Network Working Group   J. Jeong
Internet-Draft   C. Chung
Intended status: Standards Track   Sungkyunkwan University
Expires: September 12, 2019   S. Hares
                                                               L. Xia
                                                               Huawei
                                                          H. Birkholz
                                                       Fraunhofer SIT
                                                       March 11, 2019

                   I2NSF NSF Monitoring YANG Data Model
                draft-ietf-i2nsf-nsf-monitoring-data-model-00

Abstract

   This document proposes an information model and the corresponding
   YANG data model for monitoring Network Security Functions (NSFs) in
   the Interface to Network Security Functions (I2NSF) framework. If
   the monitoring of NSFs is performed in a comprehensive way, it is
   possible to detect the indication of malicious activity, anomalous
   behavior or the potential sign of denial of service attacks in a
   timely manner. This monitoring functionality is based on the
   monitoring information that is generated by NSFs. Thus, this
   document describes not only an information model for monitoring NSFs
   along with a YANG data diagram, but also the corresponding YANG data
   model for monitoring NSFs.

Editorial Note (To be removed by RFC Editor)

   Please update these statements within the document with the RFC
   number to be assigned to this document:

      "This version of this YANG module is part of RFC 6087;"

      "RFC XXXX: I2NSF NSF Monitoring YANG Data Model"

      "reference: RFC 6087"

   Please update the "revision" date of the YANG module.

Status of This Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute

working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 12, 2019.

Copyright Notice

Table of Contents

1. Introduction

   According to [I-D.ietf-i2nsf-terminology], the interface provided by
   a Network Security Function (NSF) (e.g., Firewall, IPS, Anti-DDoS, or
   Anti-Virus function) to administrative entities (e.g., Security
   Controller) to enable remote management (i.e., configuring and
   monitoring) is referred to as an I2NSF NSF-Facing Interface

[I-D.ietf-i2nsf-nsf-facing-interface-dm]. Monitoring procedures
intent to acquire vital types of data with respect to NSFs, (e.g.,
alarms, records, and counters) via data in motion (e.g., queries,
notifications, and events). The monitoring of NSF plays an important
role in an overall security framework, if it is done in a timely and
comprehensive way. The monitoring information generated by an NSF
can be a good, early indication of anomalous behavior or malicious
activity, such as denial of service attacks (DoS).

This document defines a comprehensive NSF monitoring information
model that provides visibility for an NSF for Security Controller.
It specifies the information and illustrates the methods that enable
an NSF to provide the information required in order to be monitored
in a scalable and efficient way via the NSF-Facing Interface. The
information model for monitoring presented in this document is a
complementary information model to the information model for the
security policy provisioning functionality of the NSF-Facing
Interface specified in [I-D.ietf-i2nsf-capability].

This document also defines a YANG [RFC7950] data model for monitoring
NSFs, which is derived from the information model for NSF monitoring.

2. Terminology

2.1. Requirements Notation

   This document does not propose a protocol standard, and the use of
   words such as "should" follow their ordinary English meaning and not
   that for normative languages defined in [RFC2119] [RFC8174].

2.2. Definitions

   The terms, which are used in this document, are defined in the I2NSF
   terminology document [I-D.ietf-i2nsf-terminology].

2.3. YANG

   This document follows the guidelines of [RFC6087], uses the common
   YANG types defined in [RFC6991], and adopts the Network Management
   Datastore Architecture (NMDA). The meaning of the symbols in tree
   diagrams is defined in [RFC8340].

