
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 severity {
  type severity;
```

도면21i

```
        mandatory true;
        description
          "The severity of the alarm such as
          critical, high, middle, low.";
      }
    }
  }
}
<CODE ENDS>
```

도면22



도면22a

```
module: policy-general
    +--rw policy
    |  +--rw rule* [rule-id]
    |     +--rw rule-id                       uint16
    |     +--rw name?                         string
    |     +--rw date?                         yang:date-and-time
    |     +--rw case?                         string
    |     +--rw event* [event-id]
    |     |  +--rw event-id                   string
    |     |  +--rw name?                      string
    |     |  +--rw date?                      yang:date-and-time
    |     |  +--rw event-type?                string
    |     |  +--rw time-information?          string
    |     |  +--rw event-map-group?     -> /threat-feed/event-map-group
    |     |                                   /event-map-group-id
    |     |  +--rw enable?                    boolean
    |     +--rw condition* [condition-id]
    |     |  +--rw condition-id               string
    |     |  +--rw source?                    string
    |     |  +--rw destination?               string
    |     |  +--rw match?                     boolean
    |     |  +--rw match-direction?           string
    |     |  +--rw exception?                 string
    |     +--rw policy-action* [policy-action-id]
    |        +--rw policy-action-id           string
    |        +--rw name?                      string
    |        +--rw date?                      yang:date-and-time
    |        +--rw primary-action?            string
    |        +--rw secondary-action?          string
    |        +--rw owner?                     string
    +--rw multi-tenancy
    |  +--rw policy-domain* [policy-domain-id]
    |  |  +--rw policy-domain-id              uint16
    |  |  +--rw name                          string
    |  |  +--rw address?                      string
    |  |  +--rw contact                       string
    |  |  +--rw date                          yang:date-and-time
    |  |  +--rw policy-tenant* [policy-tenant-id]
    |  |  |  +--rw policy-tenant-id           uint16
```

도면22b

```
|  |  |  +--rw name                          string
|  |  |  +--rw date                          yang:date-and-time
|  |  |  +--rw domain?                  ->   /multi-tenancy
|  |  |                                      /policy-domain
|  |  |                                      /policy-domain-id
|  |  +--rw authentication-method?      ->   /multi-tenancy
|  |                                         /policy-mgnt-auth-method
|  |                                         /policy-mgnt-auth-method-id
|  +--rw policy-role* [policy-role-id]
|  |  +--rw policy-role-id                 uint16
|  |  +--rw name                           string
|  |  +--rw date                           yang:date-and-time
|  |  +--rw access-profile                 string
|  +--rw policy-user* [policy-user-id]
|  |  +--rw policy-user-id                 uint16
|  |  +--rw name                           string
|  |  +--rw date                           yang:date-and-time
|  |  +--rw password                       string
|  |  +--rw email                          string
|  |  +--rw scope-type?                    string
|  |  +--rw scope-reference?               string
|  |  +--rw role                           string
|  +--rw policy-mgnt-auth-method* [policy-mgnt-auth-method-id]
|     +--rw policy-mgnt-auth-method-id     uint16
|     +--rw name                           string
|     +--rw date                           yang:date-and-time
|     +--rw authentication-method          enumeration
|     +--rw mutual-authentication          boolean
|     +--rw token-server                   inet:ipv4-address
|     +--rw certificate-server             inet:ipv4-address
|     +--rw single-sing-on-server          inet:ipv4-address
+--rw endpoint-group
|  +--rw meta-data-source* [meta-data-source-id]
|  |  +--rw meta-data-source-id            uint16
|  |  +--rw name                           string
|  |  +--rw date                           yang:date-and-time
|  |  +--rw tag-type?                      boolean
|  |  +--rw tag-server-information?        inet:ipv4-address
|  |  +--rw tag-application-protocol?      string
|  |  +--rw tag-server-credential?         string
|  +--rw user-group* [user-group-id]
|  |  +--rw user-group-id                  uint16
|  |  +--rw name?                          string
|  |  +--rw date?                          yang:date-and-time
|  |  +--rw group-type?                    enumeration
|  |  +--rw meta-data-server?              inet:ipv4-address
|  |  +--rw group-member?                  string
|  |  +--rw risk-level?                    uint16
```

도면22c

```
|  +--rw device-group* [device-group-id]
|  |  +--rw device-group-id                uint16
|  |  +--rw name?                          string
|  |  +--rw date?                          yang:date-and-time
|  |  +--rw group-type?                    enumeration
|  |  +--rw meta-data-server?              inet:ipv4-address
|  |  +--rw group-member?                  string
|  |  +--rw risk-level?                    uint16
|  +--rw application-group* [application-group-id]
|  |  +--rw application-group-id           uint16
|  |  +--rw name?                          string
|  |  +--rw date?                          yang:date-and-time
|  |  +--rw group-type?                    enumeration
|  |  +--rw meta-data-server?              inet:ipv4-address
|  |  +--rw group-member?                  string
|  |  +--rw risk-level?                    uint16
|  +--rw location-group* [location-group-id]
|     +--rw location-group-id              uint16
|     +--rw name?                          string
|     +--rw date?                          yang:date-and-time
|     +--rw group-type?                    enumeration
|     +--rw meta-data-server?              inet:ipv4-address
|     +--rw group-member?                  string
|     +--rw risk-level?                    uint16
+--rw threat-feed
|  +--rw threat-feed* [threat-feed-id]
|  |  +--rw threat-feed-id                 uint16
|  |  +--rw name?                          string
|  |  +--rw date?                          yang:date-and-time
|  |  +--rw feed-type                      enumeration
|  |  +--rw feed-server?                   inet:ipv4-address
|  |  +--rw feed-priority?                 uint16
|  +--rw custom-list* [custom-list-id]
|  |  +--rw custom-list-id                 uint16
|  |  +--rw name?                          string
|  |  +--rw date?                          yang:date-and-time
|  |  +--rw list-type                      enumeration
|  |  +--rw list-property                  enumeration
|  |  +--rw list-content?                  string
|  +--rw malware-scan-group* [malware-scan-group-id]
|  |  +--rw malware-scan-group-id          uint16
|  |  +--rw name?                          string
|  |  +--rw date?                          yang:date-and-time
|  |  +--rw signature-server?              inet:ipv4-address
|  |  +--rw file-types?                    string
|  |  +--rw malware-signatures?            string
|  +--rw event-map-group* [event-map-group-id]
|     +--rw event-map-group-id             uint16
```

도면22d

```
|     +--rw name?                          string
|     +--rw date?                          yang:date-and-time
|     +--rw security-events?               string
|     +--rw threat-map?                    string
+--rw telemetry-data
   +--rw telemetry-data* [telemetry-data-id]
   |  +--rw telemetry-data-id               uint16
   |  +--rw name?                           string
   |  +--rw date?                           yang:date-and-time
   |  +--rw logs?                           boolean
   |  +--rw syslogs?                        boolean
   |  +--rw snmp?                           boolean
   |  +--rw sflow?                          boolean
   |  +--rw netflow?                        boolean
   |  +--rw interface-stats?                boolean
   +--rw telemetry-source* [telemetry-source-id]
   |  +--rw telemetry-source-id             uint16
   |  +--rw name?                           string
   |  +--rw date?                           yang:date-and-time
   |  +--rw source-type?                    enumeration
   |  +--rw nsf-source?                     inet:ipv4-address
   |  +--rw nsf-credentials?                string
   |  +--rw collection-interval?            uint16
   |  +--rw collection-method?              enumeration
   |  +--rw heartbeat-interval?             uint16
   |  +--rw qos-marking?                    uint16
   +--rw telemetry-destination* [telemetry-destination-id]
      +--rw telemetry-destination-id       uint16
      +--rw name?                          string
      +--rw date?                          yang:date-and-time
      +--rw collector-source?              inet:ipv4-address
      +--rw collector-credentials?         string
      +--rw data-encoding?                 string
      +--rw data-transport?                enumeration
```

도면23a

```
module ietf-policy-general {
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-policy-general";
  prefix
    cf-interface;

  import ietf-yang-types{
    prefix yang;
  }

  import ietf-inet-types{
    prefix inet;
  }

  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: Adrian Farrel
     <mailto:Adrain@olddog.co.uk>

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

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

  description
    "This module defines a YANG data module for consumer-facing
     interface to security controller.";

  revision "2018-07-02"{
    description "fourth revision";
    reference
      "draft-kumar-i2nsf-client-facing-interface-im-04";
  }

  //Groupings
  container policy {
    description
    "This object is a policy instance to have
    complete information such as where and when
    a policy need to be applied.";

    list rule {
     key "rule-id";
```

도면23b

```
leaf rule-id {
type uint16;
description
"This is ID for rules.";
}
 description
 "This is a container for rules.";
 leaf name {
  type string;
   description
    "This field idenfifies the name of this object.";
 }

 leaf date {
  type yang:date-and-time;
   description
    "Date this object was created or last
     modified";
 }

 leaf case {
  type string;
   description
    "to identify whether the rule belongs to
    web filter or enterprise mode.";
 }

 list event {
  key "event-id";
  description
  "This represents the security event of a
  policy-rule.";

  leaf event-id {
  type string;
  mandatory true;
  description
  "This represents the event-id.";
  }

  leaf name {
   type string;
    description
     "This field idenfifies the name of this object.";
  }

  leaf date {
   type yang:date-and-time;
```

도면23c

```
  description
   "Date this object was created or last
    modified";
 }

 leaf event-type {
  type string;
   description
    "This field identifies the event of
    policy enforcement trigger type.";
 }

 leaf time-information {
  type string;
  description
   "This field contains time calendar such as
   BEGIN-TIME and END-TIME for one time
   enforcement or recurring time calendar for
   periodic enforcement.";
 }

 leaf event-map-group {
  type leafref {
  path "/threat-feed/event-map-group/event-map-group-id";
  }
  description
  "This field contains security events or threat
  map in order to determine when a policy need
  to be activated. This is a reference to
  Evnet-Map-Group.";
 }

 leaf enable {
  type boolean;
   description
    "This determines whether the condition
    matches the security event or not.";
  }
}

list condition {
 key "condition-id";
 description
 "This represents the condition of a
 policy-rule.";

 leaf condition-id {
  type string;
```

도면23d

```
 description
 "This represents the condition-id.";
}

leaf source {
 type string;
  description
  "This field identifies the source of
  the traffic. This could be reference to
  either 'Policy Endpoint Group' or
  'Threat-Feed' or 'Custom-List' if Security
  Admin wants to specify the source; otherwise,
  the default is to match all traffic.";
}

leaf destination {
 type string;
  description
   "This field identifies the source of
    the traffic. This could be reference to
    either 'Policy Endpoint Group' or
    'Threat-Feed' or 'Custom-List' if Security
    Admin wants to specify the source; otherwise,
    the default is to match all traffic.";
}

leaf match {
 type boolean;
  description
   "This field identifies the match criteria used to
   evaluate whether the specified action need to be
   taken or not.  This could be either a Policy-
   Endpoint-Group identifying a Application set or a
   set of traffic rules.";
}

leaf match-direction {
 type string;
  description
   "This field identifies if the match criteria is
   to evaluated for both direction of the traffic or
   only in one direction with default of allowing in
   the other direction for stateful match conditions.
   This is optional and by default rule should apply
   in both directions.";
}

leaf exception {
```

도면23e

```
  type string;
   description
    "This field identifies the exception
    consideration when a rule is evaluated for a
    given communication.  This could be reference to
    Policy-Endpoint-Group object or set of traffic
    matching criteria.";
 }
}

list policy-action {
 key "policy-action-id";

 leaf policy-action-id {
 type string;
 mandatory true;
  description
   "this represents the policy-action-id.";
 }
 description
  "This object represents actions that a
  Security Admin wants to perform based on
  a certain traffic class.";

 leaf name {
  type string;
   description
    "The name of the policy-action object.";
 }

 leaf date {
  type yang:date-and-time;
   description
    "When the object was created or last
    modified.";
 }

 leaf primary-action {
  type string;
   description
    "This field identifies the action when a rule
    is matched by NSF. The action could be one of
    'PERMIT', 'DENY', 'RATE-LIMIT', 'TRAFFIC-CLASS',
    'AUTHENTICATE-SESSION', 'IPS, 'APP-FIREWALL', etc.";
 }

 leaf secondary-action {
  type string;
```

도면23f

```
      description
       "This field identifies additional actions if
       a rule is matched. This could be one of 'LOG',
       'SYSLOG', 'SESSION-LOG', etc.";
     }

     leaf owner {
      type string;
      description
       "This field defines the owner of this
       policy. Only the owner is authorized to
       modify the contents of the policy.";
     }
    }
   }
}

    container multi-tenancy {
       description
         "The descriptions of multi-tenancy.";

       list policy-domain {
         key "policy-domain-id";

         leaf policy-domain-id {
           type uint16;
           description
             "This represents the list of domains.";
         }
         description
         "this represent the list of policy domains";
         leaf name {
           type string;
           mandatory true;
           description
             "Name of the organization or customer representing
             this domain.";
         }

         leaf address {
           type string;
           description
             "address of an organization or customer.";
         }

         leaf contact {
           type string;
```

도면23g

```
  mandatory true;
  description
    "contact information of the organization
     or customer.";
}

leaf date {
  type yang:date-and-time;
  mandatory true;
  description
    "The date when this account was created
    or last modified.";
}
list policy-tenant {
  key "policy-tenant-id";
  leaf policy-tenant-id {
    type uint16;
    description
      "The policy tenant id.";
  }
  description
  "This represents the list of tenants";

  leaf name {
    type string;
    mandatory true;
    description
      "Name of the Department or Division within
       an organization.";
  }

  leaf date {
    type yang:date-and-time;
    mandatory true;
    description
      "Date this account was created or last modified.";
  }

  leaf domain {
    type leafref {
      path "/multi-tenancy/policy-domain/policy-domain-id";
    }
    description
    "This field identifies the domain to which this
    tenant belongs. This should be reference to a
    'Policy-Domain' object.";
  }
}
```

도면23h

```
  leaf authentication-method {
    type leafref {
      path "/multi-tenancy/policy-mgmt-auth-method/policy-mgmt-auth-method-id";
    }

    description
    "Authentication method to be used for this domain.
    It should be a reference to a 'policy-mgmt-auth-method'
    object.";
  }
}

list policy-role {
  key "policy-role-id";

  leaf policy-role-id {
  type uint16;
  mandatory true;
  description
    "This defines a set of permissions assigned
    to a user in an organization that want to manage
    its own Security Policies.";
  }
  description
  "This represents the list of policy roles.";

  leaf name {
    type string;
    mandatory true;
    description
      "This field identifies name of the role.";
  }

  leaf date {
    type yang:date-and-time;
    mandatory true;
    description
      "Date this role was created or last modified.";
  }

  leaf access-profile {
    type string;
    mandatory true;
    description
      "This field identifies the access profile for the
      role. The profile grants or denies access to policy
      objects.  Multiple access profiles can be
      concatenated together.";
```

도면23i

```
  }
}

list policy-user {
  key "policy-user-id";

  leaf policy-user-id {
  type uint16;
  description
    "This represents the policy-user-id.";
  }
  description
  "This represents the list of policy users.";
  leaf name {
    type string;
    mandatory true;
    description
      "The name of a user.";
  }

  leaf date {
    type yang:date-and-time;
    mandatory true;
    description
      "Date this user was created or last modified";
  }

  leaf password {
    type string;
    mandatory true;
    description
      "User password for basic authentication";
  }

  leaf email {
    type string;
    mandatory true;
    description
      "The email account of a user";
  }

  leaf scope-type {
    type string;
    description
      "identifies whether a user has domain-wide
      or tenant-wide privileges";
  }
```

도면23j

```
  leaf scope-reference {
    type string;
    description
      "This references policy-domain or policy-tenant
      to identify the scope.";
  }

  leaf role {
    type string;
    mandatory true;
    description
      "This references policy-role to define specific
      permissions";
  }
}

list policy-mgnt-auth-method {
  key "policy-mgnt-auth-method-id";

  leaf policy-mgnt-auth-method-id {
  type uint16;
  description
    "This represents the authentication method id.";
  }
  description
  "The descriptions of policy management
    authentication methods.";
  leaf name {
    type string;
    mandatory true;
    description
      "name of the authentication method";
  }

  leaf date {
    type yang:date-and-time;
    mandatory true;
    description
      "date when the authentication method
      was created";
  }

  leaf authentication-method {
    type enumeration{
      enum password{
        description
          "password-based authentication.";
      }
```

도면23k

```
      enum token{
        description
          "token-based authentication.";
      }
      enum certificate{
        description
          "certificate-based authentication.";
      }
    }
    mandatory true;
    description
      "The description of authentication method;
      token-based, password, certificate,
      single-sign-on";
  }

  leaf mutual-authentication {
    type boolean;
    mandatory true;
    description
      "To identify whether the authentication
       is mutual";
  }

  leaf token-server {
    type inet:ipv4-address;
    mandatory true;
    description
      "The token-server information if the
      authentication method is token-based";
  }

  leaf certificate-server {
    type inet:ipv4-address;
    mandatory true;
    description
      "The certificate-server information if
      the authentication method is certificate-based";
  }

  leaf single-sing-on-server {
    type inet:ipv4-address;
    mandatory true;
    description
      "The single-sign-on-server information
      if the authentication method is
      single-sign-on-based";
  }
```

도면231

```
      }
    }
container endpoint-group {
      description
        "A logical entity in their business
        environment, where a security policy
        is to be applied.";

      list meta-data-source {
        key "meta-data-source-id";
        leaf meta-data-source-id {
        type uint16;
        mandatory true;
        description
          "This represents the meta-data source id.";
        }
        description
        "This represents the meta-data source.";

        leaf name {
          type string;
          mandatory true;
          description
            "This identifies the name of the
            meta-datas-ource.";
        }

        leaf date {
          type yang:date-and-time;
          mandatory true;
          description
            "This identifies the date this object was
            created or last modified.";
        }

        leaf tag-type {
          type boolean;
          description
            "This identifies the group type; user group,
            app group or device group.";
        }

        leaf tag-server-information {
          type inet:ipv4-address;
          description
            "The description of authentication method;
            token-based, password, certificate,
            single-sign-on";
```

도면23m

```
  }

  leaf tag-application-protocol {
    type string;
    description
      "This filed identifies the protocol e.g. LDAP,
      Active Directory, or CMDB";
  }

  leaf tag-server-credential {
    type string;
    description
      "This field identifies the credential
      information needed to access the tag server";
  }
}

list user-group{
  key "user-group-id";

  leaf user-group-id {
  type uint16;
  mandatory true;
  description
    "This represents the the user group id.";
  }
  description
  "This represents the user group.";

  leaf name {
    type string;
    description
      "This field identifies the name of user-group.";
  }

  leaf date {
    type yang:date-and-time;
    description
      "when this user-group was created or last modified.";
  }

  leaf group-type {
    type enumeration{
      enum user-tag{
        description
          "The user group is based on user-tag.";
      }
      enum user-name{
```

도면23n

```
        description
          "The user group is based on user-name.";
        }
        enum ip-address{
          description
            "The user group is based on ip-address.";
        }
      }

      description
        "This describes the group type; User-tag,
        User-name or IP-address.";
    }

    leaf meta-data-server {
      type inet:ipv4-address;
      description
        "This references metadata source";
    }

    leaf group-member {
      type string;
      description
        "This describes the user-tag information";
    }

    leaf risk-level {
      type uint16;
      description
        "This represents the threat level; valid range
         may be 0 to 9.";
    }
  }

  list device-group {
    key "device-group-id";
    leaf device-group-id {
    type uint16;
     description
      "This represents a device group id.";
    }
    description
      "This represents a device group.";
    leaf name {
      type string;
    description
        "This field identifies the name of
        a device-group.";
```

도면23o

