(12) 특허협력조약에 의하여 공개된 국제출원

(19) 세계지식재산권기구
국제사무국

(43) 국제공개일
2018 년 9 월 20 일 (20.09.2018)

WIPO | PCT

(10) 국제공개번호
WO 2018/169294 A1

(51) 국제특허분류:
H04L 29/06 (2006.01) *H04L 12/24* (2006.01)

(21) 국제출원번호: PCT/KR2018/002957

(22) 국제출원일: 2018 년 3 월 13 일 (13.03.2018)

(25) 출원언어: 한국어

(26) 공개언어: 한국어

(30) 우선권정보:
10-2017-0031427 2017 년 3 월 13 일 (13.03.2017) KR

(71) 출원인: 성균관대학교 산학협력단 (RESEARCH&BUSINESS FOUNDATION SUNGKYUNKWAN UNIVERSITY) [KR/KR]; 16419 경기도 수원시 장안구 서부로 2066, Gyeonggi-do (KR).

(72) 발명자: 정재훈 (JEONG, Jaehoon); 46294 부산시 금정구 금강로 225, 204-1501, Busan (KR). 김진용 (KIM, Jinyong); 15266 경기도 안산시 단원구 와동1로 51-18, Gyeonggi-do (KR).

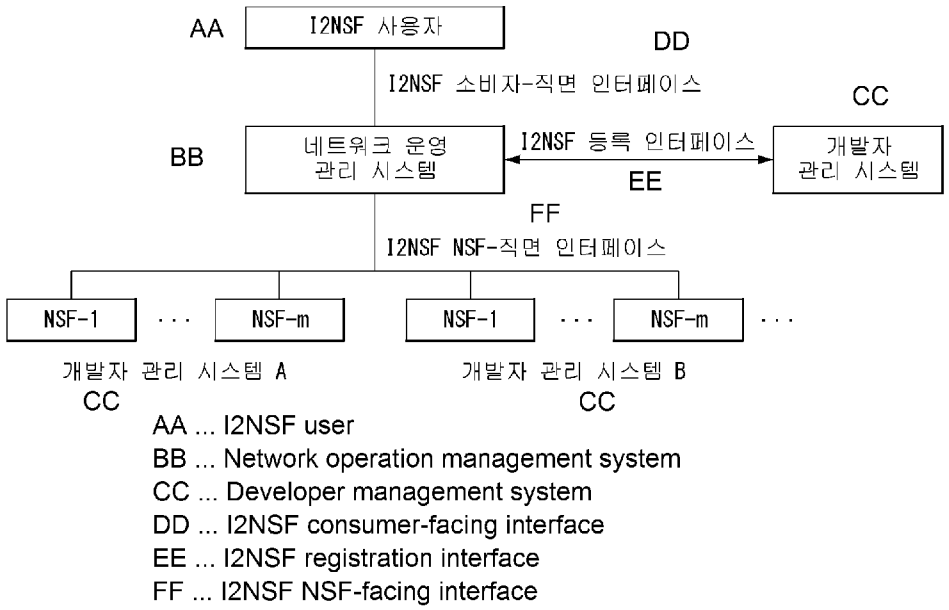
(74) 대리인: 특허법인 로얄 (ROYAL PATENT & LAW OFFICE); 06648 서울시 서초구 반포대로 104 서일빌딩 4 층, Seoul (KR).

(81) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 국내 권리의 보호를 위하여): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 역내 권리의 보호를 위하여): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 유라시아 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 유럽 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI

(54) **Title:** I2NSF NETWORK SECURITY FUNCTION-FACING INTERFACE YANG DATA MODEL

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



(57) **Abstract:** The present invention relates to a method and an apparatus for providing a security service by a network operation management system in a security management system. The present invention may provide a method and an apparatus which receive a first security policy of a high level from an interface to network security function (I2NSF) user, generate a second security policy of a low level corresponding to the first security policy, and transmit a packet including the generated second security policy so as to configure the second security policy for each of multiple network security functions (NSFs), wherein the network operation management system is connected to each of the multiple NSFs through an I2NSF NSF-facing interface.

WO 2018/169294 A1

[다음 쪽 계속]

(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

공개:

— 국제조사보고서와 함께 (조약 제21조(3))

(57) 요약서: 보안 관리 시스템에서 네트워크 운영 관리 시스템이 보안 서비스를 제공하기 위한 방법 및 장치에 관한 것이다. 본 발명에 의하면, I2NSF(Interface to Network Security Function) 사용자로부터 상위 레벨(High-Level)의 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 전송하되, 상기 네트워크 운영 관리 시스템과 상기 복수의 NSF 각각은 I2NSF NSF-직면 인터페이스로 연결되어 있는 방법 및 장치를 제공할 수 있다.

명세서

발명의 명칭: I2NSF 네트워크 보안 능력에 직면한 인터페이스 YANG 데이터 모델

기술분야

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

배경기술

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

[3] 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 상세한 설명

기술적 과제

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

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

[6] 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.

과제 해결 수단

[7] 상술한 기술적 과제를 해결하기 위하여, 본 발명의 실시 예에 따른 보안 관리 시스템에서 네트워크 운영 관리 시스템이 보안 서비스를 제공하기 위한 방법은 I2NSF(Interface to Network Security Function) 사용자로부터 상위 레벨(High-Level)의 제 1 보안 정책을 수신하는 단계; 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 포함하되,

상기 네트워크 운영 관리 시스템과 상기 복수의 NSF 각각은 I2NSF NSF-직면 인터페이스로 연결된다.

[8] 또한, 본 발명에서, 상기 제 2 보안 정책은 적용되는 정책 규칙, 및 일반적인 보안 기능을 위한 동작을 나타나내는 기본 동작 정보를 포함한다.

[9] 또한, 본 발명에서, 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 포함한다.

[10] 또한, 본 발명에서, 상기 정책 규칙은 상기 정책 규칙이 적용되는 시간을 나타내는 시간 정보를 더 포함한다.

[11] 또한, 본 발명에서, 상기 시간 정보는 상기 정책 규칙이 적용되는 절대적인 시간을 나타내는 절대 시간 정보 또는 상기 정책 규칙이 적용되는 주기적인 시간을 나타내는 주기 정보 중 적어도 하나를 더 포함한다.

[12] 또한, 본 발명에서, 상기 이벤트 절은 상기 조건 절을 평가할 수 있는지 여부를 결정하는데 사용된다.

[13] 또한, 본 발명에서, 상기 조건 절은 포트 번호와 관련된 룰의 설정을 위한 포트 번호 값을 포함한다.

[14] 또한, 본 발명에서, 상기 동작 절은 입력 동작(ingress action), 전송 동작(egress action) 및 적용 프로파일 동작(apply profile action)을 포함한다.

[15] 또한, 본 발명에서, 상기 프로파일 동작은 응용 프로그램 계층에 적용되는 보안 기능을 나타내는 컨텐츠 보안 제어, 및 네트워크 공격의 탐지 및 완화를 위한 공격 완화 제어를 포함한다.

[16] 또한, 본 발명에서, 상기 조건 절(Condition Clause), 및 상기 동작 절(Action Clause)은 다중 조건의 적용을 위한 컨테이너 구조로 구성된다.

[17] 또한, 본 발명은, 상위 레벨(High-Level)의 제 1 보안 정책을 생성하는 I2NSF(Interface to Network Security Function) 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 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 연결되어 있는 아키텍처를 제공한다.

발명의 효과

[18] 본 발명의 실시 예에 따르면, I2NSF(Interface to Network Security Functions)에서 네트워크 보안 능력(Network Security Functions: NSF)에 직면한 인터페이스를

위한 정보 모델과 보안 서비스를 위한 YANG 데이터 모델을 설계할 수 있다.

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

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

[21] 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

[22] 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.

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

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

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

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

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

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

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

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

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

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

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

[34] 도 12a 및 도 12b는 본 발명의 일 실시 예에 따른 네트워크 보안 정책 식별을 위한 데이터 모델 구조를 예시한다.

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

예시한다.

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

[37] 도 15a 및 도 15b는 본 발명의 일 실시 예에 따른 액션 규칙을 위한 데이터 모델 구조를 예시한다.

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

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

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

발명의 실시를 위한 형태

[41] 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.

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

[43] 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.

[44]

[45] 최근에는, 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.

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

[47] 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.

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

[49] 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(Interface to Network Security Function)을 포함한다.

[50] 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.

[51] **I2NSF 사용자**

[52] 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.

[53] 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.

[54] 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.

[55] 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.

[56] 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.

[57] **네트워크 운영 관리 시스템**

[58] 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.

[59] 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.

[60] 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.

[61] **NSF**

[62] 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.

[63] **개발자 관리 시스템**

[64] 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.

[65] 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.

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

[67] 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.

[68] 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.

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

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

[71] 이 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:

[72] - 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 능력을 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.

[73] - 모니터링 인터페이스(Monitoring Interface): 하나 이상의 선택된 NSF로부터의 모니터링 정보를 획득하기 위해 I2NSF 관리 시스템에 의해 시용되는 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 능력은 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.

[74] 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.

[75] 한편, I2NSF 관리 시스템은 주어진 NSF의 모든 능력들을 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 능력을 자유롭게 사용할 수 있게 된다.

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

[77] 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 능력을 정의하기 위한 전용 인터페이스를 가질 필요가 있다. 이러한 전용 인터페이스는 I2NSF 등록 인터페이스(RI)로 지칭될 수 있다.

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

[79]

[80] 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.

[81] 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 능력 관리자(NSF capability manager)를 컴포넌트로서 포함한다.

[82] 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 NFC 인스턴스 계층과 통신한다. 예를 들면, 보안 관리 시스템 계층의 보안 제어기는 NSF-직면 인터페이스를 통해 NFC 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.

[83] 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.

[84] **I2NSF 사용자**

[85] 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.

[86] 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.

[87] 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.

[88] **보안 관리 시스템**

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

[90] 보안 정책 관리자는 CFI를 통해 정책 업데이터로부터 고수준 정책을 수신하고, 이 정책을 여러 저수준 정책으로 맵핑할 수 있다. 이 저수준 정책은 NSF 능력 관리자에 등록된 주어진 NSF 능력과 관련된다. 또한, 보안 정책 관리자는 이

정책을 NFI를 통해 NSF(들)로 전달할 수 있다.

[91] NSF 능력 관리자는 주어진 NSF 능력과 관련된 저수준 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 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.

[92] 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.

[93] **NSF 인스턴스(NSF Instances)**

[94] 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, NFC는 수신된 저수준 보안 정책에 기초하여 악의적인 네트워크 트래픽을 감지하고, 이를 차단하거나 완화할 수 있다.