3. Use Cases for NSF Monitoring Data

   As mentioned earlier, monitoring plays a critical role in an overall
   security framework. The monitoring of the NSF provides very valuable
   information to the security controller in maintaining the provisioned

security posture. Besides this, there are various other reasons to
monitor the NSF as listed below:

o The security administrator with I2NSF User can configure a policy
   that is triggered on a specific event occurring in the NSF or the
   network [RFC8329] [I-D.ietf-i2nsf-consumer-facing-interface-dm].
   If a security controller detects the specified event, it
   configures additional security functions as defined by policies.

o The events triggered by an NSF as a result of security policy
   violation can be used by Security Information and Event Management
   (SIEM) to detect any suspicious activity in a larger correlation
   context.

o The events and activity logs from an NSF can be used to build
   advanced analytics, such as behavior and predictive models to
   improve security posture in large deployments.

o The security controller can use events from the NSF for achieving
   high availability. It can take corrective actions such as
   restarting a failed NSF and horizontally scaling up the NSF.

o The events and activity logs from the NSF can aid in the root
   cause analysis of an operational issue, so it can improve
   debugging.

o The activity logs from the NSF can be used to build historical
   data for operational and business reasons.

4. Classification of NSF Monitoring Data

   In order to maintain a strong security posture, it is not only
   necessary not only to configure an NSF's security policies but also
   to continuously monitor the NSF by consuming acquirable and
   observable information. This enables security administrators to
   assess the state of the network topology in a timely fashion. It is
   not possible to block all the internal and external threats based on
   static security posture. A more practical approach is supported by
   enabling dynamic security measures, for which continuous visibility
   is required. This document defines a set of information elements
   (and their scope) that can be acquired from an NSF and can be used as
   NSF monitoring information. In essence, these types of monitoring
   information can be leveraged to support constant visibility on
   multiple levels of granularity and can be consumed by the
   corresponding functions.

   Three basic domains about the monitoring information originating from
   a system entity [RFC4949] or an NSF are highlighted in this document.

o Retention and Emission

o Notifications and Events

o Unsolicited Poll and Solicited Push

The Alarm Management Framework in [RFC3877] defines an Event as
something that happens which may be of interest. It defines a fault
as a change in status, crossing a threshold, or an external input to
the system. In the I2NSF domain, I2NSF events
[I-D.ietf-i2nsf-terminology] are created and the scope of the Alarm
Management Framework's Events is still applicable due to its broad
definition. The model presented in this document elaborates on the
workflow of creating I2NSF events in the context of NSF monitoring
and on the way initial I2NSF events are created.

As with I2NSF components, every generic system entity can include a
set of capabilities [I-D.ietf-i2nsf-terminology] that creates
information about the context, composition, configuration, state or
behavior of that system entity. This information is intended to be
provided to other consumers of information and in the scope of this
document, which deals with NSF information monitoring in an automated
fashion.

4.1. Retention and Emission

Typically, a system entity populates standardized interface, such as
SNMP, NETCONF, RESTCONF or CoMI to provide and emit created
information directly via NSF-Facing Interface
[I-D.ietf-i2nsf-terminology]. Alternatively, the created information
is retained inside the system entity (or a hierarchy of system
entities in a composite device) via records or counters that are not
exposed directly via NSF-Facing Interfaces.

Information emitted via standardized interfaces can be consumed by an
I2NSF User [I-D.ietf-i2nsf-terminology] that includes the capability
to consume information not only via an I2NSF Interface(e.g.,
[I-D.ietf-i2nsf-consumer-facing-interface-dm]) but also via
interfaces complementary to the standardized interfaces a generic
system entity provides.

Information retained on a system entity requires a corresponding
I2NSF User to access aggregated records of information, typically in
the form of log-files or databases. There are ways to aggregate
records originating from different system entities over a network,
for examples via Syslog Protocol [RFC5424] or Syslog over TCP
[RFC6587]. But even if records are conveyed, the result is the same

kind of retention in form of a bigger aggregate of records on another
system entity.

An I2NSF User is required to process fresh [RFC4949] records created
by I2NSF Functions in order to provide them to other I2NSF Components
via the corresponding I2NSF Interfaces in a timely manner. This
process is effectively based on homogenizing functions, which can
access and convert specific kinds of records into information that
can be provided and emitted via I2NSF interfaces.