```
    }
    leaf date {
      type yang:date-and-time;
      description
      "The date when this group was create or
      last modified.";
    }

    leaf group-type {
      type enumeration{
        enum device-tag{
          description
            "The device group is based on device-tag.";
        }
        enum device-name{
          description
            "The device group is based on device-name.";
        }
        enum ip-address{
          description
            "The device group is based on ip-address.";
        }
      }
      description
        "This describes the group type; device-tag,
        device-name or IP-address.";
    }

    leaf meta-data-server {
      type inet:ipv4-address;
      description
        "This references meta-data-source
        object.";
    }

    leaf group-member {
      type string;
      description
        "This describes the device-tag, device-name or
        IP-address information";
    }

    leaf risk-level {
      type uint16;
      description
        "This represents the threat level; valid range
        may be 0 to 9.";
    }
```

도면23p

```
}
list application-group{
  key "application-group-id";
  leaf application-group-id {
  type uint16;
  description
  "This represents an application group id.";
  }
  description
  "This represents an application group.";
  leaf name {
    type string;
    description
    "This field identifies the name of
    an application group";
  }

  leaf date {
    type yang:date-and-time;
    description
    "The date when this group was created or
    last modified.";
  }

  leaf group-type {
    type enumeration{
      enum application-tag{
        description
          "The application group is based on application-tag.";
      }
      enum device-name{
        description
          "The application group is based on application-name.";
      }
      enum ip-address{
        description
          "The application group is based on ip-address.";
      }
    }
    description
      "This identifies the group type;
      application-tag, application-name or
      IP-address.";
  }

  leaf meta-data-server {
    type inet:ipv4-address;
```

도면23q

```
    description
      "This references meta-data-source
      object.";
  }

  leaf group-member {
    type string;
    description
      "This describes the application-tag,
      application-name or IP-address information";
  }

  leaf risk-level {
    type uint16;
    description
      "This represents the threat level; valid range
       may be 0 to 9.";
  }
}
list location-group{
  key "location-group-id";
  leaf location-group-id {
  type uint16;
  description
  "This represents a location group id.";
  }
  description
  "This represents a location group.";

  leaf name {
    type string;
    description
    "This field identifies the name of
    a location group";

  }

  leaf date {
    type yang:date-and-time;
    description
    "The date when this group was created or
    last modified.";
  }

  leaf group-type {
    type enumeration{
      enum location-tag{
```

도면23r

```
            description
              "The location group is based on location-tag.";
          }
          enum location-name{
            description
              "The location group is based on location-name.";
          }
          enum ip-address{
            description
              "The location group is based on ip-address.";
          }
        }
        description
          "This identifies the group type;
          location-tag, location-name or
          IP-address.";
      }

      leaf meta-data-server {
        type inet:ipv4-address;
        description
          "This references meta-data-source
          object.";
      }

      leaf group-member {
        type string;
        description
          "This describes the location-tag,
          location-name or IP-address information";
      }

      leaf risk-level {
        type uint16;
        description
          "This represents the threat level; valid range
           may be 0 to 9.";
      }
    }
}

container threat-feed {
  description
  "this describes the list of threat-feed.";

    list threat-feed {
      key "threat-feed-id";
      leaf threat-feed-id {
```

도면23s

```
type uint16;
mandatory true;
description
  "This represents the threat-feed-id.";
}
description
  "This represents the threat feed within the
  threat-prevention-list.";
leaf name {
  type string;
  description
    "Name of the theat feed.";
}

leaf date {
  type yang:date-and-time;
  description
    "when the threat-feed was created.";
}

leaf feed-type {
  type enumeration {
    enum unknown {
      description
        "feed-type is unknown.";
    }
    enum ip-address {
      description
        "feed-type is IP address.";
    }
    enum url {
      description
        "feed-type is URL.";
    }
  }
  mandatory true;
  description
    "This determined whether the feed-type is IP address
    based or URL based.";
}

leaf feed-server {
  type inet:ipv4-address;
  description
    "this contains threat feed server information.";
}

leaf feed-priority {
```

도면23t

```
      type uint16;
      description
        "this describes the priority of the threat from
        0 to 5, where 0 means the threat is minimum and
        5 meaning the maximum.";
    }
}

list custom-list {
    key "custom-list-id";
    leaf custom-list-id {
    type uint16;
    description
    "this describes the custom-list-id.";
    }
    description
    "this describes the threat-prevention custom list.";
    leaf name {
      type string;
      description
        "Name of the custom-list.";
    }

    leaf date {
      type yang:date-and-time;
      description
        "when the custom list was created.";
    }

    leaf list-type {
      type enumeration {
        enum unknown {
          description
            "list-type is unknown.";
        }
        enum ip-address {
          description
            "list-type is IP address.";
        }
        enum mac-address {
          description
            "list-type is MAC address.";
        }
        enum url {
          description
            "list-type is URL.";
        }
      }
```

도면23u

```
      mandatory true;
      description
        "This determined whether the feed-type is IP address
        based or URL based.";
    }

    leaf list-property {
      type enumeration {
        enum unknown {
          description
            "list-property is unknown.";
        }
        enum blacklist {
          description
            "list-property is blacklist.";
        }
        enum whitelist {
          description
            "list-property is whitelist.";
        }
      }
      mandatory true;
      description
        "This determined whether the list-type is blacklist
        or whitelist.";
    }

    leaf list-content {
      type string;
      description
        "This describes the contents of the custom-list.";
    }
}

list malware-scan-group {
    key "malware-scan-group-id";
    leaf malware-scan-group-id {
    type uint16;
    mandatory true;
    description
    "This is the malware-scan-group-id.";
    }
    description
    "This represents the malware-scan-group.";
    leaf name {
      type string;
      description
        "Name of the malware-scan-group.";
```

도면23v

```
    }

    leaf date {
      type yang:date-and-time;
      description
        "when the malware-scan-group was created.";
    }

    leaf signature-server {
      type inet:ipv4-address;
      description
        "This describes the signature server of the
        malware-scan-group.";
    }

    leaf file-types {
      type string;
      description
        "This contains a list of file types needed to
        be scanned for the virus.";
    }

    leaf malware-signatures {
      type string;
      description
        "This contains a list of malware signatures or hash.";
    }
  }

  list event-map-group {
      key "event-map-group-id";
      leaf event-map-group-id {
      type uint16;
      mandatory true;
      description
      "This is the event-map-group-id.";
      }
      description
      "This represents the event map group.";

      leaf name {
        type string;
        description
          "Name of the event-map.";
      }

      leaf date {
        type yang:date-and-time;
```

도면23w

```
        description
          "when the event-map was created.";
      }

      leaf security-events {
        type string;
        description
          "This contains a list of security events.";
      }

      leaf threat-map {
        type string;
        description
          "This contains a list of threat levels.";
      }
  }
}

container telemetry-data {
  description
  "Telemetry provides visibility into the network
  activities which can be tapped for further
  security analytics, e.g., detecting potential
  vulnerabilities, malicious activities, etc.";

  list telemetry-data {
    key "telemetry-data-id";

    leaf telemetry-data-id {
    type uint16;
    mandatory true;
    description
    "This is ID for telemetry-data-id.";
    }
    description
    "This is ID for telemetry-data.";

    leaf name {
      type string;
      description
        "Name of the telemetry-data object.";
    }

    leaf date {
      type yang:date-and-time;
      description
        "This field states when the telemery-data
        object was created.";
```

도면23x

```
    }

    leaf logs {
      type boolean;
      description
        "This field identifies whether logs
        need to be collected.";
    }

    leaf syslogs {
      type boolean;
      description
        "This field identifies whether System logs
        need to be collected.";
    }

    leaf snmp {
      type boolean;
      description
        "This field identifies whether 'SNMP traps' and
        'SNMP alarms' need to be collected.";
    }

    leaf sflow {
      type boolean;
      description
        "This field identifies whether 'sFlow' data
        need to be collected.";
    }

    leaf netflow {
      type boolean;
      description
        "This field identifies whether 'NetFlow' data
        need to be collected.";
    }

    leaf interface-stats {
      type boolean;
      description
        "This field identifies whether 'Interface' data
        such as packet bytes and counts need to be
        collected.";
    }
  }

  list telemetry-source {
    key "telemetry-source-id";
```

도면23y

```
leaf telemetry-source-id {
type uint16;
mandatory true;
description
"This is ID for telemetry-source-id.";
}
description
"This is ID for telemetry-source.";

leaf name {
  type string;
  description
    "This identifies the name of this object.";
}

leaf date {
  type yang:date-and-time;
    description
      "Date this object was created or last modified";
}

leaf source-type {
  type enumeration {
    enum network-nsf {
      description
        "NSF telemetry source type is network-nsf.";
    }

    enum firewall-nsf {
      description
        "NSF telemetry source type is firewall-nsf.";
    }
    enum ids-nsf {
      description
        "NSF telemetry source type is ids-nsf.";
    }
    enum ips-nsf {
      description
        "NSF telemetry source type is ips-nsf.";
    }
    enum proxy-nsf {
      description
        "NSF telemetry source type is proxy-nsf.";
    }
    enum other-nsf {
      description
        "NSF telemetry source type is other-nsf.";
    }
```

도면23z

```
    }
    description
      "This should have one of the following type of
      the NSF telemetry source: NETWORK-NSF,
      FIREWALL-NSF, IDS-NSF, IPS-NSF,
      PROXY-NSF, VPN-NSF, DNS, ACTIVE-DIRECTORY,
      IP Reputation Authority, Web Reputation
      Authority, Anti-Malware Sandbox, Honey Pot,
      DHCP, Other Third Party, ENDPOINT";
  }

  leaf nsf-source {
    type inet:ipv4-address;
     description
      "This field contains information such as
      IP address and protocol (UDP or TCP) port
      number of the NSF providing telemetry data.";
  }

  leaf nsf-credentials {
    type string;
    description
      "This field contains username and password
      to authenticate with the NSF.";
  }

  leaf collection-interval {
    type uint16;
    units seconds;
    default 5000;
    description
      "This field contains time in milliseconds
       between each data collection. For example,
       a value of 5000 means data is streamed to
       collector every 5 seconds. Value of 0 means
       data streaming is event-based";
  }

  leaf collection-method {
    type enumeration {
      enum unknown {
        description
          "collection-method is unknown.";
      }
      enum push-based {
        description
          "collection-method is PUSH-based.";
      }
```

도면23za

```
      enum pull-based {
        description
          "collection-method is PULL-based.";
      }
    }
    description
    "This field contains a method of collection,
    i.e., whether it is PUSH-based or PULL-based.";
  }
  leaf heartbeat-interval {
    type uint16;
    units seconds;
    description
    "time in seconds the source sends telemetry
    heartbeat.";
  }

  leaf qos-marking {
    type uint16;
    description
    "DSCP value must be contained in this field.";
  }
}

list telemetry-destination {
  key "telemetry-destination-id";

  leaf telemetry-destination-id {
  type uint16;
  description
  "this represents the telemetry-destination-id";
  }
  description
  "This object contains information related to
  telemetry destination. The destination is
  usually a collector which is either a part of
  Security Controller or external system
  such as Security Information and Event
  Management (SIEM).";

  leaf name {
    type string;
    description
      "This identifies the name of this object.";
  }

  leaf date {
    type yang:date-and-time;
```

도면23zb

```
        description
          "Date this object was created or last
          modified";
      }

        leaf collector-source {
          type inet:ipv4-address;
            description
              "This field contains information such as
              IP address and protocol (UDP or TCP) port
              number for the collector's destination.";
        }

        leaf collector-credentials {
          type string;
            description
            "This field contains the username and
           password for the collector.";
        }

        leaf data-encoding {
          type string;
           description
            "This field contains the telemetry data encoding
            in the form of schema.";
        }

        leaf data-transport {
          type enumeration{
            enum grpc {
              description
                "telemetry data protocol is grpc.";
            }
            enum buffer-over-udp{
              description
                "telemetry data protocol is buffer over UDP.";
            }
          }
          description
          "This field contains streaming telemetry data
          protocols. This could be gRPC, protocol
          buffer over UDP, etc.";
        }
    }
  }
}
<CODE ENDS>
```

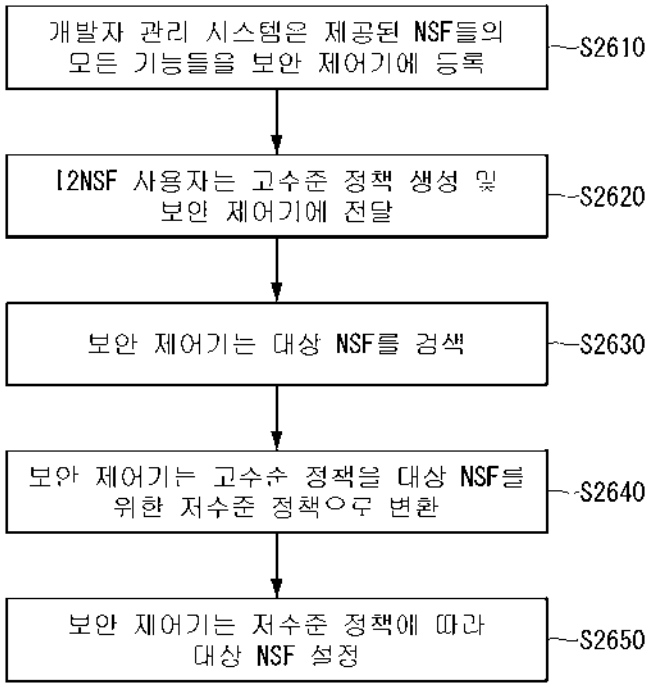
도면24

```
<I2NSF>
  <rule-name>block_web</rule-name>
  <condition>
    <src>Son's_PC</src>
    <dest>malicious<dest>
  </condition>
  <action>block</action>
</I2NSF>
```

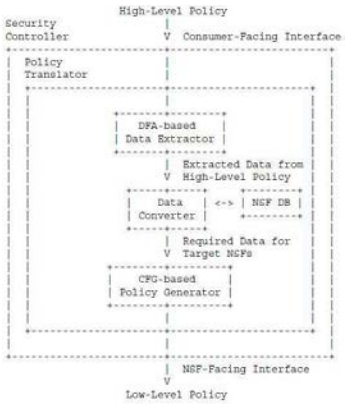
도면25