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

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

[97] 보안 능력은 보안 정책 시행 목적으로 사용할 수 있는 일련의 네트워크의 보안과 관련된 능력을 말한다. 보안 능력은 실제 구현되는 보안 제어 메커니즘과는 독립적이며, 모든 NSF는 NSF에서 제공할 수 있는 능력들의 세트가 등록되어 있다.

[98] 보안 능력은 특정 NSF가 제공하는 보안 능력을 모호하지 않게 설명함으로써 맞춤형 보안 보호를 정의 할 수 있는 방법을 제공하는 마켓 리더이다. 또한, 보안 능력을 통해 보안 능력의 공급 업체의 중립적인 방식으로 설명 할 수 있다.

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

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

[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 서비스 지향

인터페이스.

[102] 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.

[103] - 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.

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

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

[106] AAA: Access control, Authorization, Authentication

[107] ACL: Access Control List

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

[109] ECA: Event-Condition-Action

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

[111] FW: Firewall

[112] GNSF: Generic Network Security Function

[113] HTTP: HyperText Transfer Protocol

[114] I2NSF: Interface to Network Security Functions

[115] IPS: Intrusion Prevention System

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

[117] MIME: Multipurpose Internet Mail Extensions

[118] NAT: Network Address Translation

[119] NSF: Network Security Function

[120] RPC: Remote Procedure Call

[121] SMA: String Matching Algorithm

[122] URL: Uniform Resource Locator

[123] VPN: Virtual Private Network

[124] <u>정보 모델 설계</u>

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

[126] 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.

[127]

[128] 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.

[129] 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.

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

[131]

[132] 능력 정보 모델 오버뷰(Capability Information Model Overview)

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

[134]

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

[136]

[137] - 독립성(Independence): 각 보안 능력은 다른 능력에 최소한의 중첩 또는 종속성을 갖는 독립적 인 능력이어야 한다. 이를 통해 각 보안 능력을 자유롭게 사용 및 조합 할 수 있다. 더 중요한 것은, 하나의 능력으로의 변경이 다른 능력에 영향을 미치지 않는다는 것이다.

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

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

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

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

[142] - 확장성: 관리 시스템에는 scale up/down 또는 scale in/out 능력이 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 능력은 보안 컨트롤러에 보고통계를 지원해야 스케일링을 호출해야 하는지 여부를 결정하는 데 도움이 될 수 있다.

[143]

[144] 위의 원칙에 따라 표준 인터페이스를 갖춘 추상 및 벤더 중립 능력 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.

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

[146] 또한, NSF에 의해 알려지지 않은 위협(예를 들어, zero-day exploits 및 unknown malware)이 보고 될 때, 새로운 능력이 생성될 수 있고 및/또는 기존의 능력이 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.

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

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

[149] "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):

[150] - 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.

[151] - 조건(Condition): 조건은 알려진 속성, 특징 및/또는 값의 세트와 비교될 속성, 능력 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.

[152] - 동작(Action): 동작은 이벤트 및 조건 절이 충족될 때 흐름 기반 NSF의 측면을 제어하고 모니터링하는데 사용된다. NSF는 다양한 액션을 실행하여 보안 능력을 제공한다. I2NSF 작업의 예에는 침입 탐지 및 / 또는 보호, 웹 및 플로우 필터링, 패킷 및 플로우에 대한 심층 패킷 검사 제공이 포함될 수 있다.

[153]

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

[155] Boolean 절은 TRUE 또는 FALSE로 평가되는 논리문을 의미하며, 하나 이상의

용어로 구성 될 수 있습니다.

[156] 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.

[157] [표1]

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

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

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

[160] **외부 정보 모델과의 관계**

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

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

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

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

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

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

[167] 5) 제어 및 모니터링을 시작할 수 있습니다.

[168] 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).

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

것이다.

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

[171] 능력은 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).

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

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

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

[175] 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.

[176] 대신 사용자는 NSF가 가지고 있는 패킷 필터링이나 딥 패킷 검사와 같은 능력 세트만을 고려한다.

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

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

[179] 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.

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

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

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

[183] 이하 정보 모델의 각 부분에 대해서 살펴보도록 한다.

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

[185] 능력은 일반적으로 호출할 수 있는 NSF 함수를 나타내는 데 사용된다. 능력은

객체이므로 I2NSF ECA 정책 규칙의 이벤트, 조건 및/또는 액션을 설명하는 절에서 사용할 수 있다.

[186] I2NSF 능력 정보 모델은 사전 정의된 메타 데이터 모델을 구체화한다. I2NSF 능력의 적용은 능력 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.

[187] 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.

[188] 초기 NSFs 능력 카테고리(Initial NSFs Capability Categories)

[189] 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.

[190] **네트워크 보안 능력(Network Security Capabilities)**

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

[192] 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 깊이, 유지할 수 있는 다양한 흐름 및 컨텍스트 상태, 패킷 또는 흐름에 적용 할 수 있는 동작이 구현에 따라 달라질 수 있다.

[193] **콘텐츠 보안 능력(Content Security Capabilities)**

[194] 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 식별 할 수 있다.

[195] 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.

[196] 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.

[197] **공격 완화 능력(Attack Mitigation Capabilities)**

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

[199] - DDoS 공격:

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

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

[202] - 단일 패킷 공격:

[203] 스캐닝 및 스니핑 공격: IP 스윕, 포트 스캐닝 등

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

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

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

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

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

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

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

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

[212] 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.

[213] 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.

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

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

[216] 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.

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

[218] 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.

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

[220] 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.

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

[222] 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.

[223] 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.

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

[225] 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.

[226] 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.

[227] 4. 실행은 다음 세 가지 형식 중 하나를 취합니다.

[228] a. 하나 이상의 행동이 선택되면, NSF는 해결 전략에 의해 정의 된 행동을 수행할 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).

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

[230] 보안 액션 객체를 사용하여 아래에서 설명하는 실행의 전부 또는 일부를 수행할 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.

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

[232] c. 그렇지 않으면 트래픽이 거부됩니다.

[233]

[234] 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).

[235] 이후, 실행이 종료될 수 있다.

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

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

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

[239] 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.

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

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

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

[243] 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.

[244] 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.

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

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

[247] 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.

[248] 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.

[249] 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.

[250] **I2NSF** 능력을 위한 정보 모델(Information Model for I2NSF Capabilities)

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

[252] 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.

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

[254] 컨텐츠 보안 능력을 위한 정보 모델(Information Model for Content Security Capabilities)

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

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

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

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

[259] 공격완화 능력을 위한 정보 모델(Information Model for Attack Mitigation Capabilities)

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

[261] 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.

[262] **I2NSF** 보안 정책 규칙(I2NSF Security Policy Rule)

[263] 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.

[264] **Event Clause**

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

[266] 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.

[267] **Condition Clause**

[268] 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.

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

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

[271] **Action Clause**

[272] 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.

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

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

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

[276] - 능력 대수의 고려.

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

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

[279] 도 12a 및 도 12b는 본 발명의 일 실시 예에 따른 네트워크 보안 정책 식별을 위한 데이터 모델 구조를 예시한다.

[280] 네트워크 보안 정책을 식별하기 위한 데이터 모델은 도 12a 및 도 12b에 도시된 바와 같은 구조로 구성될 수 있다

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

[282] 보안 정책필드는 정책 이름, 정책 규칙들, 해결 전략, 및 고정 동작으로 구성될 수 있다.

[283] 정책 규칙은 규칙을 식별하기 위한 id, 규칙을 설명하기 위한 description, rev, priority, policy-event-clause-agg-ptr*, policy-condition-clause-agg-ptr*, policy-action-clause-agg-ptr* 및 time-zone으로 구성될 수 있다.

[284] 이때, time-zone은 적용되는 룰을 절대적인 시간 외에 주기적인 시간으로 설정할 수 있도록 absolute-time-zone 및 periodic-time-zone을 포함할 수 있다.

[285] 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.

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

[287] 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.

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

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

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

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

[292] 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.

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

[294] 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.

[295] 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.

[296] 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.

[297] 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.

[298] 도 15a 및 도 15b는 본 발명의 일 실시 예에 따른 액션 규칙을 위한 데이터 모델 구조를 예시한다.

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

[300] 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.

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

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

[303] 도 16a 내지 도 18j를 참조하면, 도 12a 내지 도 15b에서 설명한 데이터 모델을 이용하여 네트워크 보안 기능들의 정보 모델을 위한 YANG 데이터 모델을 설정할 수 있다.

[304] 도 16a 내지 18j에 도시된 모듈은 네트워크 보안 기능들을 위한 양 데이터 모듈로 정의될 수 있다.

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

[306] 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).

[307] 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.

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

[309] 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.

[310] - 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.

[311] - 보안 정책 위반의 결과로 NSF에 의해 촉발된 사건은 의심스러운 활동을

탐지하기 위해 SIEM에 의해 사용될 수 있다.

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

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

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

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

[316] **NSF 모니터링 데이터의 분류**

[317] 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.

[318] 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.

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

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

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

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

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

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

[325] - vendor_name: NSF 공급 업체의 이름.

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

[327] - Module_name: 메시지를 출력하는 모듈 이름

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

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

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

[331] **시스템 알람(System Alarm)**

[332] 메모리 알람(Memory Alarm)

[333] 다음 정보가 메모리 알람에 포함되어야 한다.

[334] - event_name: 'MEM_USAGE_ALARM'

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

[336] - usage: 사용 된 메모리 양을 지정함.

[337] - 임계 값: 경보를 트리거 하는 임계 값

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

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

[340]

[341] CPU 알람(CPU Alarm)

[342] 다음과 같은 정보가 CPU 알람에 포함될 수 있다.

[343] - event_name: 'CPU_USAGE_ALARM'

[344] - usage: 사용 된 CPU의 양을 지정합니다.

[345] - threshold: 이벤트를 트리거 하는 임계 값

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

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

[348]

[349] 디스크 알람(Disk Alarm)

[350] 다음과 같은 정보가 디스크 알람에 포함될 수 있다.

[351] - event_name: 'DISK_USAGE_ALARM'

[352] - usage: 사용 된 디스크 공간의 양을 지정합니다.

[353] - threshold: 이벤트를 트리거 하는 임계 값

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

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

[356]

[357] 하드웨어 알람(Hardware Alarm)

[358] 다음과 같은 정보가 하드웨어 알람에 포함될 수 있다.

[359] - event_name: 'HW_FAILURE_ALARM'

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

[361] - 임계 값: 경보를 트리거 하는 임계 값

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

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

[364]

[365] 인터페이스 알람(Interface Alarm)

[366] 다음과 같은 정보가 인터페이스 알람에 포함될 수 있다.

[367] - event_name: 'IFNET_STATE_ALARM'

[368] - interface_Name: 인터페이스 이름

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

[370] - threshold: 이벤트를 트리거 하는 임계 값

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

[372] - 메시지: '현재 인터페이스 상태'를 출력함.