When retained or emitted, the information required to support
monitoring processes has to be processed by an I2NSF User at some
point in the workflow. Typical locations of these I2NSF Users are:

o a system entity that creates the information

o a system entity that retains an aggregation of records

o an I2NSF Component that includes the capabilities of using
   standardized interfaces provided by other system entities that are
   not I2NSF Components

o an I2NSF Component that creates the information

4.2. Notifications and Events

A specific task of I2NSF User is to process I2NSF Policy Rules
[I-D.ietf-i2nsf-terminology]. The rules of a policy are composed of
three clauses: Events, Conditions, and Actions. In consequence, an
I2NSF Event is specified to trigger an I2NSF Policy Rule. Such an
I2NSF Event is defined as any important occurrence over time in the
system being managed, and/or in the environment of the system being
managed in [I-D.ietf-i2nsf-terminology], which aligns well with the
generic definition of Event from [RFC3877].

The model illustrated in this document introduces a complementary
type of information that can be a conveyed notification.

Notification: An occurrence of a change of context, composition,
   configuration, state or behavior of a system entity that can be
   directly or indirectly observed by an I2NSF User and can be used
   as input for an event-clause in I2NSF Policy Rules.

   A notification is similar to an I2NSF Event with the exception
   that it is created by a system entity that is not an I2NSF
   Component and that its importance is yet to be assessed.
   Semantically, a notification is not an I2NSF Event in the context
   of I2NSF, although they can potentially use the exact same

information or data model. In respect to [RFC3877], a
Notification is a specific subset of events, because they convey
information about something that happens which may be of interest.
In consequence, Notifications may contain information with very
low expressiveness or relevance. Hence, additional post-
processing functions, such as aggregation, correlation or simple
anomaly detection, might have to be employed to satisfy a level of
expressiveness that is required for an event-clause of an I2NSF
Policy Rule.

It is important to note that the consumer of a notification (the
observer) assesses the importance of a notification and not the
producer. The producer can include metadata in a notification that
supports the observer in assessing the importance (even metadata
about severity), but the deciding entity is an I2NSF User.

4.3. Unsolicited Poll and Solicited Push

The freshness of the monitored information depends on the acquisition
method. Ideally, an I2NSF User is accessing every relevant
information about the I2NSF Component and is emitting I2NSF Events to
a monitor entity(e.g., Security Controller and I2NSF User) NSF
timely. Publication of events via a pubsub/broker model, peer-2-peer
meshes, or static defined channels are only a few examples on how a
solicited push of I2NSF Events can be facilitated. The actual
mechanic implemented by an I2NSF Component is out of the scope of
this document.

Often, the corresponding management interfaces have to be queried in
intervals or on-demand if required by an I2NSF Policy rule. In some
cases, a collection of information has to be conducted via login
mechanics provided by a system entity. Accessing records of
information via this kind of unsolicited polls can introduce a
significant latency in regard to the freshness of the monitored
information. The actual definition of intervals implemented by an
I2NSF Component is also out of scope of this document.

4.4. I2NSF Monitoring Terminology for Retained Information

Records: Unlike information emitted via notifications and events,
records do not require immediate attention from an analyst but may
be useful for visibility and retroactive cyber forensic.
Depending on the record format, there are different qualities in
regard to structure and detail. Records are typically stored in
log-files or databases on a system entity or NSF. Records in the
form of log-files usually include less structures but potentially
more detailed information in regard to the changes of a system
entity's characteristics. In contrast, databases often use more

strict schemas or data models, therefore enforcing a better
structure. However, they inhibit storing information that do not
match those models ("closed world assumption"). Records can be
continuously processed by I2NSF Agents that act as I2NSF Producer
and emit events via functions specifically tailored to a certain
type of record. Typically, records are information generated
either by an NSF or a system entity about operational and
informational data, or various changes in system characteristics,
such as user activities, network/traffic status, and network
activity. They are important for debugging, auditing and security
forensic.

Counters: A specific representation of continuous value changes of
information elements that potentially occur in high frequency.
Prominent example are network interface counters, e.g., PDU amount
or byte amount, drop counters, and error counters. Counters are
useful in debugging and visibility into operational behavior of an
NSF. An I2NSF Agent that observes the progression of counters can
act as an I2NSF Producer and emit events in respect to I2NSF
Policy Rules.