```
<I2NSF>
  <rule-name>block_web</rule-name>
  <rules>
    <condition>
      <packet-condition>
        <ipv4>10.0.0.1</ipv4>
        <ipv4>10.0.0.3</ipv4>
      </packet-condition>
    </condition>
    <payload>
      <content>www.malicious.com</content>
      <content>www.illegal.com</content>
    </payload>
    <action>drop</action>
  </rules>
</I2NSF>
```

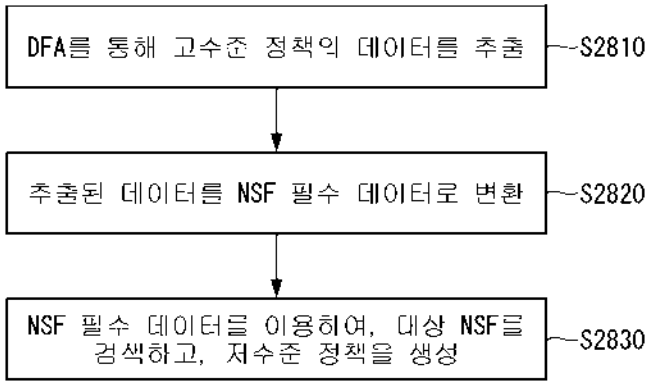
도면26


개발자 관리 시스템은 제공된 NSF들의 모든 기능들을 보안 제어기에 등록
S2610
I2NSF 사용자는 고수준 정책 생성 및 보안 제어기에 전달
S2620
보안 제어기는 대상 NSF를 검색
S2630
보안 제어기는 고수준 정책을 대상 NSF를 위한 저수준 정책으로 변환
S2640
보안 제어기는 저수준 정책에 따라 대상 NSF 설정
S2650

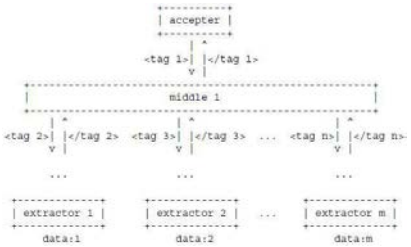
도면27


High-Level Policy
Security Controller
Consumer-Facing Interface
Policy Translator
DFA-based Data Extractor
Extracted Data from High-Level Policy
Data Converter
NSF DB
Required Data for Target NSFs
CFG-based Policy Generator
NSF-Facing Interface
Low-Level Policy

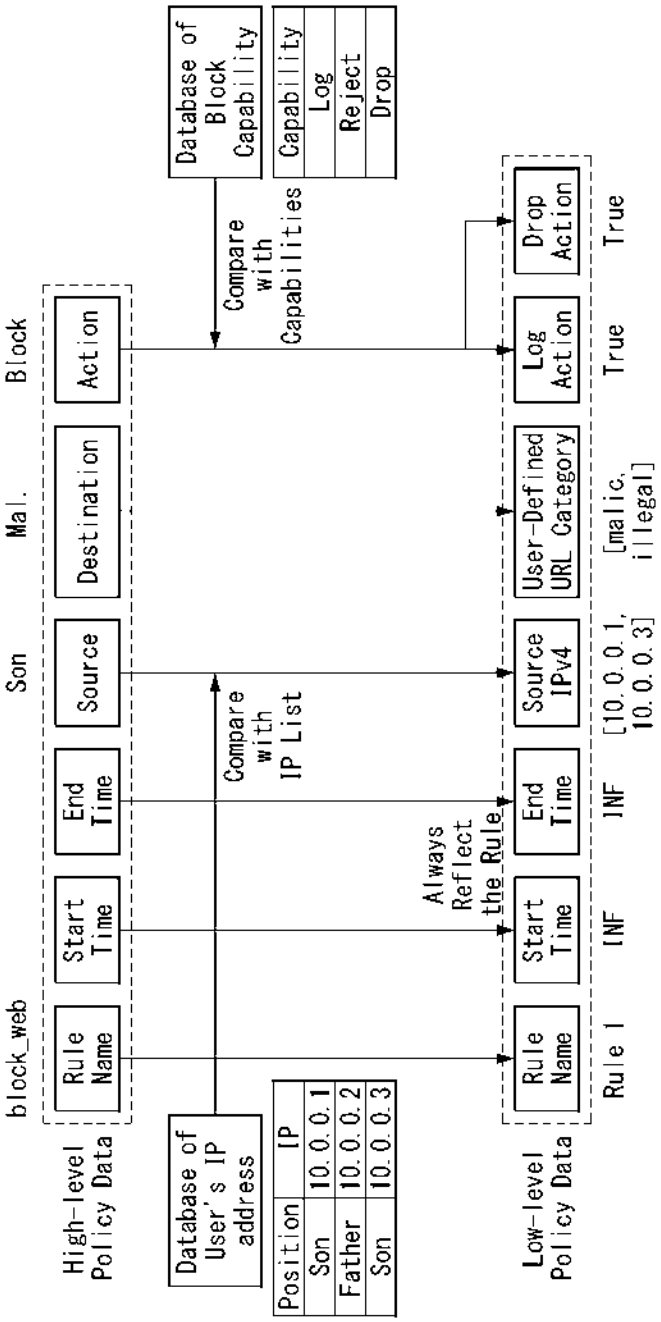
도면28


DFA를 통해 고수준 정책의 데이터를 추출
S2810
추출된 데이터를 NSF 필수 데이터로 변환
S2820
NSF 필수 데이터를 이용하여, 대상 NSF를 검색하고, 저수준 정책을 생성
S2830

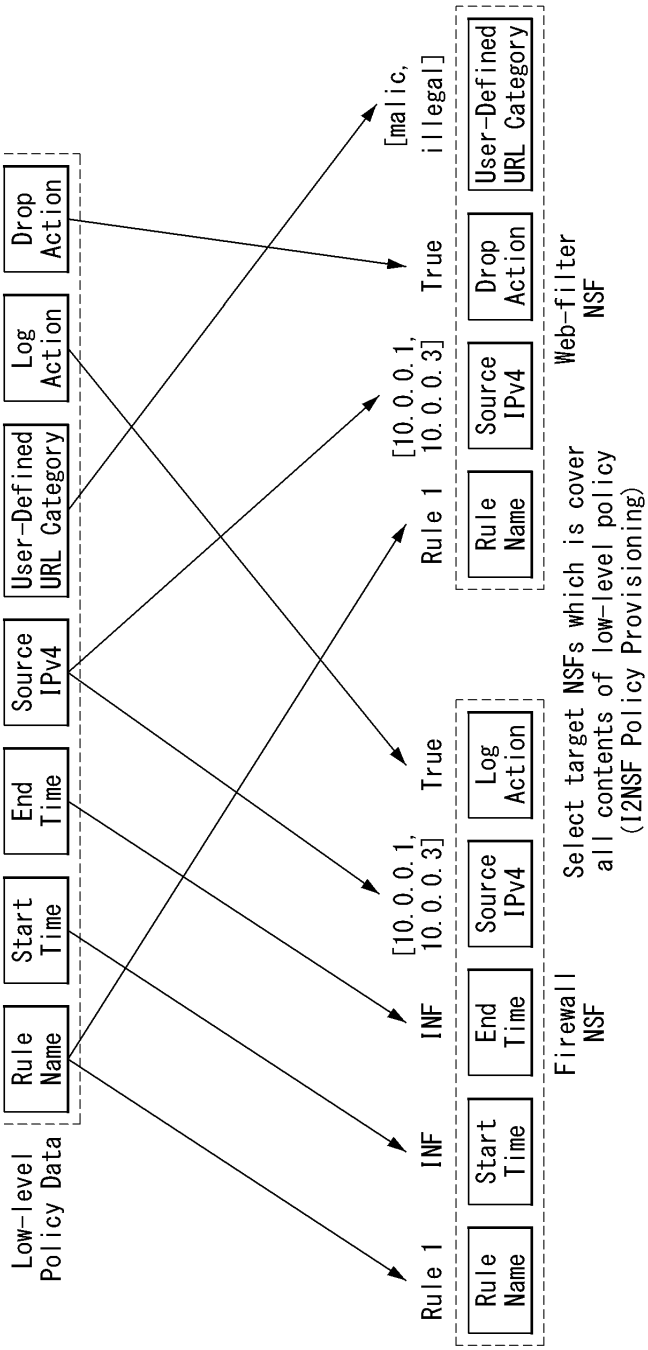
도면29


accepter
middle 1
extractor 1
extractor 2
extractor m
data:1
data:2
data:m

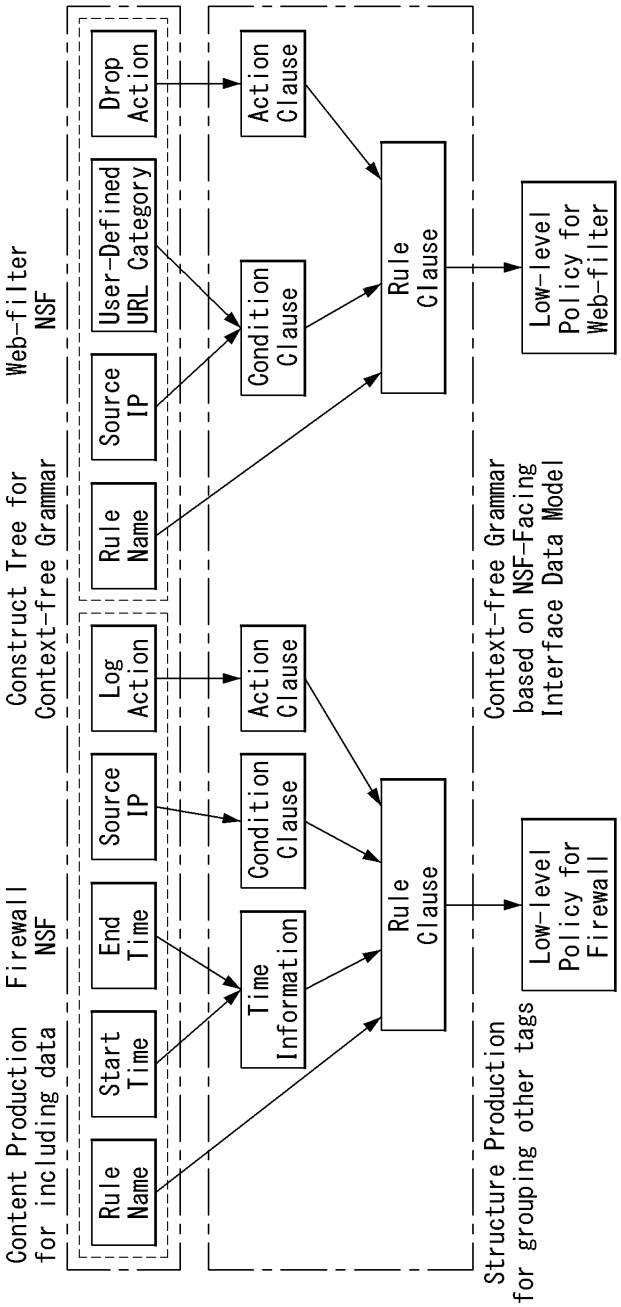
도면30


High-level Policy Data
block_web
Son
Mal.
Block
Rule Name
Start Time
End Time
Source
Destination
Action
Database of User's IP address
Position
IP
Son
10.0.0.1
Father
10.0.0.2
Son
10.0.0.3
Compare with IP List
Compare with Capabilities
Database of Block Capability
Capability
Log
Reject
Drop
Always Reflect the Rule
Low-level Policy Data
Rule Name
Start Time
End Time
Source IPv4
User-Defined URL Category
Log Action
Drop Action
Rule 1
INF
INF
[10.0.0.1, 10.0.0.3]
[malic, illegal]
True
True
True

도면31


Low-level Policy Data
Rule Name
Start Time
End Time
Source IPv4
User-Defined URL Category
Log Action
Drop Action
Rule 1
INF
INF
[10.0.0.1, 10.0.0.3]
[malic, illegal]
True
True
Firewall NSF
Rule Name
Start Time
End Time
Source IPv4
Log Action
Rule 1
INF
INF
[10.0.0.1, 10.0.0.3]
True
Web-filter NSF
Rule Name
Source IPv4
Drop Action
User-Defined URL Category
Rule 1
[10.0.0.1, 10.0.0.3]
True
[malic, illegal]
Select target NSFs which is cover all contents of low-level policy
(I2NSF Policy Provisioning)

도면32


Content Production for including data
Firewall NSF
Rule Name
Start Time
End Time
Source IP
Log Action
Time Information
Condition Clause
Action Clause
Rule Clause
Low-level Policy for Firewall
Construct Tree for Context-free Grammar
Web-filter NSF
Rule Name
Source IP
User-Defined URL Category
Drop Action
Condition Clause
Action Clause
Rule Clause
Low-level Policy for Web-filter
Structure Production for grouping other tags
Context-free Grammar based on NSF-Facing Interface Data Model

(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2019-0049579
(43) 공개일자 2019년05월09일

(51) 국제특허분류(Int. Cl.)
H04L 29/06 (2006.01) *H04L 12/24* (2006.01)
(52) CPC특허분류
H04L 63/20 (2013.01)
H04L 41/145 (2013.01)
(21) 출원번호 10-2018-0131256
(22) 출원일자 2018년10월30일
심사청구일자 2018년10월30일
(30) 우선권주장
1020170142840 2017년10월30일 대한민국(KR)

(71) 출원인
성균관대학교산학협력단
경기도 수원시 장안구 서부로 2066 (천천동, 성균관대학교내)
(72) 발명자
정재훈
부산광역시 금정구 금강로 225, 109동 1603호(장전동, 벽산블루밍디자인시티)
현상원
서울특별시 동작구 보라매로5길 51, 2105호(신대방동, 롯데타워)
(뒷면에 계속)
(74) 대리인
특허법인로얄

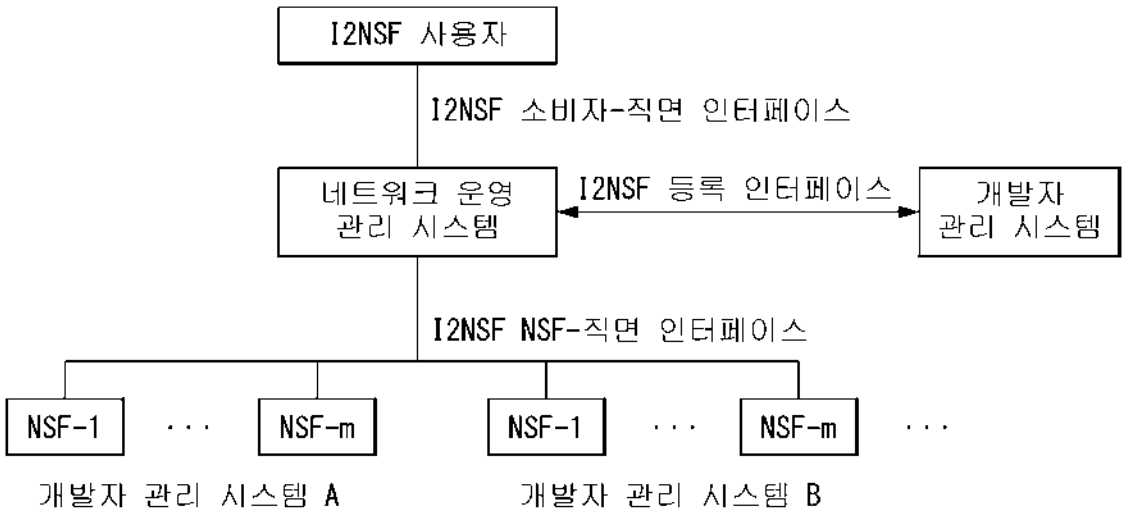
전체 청구항 수 : 총 9 항

(54) 발명의 명칭 **네트워크 보안 서비스를 제공하기 위한 방법 및 이를 위한 장치**

(57) 요 약

본 발명에서는 등록 인터페이스(Registration Interface)를 통해 네트워크 보안 기능(NSF: Network Secure Function)을 관리하는 보안 관리 시스템이 개시된다. 구체적으로, 보안 컨트롤러(Security Controller)에 의해 수행되는 방법은, 상기 보안 관리 시스템에 필요한 NSF에 대한 인스턴스화(instantiation) 요청 메시지를 개발자 관리 시스템(Developer's Management System)으로 전송하는 단계; 및 상기 요청 메시지에 대한 응답으로 상기 필요한 NSF에 대한 NSF 인스턴스(instance)의 등록을 지시하는 등록 메시지를 상기 개발자 관리 시스템으로부터 수신하는 단계를 포함하되, 상기 NSF 인스턴스는 상기 인스턴스화 요청 메시지에 기초하여 개발자 관리 시스템에 의해 생성될 수 있다.

대 표 도 - 도1



(72) 발명자

노태균

경기도 수원시 영통구 봉영로1482번길 18(영통동, 영통3차 풍림아이원아파트)

위사랑

서울특별시 송파구 오금로38가길 25-1, 302호(가락동, 로데오빌리지)

이 발명을 지원한 국가연구개발사업

과제고유번호	20160000780032005
부처명	과학기술정보통신부
연구관리전문기관	정보통신기술진흥센터
연구사업명	정보통신.방송 연구개발사업(정보보호핵심원천기술개발사업) 3/4
연구과제명	맞춤형 보안서비스 제공을 위한 클라우드 기반 지능형 보안 기술 개발
기 여 율	1/1
주관기관	성균관대학교
연구기간	2018.01.01 ~ 2018.12.31

명 세 서

청구범위

청구항 1

등록 인터페이스(Registration Interface)를 통해 네트워크 보안 기능(NSF: Network Secure Function)을 관리하는 보안 관리 시스템에서, 보안 컨트롤러(Security Controller)에 의해 수행되는 방법은,

상기 보안 관리 시스템에 필요한 NSF에 대한 인스턴스화(instantiation) 요청 메시지를 개발자 관리 시스템(Developer's Management System)으로 전송하는 단계; 및

상기 요청 메시지에 대한 응답으로 상기 필요한 NSF에 대한 NSF 인스턴스(instance)의 등록을 지시하는 등록 메시지를 상기 개발자 관리 시스템으로부터 수신하는 단계를 포함하되,

상기 NSF 인스턴스는 상기 인스턴스화 요청 메시지에 기초하여 개발자 관리 시스템에 의해 생성되는 방법.

청구항 2

제1 항에 있어서,

상기 보안 관리 시스템에 필요한 NSF의 프로파일(Profile) 또는 서명(signature)을 인식하는 단계를 더 포함하고,

상기 인스턴스화 요청 메시지는 상기 능력 정보 또는 상기 서명을 포함하는 방법.

청구항 3

제2 항에 있어서,

상기 개발자 관리 시스템에 의해 생성되는 상기 NSF 인스턴스는 상기 능력 정보 또는 상기 서명과, 기 설정된 정보 모델에 기반한 NSF와 일치하는 것을 특징으로 하는 방법.

청구항 4

제1 항에 있어서,

상기 보안 관리 시스템에서 불필요한 NSF에 대한 역인스턴스화(de-instantiation) 요청 메시지를 개발자 관리 시스템으로 전송하는 단계를 더 포함하고,

상기 역인스턴스화 요청 메시지에 대응되는 NSF 인스턴스가 상기 개발자 관리 시스템에 의해 삭제되는 것을 특징으로 하는 방법.

청구항 5

제1 항에 있어서,

상기 등록 메시지는 NSF 인스턴스의 보안 능력을 나타내는 NSF 능력 정보(Capability Information), 새로운 인스턴스에 대한 네트워크 접근을 위해 이용되는 NSF 접근 정보(Access Information) 또는 엔티티(entity)에 부여된 역할에 따라 NSF에 대한 엔티티 접근을 허용할지 결정하기 위하여 NSF의 액세스 정책을 지정하는 NSF 역할 기반 ACL(Access Control List) 정보 중 적어도 하나를 포함하는 방법.

청구항 6

제5 항에 있어서,

상기 NSF 능력 정보는 네트워크 보안 능력(Network-Security Capabilities) 필드, 컨텐츠 보안 능력(Content-Security Capabilities) 필드, 공격 완화 능력(Attack Mitigation Capabilities) 필드 또는 퍼포먼스 능력(performance capabilities) 중 적어도 하나를 포함하는 방법.

청구항 7

제6 항에 있어서,

상기 수행 능력은 처리(Processing) 정보 및 대역폭(Bandwidth) 정보를 포함하는 것을 특징으로 하는 방법.

청구항 8

제7 항에 있어서,

상기 역할 기반 접근 제어 리스트는 엔티티의 역할 식별에 이용되는 하나 이상의 역할 ID를 포함하고,

상기 역할 ID는 특정 유형의 접근 요청을 식별하는데 사용되는 하나 이상의 접근 유형을 포함하는 방법.

청구항 9

등록 인터페이스(Registration Interface)를 통해 네트워크 보안 기능(NSF: Network Secure Function)을 관리하는 보안 컨트롤러(Security Controller)에 있어서,

외부 장치와 무선 또는 유선으로 통신하기 위한 통신부; 및

상기 통신부와 기능적으로 연결되는 프로세서를 포함하되, 상기 프로세서는,

상기 보안 관리 시스템에 필요한 NSF에 대한 인스턴스화(instantiation) 요청 메시지를 개발자 관리 시스템(Developer's Management System)으로 전송하고, 그리고

상기 요청 메시지에 대한 응답으로 상기 필요한 NSF에 대한 NSF 인스턴스(instance)의 등록을 지시하는 등록 메시지를 상기 개발자 관리 시스템으로부터 수신하되,

상기 NSF 인스턴스는 상기 인스턴스화 요청 메시지에 기초하여 개발자 관리 시스템에 의해 생성되는 장치.

발명의 설명

기 술 분 야

[0001] 본 발명은 네트워크 보안 서비스를 제공하기 위한 시스템, 방법 및 이를 위한 장치에 관한 것으로서, 보다 상세하게 I2NSF(Interface to Network Security Functions)에서 네트워크 보안 능력(Network Security Functions: NSF)의 등록 인터페이스를 위한 정보 모델 및 데이터 모델에 관한 것이다.

배 경 기 술

[0002] 오늘날 통신 사업자 및 인터넷 서비스 제공 업체와 같은 다양한 기업에서 운영 비용을 줄이고 보다 효율적이고 유연한 방법으로 자원을 활용하기 위해 네트워크 기능 가상화(NFV: Network Functions Virtualization) 기술을 널리 채택하고 있다. 또한, 제3자(third-party)의 서비스 공급 업체에 의해 제공되는 네트워크 기능 및 자원의 사용이 점차 대중화되고 있다. 기업들은 자신의 네트워크 시스템 및 정보 자산을 보호하기 위하여, 보안 기기(security appliance)를 직접 운영하는 대신에 third-party 보안 공급 업체에 의해 제공되는 보안 기능을 가입하여 사용하기 시작하였다.

[0003] 이러한 운영 모델은 다양한 이점을 제공한다. 회사는 물리적인 보안 장비 구매에 비용을 지불하지 않아도 되기 때문에 자본 지출(capital outlay)을 줄일 수 있다. 또한, 다양한 공격 시그니처(attack signature)에 대한 최신(up-to-date) 데이터베이스를 항상 유지할 수 있다.

[0004] 다만, 보안 기능(security function)은 다수의 솔루션 공급 업체(solution vendor)에 의해 개발되고, 다수의 네트워크 운영자에 의해 관리되기 때문에, NFV 기반 보안 기능(NFV-based security function)을 성공적으로 배포하기 위해서는 효율적인 표준화가 요구된다.

발명의 내용

해결하려는 과제

[0005] 본 발명의 목적은, 보안 서비스를 제공하는 시스템의 인터페이스에 대한 정보 모델 및 데이터 모델을 제안한다.

[0006] 또한, 본 발명의 목적은, NSF의 검색, 인스턴스화 및 등록을 지원하기 위하여 보안 컨트롤러와 개발자 관리 시스템간 등록 인터페이스에 필요한 정보 모델을 제안한다.

[0007] 또한, 본 발명의 목적은, 등록 인터페이스를 통해 보안 컨트롤러와 개발자 관리 시스템에 의해 수행되는 정보 모델에 기반한 절차를 제안한다.

[0008] 또한, 본 발명의 목적은, 보안 컨트롤러와 개발자 관리 시스템간 I2NSF 등록 인터페이스를 위한 YANG 데이터 모델을 제안한다.

[0009] 본 발명에서 이루고자 하는 기술적 과제들은 이상에서 언급한 기술적 과제들로 제한되지 않으며, 언급하지 않은 또 다른 기술적 과제들은 아래의 기재로부터 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0010] 본 발명의 일 양상은, 등록 인터페이스(Registration Interface)를 통해 네트워크 보안 기능(NSF: Network Secure Function)을 관리하는 보안 관리 시스템에서, 보안 컨트롤러(Security Controller)에 의해 수행되는 방법은, 상기 보안 관리 시스템에 필요한 NSF에 대한 인스턴스화(instantiation) 요청 메시지를 개발자 관리 시스템(Developer's Management System)으로 전송하는 단계; 및 상기 요청 메시지에 대한 응답으로 상기 필요한 NSF에 대한 NSF 인스턴스(instance)의 등록을 지시하는 등록 메시지를 상기 개발자 관리 시스템으로부터 수신하는 단계를 포함하되, 상기 NSF 인스턴스는 상기 인스턴스화 요청 메시지에 기초하여 개발자 관리 시스템에 의해 생성될 수 있다.

[0011] 바람직하게, 상기 보안 관리 시스템에 필요한 NSF의 프로파일(Profile) 또는 서명(signature)을 인식하는 단계를 더 포함하고, 상기 인스턴스화 요청 메시지는 상기 능력 정보 또는 상기 서명을 포함할 수 있다.

[0012] 바람직하게, 상기 개발자 관리 시스템에 의해 생성되는 상기 NSF 인스턴스는 상기 능력 정보 또는 상기 서명과, 기 설정된 정보 모델에 기반한 NSF와 일치하는 것을 특징으로 할 수 있다.

[0013] 바람직하게, 상기 보안 관리 시스템에서 불필요한 NSF에 대한 역인스턴스화(de-instantiation) 요청 메시지를 개발자 관리 시스템으로 전송하는 단계를 더 포함하고, 상기 역인스턴스화 요청 메시지에 대응되는 NSF 인스턴스가 상기 개발자 관리 시스템에 의해 삭제되는 것을 특징으로 할 수 있다.

[0014] 바람직하게, 상기 등록 메시지는 NSF 인스턴스의 보안 능력을 나타내는 NSF 능력 정보(Capability Information), 새로운 인스턴스에 대한 네트워크 접근을 위해 이용되는 NSF 접근 정보(Access Information) 또는 엔티티(entity)에 부여된 역할에 따라 NSF에 대한 엔티티 접근을 허용할지 결정하기 위하여 NSF의 액세스 정책을 지정하는 NSF 역할 기반 ACL(Access Control List) 정보 중 적어도 하나를 포함할 수 있다.

[0015] 바람직하게, 상기 NSF 능력 정보는 네트워크 보안 능력(Network-Security Capabilities) 필드, 컨텐츠 보안 능력(Content-Security Capabilities) 필드, 공격 완화 능력(Attack Mitigation Capabilities) 필드 또는 퍼포먼스 능력(performance capabilities) 중 적어도 하나를 포함할 수 있다.

[0016] 바람직하게, 상기 수행 능력은 처리(Processing) 정보 및 대역폭(Bandwidth) 정보를 포함할 수 있다.

[0017] 바람직하게, 상기 역할 기반 접근 제어 리스트는 엔티티의 역할 식별에 이용되는 하나 이상의 역할 ID를 포함하고, 상기 역할 ID는 특정 유형의 접근 요청을 식별하는데 사용되는 하나 이상의 접근 유형을 포함할 수 있다.