[373]

[374] **시스템 이벤트(System Events)**

[375] 액세스 위반(Access Violation)

[376] 다음과 같은 정보가 이벤트에 포함될 수 있다.

[377] - event_name: 'ACCESS_DENIED'

[378] - user: 사용자 이름

[379] - group: 사용자가 속한 그룹

[380] - login_ip_address: 사용자의 로그인 IP 주소

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

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

[383]

[384] 구성 변경(Configuration Change)

[385] 다음과 같은 정보가 이벤트에 포함될 수 있다.

[386] - event_name: 'CONFIG_CHANGE'

[387] - user: 사용자 이름

[388] - group: 사용자가 속한 그룹

[389] - login_ip_address: 사용자의 로그인 IP 주소

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

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

[392] **시스템 로그(System Log)**

[393] 접속 로그(Access Logs)

[394] 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.

[395] - 관리자: 장치에서 작동하는 관리자

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

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

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

[399] - 결과: 명령 실행 결과

[400] - content: 로그인 후 관리자가 수행 한 작업.

[401] 자원 사용률 로그(Resource Utilization Logs)

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

[403] - system_status: 현재 시스템의 실행 상태

[404] - CPU_usage: CPU 사용량을 지정합니다.

[405] - memory_usage: 메모리 사용량을 지정합니다.

[406] - disk_usage: 디스크 사용량을 지정합니다.

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

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

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

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

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

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

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

[414] 사용자 활동 로그(User Activity Logs)

[415] 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.

[416] - group: 사용자가 속한 그룹

[417] - login_ip_address: 사용자의 로그인 IP 주소

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

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

[420] - online_duration: 온라인 기간

[421] - lockout_duration: 잠금 기간

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

[423] - 원인: 사용자 작업에 실패했습니다.

[424] **시스템 카운터(System Counter)**

[425] 인터페이스 카운터(Interface counters)

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

[427] - interface_name : NSF에서 구성된 네트워크 인터페이스 이름

[428] - in_total_traffic_pkts : 전체 인바운드 패킷

[429] - out_total_traffic_pkts : 총 아웃 바운드 패킷

[430] - in_total_traffic_bytes : 총 인바운드 바이트

[431] - out_total_traffic_bytes : 총 아웃 바운드 바이트

[432] - in_drop_traffic_pkts : 총 인바운드 드롭 패킷

[433] - out_drop_traffic_pkts : 총 아웃 바운드 드롭 패킷

[434] - in_drop_traffic_bytes : 총 인바운드 드롭 바이트

[435] - out_drop_traffic_bytes : 총 아웃 바운드 삭제 바이트

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

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

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

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

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

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

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

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

[444] NSF 이벤트(NSF Events)

[445] DDos 이벤트는 다음과 같은 정보를 포함할 수 있다.

[446] - event_name : 'SEC_EVENT_DDoS'

[447] - 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

[448] - dst_ip: 공격 받고있는 victum의 IP 주소

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

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

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

[452] - attack_rate: 공격 트래픽의 PPS

[453] - attack_speed: 공격 트래픽의 bps

[454] - rule_id: 트리거되는 규칙의 ID입니다.

[455] - rule_name: 트리거되는 규칙의 이름

[456] - 프로필: 트래픽이 일치하는 보안 프로필입니다.

[457] 세션 테이블 이벤트(session Table Event)

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

[459] - event_name: 'SESSION_USAGE_HIGH'

[460] - current: 동시 세션 수

[461] - max: 세션 테이블이 지원할 수있는 최대 세션 수

[462] - threshold: 이벤트를 트리거하는 임계 값

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

[464]

[465] 바이러스 이벤트(Virus Event)

[466] 아래와 같은 정보가 바이러스 이벤트에 포함될 수 있다.

[467] - event_Name : 'SEC_EVENT_VIRUS'

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

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

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

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

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

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

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

[475] - file_type : 바이러스가 숨겨진 파일의 유형

[476] - file_name : 바이러스가 숨겨진 파일의 이름

[477] - virus_info : 바이러스의 간단한 소개

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

[479] - rule_id : 트리거되는 규칙의 ID입니다.

[480] - rule_name : 트리거되는 규칙의 이름

[481] - 프로필 : 트래픽이 일치하는 보안 프로필입니다.

[482]

[483] 침입 이벤트(Intrusion Event)

[484] - Intrustion Event에는 다음 정보가 포함되어야합니다.

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

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

[487] - src_ip: 패킷의 소스 IP 주소

[488] - dst_ip: 패킷의 목적지 IP 주소

[489] - src_port: 패킷의 소스 포트 번호

[490] - dst_port : 패킷의 목적지 포트 번호

[491] - src_zone: 패킷의 소스 보안 영역

[492] - dst_zone: 패킷의 대상 보안 영역

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

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

[495] - rule_id: 트리거 되는 규칙의 ID입니다.

[496] - rule_name: 트리거 되는 규칙의 이름

[497] - 프로필: 트래픽이 일치하는 보안 프로필

[498] - intrusion_info: 침입에 대한 간단한 설명

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

[500] 봇넷 이벤트(Botnet Event)

[501] 아래와 같은 정보는 봇넷 이벤트에 포함될 수 있다.

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

[503] - botnet_name : 탐지 된 봇넷의 이름

[504] - src_ip : 패킷의 소스 IP 주소

[505] - dst_ip : 패킷의 목적지 IP 주소

[506] - src_port : 패킷의 소스 포트 번호

[507] - dst_port : 패킷의 목적지 포트 번호

[508] - src_zone : 패킷의 소스 보안 영역

[509] - dst_zone : 패킷의 대상 보안 영역

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

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

[512] - 역할 : 봇넷 내 통신 당사자의 역할 :

[513] 1. 좀비 호스트에서 공격자까지의 패킷

[514] 2. 공격자에서 좀비 호스트로 가는 패킷

[515] 3. IRC / WEB 서버에서 좀비 호스트로 가는 패킷

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

[517] 5. 공격자에서 IRC / WEB 서버로 보낸 패킷

[518] 6. IRC / WEB 서버에서 공격자로 가는 패킷

[519] 7. 좀비 호스트에서 희생자까지의 패킷

[520] - botnet_info : Botnet에 대한 간단한 설명

[521] - rule_id : 트리거 되는 규칙의 ID입니다.

[522] - rule_name : 트리거 되는 규칙의 이름

[523] - 프로필: 트래픽이 일치하는 보안 프로필

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

[525] 웹 공격 이벤트(Web Attack Event)

[526] 아래와 같은 정보가 웹 공격 이벤트에 포함될 수 있다.

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

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

[529] - src_ip : 패킷의 소스 IP 주소

[530] - dst_ip : 패킷의 목적지 IP 주소

[531] - src_port : 패킷의 소스 포트 번호

[532] - dst_port : 패킷의 목적지 포트 번호

[533] - src_zone : 패킷의 소스 보안 영역

[534] - dst_zone : 패킷의 대상 보안 영역

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

[536] - req_url : 요청 된 URL

[537] - url_category : 일치하는 URL 카테고리

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

[539] - rule_id : 트리거되는 규칙의 ID입니다.

[540] - rule_name : 트리거되는 규칙의 이름

[541] - 프로필 : 트래픽이 일치하는 보안 프로필입니다.

[542]

[543] **NSF 로그(NSF Logs)**

[544] DDoS 로그(DDoS Logs)

[545] DDoS 경보의 필드 외에도 필드 외에도 아래와 같은 정보가 DDoS 로그에 포함될 수 있다.

[546] - 공격 유형 : DDoS

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

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

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

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

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

[552]

[553] 바이러스 로그(Virus Logs)

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

[555] - 공격 유형 : 바이러스

[556] - 프로토콜 : 전송 계층 프로토콜

[557] - app : 응용 프로그램 계층 프로토콜의 이름

[558] - times : 바이러스 탐지 시간

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

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

[561]

[562] 침입 로그(Intrusion Logs)

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

[564] - 공격 유형 : 침입

[565] - 시간 : 기록된 시간에 침입 시간이 발생했습니다.

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

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

[568] - attack_rate : 공격 트래픽의 pps NUM

[569] - attack_speed : NUM 공격 트래픽의 bps

[570] 봇넷 로그(Botnet Logs)

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

[572] - attack_type : 봇넷

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

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

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

[576] DPI 로그(DPI Logs)

[577] 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.

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

[579] - file_name : 파일 이름

[580] - file_type : 파일 형식

[581] - src_zone : 트래픽 소스 보안 영역

[582] - dst_zone : 트래픽의 대상 보안 영역

[583] - src_region : 트래픽 소스 영역

[584] - dst_region : 트래픽의 대상 영역

[585] - src_ip : 트래픽 소스 IP 주소

[586] - src_user : 트래픽을 생성 한 사용자

[587] - dst_ip : 트래픽의 대상 IP 주소

[588] - src_port : 트래픽 소스 포트

[589] - dst_port : 트래픽의 대상 포트

[590] - 프로토콜 : 트래픽의 프로토콜 유형

[591] - 앱 : 트래픽의 애플리케이션 유형

[592] - policy_id : 트래픽이 일치하는 보안 정책 ID

[593] - policy_name : 트래픽이 일치하는 보안 정책 이름

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

[595]

[596] Vulnerabillity 검색 로그

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

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

[599] - 취약점 ID : 취약점 ID

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

[601] - 운영 체제 : 대상 호스트의 운영 체제

[602] - 서비스 : 피해자 호스트에 취약성이 있는 서비스

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

[604] - port : 포트 번호

[605] - vulnerability_info : 취약점에 대한 정보

[606] - fix_suggestion : 취약점에 대한 수정 제안.