5. Conveyance of NSF Monitoring Information

As per the use cases of NSF monitoring data, information needs to be
conveyed to various I2NSF Consumers based on requirements imposed by
I2NSF Capabilities and workflows. There are multiple aspects to be
considered in regard to the emission of monitoring information to
requesting parties as listed below:

o Pull-Push Model: A set of data can be pushed by an NSF to a
   requesting party or pulled by a requesting party from an NSF.
   Specific types of information might need both the models at the
   same time if there are multiple I2NSF Consumers with varying
   requirements. In general, any I2NSF Event including a high
   severity assessment is considered to be of great importance and
   should be processed as soon as possible (push-model). Records, in
   contrast, are typically not as critical (pull-model). The I2NSF
   Architecture does not mandate a specific scheme for each type of
   information and is therefore out of scope of this document.

o Pub-Sub Model: In order for an I2NSF Provider to push monitoring
   information to multiple appropriate I2NSF Consumers, a
   subscription can be maintained by both I2NSF Components.
   Discovery of available monitoring information can be supported by
   an I2NSF Controller that takes the role of a broker and therefore
   includes I2NSF Capabilities that support registration.

o Export Frequency: Monitoring information can be emitted
   immediately upon generation by an NSF to requesting I2NSF
   Consumers or can be pushed periodically. The frequency of
   exporting the data depends upon its size and timely usefulness.
   It is out of the scope of I2NSF and left to each NSF
   implementation.

o Authentication: There may be a need for authentication between an
   I2NSF Producer of monitoring information and its corresponding
   I2NSF Consumer to ensure that critical information remains
   confidential. Authentication in the scope of I2NSF can also
   require its corresponding content authorization. This may be
   necessary, for example, if an NSF emits monitoring information to
   an I2NSF Consumer outside its administrative domain. The I2NSF
   Architecture does not mandate when and how specific authentication
   has to be implemented.

o Data-Transfer Model: Monitoring information can be pushed by an
   NSF using a connection-less model that does require a persistent
   connection or streamed over a persistent connection. An
   appropriate model depends on the I2NSF Consumer requirements and
   the semantics of the information to be conveyed.

o Data Model and Interaction Model for Data in Motion: There are a
   lot of transport mechanisms such as IP, UDP, and TCP. There are
   also open source implementations for specific set of data such as
   systems counter, e.g. IPFIX [RFC7011] and NetFlow [RFC3954]. The
   I2NSF does not mandate any specific method for a given data set,
   so it is up to each implementation.

5.1. Information Types and Acquisition Methods

   In this document, most defined information types defined benefit from
   high visibility with respect to value changes, e.g., alarms and
   records. In contrast, values that change monotonically in a
   continuous way do not benefit from this high visibility. On the
   contrary, emitting each change would result in a useless amount of
   value updates. Hence, values, such as counter, are best acquired in
   periodic intervals.

   The mechanisms provided by YANG Push [I-D.ietf-netconf-yang-push] and
   YANG Subscribed Notifications
   [I-D.ietf-netconf-subscribed-notifications] address exactly these set
   of requirements. YANG also enables semantically well-structured
   information, as well as subscriptions to datastores or event streams
   - by changes or periodically.

In consequence, this information model in this document is intended to support data models used in solicited or unsolicited event streams that potentially are facilitated by a subscription mechanism. A subset of information elements defined in the information model address this domain of application.

6. Basic Information Model for All Monitoring Data

As explained in the above section, there is a wealth of data available from the NSF that can be monitored. Firstly, there must be some general information with each monitoring message sent from an NSF that helps a consumer to identify meta data with that message, which are listed as below:

o message_version: It indicates the version of the data format and is a two-digit decimal numeral starting from 01.

o message_type: Event, Alert, Alarm, Log, Counter, etc.

o time_stamp: It indicates the time when the message is generated.

o vendor_name: The name of the NSF vendor.

o NSF_name: The name (or IP) of the NSF generating the message.

o Module_name: The module name outputting the message.

o Severity: It indicates the level of the logs. There are total eight levels, from 0 to 7. The smaller the numeral is, the higher the severity is.