[0018] 본 발명의 다른 일 양상은, 등록 인터페이스(Registration Interface)를 통해 네트워크 보안 기능(NSF: Network Secure Function)을 관리하는 보안 컨트롤러(Security Controller)에 있어서, 외부 장치와 무선 또는 유선으로 통신하기 위한 통신부; 및 상기 통신부와 기능적으로 연결되는 프로세서를 포함하되, 상기 프로세서는, 상기 보안 관리 시스템에 필요한 NSF에 대한 인스턴스화(instantiation) 요청 메시지를 개발자 관리 시스템(Developer's Management System)으로 전송하고, 그리고 상기 요청 메시지에 대한 응답으로 상기 필요한 NSF에 대한 NSF 인스턴스(instance)의 등록을 지시하는 등록 메시지를 상기 개발자 관리 시스템으로부터 수신하되, 상기 NSF 인스턴스는 상기 인스턴스화 요청 메시지에 기초하여 개발자 관리 시스템에 의해 생성될 수 있다.

발명의 효과

[0019] 본 발명의 실시예에 따르면, I2NSF(Interface to Network Security Functions) 등록 인터페이스의 활용도를 더

욱 확장할 수 있다.

[0020] 또한, 본 발명의 실시예에 따르면, NSF 능력 정보(capability information) 모델을 정의함으로써, I2NSF 프레임 워크의 구성 요소들이 NSF 능력(Capability) 세트를 표준화된 방식으로 교환할 수 있다.

[0021] 본 발명에서 얻을 수 있는 효과는 이상에서 언급한 효과로 제한되지 않으며, 언급하지 않은 또 다른 효과들은 아래의 기재로부터 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

[0022] 본 발명에 관한 이해를 돕기 위해 상세한 설명의 일부로 포함되는, 첨부 도면은 본 발명에 대한 실시예를 제공하고, 상세한 설명과 함께 본 발명의 기술적 특징을 설명한다.

도 1은 본 발명의 일 실시예에 따른 I2NSF(Interface to Network Security Functions) 시스템을 예시한다.

도 2는 본 발명의 다른 실시예에 따른 I2NSF 시스템의 아키텍처를 예시한다.

도 3은 본 발명이 적용되는 실시예로서, 등록 인터페이스를 통해 NSF 인스턴스를 등록하는 방법을 나타내는 흐름도이다.

도 4는 본 발명의 일 실시예에 따른 등록 인터페이스의 정보 모델을 예시하는 도면이다.

도 5는 본 발명의 일 실시예에 따른 인스턴스 관리 서브 모델(Instance Management Sub-Model)을 예시하는 도면이다.

도 6은 본 발명의 일 실시예에 따른 등록 서브 모델(Registration Sub-Model)을 예시하는 도면이다.

도 7은 본 발명의 일 실시예에 따른, NSF 프로파일을 예시하는 도면이다.

도 8은 본 발명의 일 실시예에 따른, 퍼포먼스 능력(Performance Capabilities) 정보를 개략적으로 나타내는 도면이다.

도 9 및 도 10은 본 발명의 일 실시예에 따른 역할 기반 ACL(Role-based Access Control List)을 예시하는 도면이다.

도 11은 본 발명의 실시예에 따른 등록 인터페이스의 상위 레벨 YANG 데이터 모델을 예시하는 도면이다.

도 12는 본 발명의 실시예에 따른 등록 요청의 상위 레벨 YANG 데이터 모델을 예시하는 도면이다.

도 13은 본 발명의 실시예에 따른 인스턴스 관리 요청의 상위 레벨 YANG 데이터 모델을 예시하는 도면이다.

도 14는 본 발명의 실시예에 따른 NSF 능력 정보의 상위 레벨 YANG 데이터 모델을 예시하는 도면이다.

도 15는 본 발명의 실시예에 따른 NSF 액세스 정보의 상위 레벨 YANG 데이터 모델을 예시하는 도면이다.

도 16은 본 발명의 실시예에 따른 NSF 퍼포먼스 능력의 상위 레벨 YANG 데이터 모델을 예시하는 도면이다.

도 17은 본 발명의 실시예에 따른 역할 기반 ACL의 상위 레벨 YANG 데이터 모델을 예시하는 도면이다.

도 18은 본 발명의 실시예에 따른 I2NSF 등록 인터페이스의 데이터 모델을 예시하는 도면이다.

도 19는 본 발명의 실시예에 따른 등록 인터페이스에 대한 XML 출력을 예시하는 도면이다.

도 20은 본 발명의 일 실시예에 따른 네트워크 장치의 블록 구성도를 예시한다.

발명을 실시하기 위한 구체적인 내용

[0023] 이하, 본 발명에 따른 바람직한 실시 형태를 첨부된 도면을 참조하여 상세하게 설명한다. 첨부된 도면과 함께 이하에 개시될 상세한 설명은 본 발명의 예시적인 실시형태를 설명하고자 하는 것이며, 본 발명이 실시될 수 있는 유일한 실시형태를 나타내고자 하는 것이 아니다. 이하의 상세한 설명은 본 발명의 완전한 이해를 제공하기 위해서 구체적 세부사항을 포함한다. 그러나, 당업자는 본 발명이 이러한 구체적 세부사항 없이도 실시될 수 있음을 안다.

[0024] 몇몇 경우, 본 발명의 개념이 모호해지는 것을 피하기 위하여 공지의 구조 및 장치는 생략되거나, 각 구조 및

장치의 핵심기능을 중심으로 한 블록도 형식으로 도시될 수 있다.

[0025] 아울러, 본 발명에서 사용되는 용어는 가능한 한 현재 널리 사용되는 일반적인 용어를 선택하였으나, 특정한 경우는 출원인이 임의로 선정한 용어를 사용하여 설명한다. 그러한 경우에는 해당 부분의 상세 설명에서 그 의미를 명확히 기재하므로, 본 발명의 설명에서 사용된 용어의 명칭만으로 단순 해석되어서는 안 될 것이며 그 해당 용어의 의미까지 파악하여 해석되어야 함을 밝혀두고자 한다.

[0026] 이하의 설명에서 사용되는 특정 용어들은 본 발명의 이해를 돕기 위해서 제공된 것이며, 이러한 특정 용어의 사용은 본 발명의 기술적 사상을 벗어나지 않는 범위에서 다른 형태로 변경될 수 있다.

[0027] 명세서 전체에서, 어떤 부분이 다른 부분과 "연결"되어 있다고 할 때, 이는 "직접적으로 연결"되어 있는 경우뿐 아니라, 그 중간에 다른 소자를 사이에 두고 "전기적으로 연결"되어 있는 경우도 포함한다.

[0028] 본 명세서 전체에서, 어떤 부재가 다른 부재 “상에” 위치하고 있다고 할 때, 이는 어떤 부재가 다른 부재에 접해 있는 경우뿐 아니라 두 부재 사이에 또 다른 부재가 존재하는 경우도 포함한다.

[0029] 본 명세서 전체에서, 어떤 부분이 어떤 구성요소를 "포함" 한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성 요소를 더 포함할 수 있는 것을 의미한다. 본원 명세서 전체에서 사용되는 정도의 용어 "~(하는) 단계" 또는 "~의 단계"는 "~ 를 위한 단계"를 의미하지 않는다.

[0030] 최근에는, NFV(Network Functions Virtualization)-based security function을 위한 기본 표준 인터페이스가 I2NSF(Interface to Network Security Functions) 워킹 그룹에 의해 개발되고 있다. 이는 인터넷 엔지니어링 태스크 포스(IETF: Internet Engineering Task Force)로 불리는 국제 인터넷 표준 기구의 일부이다.

[0031] I2NSF의 목적은 다수의 보안 솔루션 벤더(security solution vendor)들에 의해 제공되는 이종의(heterogeneous) 네트워크 보안 기능(들)(NSF: network security function)을 위한 표준화된 인터페이스를 정의하기 위함이다.

[0032] I2NSF 아키텍처(architecture)에서, NSF(들)의 관리에 대하여 상세히 고려할 필요 없이(NSF의 관리는 결국 보안 정책의 시행(enforce)을 요구한다), 사용자는 사용자의 네트워크 시스템 내 네트워크 자원을 보호하기 위한 보호 정책을 정의할 수 있다. 또한, 다수의 vendor들로부터 NSF(들)로의 표준화된 인터페이스는 이종의 NSF(들)에 대한 태스크(task)의 설정 및 관리를 단순화할 수 있다.

[0033] 본 명세서는, NSF의 검색, 인스턴스화 및 등록을 지원하기 위하여 보안 컨트롤러와 개발자 관리 시스템간 등록 인터페이스에 필요한 정보 모델을 제안한다. 또한, 본 명세서는, I2NSF(Interface to Network Security Functions) 등록 인터페이스에 대한 YANG 데이터 모델을 제안한다. 또한, 본 명세서는, NSF 인스턴스 풀을 동적으로 운영하기 위해 보안 컨트롤러와 개발자 관리 시스템간 등록 인터페이스에 필요한 데이터를 정의하는 YANG 데이터 모델을 제공한다.

[0034] 또한, 본 명세서는, I2NSF 프레임워크에 기초한 보안 관리 아키텍처를 제안한다. 실시예로서, 보안 관리 아키텍처는 I2NSF 사용자, 보안 관리 시스템(Security Management System) 및/또는 프레임워크의 최하위 계층의 NSF(들)의 인스턴스(들)을 포함할 수 있다. 예를 들면, 보안 관리 시스템은 보안 컨트롤러(security Controller) 및 개발자 관리 시스템(Developer's Management System)을 포함할 수 있다. 보안 컨트롤러는 보안 정책 관리자(Security Policy Manager) 및 NSF 능력 관리자(NSF Capability Manager)를 포함할 수 있다.

[0035] 또한, 본 명세서는, I2NSF 보안 관리 시스템에서 보안 서비스(예컨대, VoIP-VoLTE)에 대한 임무(mission)를 수행하기 위한 데이터 모델을 제안한다.

[0036] 본 명세서에서 사용될 수 있는 용어(terminology)들은 다음과 같이 정의된다.

[0037] - 어플리케이션 로직(Application Logic): 보안 공격을 차단하거나 완화(mitigate)하기 위한 사용자 관점 보안 정책을 생성하는 보안 관리 아키텍처의 구성요소.

[0038] - 정책 업데이터(Policy Updater): 사용자 관점 보안 정책을 보안 컨트롤러로 전달하는 구성요소. 사용자 관점 정책은 어플리케이션 로직으로부터 검색된다.

[0039] - 보안 정책 관리자(Security Policy Manager): 정책 업데이터로부터 수신된 사용자 관점 보안 정책을 하위 레벨 보안 정책으로 맵핑하거나, 그 반대로 맵핑하는 구성요소.

[0040] - NSF 능력 관리자(NSF Capability Manager): 등록 인터페이스를 통해 개발자 관리 시스템에 의해 등록된 NSF

능력을 저장하고, 대응하는 하위 레벨 보안 정책을 생성하기 위해 이를 보안 정책 관리자와 공유하는 구성요소.

[0041] - 이벤트 수집기(Event Collector): 어플리케이션 로직에서 사용자 관점 정책을 업데이트(또는 생성)하기 위해 사용되는, 보안 컨트롤러로부터 이벤트를 수신하는 구성요소.

[0042] - 네트워크 보안 기능(NSF): 수신된 패킷(packet)의 특정 취급을 담당하는 기능 또는 이를 위한 장치를 의미한다. NSF는 다양한 프로토콜 스택(stack)의 다양한 계층(예컨대, 네트워크 계층 또는 다른 OSI(Open System Interconnection) 계층 등)에서 동작할 수 있다.

[0043] 예를 들어, NSF의 일례로서, 방화벽(firewall), 침입 방지 시스템(IPS: Intrusion Prevention System)/침입 탐지 시스템(IDS: Intrusion Detection System), 강한 패킷 검사(DPI: Deep Packet Inspection), 애플리케이션 가시성 및 제어(AVC: Application Visibility and Control), 네트워크 바이러스 및 악성 코드 스캐닝, 샌드박스(sandbox), 데이터 손실 방지(DLP: Data Loss Prevention), 분산 서비스 거부(DDoS: Distribute Denial of Service) 완화(mitigation), 전송 계층 보안(TLS: Transport Layer Security) 프록시, 안티스푸핑(Anti-Spoofing) 등이 포함될 수 있다.

[0044] 본 발명의 일 실시예에 따른 NSF는 상술한 예시 중 어느 하나로 구현될 수 있으며, 다양한 타입의 NSF가 이용될 수 있다. 또한, 동일한 타입의 NSF가 다수로 구현될 수도 있다. 또한, 본 발명에 따른 NSF는 상술한 예시 중 어느 하나 이상이 결합되어 구현될 수도 있다.

[0045] 이하에서는 I2NSF 시스템의 아키텍처/프레임워크 및 I2NSF 시스템의 각 컴포넌트들에 대하여 설명한다. 또한, 어떻게 I2NSF가 SDN(Software-Defined Networking) 및 NFV(Network Functions Virtualization) 환경에서 기술 및 벤더 독립적인 방식으로 보안 기능을 구현하는 것을 용이하게 하면서, NFS들의 기능을 제한할 수 있는 잠재적 제약(constraint)을 피하게 하는지를 설명한다.

[0046] I2NSF 프레임워크는 I2NSF 시스템의 사용자(예컨대, 어플리케이션, 오버레이 또는 클라우드 네트워크 관리 시스템, 또는 기업 네트워크 관리자 또는 관리 시스템)가 어떤 I2NSF 기능이 어떤 트래픽(또는, 트래픽 패턴)에 적용되어야 하는지를 I2NFS 시스템에 알리기 위한 표준 인터페이스를 요구한다. I2NSF 시스템은 이 표준 인터페이스를 상이한 트래픽의 동작(behavior)을 모니터링하고 제어하기 위한 보안 규칙들의 세트로서 인식할 수 있다. I2NSF 프레임워크는 또한 사용자가 상이한 관리 도메인에 의해 호스팅되고 관리되는 흐름-기반(flow-based) 보안 기능을 모니터링하기 위한 표준 인터페이스를 제공한다.

[0047] 도 1은 본 발명의 일 실시예에 따른 I2NSF(Interface to Network Security Functions) 시스템을 예시한다.

[0048] 도 1을 참조하면, I2NSF 시스템은 I2NSF 사용자(user), 네트워크 운영 관리 시스템(Network Operator Management System), 개발자 관리 시스템(Developer's Management System) 및/또는 적어도 하나의 NSF(Network Security Function)을 포함한다.

[0049] I2NSF 사용자는 I2NSF 소비자-직면 인터페이스(I2NSF Consumer-Facing Interface)를 통해 네트워크 운영 관리 시스템과 통신한다. 네트워크 운영 관리 시스템은 I2NSF NSF-직면 인터페이스(I2NSF NSF-Facing Interface)를 통해 NSF(들)과 통신한다. 개발자 관리 시스템은 I2NSF 등록 인터페이스(I2NSF Registration Interface)를 통해 네트워크 운영 관리 시스템과 통신한다. 이하에서는 I2NSF 시스템의 각 컴포넌트(I2NSF 컴포넌트) 및 각 인터페이스(I2NSF 인터페이스)에 설명한다.

[0050] I2NSF 사용자

[0051] I2NSF 사용자는 다른 I2NSF 컴포넌트(예컨대, 네트워크 운영 관리 시스템)에서 정보(예컨대, NSF의 정보)를 요청하거나 및/또는 다른 I2NSF 컴포넌트(예컨대, 개발자 관리 시스템)에 의해 제공되는 보안 서비스(예컨대, 네트워크 보안 서비스)를 사용하는 I2NSF 컴포넌트이다. 예를 들면, I2NSF 사용자는 오버레이 네트워크 관리 시스템, 기업 네트워크 관리자 시스템, 다른 네트워크 도메인 관리자 등일 수 있다. I2NSF 사용자는 I2NSF 클라이언트로 지칭될 수 있다.

[0052] 이러한 I2NSF 사용자 컴포넌트에 할당된 역할을 수행하는 대상은 I2NSF 소비자로 지칭될 수 있다. I2NSF 소비자의 예로는, 일정 기간(time span) 동안 패킷의 특정 필드에 기초하여 흐름을 허용, 속도-제한(rate-limit), 또는 거부하기 위해 언더레이 네트워크(underlay network)에 동적으로 알릴 필요가 있는 화상 회의 네트워크 관리자(video-conference network manager), 특정 흐름에 대한 특정 I2NSF 정책을 시행(enforce)하기 위해 제공자 네트워크를 요청할 필요가 있는 기업 네트워크 관리자(Enterprise network administrators) 및 관리 시스템(management systems), 특정 조건의 세트와 일치하는 흐름을 차단하기 위해 언더레이 네트워크에 요청을 전송하

는 IoT 관리 시스템(IoT management system)이 포함될 수 있다.

[0053] I2NSF 사용자는 상위 레벨(high-level) 보안 정책(security policy)을 생성 및 배포할 수 있다. 구체적으로 설명하면, I2NSF 사용자는 다양한 악의적인(malicious) 공격으로부터 네트워크 트래픽(traffic)을 보호하기 위하여 네트워크 보안 서비스(network security service)를 이용할 필요가 있다. 이 보안 서비스를 요청하기 위하여, I2NSF 사용자는 자신이 원하는 보안 서비스에 대한 상위 레벨 보안 정책을 생성하고 네트워크 운영 관리 시스템에게 이를 알릴 수 있다.

[0054] 한편, 상위 레벨 보안 정책을 준비하는 과정에서, I2NSF 사용자는 각 NSF(들)를 위한 보안 서비스 또는 보안 정책 규칙 구성(security policy rule configuration)을 실현하기 위하여 요구되는 NSF(들)의 타입에 대하여 고려하지 않을 수 있다.

[0055] 또한, I2NSF 사용자는 네트워크 운영 관리 시스템에 의해 기본적인(underlying) NSF(들) 내에서 발생되는 보안 이벤트(들)(security event)를 통지 받을 수 있다. 이들의 보안 이벤트(들)을 분석함으로써, I2NSF 사용자는 새로운 공격을 식별하고, 새로운 공격에 대처하기 위한 상위 레벨 보안 정책을 업데이트(또는 생성)할 수 있다. 이와 같이, I2NSF 사용자는 보안 정책을 정의, 관리 및 모니터링할 수 있다.

[0056] 네트워크 운영 관리 시스템

[0057] 네트워크 운영 관리 시스템은 보안 제공, 모니터링 및 기타 동작을 위한 수집(collection) 및 배포(distribution) 지점(point)의 역할을 수행하는 컴포넌트이다. 예를 들면, 네트워크 운영 관리 시스템은 보안 컨트롤러(Security Controller)에 해당하거나, 또는 보안 컨트롤러를 포함하는 컴포넌트일 수 있다. 이러한 네트워크 운영 관리 시스템은 네트워크 운영자에 의해 관리될 수 있고, I2NSF 관리 시스템으로 지칭될 수도 있다.