[607] - 8.6.7. 웹 공격 로그

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

[609] - attack_type : 웹 공격

[610] - rsp_code : 응답 코드

[611] - req_clientapp : 클라이언트 응용 프로그램

[612] - req_cookies : 쿠키

[613] - req_host : 요청한 호스트의 도메인 이름

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

[615]

[616] **NSF 카운터(NSF Counter)**

[617] 방화벽 카운터(Firewall Counters)

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

[619] - src_zone : 트래픽 소스 보안 영역

[620] - dst_zone : 트래픽의 대상 보안 영역

[621] - src_region : 트래픽 소스 영역

[622] - dst_region : 트래픽의 대상 영역

[623] - src_ip : 트래픽 소스 IP 주소

[624] - src_user : 트래픽을 생성 한 사용자

[625] - dst_ip : 트래픽의 대상 IP 주소

[626] - src_port : 트래픽 소스 포트

[627] - dst_port : 트래픽의 대상 포트

[628] - 프로토콜 : 트래픽의 프로토콜 유형

[629] - 앱 : 트래픽의 애플리케이션 유형

[630] - policy_id : 트래픽이 일치하는 보안 정책 ID

[631] - policy_name : 트래픽이 일치하는 보안 정책 이름

[632] - in_interface : 트래픽의 인바운드 인터페이스

[633] - out_interface : 트래픽의 아웃 바운드 인터페이스

[634] - total_traffic : 총 트래픽 양

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

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

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

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

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

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

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

[642]

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

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

[645] - src_zone : 트래픽 소스 보안 영역

[646] - dst_zone : 트래픽의 대상 보안 영역

[647] - src_region : 트래픽 소스 영역

[648] - dst_region : 트래픽의 대상 영역

[649] - src_ip : 트래픽 소스 IP 주소

[650] - src_user : 트래픽을 생성 한 사용자

[651] - dst_ip : 트래픽의 대상 IP 주소

[652] - src_port : 트래픽 소스 포트

[653] - dst_port : 트래픽의 대상 포트

[654] - 프로토콜 : 트래픽의 프로토콜 유형

[655] - 앱 : 트래픽의 애플리케이션 유형

[656] - policy_id : 트래픽이 일치하는 보안 정책 ID

[657] - policy_name : 트래픽이 일치하는 보안 정책 이름

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

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

[660] 도 19a 내지 도 19j를 참조하면, 앞에서 살펴본 NSF를 모니터링하기 위한 정보 모델을 이용하여 데이터 모델을 설계할 수 있다

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

[662] 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.

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

[664] 도 20a 내지 도 21i를 참조하면 앞에서 살펴본 NSF 모니터링을 위한 정보 모델 및 데이터 모델을 이용하여 해당 YANG 데이터 모델을 설계할 수 있다.

[665] 도 1 내지 도 21i에서 설명한 정보 모델, 데이터 모델 및 YANG 데이터 모델은 선택적으로 조합되어 사용될 수 있다.

[666] 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.

[667] 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.

[668] 펌웨어나 소프트웨어에 의한 구현의 경우, 본 발명의 일 실시 예는 이상에서 설명된 능력 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.

[669] 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

[670] 본 발명은 다양한 보안 관리 시스템에 적용될 수 있다.

청구범위

[청구항 1] 보안 관리 시스템에서 네트워크 운영 관리 시스템이 보안 서비스를 제공하기 위한 방법에 있어서,
I2NSF(Interface to Network Security Function) 사용자로부터 상위 레벨(High-Level)의 제 1 보안 정책을 수신하는 단계;
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 포함하되,
상기 네트워크 운영 관리 시스템과 상기 복수의 NSF 각각은 I2NSF NSF-직면 인터페이스로 연결되어 있는 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 중 적어도 하나를 더 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 포트 번호와 관련된 룰의 설정을 위한 포트 번호 값을 포함하는 방법.

[청구항 8] 제 3 항에 있어서,
상기 동작 절은 입력 동작(ingress action), 전송 동작(egress action) 및 적용 프로파일 동작(apply 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 절(Condition Clause), 및 상기 동작 절(Action Clause)은 다중 조건의 적용을 위한 컨테이너 구조로 구성되는 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 Function) 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 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 연결되어 있는 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 중 적어도 하나를 더 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.

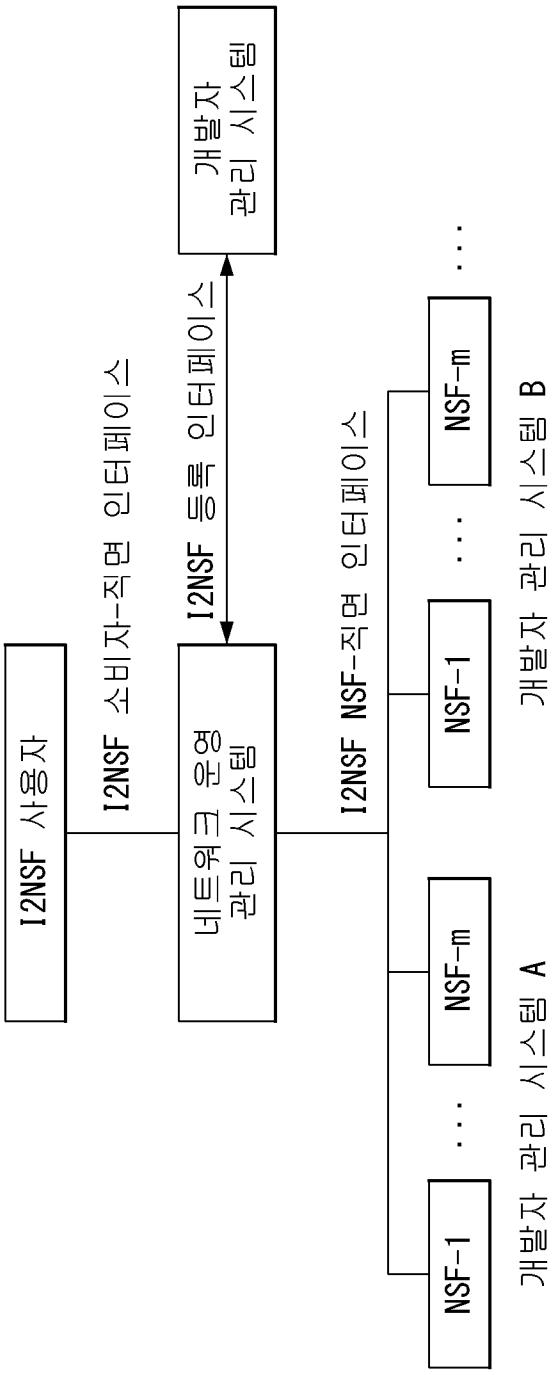
[청구항 17] 제 13 항에 있어서,
상기 조건 절은 포트 번호와 관련된 룰의 설정을 위한 포트 번호 값을 포함하는 보안 관리 시스템.

[청구항 18] 제 13 항에 있어서,
상기 동작 절은 입력 동작(ingress action), 전송 동작(egress action) 및 적용 프로파일 동작(apply profile action)을 포함하는 보안 관리 시스템.

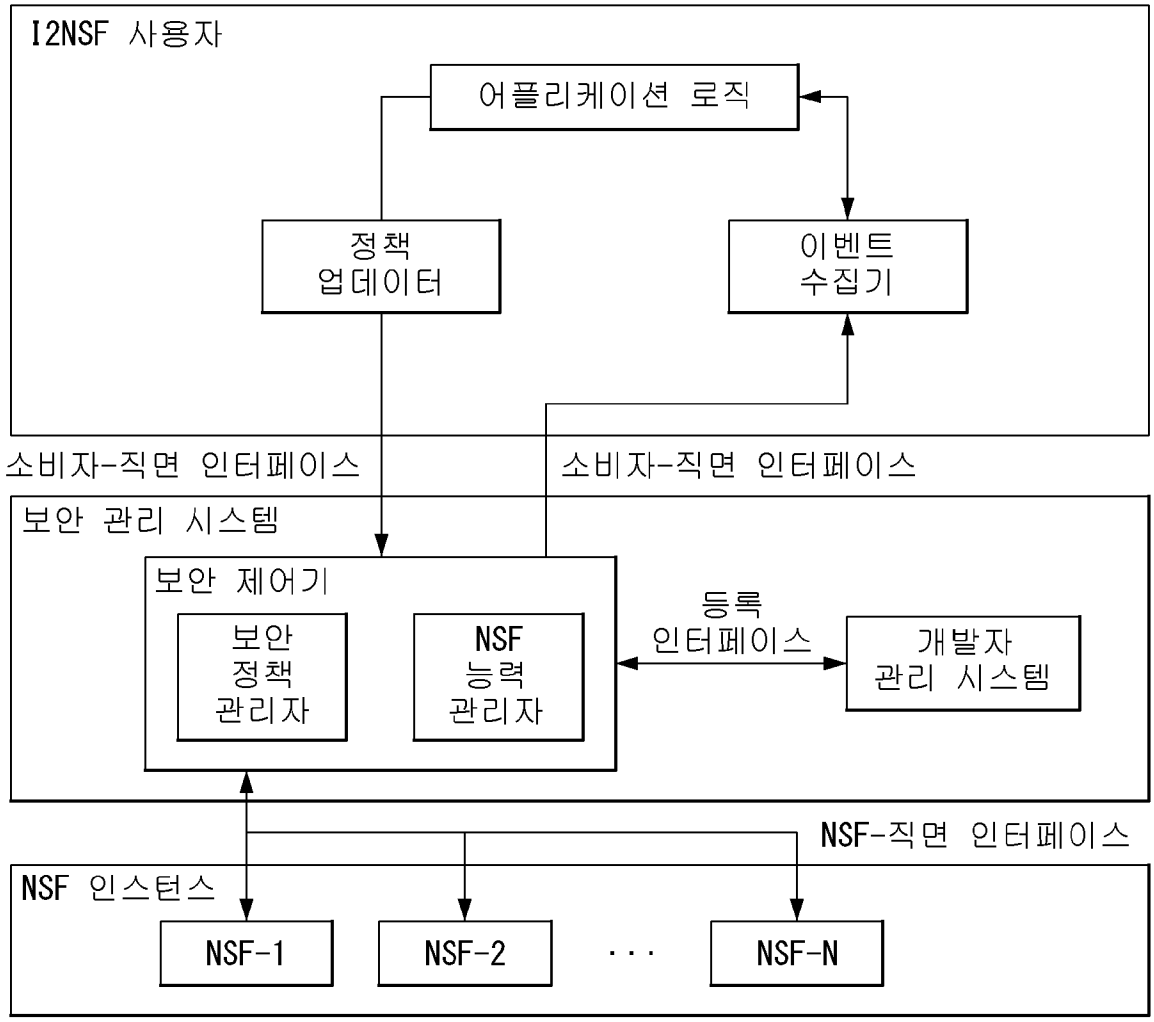
[청구항 19] 제 18 항에 있어서,
상기 프로파일 동작은 응용 프로그램 계층에 적용되는 보안 기능을 나타내는 컨텐츠 보안 제어, 및 네트워크 공격의 탐지 및 완화를 위한 공격 완화 제어를 포함하는 보안 관리 시스템.

[청구항 20] 제 13 항에 있어서,
상기 조건 절(Condition Clause), 및 상기 동작 절(Action Clause)은 다중 조건의 적용을 위한 컨테이너 구조로 구성되는 보안 관리 시스템.