7. Extended Information Model for Monitoring Data

This section covers the additional information associated with the system messages. The extended information model is only for the structured data such as alarm. Any unstructured data is specified with basic information model only.

7.1. System Alarm

Characteristics:

o acquisition_method: subscription

o emission_type: on-change

o dampening_type: no-dampening

7.1.1. Memory Alarm

   The following information should be included in a Memory Alarm:

   o event_name: MEM_USAGE_ALARM

   o module_name: It indicates the NSF module responsible for
      generating this alarm.

   o usage: specifies the amount of memory used.

   o threshold: The threshold triggering the alarm

   o severity: The severity of the alarm such as critical, high,
      medium, low

   o message: The memory usage exceeded the threshold

7.1.2. CPU Alarm

   The following information should be included in a CPU Alarm:

   o event_name: CPU_USAGE_ALARM

   o usage: Specifies the amount of CPU used.

   o threshold: The threshold triggering the event

   o severity: The severity of the alarm such as critical, high,
      medium, low

   o message: The CPU usage exceeded the threshold.

7.1.3. Disk Alarm

   The following information should be included in a Disk Alarm:

   o event_name: DISK_USAGE_ALARM

   o usage: Specifies the amount of disk space used.

   o threshold: The threshold triggering the event

   o severity: The severity of the alarm such as critical, high,
      medium, low

   o message: The disk usage exceeded the threshold.

7.1.4. Hardware Alarm

   The following information should be included in a Hardware Alarm:

   o event_name: HW_FAILURE_ALARM

   o component_name: It indicates the HW component responsible for
      generating this alarm.

   o threshold: The threshold triggering the alarm

   o severity: The severity of the alarm such as critical, high,
      medium, low

   o message: The HW component has failed or degraded.

7.1.5. Interface Alarm

   The following information should be included in a Interface Alarm:

   o event_name: IFNET_STATE_ALARM

   o interface_Name: The name of interface

   o interface_state: UP, DOWN, CONGESTED

   o threshold: The threshold triggering the event

   o severity: The severity of the alarm such as critical, high,
      medium, low

   o message: Current interface state

7.2. System Events

   Characteristics:

   o acquisition_method: subscription

   o emission_type: on-change

   o dampening_type: on-repetition

7.2.1. Access Violation

   The following information should be included in this event:

   o event_name: ACCESS_DENIED

o user: Name of a user

o group: Group to which a user belongs

o login_ip_address: Login IP address of a user

o authentication_mode: User authentication mode. e.g., Local
   Authentication, Third-Party Server Authentication, Authentication
   Exemption, Single Sign-On (SSO) Authentication

o message: access is denied.

7.2.2. Configuration Change

   The following information should be included in this event:

o event_name: CONFIG_CHANGE

o user: Name of a user

o group: Group to which a user belongs

o login_ip_address: Login IP address of a user

o authentication_mode: User authentication mode. e.g., Local
   Authentication, Third-Party Server Authentication, Authentication
   Exemption, SSO Authentication

o message: Configuration is modified.

7.3. System Log

   Characteristics:

o acquisition_method: subscription

o emission_type: on-change

o dampening_type: on-repetition

7.3.1. Access Logs

   Access logs record administrators' login, logout, and operations on a
   device. By analyzing them, security vulnerabilities can be
   identified. The following information should be included in an
   operation report:

o Administrator: Administrator that operates on the device

o login_ip_address: IP address used by an administrator to log in

o login_mode: Specifies the administrator logs in mode e.g. root, user

o operation_type: The operation type that the administrator execute, e.g., login, logout, and configuration.

o result: Command execution result

o content: Operation performed by an administrator after login.