[0058] 네트워크 운영 관리 시스템(또는 보안 컨트롤러)의 주요한 역할 중 하나는 I2NSF 사용자로부터의 상위 레벨 보안 정책(또는 정책 규칙)을 특정 NSF(들)을 위한 하위 레벨(low-level) 보안 정책 규칙으로 번역(translate)하는 것이다. 네트워크 운영 관리 시스템(또는 보안 컨트롤러)은 상위 레벨 보안 정책을 I2NSF 사용자로부터 수신한 후, 우선 I2NSF 사용자에 의해 요구되는 정책을 시행하기 위하여 요구되는 NSF(들)의 타입을 결정할 수 있다. 그리고, 네트워크 운영 관리 시스템(또는 보안 컨트롤러)은 요구되는 각 NSF(들)을 위한 하위 레벨(low-level) 보안 정책을 생성할 수 있다. 결국, 네트워크 운영 관리 시스템(또는 보안 컨트롤러)는 생성된 하위 레벨 보안 정책을 각 NSF(들)에게 설정할 수 있다.

[0059] 또한, 네트워크 운영 관리 시스템(또는 보안 컨트롤러)은 I2NSF 시스템 내 구동 중인 NSF(들)을 모니터링하고, 각 NSF(들)에 대한 다양한 정보(예를 들어, 네트워크 액세스(access) 정보 및 작업로드(workload) 상태 등)를 유지할 수 있다. 또한, 네트워크 운영 관리 시스템(또는 보안 컨트롤러)은 개발자 관리 시스템의 도움을 받아 NSF 인스턴스의 동적인 수명시간(life-cycle) 관리를 통해 NSF 인스턴스(instance)의 풀(pool)을 동적으로 관리할 수 있다.

[0060] NSF

[0061] NSF는 보안 관련 서비스를 제공하는 논리적 엔티티(logical entity) 또는 소프트웨어 컴포넌트이다. 예를 들면, NSF(예컨대, 방화벽)는 하위 레벨 보안정책을 수신하고, 이에 기초하여 악의적인 네트워크 트래픽을 감지하고, 이를 차단하거나 완화할 수 있다. 이를 통해, 네트워크 통신 스트림의 무결성(integrity) 및 기밀성(confidentiality)이 보장될 수 있다.

[0062] 개발자 관리 시스템

[0063] 개발자 관리 시스템은 다른 I2NSF 컴포넌트(예컨대, 네트워크 운영 관리 시스템)으로 정보(예컨대, NSF의 정보)를 보내거나, 및/또는 보안 서비스(예컨대, 네트워크 보안 서비스)를 제공하는 I2NSF 컴포넌트이다. 개발자 관리 시스템은 벤더 관리 시스템(Vendor's Management System)으로 지칭될 수도 있다. 이러한 개발자 관리 시스템에 할당된 역할을 수행하는 대상은 I2NSF 생산자(producer)로 지칭될 수 있다.

[0064] 개발자 관리 시스템은 네트워크 운영자에게 NSF(들)을 제공하는 제3자(third-party) 보안 벤더에 의해 관리될 수 있다. 다양한 보안 벤더의 다수의 개발자 관리 시스템(들)이 존재할 수 있다.

[0065] I2NSF 소비자-직면 인터페이스(간단히, 소비자-직면 인터페이스(CFI))

[0066] CFI는 I2NSF 사용자와 네트워크 운영 관리 시스템 사이에 위치하는, 사용자의 I2NSF 시스템으로의 인터페이스이다. 이렇게 설계됨으로써, I2NSF 시스템은 하위(underlying) NSF(들)의 상세한 내용을 숨기고, 사용자에게 NSF

(들)의 추상적인 시각(abstract view)만을 제공할 수 있다.

[0067] 이 CFI는 주어진 I2NSF 시스템의 상이한 사용자가 관리 도메인 내의 특정 흐름(flow)에 대한 보안 정책을 정의, 관리 및 모니터링할 수 있게 하기 위해 사용될 수 있다. I2NSF 사용자에 의해 생성된 상위 레벨 보안 정책(또는 정책 규칙)은 이 CFI를 통해 네트워크 운영 관리 시스템으로 전달될 수 있다. 또한, NSF(들)에 의한 보안 경보(alert)는 이 CFI를 통해 네트워크 운영 관리 시스템으로부터 I2NSF 사용자로 전달될 수 있다.

[0068] I2NSF NSF-직면 인터페이스(간단히, NSF-직면 인터페이스(NFI))

[0069] NFI는 네트워크 운영 관리 시스템(또는 보안 컨트롤러)과 NSF(들) 사이에 위치하는 인터페이스이다.

[0070] NFI의 주요한 목적은 NSF(들)로부터 보안 관리 기법을 분리(decouple)함으로써 다양한 보안 솔루션 벤더들의 NSF(들)을 제어하고 관리하기 위한 표준화된 인터페이스를 제공하기 위함이다. NFI는 NSF(들)의 상세한 내용(예를 들어, 벤더, 유형 인자(form factor) 등)과 독립된다. 따라서, 보안 정책 규칙을 NSF에게 설정할 때, 네트워크 운영 관리 시스템은 벤더 특정한 차이 및/또는 NSF의 폼 팩터(form factor)를 고려할 필요가 없다. 이 NFI는 하나 이상의 NSF에 의해 시행되는 흐름-기반(flow-based) 보안 정책을 지정하고 모니터링하기 위해 사용될 수 있다. 예를 들면, 네트워크 운영 관리 시스템은 I2NSF 사용자에 의한 상위 레벨 보안 정책을 시행하기 위하여 흐름-기반(flow-based) 보안 정책을 NFI 인터페이스를 통해, 각 흐름-기반 NSF에게 전달할 수 있다. 여기서, 흐름-기반 NSF는 보안 특성을 강화하기 위해 정책의 세트에 따라 네트워크 흐름을 검사하는 NSF이다. 이러한 흐름-기반 NSF에 의한 흐름-기반 보안은 수신된 순서대로 패킷들이 검사되고, 검사 프로세스에 따라 패킷에 대한 수정이 없는 것을 의미한다. 흐름-기반 NSF에 대한 인터페이스는 다음과 같이 분류될 수 있다:

[0071] - NSF 운영 및 관리 인터페이스(NSF Operational and Administrative Interface): NSF의 운영 상태를 프로그래밍하기 위해 I2NSF 관리 시스템에 의해 사용되는 인터페이스 그룹; 이 인터페이스 그룹은 또한 관리 제어 기능을 포함한다. I2NSF 정책 규칙은 일관된 방식으로 이 인터페이스 그룹을 변경하는 한가지 방법을 나타낸다. 어플리케이션 및 I2NSF 컴포넌트가 그들이 송신 및 수신하는 트래픽의 동작을 동적으로 제어할 필요가 있기 때문에, I2NSF 노력(effort)의 대부분이 이 인터페이스 그룹에 집중된다.

[0072] - 모니터링 인터페이스(Monitoring Interface): 하나 이상의 선택된 NSF로부터의 모니터링 정보를 획득하기 위해 I2NSF 관리 시스템에 의해 사용되는 인터페이스 그룹; 이 인터페이스 그룹의 각 인터페이스는 쿼리 또는 리포트 기반 인터페이스일 수 있다. 둘 사이의 차이점은 쿼리 기반 인터페이스는 정보를 획득하기 위해 I2NSF 관리 시스템에 의해 사용되고, 이에 반하여 리포트 기반 인터페이스는 정보를 제공하기 위해 NSF에 의해 사용된다는 것이다. 이 인터페이스 그룹의 기능은 또한 SYSLOG 및 DOTS와 같은 다른 프로토콜에 의해 정의될 수 있다. I2NSF 관리 시스템은 정보의 수신에 기초하여 하나 이상의 동작(action)을 취할 수 있다. 이는 I2NSF 정책 규칙에 의해 지정되어야 한다. 이 인터페이스 그룹은 NSF의 운영 상태를 변경하지 않는다.

[0073] 이와 같이, NFI는 흐름-기반 패러다임을 사용하여 개발될 수 있다. 흐름-기반 NSF의 공동 특성(common trait)은 수신된 패킷의 컨텐츠(예컨대, 헤더/페이로드) 및/또는 컨텍스트(예컨대, 세션 상태 및 인증 상태)에 기초하여 패킷을 처리하는 것이다. 이 특징은 I2NSF 시스템의 동작을 정의하기 위한 요구사항(requirement) 중 하나이다.

[0074] 한편, I2NSF 관리 시스템은 주어진 NSF의 모든 기능들을 사용할 필요가 없으며, 모든 사용 가능한 NSF들을 사용할 필요도 없다. 따라서, 이 추상화(abstraction)는 NSF 특징(feature)을 NSF 시스템에 의해 빌딩 블록(building block)으로 취급될 수 있게 해준다. 그러므로, 개발자는 벤더 및 기술에 독립적인 NSF에 의해 정의되는 보안 기능을 자유롭게 사용할 수 있게 된다.

[0075] I2NSF 등록 인터페이스(간단히, 등록 인터페이스(RI))

[0076] RI는 네트워크 운영 관리 시스템 및 개발자 관리 시스템 사이에 위치하는 인터페이스이다. 상이한 벤더에 의해 제공되는 NSF는 상이한 능력(capability)을 가질 수 있다. 따라서, 상이한 벤더에 의해 제공되는 여러 유형의 보안 능력을 이용하는 프로세스를 자동화하기 위해, 벤더가 그들의 NSF의 기능을 정의하기 위한 전용 인터페이스를 가질 필요가 있다. 이러한 전용 인터페이스는 I2NSF 등록 인터페이스(RI)로 지칭될 수 있다.

[0077] NSF의 능력은 미리 구성되거나 또는 I2NSF 등록 인터페이스를 통해 동적으로 검색될 수 있다. 만일 소비자에게 노출되는 새로운 기능이 NSF에 추가된다면, 관심 있는(interested) 관리 및 제어 엔티티가 그것들을 알 수 있도록, 그 새로운 기능의 능력이 이 RI를 통해 I2NSF 레지스트리(registry)에 등록될 필요가 있다.

[0079] 도 2는 본 발명의 다른 실시예에 따른 I2NSF 시스템의 아키텍처를 예시한다. 도 2의 I2NSF 시스템은 도 1의 I2NSF 시스템에 비하여 I2NSF 사용자 및 네트워크 운영 관리 시스템의 구성을 더 구체적으로 나타낸다. 도 2에서는 도 1에서 상술한 설명과 중복된 설명은 생략한다.

[0080] 도 2를 참조하면, I2NSF 시스템은 I2NSF 사용자, 보안 관리 시스템(Security Management System), 및 NSF 인스턴스(instances) 계층을 포함한다. I2NSF 사용자 계층은 어플리케이션 로직(Application Logic), 정책 업데이터(Policy Updater), 및 이벤트 수집기(Event Collector)을 컴포넌트로서 포함한다. 보안 관리 시스템 계층은 보안 컨트롤러 및 개발자 관리 시스템을 포함한다. 보안 관리 시스템 계층의 보안 컨트롤러는 보안 정책 관리자(Security policy manager) 및 NSF 기능 관리자(NSF capability manager)를 컴포넌트로서 포함한다.

[0081] I2NSF 사용자 계층은 소비자-직면 인터페이스를 통해 보안 관리 시스템 계층과 통신한다. 예를 들면, I2NSF 사용자 계층의 정책 업데이터 및 이벤트 수집기는 소비자-직면 인터페이스를 통해 보안 관리 시스템 계층의 보안 컨트롤러와 통신한다. 또한, 보안 관리 시스템 계층은 NSF-직면 인터페이스를 통해 NSF 인스턴스 계층과 통신한다. 예를 들면, 보안 관리 시스템 계층의 보안 컨트롤러는 NSF-직면 인터페이스를 통해 NSF 인스턴스 계층의 NSF 인스턴스(들)과 통신한다. 또한, 보안 관리 시스템 계층의 개발자 관리 시스템은 등록 인터페이스를 통해 보안 관리 시스템 계층의 보안 컨트롤러와 통신한다.

[0082] 도 2의 I2NSF 사용자 계층, 보안 관리 시스템 계층의 보안 컨트롤러 컴포넌트, 보안 관리 시스템 계층의 개발자 관리 시스템 컴포넌트 및 NSF 인스턴스 계층은 각각 도 1의 I2NSF 사용자 컴포넌트, 네트워크 운영 관리 시스템 컴포넌트, 개발자 관리 시스템 컴포넌트 및 NSF 컴포넌트에 대응된다. 또한, 도 2의 소비자-직면 인터페이스, NSF-직면 인터페이스 및 등록 인터페이스는 도 1의 소비자-직면 인터페이스, NSF-직면 인터페이스 및 등록 인터페이스에 대응된다. 이하에서는, 각 계층에 포함된 새로 정의된 컴포넌트들에 대하여 설명한다.

[0083] I2NSF 사용자

[0084] 상술한 것처럼, I2NSF 사용자 계층은 다음 3 개의 컴포넌트를 포함한다: 어플리케이션 로직(Application Logic), 정책 업데이터(Policy Updater), 및 이벤트 수집기(Event Collector). 각각의 역할 및 동작을 설명하면 다음과 같다.

[0085] 어플리케이션 로직은 상위 레벨 보안 정책을 생성하는 컴포넌트이다. 이를 위해, 어플리케이션 로직은 이벤트 수집기로부터 상위 레벨 정책을 업데이트(또는 생성)하기 위한 이벤트를 수신하고, 수집된 이벤트에 기초하여 상위 레벨 정책을 업데이트(또는 생성)한다. 그 이후에, 상위 레벨 정책은 보안 컨트롤러로 배포하기 위해 정책 업데이터로 보내진다. 상위 레벨 정책을 업데이트(또는 생성)하기 위해, 이벤트 수집기는 보안 수집기에 의해 보내진 이벤트를 수신하고, 그들을 어플리케이션 로직으로 보낸다. 이 피드백에 기초하여, 어플리케이션 로직은 상위 레벨 보안 정책을 업데이트(또는 생성)할 수 있다.

[0086] 도 2에서는, 어플리케이션 로직, 정책 업데이터 및 이벤트 수집기를 각각 별도의 구성으로 도시하고 있으나, 본 발명의 이에 한정되지 않는다. 다시 말해, 각각은 논리적인 컴포넌트로서, I2NSF 시스템에서 하나 또는 복수의 컴포넌트로 구현될 수도 있다. 예를 들면, 도 1과 같이 단일의 I2NSF 사용자 컴포넌트에 의해 구현될 수 있다.

[0087] 보안 관리 시스템

[0088] 상술한 것처럼, 보안 관리 시스템 계층의 보안 컨트롤러는 다음 2개의 컴포넌트를 포함한다: 보안 정책 관리자(Security policy manager) 및 NSF 능력 관리자(NSF capability manager). 각각의 역할 및 동작을 설명하면 다음과 같다.

[0089] 보안 정책 관리자는 CFI를 통해 정책 업데이터로부터 상위 레벨 정책을 수신하고, 이 정책을 여러 하위 레벨 정책으로 맵핑할 수 있다. 이 하위 레벨 정책은 NSF 능력 관리자에 등록된 주어진 NSF 능력과 관련된다. 또한, 보안 정책 관리자는 이 정책을 NFI를 통해 NSF(들)로 전달할 수 있다.

[0090] NSF 능력 관리자는 주어진 NSF 능력과 관련된 하위 레벨 정책을 생성하기 위해, 개발자 관리 시스템에 의해 등록된 NSF의 능력을 지정하고, 그것을 보안 정책 관리자와 공유할 수 있다. 새로운 NSF가 등록될 때마다, NSF 능력 관리자는 등록 인터페이스를 통해 NSF 능력 관리자의 관리 테이블에 NSF의 기능/능력을 등록하도록 개발자 관리 시스템에 요청할 수 있다. 개발자 관리 시스템은 새로운 NSF의 능력을 NSF 능력 관리자로 등록하기 위한 보안 관리 시스템의 다른 부분에 해당한다.

[0091] 도 2에서는, 보안 정책 관리자 및 NSF 능력 관리자를 각각 별도의 구성으로 도시하고 있으나, 본 발명의 이에 한정되지 않는다. 다시 말해, 각각은 논리적인 컴포넌트로서, I2NSF 시스템에서 하나의 컴포넌트로 구현될 수도

있다.

[0092] NSF 인스턴스(NSF Instances)

[0093] 도 2에 도시된 것처럼, NSF 인스턴스 계층은 NSF들을 포함한다. 이때, 모든 NSF들은 이 NSF 인스턴스 계층에 위치된다. 한편, 상위 레벨 정책을 하위 레벨 정책으로 번역한 후에, 보안 정책 관리자는 NFI를 통해 정책을 NSF(들)로 전달한다. 이 경우, NSF는 수신된 하위 레벨 보안 정책에 기초하여 악의적인 네트워크 트래픽을 감지하고, 이를 차단하거나 완화할 수 있다.

[0094] 이하에서는 I2NSF에 대한 정보 및 데이터 모델에 대하여 설명한다. 특히, I2NSF 시스템 내의 등록 인터페이스에 대한 정보 및 데이터 모델(YANG 데이터 모델)을 설명한다. 본 발명의 실시예에 따르면, I2NSF 등록 인터페이스를 통해 필요한 보안 능력에 따라 NSF 검색, 인스턴스화 및 등록을 지원할 수 있다.

[0095] 이를 위해 먼저, 정보 모델과 데이터 모델의 기본 개념에 대하여 설명한다.

[0096] 기본적으로, 정보 모델 및 데이터 모델은 네트워크 관리에서 관리 객체(managed object)를 정의하기 위해 사용될 수 있다. 중복되는 세부사항(overlapped details)에도 불구하고, 정보 모델 및 데이터 모델은 네트워크 관리 관점에서 상이한 특성(character)을 가진다.

[0097] 일반적으로, 정보 모델의 주요 목적은 임의의 특정 구현 또는 프로토콜에 의존하지 않고, 개념적 수준(conceptual level)에서 관리 객체를 모델링하는 것이다. 전체 설계(overall design)를 명확히 하기 위해, 정보 모델은 관리 객체들간의 관계를 정의하는 모든 프로토콜 및 구현 세부사항을 숨겨야 한다. 이에 기초하여, 정보 모델은 상이한 방식으로 구현될 수 있고, 상이한 프로토콜에 맵핑될 수 있다. 이처럼 정보 모델은 프로토콜에 중립적이다. 일반적으로, 정보 모델은 영어와 같은 자연어(natural language)를 사용하여, 예시적인 방식으로 정의될 수 있다. 뿐만 아니라, 정보 모델을 설명하기 위해 객체-지향 기법(object-oriented technique)을 사용하는 것이 바람직할 수 있다.

[0098] 일반적으로, 데이터 모델은 더 하위 레벨(lower level)의 추상화로 정의되며, 많은(many) 세부사항을 제공한다. 데이터 모델은 구현 및 프로토콜의 사양(specification), 예컨대, 관리 객체를 하위 레벨 프로토콜 구조(construct)로 매핑하는 방법을 설명하는 규칙에 대한 세부사항을 제공한다. 개념적 모델은 다양한 방식으로 구현될 수 있기 때문에, 다중(multiple) 데이터 모델이 단일 정보 모델로부터 도출될 수 있다. 데이터 모델은 NSF 인스턴스 등록과 NSF 인스턴스의 동적 수명 주기 관리에 필요하다.

[0099] 이하에서는, 보안 컨트롤러(Security Controller)와 개발자 관리 시스템(Developer's Management System)간 I2NSF 등록 인터페이스에 대한 정보 모델을 설명한다.

[0100] 다수의 가상 NSF 인스턴스는 일반적으로 I2NSF 프레임 워크에 존재한다. 이러한 NSF 인스턴스는 서로 다른 보안 기능을 가질 수 있기 때문에, 각각의 NSF 인스턴스의 보안 기능이 생성된 이후 보안 컨트롤러에 등록하는 것이 중요하다. 또한, 필요시 일부 보안 기능의 NSF를 인스턴스화 해야 한다. 예를 들어, I2NSF 사용자가 요구하는 새로운 보안 요구 사항을 충족하기 위해 추가 보안 기능이 필요한 경우, 보안 컨트롤러는 필요한 보안 기능이 있는 NSF를 인스턴스화 하도록 개발자 관리 시스템에 요청할 수 있어야 한다.

[0101] 본 발명에서는, 필요한 보안 능력에 따라 NSF의 검색, 인스턴스화 및 등록을 지원하는 보안 컨트롤러와 개발자 관리 시스템간의 I2NSF 등록 인터페이스에 대한 정보 모델 및 YANG 데이터 모델을 제안한다. 또한, 본 발명에서는, 정의된 모델을 사용하여 등록 인터페이스를 통해 보안 컨트롤러와 개발자 관리 시스템에서 수행되는 절차를 제안한다.

[0102] 또한, 본 발명에서는, 등록 인터페이스를 통해 보안 컨트롤러와 개발자 관리 시스템에 의해 수행되는 능력 정보 모델에 기반한 절차를 제안한다.

[0103] 기존의 I2NSF 등록 인터페이스는 보안 컨트롤러에 새로운 NSF 인스턴스를 등록하는 경우에 한하여 사용되었다. 반면에, 본 발명에서는 필요한 경우 NSF 인스턴스화(instantiation)/역인스턴스화(deinstantiation)를 지원하기 위해 그 활용도를 확장하고 기능을 위하여 등록 인터페이스를 통해 교환되어야 하는 정보를 제안한다. 또한, 본 발명에서는, 등록 인터페이스의 경우 NSF 능력 정보(즉, NSF의 능력)을 명확히하여 I2NSF 프레임워크의 구성 요소가 능력들의 세트를 표준화된 방식으로 교환할 수 있어야 하기 때문에, 이를 위한 NSF 정보 모델을 정의한다.