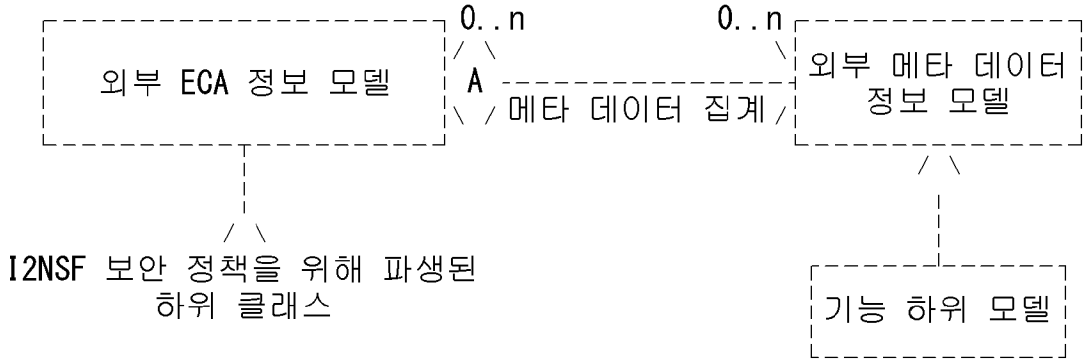
[도1]


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

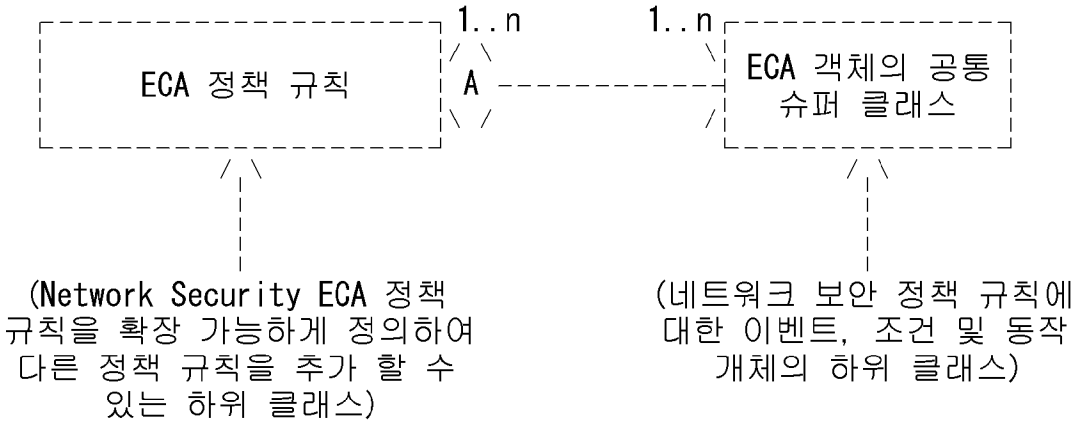
[도2]


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

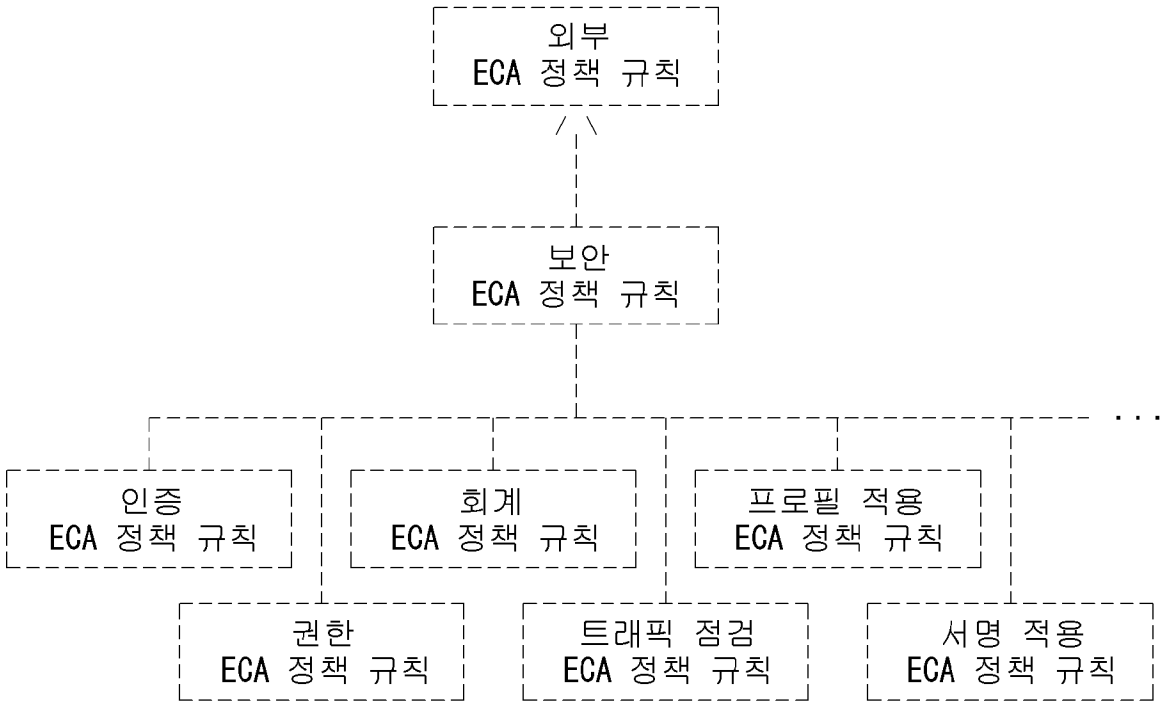
[도3]


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

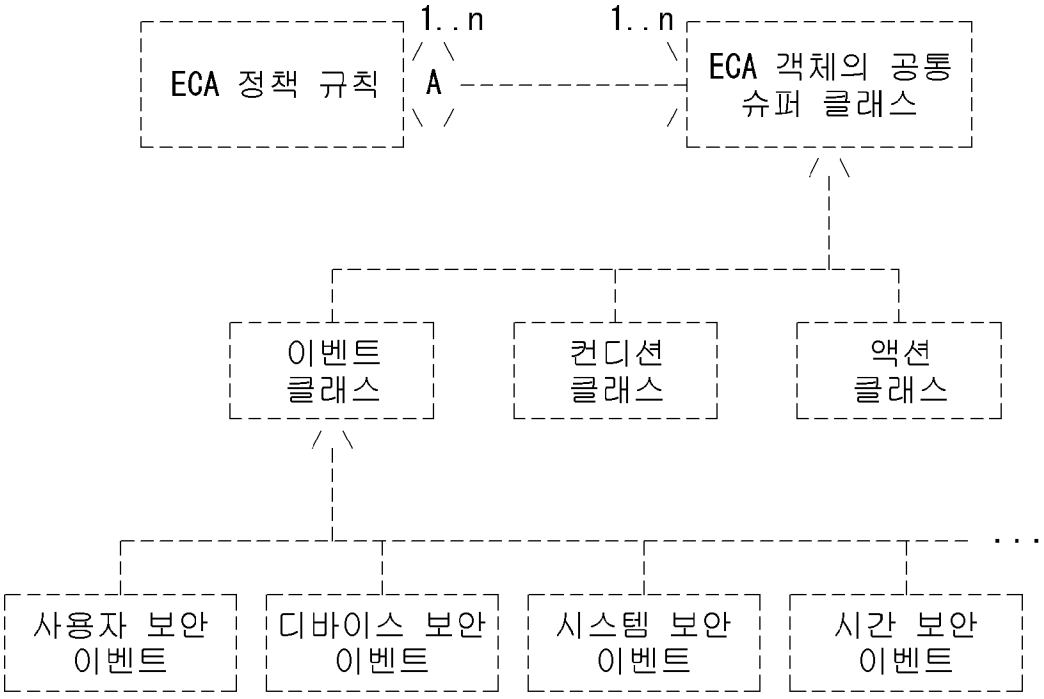
[도4]


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)

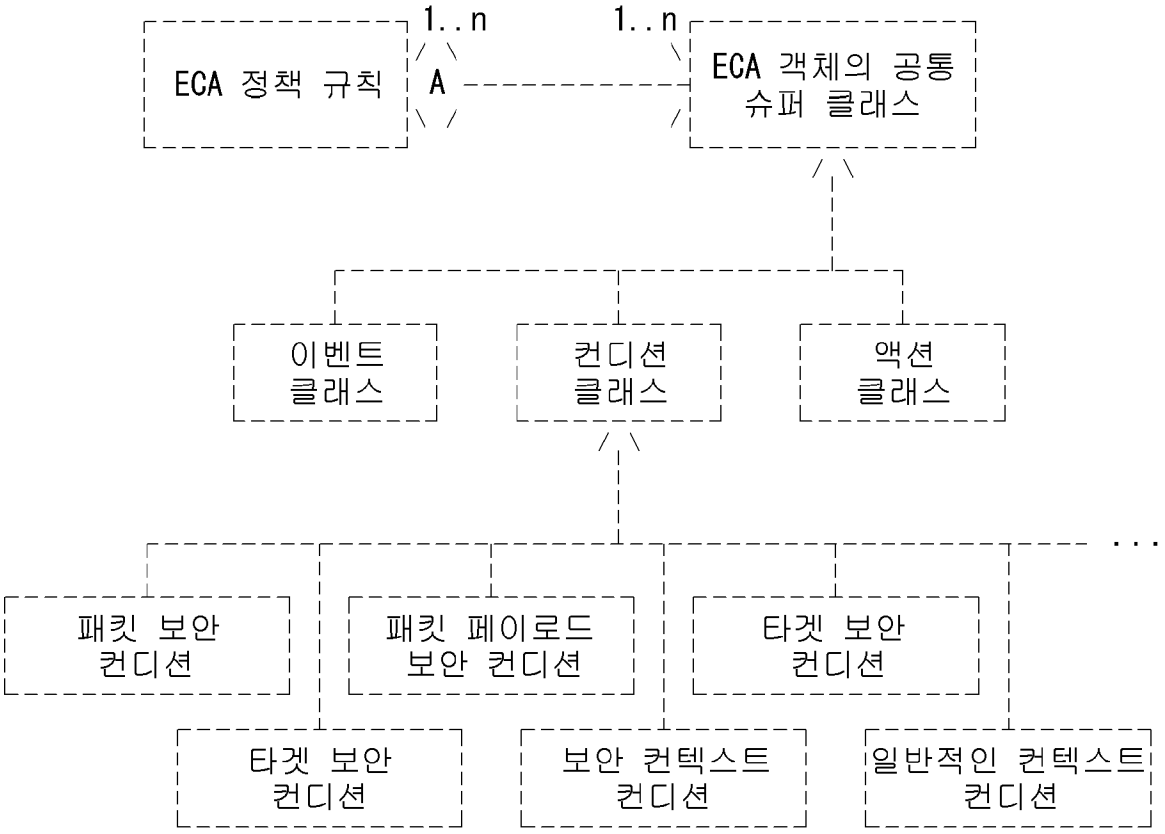
[도5]


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

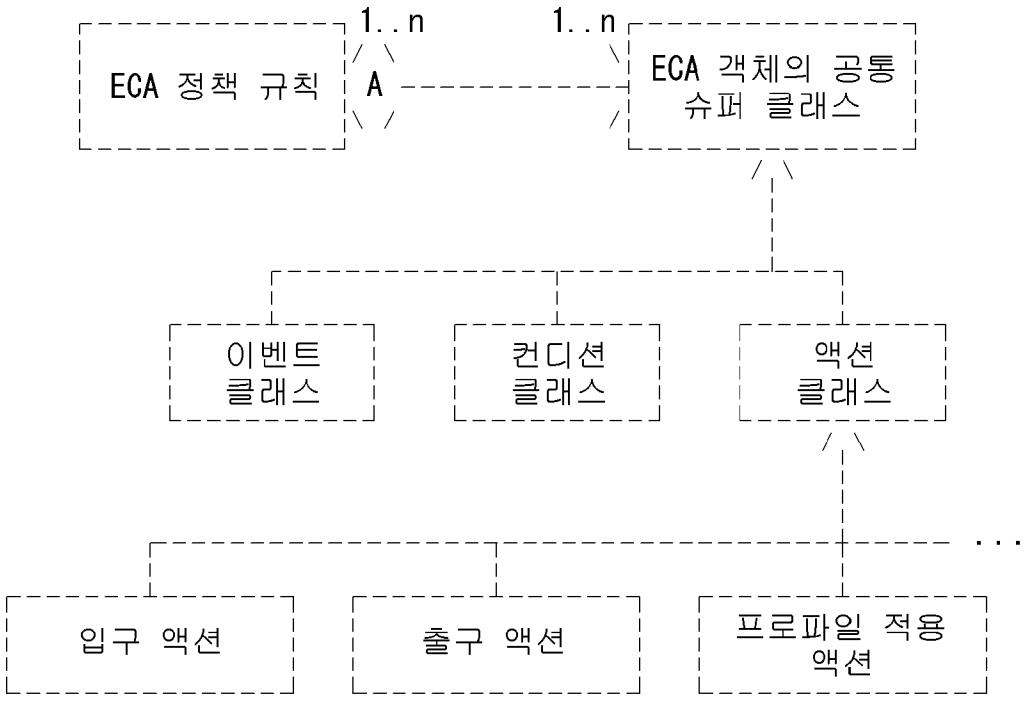
[도6]


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

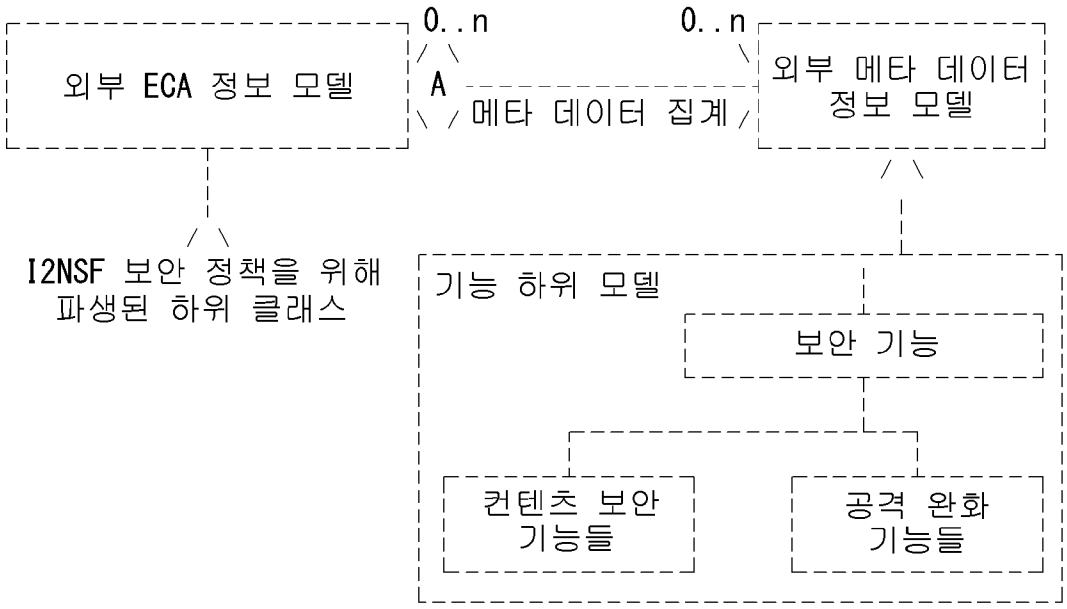
[도7]


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

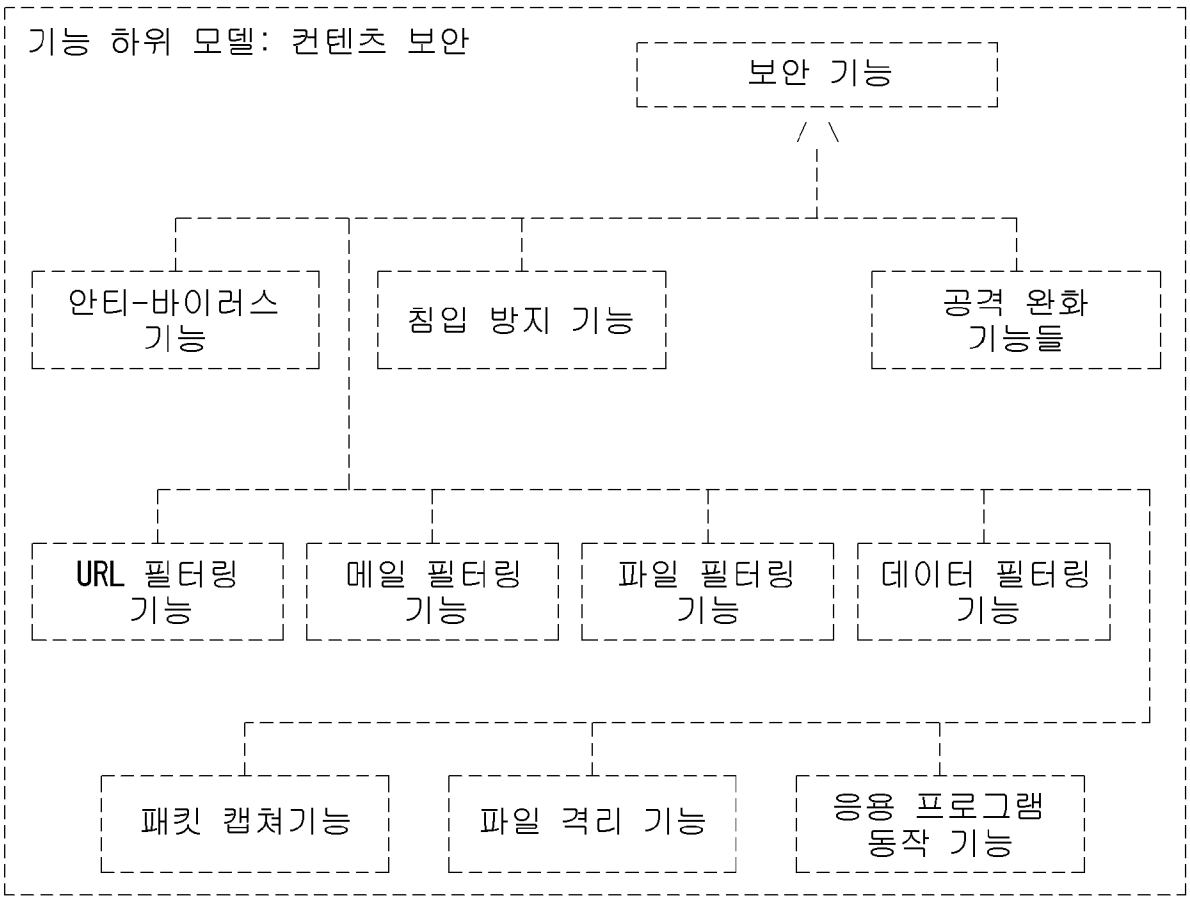
[도8]


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
...

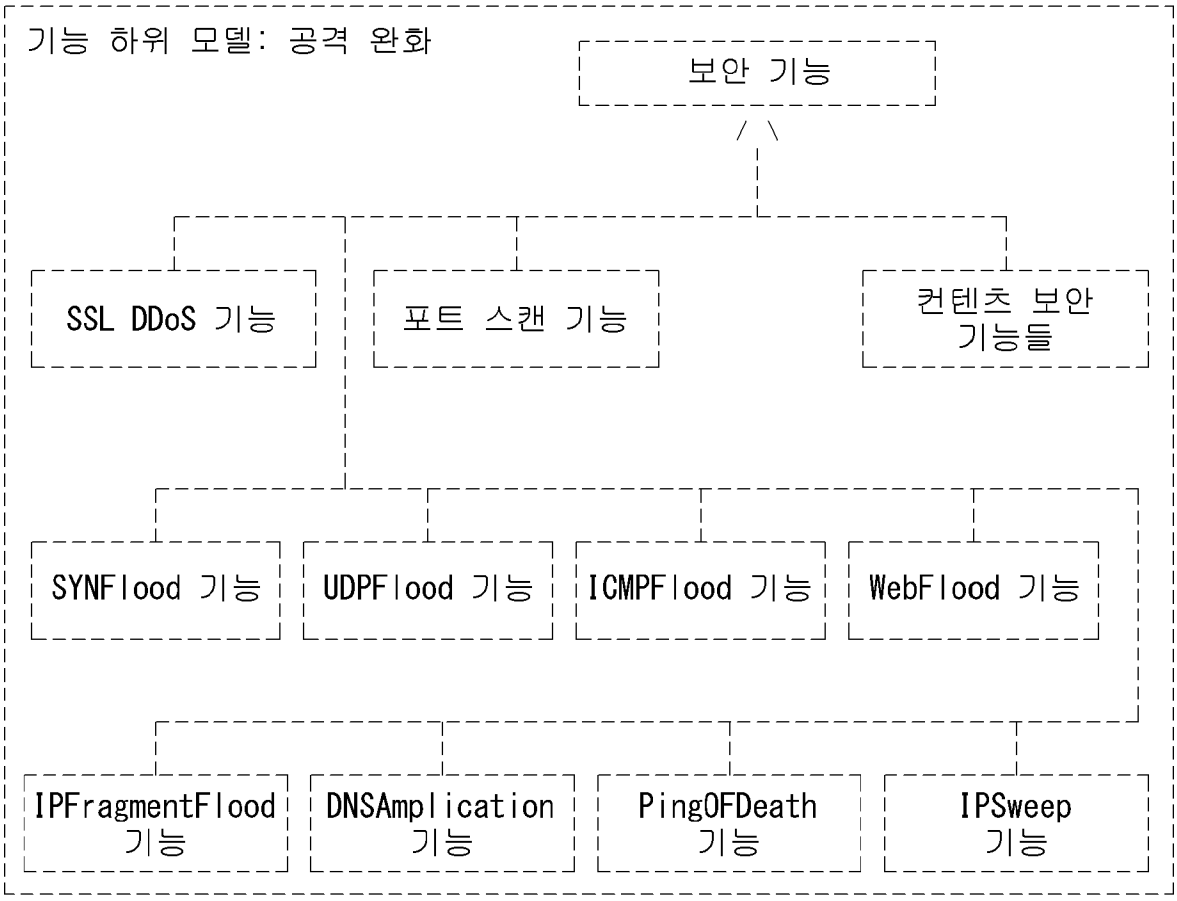
[도9]