[0104] 본 명세서에서, 네트워크 보안 기능(NSF: Network Security Function)은 수신된 패킷의 특정 처리를 담당하는 기능을 말한다. NSF는 프로토콜 스택의 다양한 계층(예를 들어, 네트워크 계층 또는 다른 개방형 시스템간 상호

접속(OSI: Open Systems Interconnection) 계층)에서 작동할 수 있다.

[0105] 또한, 본 명세서에서, NSF(또는 네트워크 보안 서비스 기능)은, 예를 들어, 방화벽, 침입 방지 시스템/침입 탐지 시스템(IPS: Intrusion Prevention System/IDS: Intrusion Detection System), 딥 패킷 검사(DPI: Deep Packet Inspection), 어플리케이션 가시성 및 제어(AVC: Application Visibility and Control), 네트워크 바이러스 및 맬웨어 검사(malware scanning), 샌드 박스(sandbox), 데이터 손실 방지(DLP: Data Loss Prevention)), 디도스(DDoS: Distributed Denial of Service) 완화(mitigation) 및 전송 계층 보안(TLS: Transport Layer Security) 프록시(proxy)와 같은 서비스를 나타낼(또는 제공할) 수 있다.

[0106] 또한, 본 명세서에서, 고급 검사/조치(Advanced Inspection/Action)는 NSF 직면 인터페이스에 대한 I2NSF 정보 모델과 마찬가지로, 보안 기능이 자체 검사 결과에 기반하여 추가 검사를 위해 다른 보안 기능을 호출함을 나타낼 수 있다.

[0107] 또한, 본 명세서에서, NSF 능력 정보(또는 NSF 프로파일)는 연관된 NSF 인스턴스의 검사 수행 능력을 나타낸다. 각각의 NSF 인스턴스는 자신이 제공하는 보안 서비스 유형과 리소스 용량 등을 지정하는 자체 NSF 능력 정보를 가진다.

[0108] 또한, 본 명세서에서, 데이터 모델은 데이터 저장소, 데이터 정의 언어, 쿼리 언어, 구현 언어 및 프로토콜에 의존하는 양식으로 환경(environment)에 대한 관심 개념의 표현을 말한다.

[0109] 또한, 본 명세서에서, 정보 모델은 데이터 저장소, 데이터 정의 언어, 쿼리 언어, 구현 언어 및 프로토콜과 독립적인 형태로 환경에 대한 관심 개념의 표현을 말한다.

[0110] 본 발명의 일 실시예에서, 보안 컨트롤러와 개발자 관리 시스템간 등록 인터페이스를 통해 NSF 인스턴스를 등록할 수 있다. 즉, 시스템의 보안 요구 사항에 따라 NSF가 필요할 수 있다. 보안 컨트롤러는 필요한 NSF 인스턴스의 등록을 요청하고, 개발자 관리 시스템은 NSF 인스턴스를 생성/등록하고 등록 인터페이스를 통해 보안 컨트롤러에 알릴 수 있다. 아래의 도면을 참조하여 설명한다.

[0112] 도 3은 본 발명이 적용되는 실시예로서, 등록 인터페이스를 통해 NSF 인스턴스를 등록하는 방법을 나타내는 흐름도이다.

[0113] 도 3을 참조하면, 보안 컨트롤러는 인스턴스화(instantiation) 요청 메시지를 개발자 관리 시스템으로 전송한다(S301). 보안 컨트롤러 및 개발자 관리 시스템은 앞서 도 2에서 설명한 I2NSF 프레임 워크의 구성 요소를 나타낸다.

[0114] 보안 컨트롤러로부터 인스턴스화 요청 메시지를 수신하면, 개발자 관리 시스템은 요청 받은 NSF에 대한 NSF 인스턴스를 생성할 수 있다. 예를 들어, S301 단계에 앞서, 보안 컨트롤러는 현재 시스템에서 필요한 능력들 세트(즉, NSF 능력 정보 또는 NSF 프로파일) 또는 특정 NSF의 서명을 인식할 수 있다. 그리고, 보안 컨트롤러는 인식된 정보를 포함하는 인스턴스화 요청 메시지를 개발자 관리 시스템으로 전송할 수 있다. 개발자 관리 시스템은 수신된 정보를 정보 모델 정의에 기초한 NSF와 일치시키고, 상기 수신된 정보와 일치하는 NSF 인스턴스를 생성할 수 있다.

[0115] 또는, 보안 컨트롤러는 역인스턴스화 요청 메시지를 개발자 관리 시스템으로 전송할 수 있다. 보안 컨트롤러로부터 역인스턴스화 요청 메시지를 수신하면, 개발자 관리 시스템은 해당 NSF 인스턴스를 제거할 수 있다.

[0116] 예를 들어, S301 단계에 앞서, 보안 컨트롤러는 현재 시스템에서 불필요한 능력들 세트(즉, NSF 액세스(또는 접근) 정보) 또는 특정 NSF의 서명을 인식할 수 있다. 보안 컨트롤러는 인식된 정보를 포함하는 역인스턴스화 요청 메시지를 개발자 관리 시스템으로 전송할 수 있다. 개발자 관리 시스템은 수신된 정보를 능력 정보 모델 정의에 기반한 NSF와 일치시키고, 상기 수신된 정보와 일치하는 NSF 인스턴스를 제거할 수 있다.

[0117] 보안 컨트롤러는 NSF 인스턴스의 등록을 위한 등록 메시지를 개발자 관리 시스템으로부터 수신한다(S302). 이 경우, 보안 컨트롤러는 등록 메시지를 수신한 후 해당 NSF 인스턴스를 시스템의 사용 가능한 NSF 인스턴스 목록에 추가할 수 있다.

[0118] 만약 S301 단계에서 보안 컨트롤러가 인스턴스화 요청 메시지가 아닌 역인스턴스화 요청 메시지를 전송한 경우, 보안 컨트롤러는 개발자 관리 시스템으로부터 삭제 메시지를 수신할 수 있다. 이 경우, 보안 컨트롤러는 해당 NSF 인스턴스를 시스템의 사용 가능한 NSF 인스턴스 목록에서 제거할 수 있다.

[0119] 또한, 일 실시예에서, 개발자 관리 시스템은 NSF 인스턴스를 등록할 수 있다. 구체적으로, 시스템의 보안 요구사항에 따라 일부 NSF가 기본적으로 요구될 수 있다. 이 경우, 개발자 관리 시스템은 보안 컨트롤러의 요청 없이 이러한 기본 NSF 인스턴스를 생성할 수 있다. 개발자 관리 시스템은 NSF 인스턴스들을 생성한 후, 등록 인터페이스를 통해 보안 컨트롤러에 NSF 인스턴스(또는 NSF 인스턴스에 관련된 정보)를 통지할 수 있다.

[0120] 또한, 일 실시예에서, 등록 인터페이스를 통해 필요한 보안 기능을 갖춘 NSF 인스턴스 요청이 수행될 수 있다. 예를 들어, 등록 인터페이스를 통해 다른 NSF에 의해 트리거된 고급 검사/조치를 제공하는 NSF 인스턴스가 생성될 수 있다. 즉, I2NSF 프레임 워크에서 NSF는 트래픽(traffic)의 고급 보안 검사를 위하여 다른 유형의 NSF를 트리거할 수 있다. 이때, 다음 NSF는 현재 NSF의 검사 결과 및 I2NSF 사용자 정책에 의해 결정될 수 있다. 그러나, 만약 이전 NSF에 의해 트리거된 고급 검사/조치를 제공할 수 있는 NSF 인스턴스가 없는 경우, 보안 컨트롤러는 등록 인터페이스를 통해 개발자 관리 시스템에 요청하여 NSF 인스턴스를 생성할 수 있다.

[0121] 또한, 일 실시예에서, I2NSF 사용자는 보안 컨트롤러에 보안 정책을 요청하고 보안 정책을 시행하려면 일련의 보안 기능이 필요할 수 있다. 또한, NSF가 특정 트래픽에 대한 고급 보안 검사를 위해 다른 유형의 NSF를 트리거하는 경우 고급 보안 검사를 수행하는 데 필요한 일부 보안 기능도 필요할 수 있다. 보안 컨트롤러에 요청된 기능으로 등록된 NSF 인스턴스가 없는 경우 보안 컨트롤러는 개발자 관리 시스템에 필요한 기능을 제공할 수 있는 새로운 NSF 인스턴스를 요청할 수 있다. 이러한 요청을 받으면 개발자 관리 시스템는 필요한 기능이 있는 NSF 인스턴스에 대한 인벤토리를 먼저 검색할 수 있다. 만약, 개발자 관리 시스템가 NSF 인스턴스를 찾지 못하면 필요한 보안 기능을 사용하여 새로운 NSF 인스턴스를 생성하고, 생성된 NSF 인스턴스를 보안 컨트롤러에 등록할 수 있다.

[0122] 또한, 일 실시예에서, I2NSF 사용자로부터 수신된 보안 정책 규칙을 시행함에 있어서 필요한 NSF 인스턴스가 생성될 수 있다. 즉, I2NSF 프레임 워크에서 I2NSF 사용자는 시스템에 필요한 보안 서비스를 결정할 수 있다. 만약 I2NSF 사용자가 요청한 보안 정책을 시행할 NSF 인스턴스가 없는 경우, 보안 컨트롤러는 등록 인터페이스를 통해 개발자 관리 시스템에 요청하여 필요한 NSF 인스턴스를 생성할 수 있다.

[0123] 또한, 일 실시예에서, 더 이상 필요하지 않은 NSF 인스턴스는 삭제될 수 있다. 즉, I2NSF 프레임 워크에서 사용자는 시스템에서 불필요한 보안 서비스를 결정할 수 있다. 또는, 다양한 유형의 NSF 인스턴스가 I2NSF 프레임 워크에서 실행될 수 있고, 시스템에서 수행될 보안 정책의 동적 변경에 따라 일부 유형의 NSF 인스턴스가 더 이상 필요하지 않을 수 있다. 이 경우, 보안 컨트롤러는 등록 인터페이스를 통해 NSF 인스턴스를 파기하도록 개발자 관리 시스템에 요청하여 기존 인스턴스를 삭제할 수 있다.

[0124] 또한, 일 실시예에서, NSF 인스턴스는 업데이트될 수 있다. NSF 인스턴스가 I2NSF 프레임 워크에 등록된 이후, NSF 인스턴스의 기능이 변경될 수 있다. 이러한 변경 사항은 보안 컨트롤러에게 알려야 한다. 이를 위해 개발자 관리 시스템은 일부 NSF 인스턴스를 업데이트한 후 등록 인터페이스를 통해 보안 컨트롤러에 업데이트를 알려줄 수 있다.

[0126] 도 4는 본 발명의 일 실시예에 따른 등록 인터페이스의 정보 모델을 예시하는 도면이다.

[0127] 종래의 I2NSF 등록 인터페이스는 보안 컨트롤러에 새로운 NSF 인스턴스를 등록하는 경우에만 사용되었다. 그러나, 본 발명의 실시예에서는, 언제든지 NSF 인스턴스화/역인스턴스화를 지원하기 위해 그 활용도를 확장할 수 있는 정보 모델을 제안한다.

[0128] 본 실시예에서 제안하는 정보 모델은 기능(functionality)을 위하여 등록 인터페이스를 통해 교환되어야 하는 정보를 설명한다. 특히, 등록 인터페이스에서는 NSF 프로파일(즉, NSF의 능력)을 명확히하여 I2NSF 프레임 워크의 구성 요소가 기능 세트를 표준화된 방식으로 교환할 수 있도록 하여야 하기 때문에, 본 실시예에서는, NSF 프로파일의 정보 모델을 정의한다.

[0129] 구체적으로, 도 4를 참조하면, 등록 인터페이스의 정보 모델은 인스턴스 관리 서브 모델(또는 하위 모델) 및/또는 등록 서브 모델을 포함할 수 있다. 인스턴스 관리 기능 및/또는 등록 기능은 목적 달성을 위하여 NSF 프로파일을 사용할 수 있다. 여기서, NSF 프로파일은 NSF 인스턴스가 제공할 수 있는 검사 능력을 설명 및/또는 규정하는 능력 객체를 나타낸다.

[0130] 즉, 등록 인터페이스 정보 모델 중에서 인스턴스 관리 서브 모델에 기초하여 NSF 인스턴스의 생성/제거가 수행될 수 있고, 등록 서브 모델에 기초하여 NSF 프로파일을 구성하는 세부 정보들이 정의될 수 있다.

[0132] NSF 인스턴스 관리 메커니즘(Instance Management Mechanism)

[0133] 도 5는 본 발명의 일 실시예에 따른 인스턴스 관리 서브 모델(Instance Management Sub-Model)을 예시하는 도면이다.

[0134] 도 5를 참조하면, I2NSF 프레임 워크의 보안 컨트롤러는 NSF의 인스턴스 관리를 위해 인스턴스화/재-인스턴스화(Instantiation/Re-instantiation) 요청 및/또는 역인스턴스화(Deinstantiation) 요청을 수행할 수 있다.

[0135] 도 5(a)를 참조하면, 보안 컨트롤러는 필요한 경우 개발자 관리 시스템에 인스턴스화/재-인스턴스화 요청 메시지를 전송할 수 있다. 이때, 상기 인스턴스화/재-인스턴스화 요청 메시지는 NSF 능력 정보를 포함할 수 있다. 보안 컨트롤러로부터 요청을 수신한 개발자 관리 시스템은 NSF 능력 정보에 기초하여 해당 NSF 인스턴스를 생성하고, 보안 컨트롤러에 처리 결과에 관련된 정보를 포함하는 응답 메시지를 전송할 수 있다.

[0136] 도 5(b)를 참조하면, 보안 컨트롤러는 필요한 경우 개발자 관리 시스템에 역인스턴스화 요청 메시지를 전송할 수 있다. 이때, 상기 역인스턴스화 요청 메시지는 NSF 액세스 정보를 포함할 수 있다. 보안 컨트롤러로부터 요청을 수신한 개발자 관리 시스템은 NSF 액세스 정보에 기반하여 해당 NSF 인스턴스를 제거하고 보안 컨트롤러에 처리 결과에 관련된 정보를 포함하는 응답 메시지를 전송할 수 있다.

[0138] NSF 등록 메커니즘(Registration Mechanism)

[0139] 도 6은 본 발명의 일 실시예에 따른 등록 서브 모델(Registration Sub-Model)을 예시하는 도면이다.

[0140] 도 6을 참조하면, 개발자 관리 시스템은 새로운 NSF 인스턴스를 등록하기 위하여 보안 컨트롤러에 등록 메시지(또는 NSF 등록 메시지, NSF 인스턴스 등록 메시지)를 전송(또는 생성)할 수 있다. 이때, 상기 등록 메시지는 NSF 능력 정보(Capability Information), NSF 액세스 정보(Access Information) 및 NSF 역할 기반 ACL(Access Control List) 중 적어도 하나를 포함할 수 있다.

[0141] NSF 능력 정보는 새로운 NSF 인스턴스의 검사 능력을 나타낸다. 그리고, NSF 액세스 정보는 다른 구성 요소의 새로운 인스턴스에 대한 네트워크 접근을 가능하게 하는 정보를 나타낸다. NSF 역할 기반 ACL 정보는 엔티티(entity)에 부여된 역할에 따라 NSF에 대한 엔티티 접근을 허용할지 또는 거부할지 결정하기 위하여 NSF의 액세스 정책을 지정한다. NSF 능력 정보, NSF 액세스 정보, NSF 역할 기반 ACL 정보의 세부 정보 모델은 자세히 후술한다.

[0142] 위와 같은 등록 프로세스 이후, I2NSF 능력 인터페이스(capability interface)는 새롭게 등록된 NSF 인스턴스를 제어하고 모니터링할 수 있다.

[0144] NSF 액세스 정보(Access Information)

[0145] NSF 액세스 정보는 NSF와 통신을 수행하기 위하여 요구되는 정보를 나타낸다. 본 발명의 일 실시예에서, NSF 액세스 정보는 IPv4(Internet Protocol version 4) 주소(address), IPv6(Internet Protocol version 6) 주소, 포트 번호(port number) 및/또는 지원되는 전송 프로토콜(transport protocol)을 포함할 수 있다.

[0146] 여기서 지원되는 전송 프로토콜은, 예를 들어, 가상 확장(Virtual Extensible) LAN(VXLAN), VXLAN를 위한 일반 프로토콜 확장(VXLAN-GPE(Generic Protocol Extension)), 일반 경로 캡슐화(GRE: Generic Route Encapsulation), 이더넷(Ethernet) 등을 포함할 수 있다.

[0147] NSF 액세스 정보는 전반적인 시스템(또는 NSF 인스턴스 리스트) 내에서 NSF 인스턴스의 서명(또는 고유 식별자)를 나타낸다. NSF 액세스 정보는 특정 NSF 인스턴스를 식별하는 데 사용될 수 있다.

[0149] NSF 능력 정보(Capability Information)(또는 NSF 인스턴스의 능력)

[0150] 도 7은 본 발명의 일 실시예에 따른, NSF 프로파일을 예시하는 도면이다.

[0151] 도 7을 참조하면, NSF 인스턴스의 검사 능력을 나타내는 NSF 프로파일(또는 NSF 능력 정보)은 다양한 NSF 인스

턴스의 능력 객체 (capability object)들을 포함할 수 있다.

[0152] 구체적으로, NSF 프로파일(또는 NSF 능력 객체)은 네트워크 보안 능력(Network-Security Capabilities), 컨텐트 보안 능력(Content-Security Capabilities), 공격 완화 능력(Attack Mitigation Capabilities) 및 퍼포먼스 능력(performance capabilities) 중 적어도 하나를 포함할 수 있다.

[0153] 여기서, 네트워크 보안 능력은 미리 정의된 보안 정책을 사용하여 네트워크 트래픽을 검사하고 처리하는 능력을 나타낸다. 컨텐츠 보안 능력 어플리케이션(application) 레이어에서 전달되는 트래픽 내용을 분석하는 능력을 나타낸다. 또한, 공격 완화 능력은 다양한 유형의 네트워크 공격을 탐지하고 완화하는 능력을 나타낸다.

[0154] 퍼포먼스 능력과 역할 기반 ACL은 이하에서 상세히 설명한다.

[0156] 퍼포먼스 능력(Performance Capabilities)

[0157] 도 8은 본 발명의 일 실시예에 따른, 퍼포먼스 능력(Performance Capabilities) 정보를 개략적으로 나타내는 도면이다.

[0158] 도 8을 참조하면, 퍼포먼스 능력(또는 수행 능력) 정보는 처리(Processing) 및/또는 대역폭(Bandwidth) 필드(또는 정보)를 포함할 수 있다. 여기서, 퍼포먼스 능력 정보는 NSF의 처리 능력(processing capability)을 나타낸다.

[0159] 퍼포먼스 능력 정보는 NSF가 현재 담당하는 작업량(workload)과 해당 정보의 비교를 통해 NSF가 정체(congestion) 상태인지 여부를 결정하는 데 사용될 수 있다. 또한, 퍼포먼스 능력 정보는 NSF에서 이용 가능한 능력과 같은 각각의 유형의 자원의 이용 가능한 양을 지정하는 데 사용될 수 있다.

[0160] 또한, 처리 정보는 NSF의 사용 가능한 처리 능력(processing power)를 나타낸다. 대역폭은 아웃 바운드(outbound), 인바운드(inbound)의 두 가지 경우에 사용 가능한 네트워크 양에 대한 정보를 나타낸다. 처리 정보 및 대역폭 정보는 보안 컨트롤러에 의한 NSF의 인스턴스(또는 인스턴스화) 요청에 사용될 수 있다.

[0161] 본 명세서 제안하는 등록 인터페이스는 생성된 인스턴스의 사용 및 제한을 제어할 수 있으며, 상태(status)에 따라 적절한 요청을 수행할 수 있다.

[0163] 역할 기반 ACL(Role-based Access Control List)

[0164] 역할 기반 ACL 정보는 엔티티(entity)에 부여된 역할에 따라 NSF에 대한 엔티티 접근을 허용할지 또는 거부할지 결정하기 위하여 NSF의 액세스 정책을 지정할 수 있다. 각각의 NSF는 역할 기반 ACL과 연결되어 엔티티의 액세스 요청을 허용할지 또는 거부할지 결정할 수 있다.

[0165] 도 9 및 도 10은 본 발명의 일 실시예에 따른 역할 기반 ACL(Role-based Access Control List)을 예시하는 도면이다.

[0166] 도 9를 참조하면, 역할 기반 ACL는 하나 이상의 역할 ID를 포함할 수 있다. 여기서, 역할 ID는 엔티티(예: 관리자, 개발자 등)의 역할을 식별하는데 이용되는 정보를 나타낸다.