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

[도10]


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

[도11]


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

[도12a]

```
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
```

[도12b]

```
|  |  ---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
   ...
```

[도13]

```
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
   ...
```

[도14a]

```
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
```

[도14b]

```
|  ...
+--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
```

[도14c]

```
|        |     +--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
```

[도14d]

```
|       |  +--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
   ...
```

[도15a]

```
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
```

[도15b]

```
|       |      +--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
```

[도16a]

```
<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;
```

[도16b]

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

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

[도16c]

```
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
```

[도16d]

```
            (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.";
}
```

[도16e]

```
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";
```

[도16f]

```
        }
      }
      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.";
}
```

[도16g]

```
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";
}
```

[도16h]

```
 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";
  }
```

[도16i]

```
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;
```

[도16j]

```
   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).";
 }
```

[도16k]

```
      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
```

[도16l]

```
"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.";
  }
```

[도16m]

```
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;
```

[도16n]

```
    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";
}
```

[도16o]

```
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";
```

[도16p]

```
    }
  }
}
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";
    }
```

[도16q]

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

[도16r]

```
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";
}
```

[도16s]

```
}
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";
}
```

[도16t]

```
        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";
        }
      }
    }
  }
}
```

[도16u]

```
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";
      }
    }

  }
```

[도16v]

```
  }

  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.";
    }
  }
}
```

[도16w]

```
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 {
```

[도16x]

```
    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";
    }
```

[도16y]

```
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
```

[도16z]

```
      "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).";
  }
}
```

[도17a]

```
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 {
```

[도17b]

```
  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.";
}
```

[도17c]

```
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).";
}
```

[도17d]

```
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.";
```

[도17e]

```
  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.";
```

[도17f]

```
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.";
}
```

[도17g]

```
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.";
}
```

[도17h]

```
  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).";
  }
}
```

[도17i]

```
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.";
  }
```

[도17j]

```
  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).";
  }
}
```

[도17k]

```
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).";
  }
```

[도17l]

```
    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.";
    }
  }
}
```

[도17m]

```
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
```

[도17n]

```
        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.";
```

[도17o]

```
  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 {
```

[도17p]

```
      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";
  }
```

[도17q]

```
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;
```

[도17r]

```
        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.";
      }
    }
  }
}
```

[도17s]

```
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.";
      }
    }
```

[도17t]

```
      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";
  }
}
```

[도17u]

```
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
```

[도17v]

```
            "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.";
```

[도17w]

```
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.";
```

[도17x]

```
  }
}
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";
  }
```

[도17y]

```
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.";
```

[도17z]

```
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.";
}
```

[도18a]

```
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.";
}
```

[도18b]

```
    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.";
    }
  }
}
```

[도18c]

```
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";
```

[도18d]

```
    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.";
    }
  }
}
```

[도18e]

```
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.";
```

[도18f]

```
        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.";
        }
      }
    }
  }
}
```

[도18g]

```
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.";
```

[도18h]

```
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.";
    }
  }
}
```

[도18i]

```
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.";
    }
  }
```

[도18j]

```
                    }

                    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.";
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

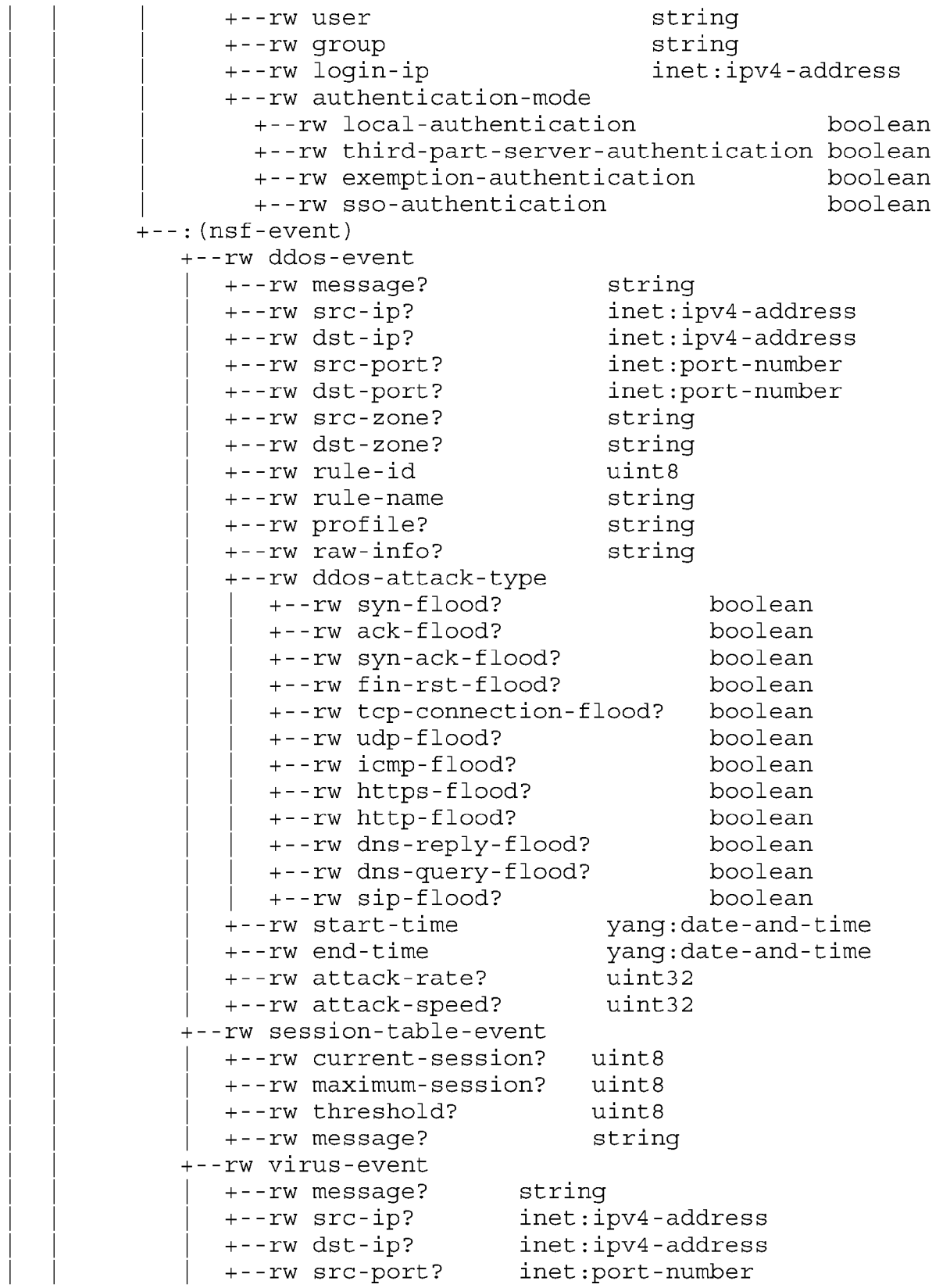
[도19a]

```
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)
```

[도19b]

```
|   |   +--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
```

[도19c]


+--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

[도19d]

```
|   |         |  +--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
```

[도19e]

```
|   |         |  +--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
```

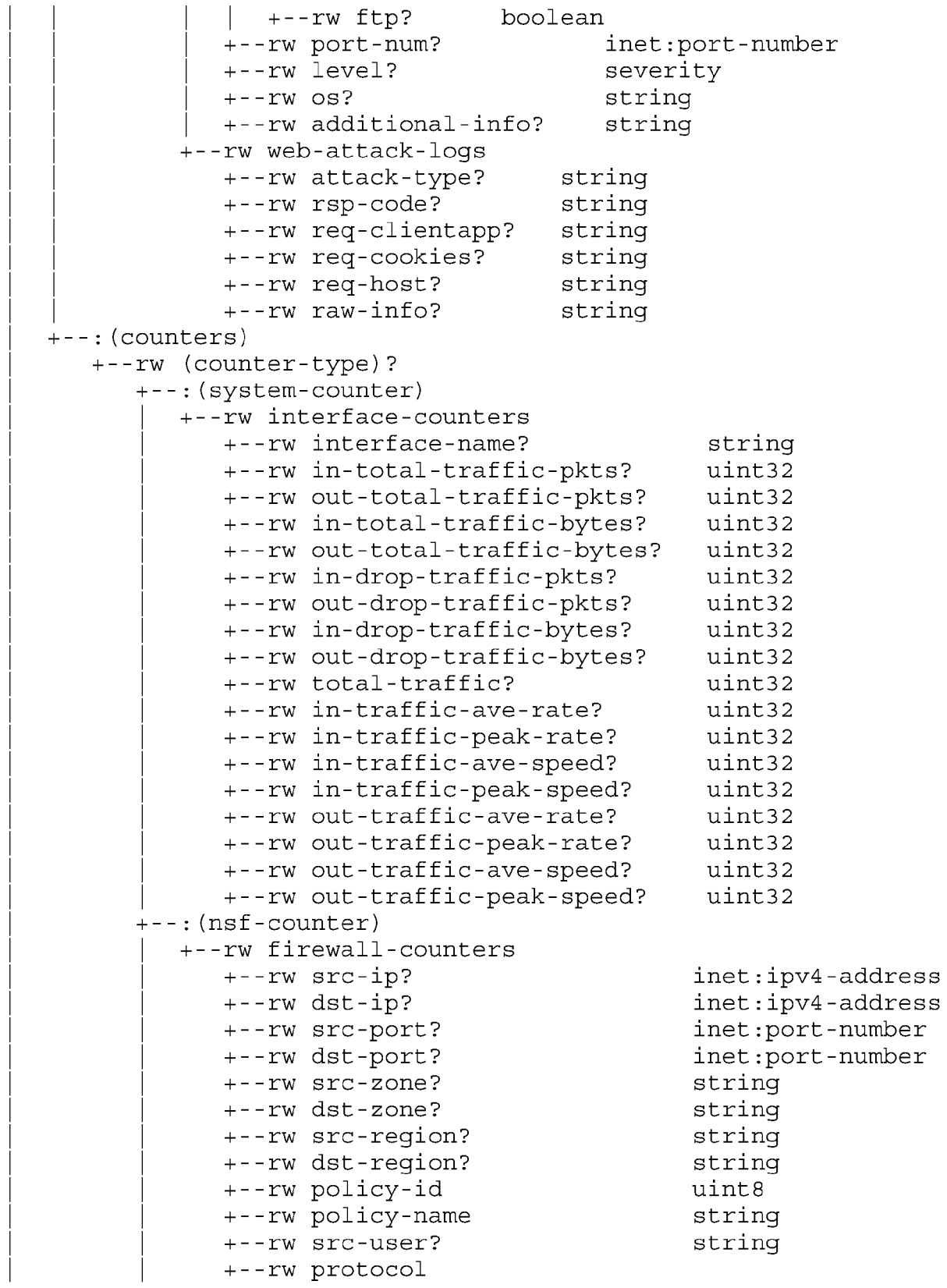
[도19f]