[0167] 도 10을 참조하면, 역할 기반 ACL에 포함되는 각각의 역할 ID는 허용/거부로 분류되는 액세스 유형들을 포함할 수 있다. 즉, 역할 기반 ACL은 각각의 역할 ID에서 허용되거나 거부되는 액세스 유형들의 집합으로 구성될 수 있다. 여기서, 액세스 유형은 NSF 규칙(rule) 구성(configuration), NSF 규칙 업데이트(update) 및/또는 NSF 모니터링(monitoring)과 같은 특정 유형의 액세스 요청을 식별하는데 사용될 수 있다.

[0169] 이하에서는, 보안 컨트롤러와 개발자 관리 시스템간 I2NSF 등록 인터페이스를 위한 YANG 데이터 모델을 설명한다. 본 명세서에서, 데이터 모델의 단순화된 그래픽 표현이 사용된다. 이러한 다이어그램에서 기호의 의미는 다음과 같다.

[0170] - 대괄호 "["및 "]"는 목록 키를 묶는다.

[0171] - 데이터 노드 이름 앞의 약어 중에서, "rw"는 읽기-쓰기 구성을 의미하고, "ro"는 읽기 전용 상태 데이터를

의미한다.

[0172] - 데이터 노드 이름 뒤에 오는 기호 중에서, "?"는 선택적 노드를 의미하고, "*"는 리스트 및/또는 리프 리스트(leaf-list)를 의미한다.

[0173] - 괄호는 선택(choice) 및 케이스 노드(case node)를 묶으며, 케이스 노드는 콜론 (":")으로 표시된다.

[0174] - 줄임표("...")는 표시되지 않은 서브 트리의 내용을 나타낸다.

[0176] 등록 인터페이스(Registration Interface)

[0177] 도 11은 본 발명의 실시예에 따른 등록 인터페이스의 상위 레벨 YANG 데이터 모델을 예시하는 도면이다.

[0178] 도 11을 참조하면, I2NSF 등록 인터페이스의 상위 레벨 YANG 데이터 모델은 등록 요청 필드(또는 객체, 정보) 및 인스턴스 관리 요청 필드(또는 객체, 정보)를 포함할 수 있다. 본 실시예에서, I2NSF 시스템은 도 1 또는 도 2에서 상술한 I2NSF 시스템의 아키텍처를 갖는다. 또한, 도 11에 도시된 YANG 데이터 모델에 포함된 객체/필드/정보와 이들간의 관계는 도 11에 도시된 내용 및/또는 앞서 도 3 내지 도 10에서 설명한 내용에 의해 설명될 수 있다. 도 1 내지 도 10에서 상술한 설명과 중복된 설명은 생략하도록 한다.

[0180] 등록 요청(Registration Request)

[0181] 도 12는 본 발명의 실시예에 따른 등록 요청의 상위 레벨 YANG 데이터 모델을 예시하는 도면이다.

[0182] 본 발명의 실시예에서, 상술한 도 11의 등록 요청 필드는 도 12에 도시된 바와 같이 확장될 수 있다. 살펴보면, 등록 요청(또는 등록 요청 객체/필드/정보)은 보안 컨트롤러에 능력을 알리기 위하여 새로 생성된 NSF의 능력 정보(capability information)를 포함할 수 있다. 또한, 상기 등록 요청은 보안 컨트롤러가 NSF에 액세스할 수 있도록 하는 네트워크 액세스 정보도 포함할 수 있다.

[0183] 본 실시예에서, I2NSF 시스템은 도 1 또는 도 2에서 상술한 I2NSF 시스템의 아키텍처를 갖는다. 또한, 도 12에 도시된 YANG 데이터 모델에 포함된 객체/필드/정보와 이들간의 관계는 도 12에 도시된 내용 및/또는 앞서 도 3 내지 도 10에서 설명한 내용에 의해 설명될 수 있다. 도 1 내지 도 10에서 상술한 설명과 중복된 설명은 생략하도록 한다.

[0185] 인스턴스 관리 요청(Instance Management Request)

[0186] 도 13은 본 발명의 실시예에 따른 인스턴스 관리 요청의 상위 레벨 YANG 데이터 모델을 예시하는 도면이다.

[0187] 본 발명의 실시예에서, 상술한 도 11의 인스턴스 관리 요청 필드는 도 13에 도시된 바와 같이 확장될 수 있다. 살펴보면, 인스턴스 관리 요청 필드는 인스턴스화 요청(instanciation request), 역인스턴스화 요청(deinstanciation request) 및/또는 업데이트 요청(updating request) 필드(또는 객체, 정보)를 포함할 수 있다. 인스턴스화 요청은 필요한 NSF 능력 정보를 지정하는 NSF 능력 정보를 사용하여 새로운 NSF 인스턴스의 생성을 요청하는데 사용될 수 있다. 역인스턴스화 요청은 NSF 액세스 정보를 사용하여 기존 NSF를 제거하는데 사용될 수 있다. 업데이트 NSF 요청은 NSF 능력으로 기존 NSf 정보를 업데이트 하는데 사용될 수 있다.

[0188] 본 실시예에서, I2NSF 시스템은 도 1 또는 도 2에서 상술한 I2NSF 시스템의 아키텍처를 갖는다. 또한, 도 13에 도시된 YANG 데이터 모델에 포함된 객체/필드/정보와 이들간의 관계는 도 13에 도시된 내용 및/또는 앞서 도 3 내지 도 10에서 설명한 내용에 의해 설명될 수 있다. 도 1 내지 도 10에서 상술한 설명과 중복된 설명은 생략하도록 한다.

[0190] NSF 능력 정보(Capability Information)

[0191] 도 14는 본 발명의 실시예에 따른 NSF 능력 정보의 상위 레벨 YANG 데이터 모델을 예시하는 도면이다.

[0192] 본 발명의 실시예에서, 상술한 도 12 및 도 13의 NSF 능력 정보 필드(또는 객체, 정보)는 도 14에 도시된 바와 같이 확장될 수 있다. 살펴보면, NSF 능력 정보 필드는 I2NSF 능력 필드(또는 객체, 정보), 퍼포먼스 능력 필드

(또는 객체, 정보)를 포함할 수 있다.

[0193] 본 실시예에서, I2NSF 시스템은 도 1 또는 도 2에서 상술한 I2NSF 시스템의 아키텍처를 갖는다. 또한, 도 14에 도시된 YANG 데이터 모델에 포함된 객체/필드/정보와 이들간의 관계는 도 14에 도시된 내용 및/또는 앞서 도 3 내지 도 10에서 설명한 내용에 의해 설명될 수 있다. 도 1 내지 도 10에서 상술한 설명과 중복된 설명은 생략하도록 한다.

[0195] NSF 액세스 정보(Access Information)

[0196] 도 15는 본 발명의 실시예에 따른 NSF 액세스 정보의 상위 레벨 YANG 데이터 모델을 예시하는 도면이다.

[0197] 본 발명의 실시예에서, 상술한 도 12 및 도 13의 NSF 액세스 정보 필드(또는 객체, 정보)는 도 15에 도시된 바와 같이 확장될 수 있다. 살펴보면, NSF 액세스 정보 필드는 NSF 주소 필드(또는 객체, 정보), NSF 포트 주소 필드(또는 객체, 정보)를 포함할 수 있다.

[0198] 본 실시예에서, I2NSF 시스템은 도 1 또는 도 2에서 상술한 I2NSF 시스템의 아키텍처를 갖는다. 또한, 도 15에 도시된 YANG 데이터 모델에 포함된 객체/필드/정보와 이들간의 관계는 도 15에 도시된 내용 및/또는 앞서 도 3 내지 도 10에서 설명한 내용에 의해 설명될 수 있다. 도 1 내지 도 10에서 상술한 설명과 중복된 설명은 생략하도록 한다.

[0200] NSF 퍼포먼스 능력(Performance Capability)

[0201] 도 16은 본 발명의 실시예에 따른 NSF 퍼포먼스 능력의 상위 레벨 YANG 데이터 모델을 예시하는 도면이다.

[0202] 본 발명의 실시예에서, 상술한 도 14의 NSF 퍼포먼스 능력 필드(또는 객체, 정보)는 도 15에 도시된 바와 같이 확장될 수 있다. 살펴보면, NSF 퍼포먼스 능력 필드는 처리 필드(또는 객체, 정보), 대역폭 필드(또는 객체, 정보)를 포함할 수 있다. 보안 컨트롤러가 새로운 NSF 인스턴스를 생성하기 위해 개발자 관리 시스템에 요청하면, 퍼포먼스 능력은 새로운 인스턴스의 퍼포먼스 요구 사항을 지정하는데 사용될 수 있다.

[0203] 본 실시예에서, I2NSF 시스템은 도 1 또는 도 2에서 상술한 I2NSF 시스템의 아키텍처를 갖는다. 또한, 도 16에 도시된 YANG 데이터 모델에 포함된 객체/필드/정보와 이들간의 관계는 도 16에 도시된 내용 및/또는 앞서 도 3 내지 도 10에서 설명한 내용에 의해 설명될 수 있다. 도 1 내지 도 10에서 상술한 설명과 중복된 설명은 생략하도록 한다.

[0205] 역할 기반 ACL(Role-Based Access Control List)

[0206] 도 17은 본 발명의 실시예에 따른 역할 기반 ACL의 상위 레벨 YANG 데이터 모델을 예시하는 도면이다.

[0207] 본 발명의 실시예에서, 등록 인터페이스의 상위 레벨 YANG 데이터 모델은 도 17에 도시된 바와 같은 역할 기반 ACL을 포함할 수 있다.

[0208] 본 실시예에서, I2NSF 시스템은 도 1 또는 도 2에서 상술한 I2NSF 시스템의 아키텍처를 갖는다. 또한, 도 17에 도시된 YANG 데이터 모델에 포함된 객체/필드/정보와 이들간의 관계는 도 17에 도시된 내용 및/또는 앞서 도 3 내지 도 10에서 설명한 내용에 의해 설명될 수 있다. 도 1 내지 도 10에서 상술한 설명과 중복된 설명은 생략하도록 한다.

[0210] YANG 모듈(Module)

[0211] 도 18은 본 발명의 실시예에 따른 I2NSF 등록 인터페이스의 데이터 모델을 예시하는 도면이다.

[0212] 본 발명의 실시예에서, 보안 컨트롤러와 개발자 관리 시스템 간의 등록 인터페이스에 필요한 데이터의 정보 모델에 대한 YANG 모듈은 도 17에 도시된 바와 같을 수 있다.

[0213] 본 실시예에서, I2NSF 시스템은 도 1 또는 도 2에서 상술한 I2NSF 시스템의 아키텍처를 갖는다. 또한, 도 18에 도시된 YANG 모듈에 포함된 객체/필드/정보와 이들간의 관계는 도 18에 도시된 내용 및/또는 앞서 도 3 내지 도

10에서 설명한 내용에 의해 설명될 수 있다. 도 1 내지 도 10에서 상술한 설명과 중복된 설명은 생략하도록 한다.

[0215] 등록 인터페이스 데이터 모델의 XML 일 예

[0216] 도 19는 본 발명의 실시예에 따른 등록 인터페이스에 대한 XML 출력을 예시하는 도면이다.

[0217] 도 19를 참조하면, 등록 인터페이스를 통해 VoIP/VoLTE 보안 능력을 사용하여 IDS NSF를 등록할 수 있다. 상술한 등록 인터페이스를 위한 구성(configuration) XML은 도 19에 도시된 바와 같다.

[0218] 본 실시예에서, I2NSF 시스템은 도 1 또는 도 2에서 상술한 I2NSF 시스템의 아키텍처를 갖는다. 또한, 도 19에 도시된 YANG 모듈에 포함된 객체/필드/정보와 이들간의 관계는 도 19에 도시된 내용 및/또는 앞서 도 3 내지 도 10에서 설명한 내용에 의해 설명될 수 있다. 도 1 내지 도 10에서 상술한 설명과 중복된 설명은 생략하도록 한다.

[0220] 도 20은 본 발명의 일 실시예에 따른 네트워크 장치의 블록 구성도를 예시한다. 네트워크 장치는 상술한 I2NSF 시스템(또는 보안 관리 시스템)에 해당하거나, I2NSF 시스템 내에 포함되는 장치일 수 있다. I2NSF 시스템 내에 포함되는 장치의 예로는 상술한 I2NSF, 보안 컨트롤러, 개발자 관리 시스템, NSF 등이 포함될 수 있다.

[0221] 도 20을 참조하면, 네트워크 장치(2000)는 프로세서(processor, 2010), 메모리(memory, 2020) 및 통신 모듈(communication module, 2030)을 포함한다.

[0222] 프로세서(2010)는 앞서 도 1 내지 도 19에서 제안된 기능, 과정 및/또는 방법을 구현한다. 메모리(2020)는 프로세서(2010)와 연결되어, 프로세서(2010)를 구동하기 위한 다양한 정보를 저장한다. 통신 모듈(2030)은 프로세서(2010)와 연결되어, 유/무선 신호를 송신 및/또는 수신한다.

[0223] 메모리(2020)는 프로세서(2010) 내부 또는 외부에 있을 수 있고, 잘 알려진 다양한 수단으로 프로세서(2010)와 연결될 수 있다.

[0224] 이상에서 설명된 실시예들은 본 발명의 구성요소들과 특징들이 소정 형태로 결합된 것들이다. 각 구성요소 또는 특징은 별도의 명시적 언급이 없는 한 선택적인 것으로 고려되어야 한다. 각 구성요소 또는 특징은 다른 구성요소나 특징과 결합되지 않은 형태로 실시될 수 있다. 또한, 일부 구성요소들 및/또는 특징들을 결합하여 본 발명의 실시예를 구성하는 것도 가능하다. 본 발명의 실시예들에서 설명되는 동작들의 순서는 변경될 수 있다. 어느 실시예의 일부 구성이나 특징은 다른 실시예에 포함될 수 있고, 또는 다른 실시예의 대응하는 구성 또는 특징과 교체될 수 있다. 특허청구범위에서 명시적인 인용 관계가 있지 않은 청구항들을 결합하여 실시예를 구성하거나 출원 후의 보정에 의해 새로운 청구항으로 포함시킬 수 있음은 자명하다.

[0225] 본 발명에 따른 실시예는 다양한 수단, 예를 들어, 하드웨어, 펌웨어(firmware), 소프트웨어 또는 그것들의 결합 등에 의해 구현될 수 있다. 하드웨어에 의한 구현의 경우, 본 발명의 일 실시예는 하나 또는 그 이상의 ASICs(application specific integrated circuits), DSPs(digital signal processors), DSPDs(digital signal processing devices), PLDs(programmable logic devices), FPGAs(field programmable gate arrays), 프로세서, 콘트롤러, 마이크로 콘트롤러, 마이크로 프로세서 등에 의해 구현될 수 있다.

[0226] 또한, 펌웨어나 소프트웨어에 의한 구현의 경우, 본 발명의 일 실시예는 이상에서 설명된 기능 또는 동작들을 수행하는 모듈, 절차, 함수 등의 형태로 구현되어, 다양한 컴퓨터 수단을 통하여 판독 가능한 기록매체에 기록될 수 있다. 여기서, 기록매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 기록매체에 기록되는 프로그램 명령은 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 예컨대 기록매체는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(Magnetic Media), CD-ROM(Compact Disk Read Only Memory), DVD(Digital Video Disk)와 같은 광 기록 매체(Optical Media), 플롭티컬 디스크(Floptical Disk)와 같은 자기-광 매체(Magneto-Optical Media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치를 포함한다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함할 수 있다. 이러한 하드웨어 장치는 본 발명의 동작을 수행하기 위해 하나 이상의 소프트웨어 모로서 작동하도록

구성될 수 있으며, 그 역도 마찬가지이다.

[0227] 아울러, 본 발명에 따른 장치나 단말은 하나 이상의 프로세서로 하여금 앞서 설명한 기능들과 프로세스를 수행하도록 하는 명령에 의하여 구동될 수 있다. 예를 들어 그러한 명령으로는, 예컨대 JavaScript나 ECMAScript 명령 등의 스크립트 명령과 같은 해석되는 명령이나 실행 가능한 코드 혹은 컴퓨터로 판독 가능한 매체에 저장되는 기타의 명령이 포함될 수 있다. 나아가 본 발명에 따른 장치는 서버 팜(Server Farm)과 같이 네트워크에 걸쳐서 분산형으로 구현될 수 있으며, 혹은 단일의 컴퓨터 장치에서 구현될 수도 있다.

[0228] 또한, 본 발명에 따른 장치에 탑재되고 본 발명에 따른 방법을 실행하는 컴퓨터 프로그램(프로그램, 소프트웨어, 소프트웨어 어플리케이션, 스크립트 혹은 코드로도 알려져 있음)은 컴파일 되거나 해석된 언어나 선험적 혹은 절차적 언어를 포함하는 프로그래밍 언어의 어떠한 형태로도 작성될 수 있으며, 독립형 프로그램이나 모듈, 컴포넌트, 서브루틴 혹은 컴퓨터 환경에서 사용하기에 적합한 다른 유닛을 포함하여 어떠한 형태로도 전개될 수 있다. 컴퓨터 프로그램은 파일 시스템의 파일에 반드시 대응하는 것은 아니다. 프로그램은 요청된 프로그램에 제공되는 단일 파일 내에, 혹은 다중의 상호 작용하는 파일(예컨대, 하나 이상의 모듈, 하위 프로그램 혹은 코드의 일부를 저장하는 파일) 내에, 혹은 다른 프로그램이나 데이터를 보유하는 파일의 일부(예컨대, 마크업 언어 문서 내에 저장되는 하나 이상의 스크립트) 내에 저장될 수 있다. 컴퓨터 프로그램은 하나의 사이트에 위치하거나 복수의 사이트에 걸쳐서 분산되어 통신 네트워크에 의해 상호 접속된 다중 컴퓨터나 하나의 컴퓨터 상에서 실행되도록 전개될 수 있다.

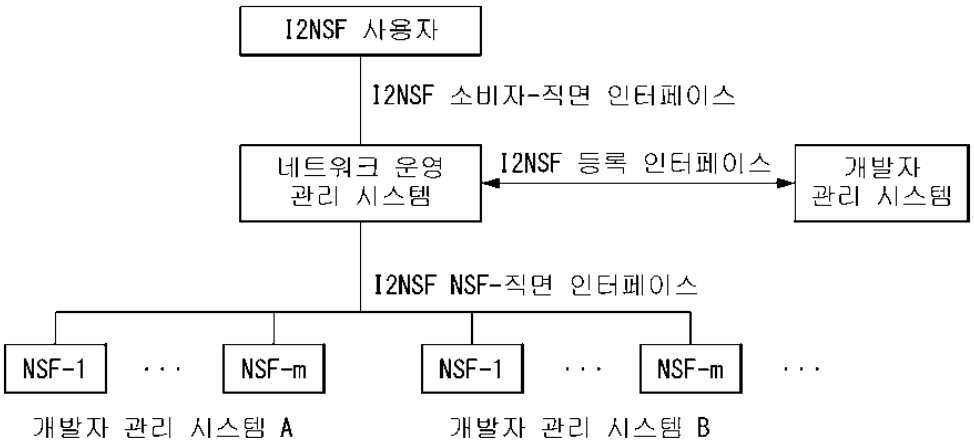
[0229] 본 발명은 본 발명의 필수적 특징을 벗어나지 않는 범위에서 다른 특정한 형태로 구체화될 수 있음은 당업자에게 자명하다. 따라서, 상술한 상세한 설명은 모든 면에서 제한적으로 해석되어서는 아니 되고 예시적인 것으로 고려되어야 한다. 본 발명의 범위는 첨부된 청구항의 합리적 해석에 의해 결정되어야 하고, 본 발명의 등가적 범위 내에서의 모든 변경은 본 발명의 범위에 포함된다.

산업상 이용가능성

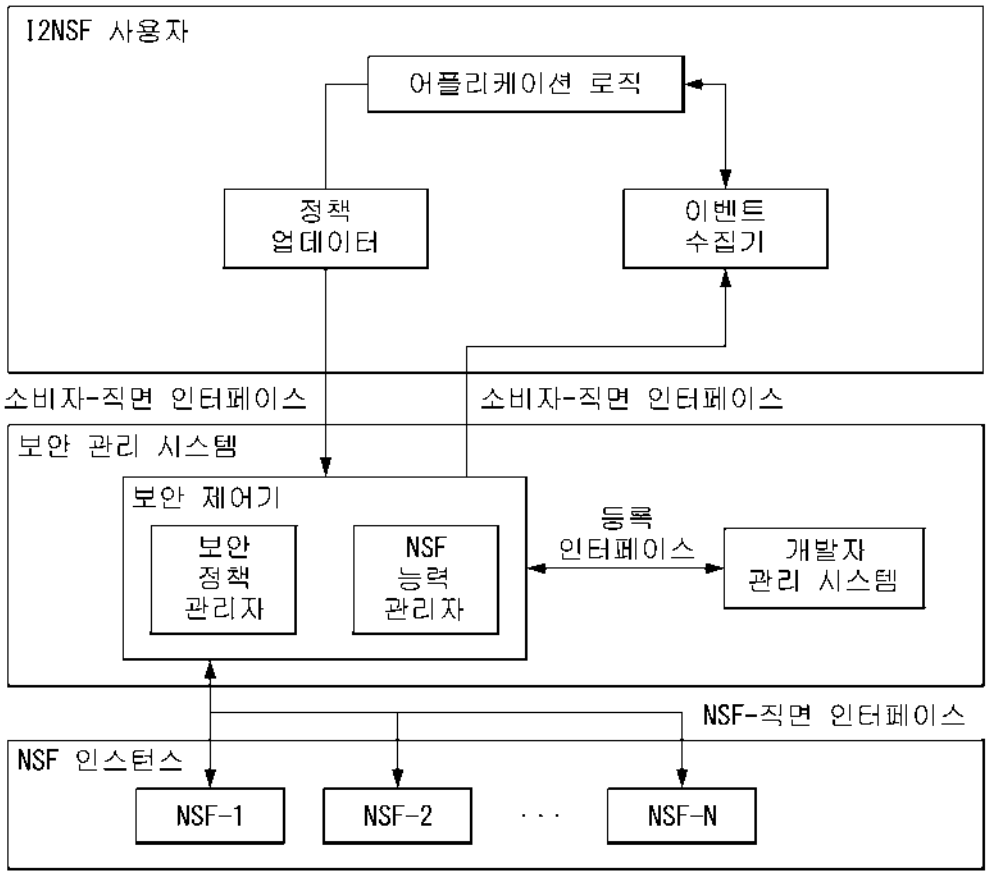
[0230] 이상, 전술한 본 발명의 바람직한 실시예는, 예시의 목적을 위해 개시된 것으로, 당업자라면 이하 첨부된 특허청구범위에 개시된 본 발명의 기술적 사상과 그 기술적 범위 내에서, 다양한 다른 실시예들을 개량, 변경, 대체 또는 부가 등이 가능할 것이다.

도면

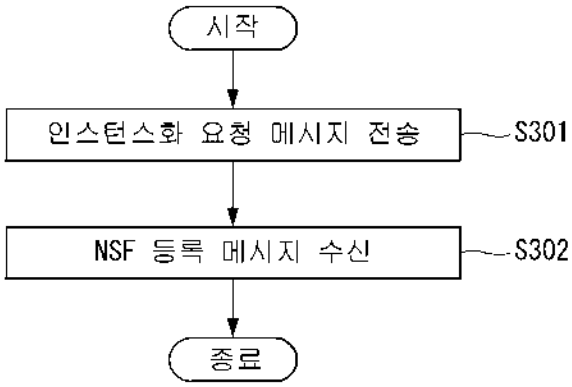
도면1



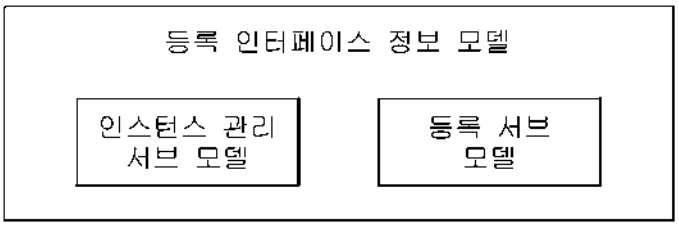
도면2


I2NSF 사용자
어플리케이션 로직
정책
업데이터
이벤트
수집기
소비자-직면 인터페이스
소비자-직면 인터페이스
보안 관리 시스템
보안 제어기
보안
정책
관리자
NSF
능력
관리자
등록
인터페이스
개발자
관리 시스템
NSF-직면 인터페이스
NSF 인스턴스
NSF-1
NSF-2
· · ·
NSF-N

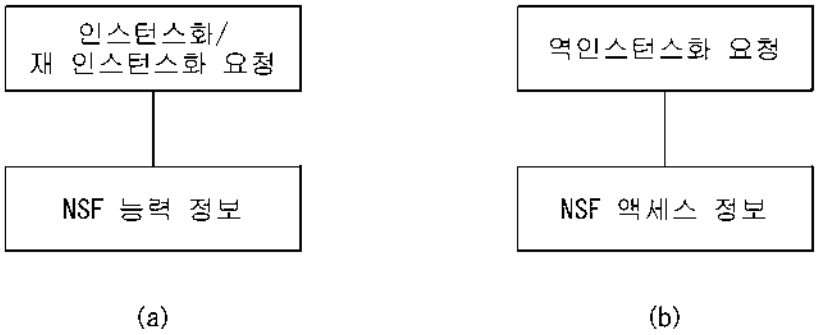
도면3


시작
인스턴스화 요청 메시지 전송
S301
NSF 등록 메시지 수신
S302
종료

도면4

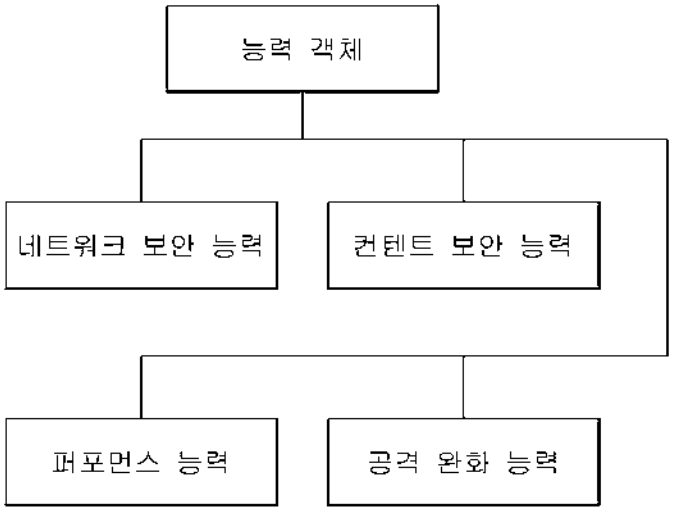

등록 인터페이스 정보 모델
인스턴스 관리
서브 모델
등록 서브
모델

도면5

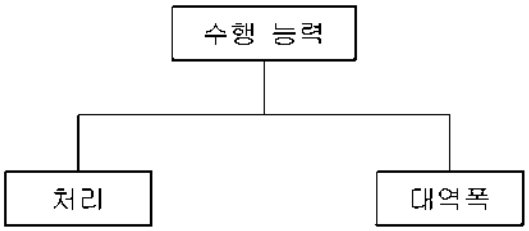


도면6

도면7

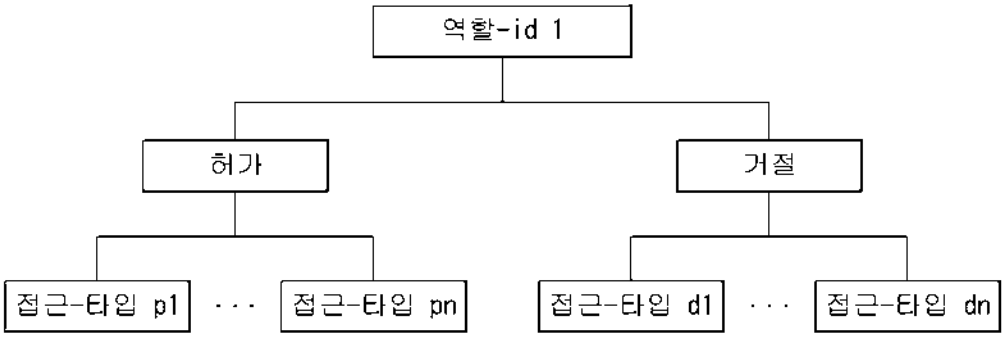


도면8



도면9

도면10



도면11

```
module : ietf-i2nsf-regs-interface-model
  +--rw regs-req
  |  uses i2nsf-regs-req
  +--rw instance-mgnt-req
  |  uses i2nsf-instance-mgnt-req
```

도면12

```
Registration Request
  +--rw i2nsf-regs-req
    +--rw nsf-capability-information
    |  uses i2nsf-nsf-capability-information
    +--rw nsf-access-info
    |  uses i2nsf-nsf-access-info
```

도면13

```
Instance Management Request
  +--rw i2nsf-instance-mgnt-req
    +--rw req-level uint16
    +--rw req-id uint64
    +--rw (req-type)?
      +--rw (instanciation-request)
        +--rw in-nsf-capability-information
        |  uses i2nsf-nsf-capability-information
      +--rw (deinstanciation-request)
        +--rw de-nsf-access-info
        |  uses i2nsf-nsf-access-info
      +--rw (updating-request)
        +--rw update-nsf-capability-information
        |  uses i2nsf-nsf-capability-information
```

도면14

```
NSF Capability Information
  +--rw i2nsf-nsf-capability-information
    +--rw i2nsf-capability
    |  uses ietf-i2nsf-capability
    +--rw performance-capability
    |  uses i2nsf-nsf-performance-caps
```

도면15

```
NSF Access Information
  +--rw i2nsf-nsf-access-info
    +--rw nsf-address  inet:ipv4-address
    +--rw nsf-port-address inet:port-number
```

도면16

```
NSF Performance Capability
  +--rw i2nsf-nsf-performance-caps
   +--rw processing
   |    +--rw processing-average uint16
   |    +--rw processing-peak uint16
   +--rw bandwidth
   |    +--rw outbound
   |    |  +--rw outbound-average uint16
   |    |  +--rw outbound-peak uint16
   |    +--rw inbound
   |    |  +--rw inbound-average uint16
   |    |  +--rw inbound-peak uint16
```

도면17

```
Role-Based ACL
   +--rw role-based-acl
         uses ietf-netmod-acl-model
```

도면18a

```
<CODE BEGINS> file "ietf-i2nsf-regs-interface@2018-07-26.yang"
    module ietf-i2nsf-regs-interface {
      namespace
       "urn:ietf:params:xml:ns:yang:ietf-i2nsf-regs-interface";
      prefix
        regs-interface;
      import ietf-inet-types{
       prefix inet;
      }

      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: Adrian Farrel
       <mailto:Adrain@olddog.co.uk>

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

       Editor: Sangwon Hyun
       <mailto:swhyun77@skku.edu>

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

       Editor: Taekyun Roh
       <mailto:tkroh0198@skku.edu>

       Editor: Sarang Wi
       <mailto:dnl9795@skku.edu>

       Editor: Jung-Soo Park
```

도면18b

```
    <mailto:pjs@etri.re.kr>";

  description
    "It defines a YANG data module for Registration Interface.";
  revision "2018-07-26"{
    description "The second revision";
    reference
    "draft-ietf-i2nsf-capability-data-model-01";
  }
list interface-container{
    key "interface-name";
    description
    "i2nsf-reg-interface-container";
    leaf interface-name{
      type string;
      description
      "interface name";
    }
    container i2nsf-regs-req {
     description
     "The capability information of newly
     created NSF to notify its
     capability to Security Controller";
     container nsf-capability-information {
      description
      "nsf-capability-information";
      uses i2nsf-nsf-capability-information;
     }
    container nsf-access-info {
     description
     "nsf-access-info";
     uses i2nsf-nsf-access-info;
    }
    container ietf-netmod-acl-model{
     description
     "netmod-acl-model";
     uses ietf-netmod-acl-model;
    }
   }
    container i2nsf-instance-mgmt-req {
     description
     "Required information for instanciation-request,
     deinstanciation-request and updating-request";
     leaf req-level {
      type uint16;
      description
      "req-level";
    }
```

도면18c

```
      leaf req-id {
       type uint64;
       mandatory true;
       description
       "req-id";
      }
      choice req-type {
       description
       "req-type";
       case instanciation-request {
        description
        "instanciation-request";
       container in-nsf-capability-information {
         description
         "nsf-capability-information";
         uses i2nsf-nsf-capability-information;
        }
       }
       case deinstanciation-request {
        description
        "deinstanciation-request";
        container de-nsf-access-info {
          description
          "nsf-access-info";
          uses i2nsf-nsf-access-info;
         }
       }
       case updating-request {
        description
         "updating nsf's information";
         container update-nsf-capability-information {
          description
          "nsf-capability-information";
          uses i2nsf-nsf-capability-information;
         }
       }
      }
     }
}
grouping i2nsf-nsf-performance-caps {
     description
     "NSF performance capailities";
     container processing{
       description
       "processing info";
       leaf processing-average{
        type uint16;
        description
```

도면18d

```
        "processing-average";
      }
      leaf processing-peak{
       type uint16;
       description
       "processing peak";
      }
    }
    container bandwidth{
      description
      "bandwidth info";
      container inbound{
       description
       "inbound";
       leaf inbound-average{
        type uint16;
        description
        "inbound-average";
       }
       leaf inbound-peak{
        type uint16;
        description
        "inbound-peak";
       }
      }
      container outbound{
       description
       "outbound";
       leaf outbound-average{
        type uint16;
        description
        "outbound-average";
       }
       leaf outbound-peak{
        type uint16;
        description
        "outbound-peak";
       }
      }
    }
}
grouping i2nsf-nsf-capability-information {
  description
  "Detail information of an NSF";
  container performance-capability {
   uses i2nsf-nsf-performance-caps;
   description
   "performance-capability";
```

도면18e

```
      }
      container i2nsf-capability {
       description
       "It refers draft-ietf-i2nsf-capability-data-model-01.txt
       later";
      }
    }
    grouping ietf-netmod-acl-model {
      description
      "Detail information";
      container role-based-acl {
       description
       "It refers draft-ietf-netmod-acl-model-19.txt
       later";
      }
    }
    grouping i2nsf-nsf-access-info {
      description
      "NSF access information";
      leaf nsf-address {
       type inet:ipv4-address;
       mandatory true;
       description
       "nsf-address";
      }
      leaf nsf-port-address {
       type inet:port-number;
       description
       "nsf-port-address";
      }
    }
}

<CODE ENDS>
```

도면19a

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc xmlns="urn:ietf:params:netconf:base:1.0" message-id="1">
  <edit-config>
    <target>
```

도면19b

```
  <running/>
</target>
<config>
<i2nsf-regs-req>
  <i2nsf-nsf-capability-information>
    <ietf-i2nsf-capability>
      <nsf-capabilities>
       <nsf-capabilities-id>1</nsf-capabilities-id>
        <con-sec-control-capabilities>
         <content-security-control>
          <ids>
           <ids-support>true</ids-support>
           <ids-fcn nc:operation="create">
            <ids-fcn-name>ids-service</ids-fcn-name>
           </ids-fcn>
          </ids>
          <voip-volte>
           <voip-volte-support>true</voip-volte-support>
           <voip-volte-fcn nc:operation="create">
            <voip-volte-fcn-name>
             ips-service
            </voip-volte-fcn-name>
           </voip-volte-fcn>
          </voip-volte>
         </content-security-control>
        </con-sec-control-capabilities>
      </nsf-capabilities>
    </ietf-i2nsf-capability>
    <i2nsf-nsf-performance-caps>
      <processing>
       <processing-average>1000</processing-average>
       <processing-peak>5000</processing-peak>
      </processing>
      <bandwidth>
       <outbound>
        <outbound-average>1000</outbound-average>
        <outbound-peak>5000</outbound-peak>
       </outbound>
       <inbound>
        <inbound-average>1000</inbound-average>
        <inbound-peak>5000</inbound-peak>
       </inbound>
      </bandwidth>
    </i2nsf-nsf-performance-caps>
  </i2nsf-nsf-capability-information>
  <nsf-access-info>
   <nsf-address>10.0.0.1</nsf-address>
   <nsf-port-address>145</nsf-port-address>
```

도면19c

```
      </nsf-access-info>
    </i2nsf-regs-req>
    </config>
  </edit-config>
</rpc>
```

도면20