```
|   |      |   |  +--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
```

[도19g]

```
|   |         |  +--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
```

[도19h]


+--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

[도19i]

```
|          |       |  +--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
```

[도19j]

```
|              +--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
```

[도20a]

```
<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.";
```

[도20b]

```
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";
```

[도20c]

```
}
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
```

[도20d]

```
          "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";
  }
```

[도20e]

```
  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 {
```

[도20f]

```
    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.";
    }
```

[도20g]

```
    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;
```

[도20h]

```
    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";
```

[도20i]

```
      }
    }
  }
}
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
```

[도20j]

```
        "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";
    }
```

[도20k]

```
    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;
```

[도20l]

```
    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";
  }
```

[도20m]

```
    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
```

[도20n]

```
      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";
```

[도20o]

```
        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 {
```

[도20p]

```
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
```

[도20q]

```
    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
```

[도20r]

```
        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
```

[도20s]

```
        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
```

[도20t]

```
        "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
```

[도20u]

```
      "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";
    }
```

[도20v]

```
  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;
```

[도20w]

```
            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;
```

[도20x]

```
    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
```

[도20y]

```
        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.";
```

[도20z]

```
}
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{
```

[도21a]

```
      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
```

[도21b]

```
        "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";
```

[도21c]

```
    }
    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";
```

[도21d]

```
  }
  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;
```

[도21e]

```
    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 {
```

[도21f]

```
    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";
  }
```

[도21g]

```
        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
```

[도21h]

```
                "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;
```

[도21i]

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

INTERNATIONAL SEARCH REPORT

International application No.
PCT/KR2018/002957

A. CLASSIFICATION OF SUBJECT MATTER

H04L 29/06(2006.01)i, H04L 12/24(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L 29/06; H04L 29/08; G06F 17/30; H04M 7/00; H04L 12/24

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean Utility models and applications for Utility models: IPC as above
Japanese Utility models and applications for Utility models: IPC as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS (KIPO internal) & Keywords: security, management, service, high-level, low-level, policy, network, function, and similar terms.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2016-0352780 A1 (LANG, Ulrich et al.) 01 December 2016 See paragraph [0003].	1-20
A	US 9521115 B1 (VARMOUR NETWORKS, INC.) 13 December 2016 See column 3, line 22-column 8, line 49; and figures 1-5.	1-20
A	US 2009-0077086 A1 (MALKIN, Peter Kenneth et al.) 19 March 2009 See paragraphs [0024]-[0062]; and figures 1-7.	1-20
A	US 2004-0042609 A1 (DELANEY, Robert J. et al.) 04 March 2004 See paragraphs [0029]-[0059]; and figures 2-7.	1-20
A	US 2015-0207813 A1 (VORSTACK, INC.) 23 July 2015 See paragraphs [0036]-[0048]; and figures 1-4B.	1-20

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
19 JUNE 2018 (19.06.2018)	**03 JULY 2018 (03.07.2018)**

Name and mailing address of the ISA/KR	Authorized officer
Korean Intellectual Property Office Government Complex-Daejeon, 189 Seonsa-ro, Daejeon 302-701, Republic of Korea Facsimile No. +82-42-481-8578	Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/KR2018/002957

Patent document cited in search report	Publication date	Patent family member	Publication date
US 2016-0352780 A1	01/12/2016	US 2009-0077621 A1	19/03/2009
		US 2015-0237073 A1	20/08/2015
		US 2017-0289198 A1	05/10/2017
		US 9043861 B2	26/05/2015
		US 9420006 B2	16/08/2016
		US 9692792 B2	27/06/2017
		WO 2009-036896 A2	26/03/2009
		WO 2009-036896 A3	28/05/2009
		WO 2009-036896 A4	16/07/2009
US 9521115 B1	13/12/2016	US 2017-279770 A1	28/09/2017
US 2009-0077086 A1	19/03/2009	US 8024771 B2	20/09/2011
US 2004-0042609 A1	04/03/2004	AU 2003-268466 A1	29/03/2004
		US 7043000 B2	09/05/2006
		WO 2004-023775 A1	18/03/2004
US 2015-0207813 A1	23/07/2015	US 2015-0222656 A1	06/08/2015
		US 2016-0164890 A1	09/06/2016
		US 2016-0269427 A1	15/09/2016
		US 2017-0279824 A1	28/09/2017
		US 2017-0316203 A1	02/11/2017
		US 8914406 B1	16/12/2014
		US 9038183 B1	19/05/2015
		US 9137258 B2	15/09/2015
		US 9167001 B1	20/10/2015
		US 9680846 B2	13/06/2017
		US 9710644 B2	18/07/2017
		US 9756082 B1	05/09/2017

국 제 조 사 보 고 서

국제출원번호
PCT/KR2018/002957

A. 발명이 속하는 기술분류(국제특허분류(IPC))

H04L 29/06(2006.01)i, H04L 12/24(2006.01)i

B. 조사된 분야

조사된 최소문헌(국제특허분류를 기재)
H04L 29/06; H04L 29/08; G06F 17/30; H04M 7/00; H04L 12/24

조사된 기술분야에 속하는 최소문헌 이외의 문헌
한국등록실용신안공보 및 한국공개실용신안공보: 조사된 최소문헌란에 기재된 IPC
일본등록실용신안공보 및 일본공개실용신안공보: 조사된 최소문헌란에 기재된 IPC

국제조사에 이용된 전산 데이터베이스(데이터베이스의 명칭 및 검색어(해당하는 경우))
eKOMPASS(특허청 내부 검색시스템) & 키워드: security, management, service, high-level, low-level, policy, network, function, 및 유사 용어.

C. 관련 문헌

카테고리*	인용문헌명 및 관련 구절(해당하는 경우)의 기재	관련 청구항
A	US 2016-0352780 A1 (ULRICH LANG 등) 2016.12.01 단락 [0003] 참조.	1-20
A	US 9521115 B1 (VARMOUR NETWORKS, INC.) 2016.12.13 컬럼 3, 라인 22 - 컬럼 8, 라인 49; 및 도면 1-5 참조.	1-20
A	US 2009-0077086 A1 (PETER KENNETH MALKIN 등) 2009.03.19 단락 [0024]-[0062]; 및 도면 1-7 참조.	1-20
A	US 2004-0042609 A1 (ROBERT J. DELANEY 등) 2004.03.04 단락 [0029]-[0059]; 및 도면 2-7 참조.	1-20
A	US 2015-0207813 A1 (VORSTACK, INC.) 2015.07.23 단락 [0036]-[0048]; 및 도면 1-4B 참조.	1-20

☐ 추가 문헌이 C(계속)에 기재되어 있습니다.

☒ 대응특허에 관한 별지를 참조하십시오.

* 인용된 문헌의 특별 카테고리:

"A" 특별히 관련이 없는 것으로 보이는 일반적인 기술수준을 정의한 문헌

"E" 국제출원일보다 빠른 출원일 또는 우선일을 가지나 국제출원일 이후에 공개된 선출원 또는 특허 문헌

"L" 우선권 주장에 의문을 제기하는 문헌 또는 다른 인용문헌의 공개일 또는 다른 특별한 이유(이유를 명시)를 밝히기 위하여 인용된 문헌

"O" 구두 개시, 사용, 전시 또는 기타 수단을 언급하고 있는 문헌

"P" 우선일 이후에 공개되었으나 국제출원일 이전에 공개된 문헌

"T" 국제출원일 또는 우선일 후에 공개된 문헌으로, 출원과 상충하지 않으며 발명의 기초가 되는 원리나 이론을 이해하기 위해 인용된 문헌

"X" 특별한 관련이 있는 문헌. 해당 문헌 하나만으로 청구된 발명의 신규성 또는 진보성이 없는 것으로 본다.

"Y" 특별한 관련이 있는 문헌. 해당 문헌이 하나 이상의 다른 문헌과 조합하는 경우로 그 조합이 당업자에게 자명한 경우 청구된 발명은 진보성이 없는 것으로 본다.

"&" 동일한 대응특허문헌에 속하는 문헌

국제조사의 실제 완료일	국제조사보고서 발송일
2018년 06월 19일 (19.06.2018)	2018년 07월 03일 (03.07.2018)

ISA/KR의 명칭 및 우편주소
대한민국 특허청
(35208) 대전광역시 서구 청사로 189,
4동 (둔산동, 정부대전청사)
팩스 번호 +82-42-481-8578

심사관
김성우
전화번호 +82-42-481-3348

국제조사보고서
대응특허에 관한 정보

국제출원번호
PCT/KR2018/002957

국제조사보고서에서 인용된 특허문헌	공개일	대응특허문헌	공개일
US 2016-0352780 A1	2016/12/01	US 2009-0077621 A1	2009/03/19
		US 2015-0237073 A1	2015/08/20
		US 2017-0289198 A1	2017/10/05
		US 9043861 B2	2015/05/26
		US 9420006 B2	2016/08/16
		US 9692792 B2	2017/06/27
		WO 2009-036896 A2	2009/03/26
		WO 2009-036896 A3	2009/05/28
		WO 2009-036896 A4	2009/07/16
US 9521115 B1	2016/12/13	US 2017-279770 A1	2017/09/28
US 2009-0077086 A1	2009/03/19	US 8024771 B2	2011/09/20
US 2004-0042609 A1	2004/03/04	AU 2003-268466 A1	2004/03/29
		US 7043000 B2	2006/05/09
		WO 2004-023775 A1	2004/03/18
US 2015-0207813 A1	2015/07/23	US 2015-0222656 A1	2015/08/06
		US 2016-0164890 A1	2016/06/09
		US 2016-0269427 A1	2016/09/15
		US 2017-0279824 A1	2017/09/28
		US 2017-0316203 A1	2017/11/02
		US 8914406 B1	2014/12/16
		US 9038183 B1	2015/05/19
		US 9137258 B2	2015/09/15
		US 9167001 B1	2015/10/20
		US 9680846 B2	2017/06/13
		US 9710644 B2	2017/07/18
		US 9756082 B1	2017/09/05