7.3.2. Resource Utilization Logs

Running reports record the device system's running status, which is useful for device monitoring. The following information should be included in running report:

o system_status: The current system's running status

o CPU_usage: Specifies the CPU usage.

o memory_usage: Specifies the memory usage.

o disk_usage: Specifies the disk usage.

o disk_left: Specifies the available disk space left.

o session_number: Specifies total concurrent sessions.

o process_number: Specifies total number of system processes.

o in_traffic_rate: The total inbound traffic rate in pps

o out_traffic_rate: The total outbound traffic rate in pps

o in_traffic_speed: The total inbound traffic speed in bps

o out_traffic_speed: The total outbound traffic speed in bps

7.3.3. User Activity Logs

User activity logs provide visibility into users' online records (such as login time, online/lockout duration, and login IP addresses) and the actions that users perform. User activity reports are helpful to identify exceptions during a user's login and network access activities.

o user: Name of a user

o group: Group to which a user belongs

o login_ip_address: Login IP address of a user

o authentication_mode: User authentication mode. e.g., Local
   Authentication, Third-Party Server Authentication, Authentication
   Exemption, SSO Authentication

o access_mode: User access mode. e.g., PPP, SVN, LOCAL

o online_duration: Online duration

o lockout_duration: Lockout duration

o type: 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

o cause: Cause of a failed user activity

7.4. System Counters

Characteristics:

o acquisition_method: subscription or query

o emission_type: periodical

o dampening_type: none

7.4.1. Interface counters

Interface counters provide visibility into traffic into and out of an
NSF, and bandwidth usage.

o interface_name: Network interface name configured in NSF

o in_total_traffic_pkts: Total inbound packets

o out_total_traffic_pkts: Total outbound packets

o in_total_traffic_bytes: Total inbound bytes

o out_total_traffic_bytes: Total outbound bytes

o in_drop_traffic_pkts: Total inbound drop packets

o out_drop_traffic_pkts: Total outbound drop packets

o in_drop_traffic_bytes: Total inbound drop bytes

o out_drop_traffic_bytes: Total outbound drop bytes

o in_traffic_ave_rate: Inbound traffic average rate in pps

o in_traffic_peak_rate: Inbound traffic peak rate in pps

o in_traffic_ave_speed: Inbound traffic average speed in bps

o in_traffic_peak_speed: Inbound traffic peak speed in bps

o out_traffic_ave_rate: Outbound traffic average rate in pps

o out_traffic_peak_rate: Outbound traffic peak rate in pps

o out_traffic_ave_speed: Outbound traffic average speed in bps

o out_traffic_peak_speed: Outbound traffic peak speed in bps

7.5. NSF Events

Characteristics:

o acquisition_method: subscription

o emission_type: on-change

o dampening_type: none

7.5.1. DDoS Event

The following information should be included in a DDoS Event:

o event_name: SEC_EVENT_DDoS

o sub_attack_type: Any one of SYN flood, ACK flood, SYN-ACK flood,
   FIN/RST flood, TCP Connection flood, UDP flood, ICMP flood, HTTPS
   flood, HTTP flood, DNS query flood, DNS reply flood, SIP flood,
   and etc.

o dst_ip: The IP address of a victum under attack

o dst_port: The port number that the attrack traffic aims at.

o start_time: The time stamp indicating when the attack started

o end_time: The time stamp indicating when the attack ended. If the
   attack is still undergoing when sending out the alarm, this field
   can be empty.

o attack_rate: The PPS of attack traffic

o attack_speed: the bps of attack traffic

o rule_id: The ID of the rule being triggered

o rule_name: The name of the rule being triggered

o profile: Security profile that traffic matches.

7.5.2. Session Table Event

   The following information should be included in a Session
   Table Event:

o event_name: SESSION_USAGE_HIGH

o current: The number of concurrent sessions

o max: The maximum number of sessions that the session table can
   support

o threshold: The threshold triggering the event

o message: The number of session table exceeded the threshold.

7.5.3. Virus Event

   The following information should be included in a Virus Event:

o event_Name: SEC_EVENT_VIRUS

o virus_type: Type of the virus. e.g., trojan, worm, macro virus
   type

o virus_name: Name of the virus

o dst_ip: The destination IP address of the packet where the virus
   is found

o src_ip: The source IP address of the packet where the virus is
   found

o src_port: The source port of the packet where the virus is found

o dst_port: The destination port of the packet where the virus is
   found

o src_zone: The source security zone of the packet where the virus
   is found

o dst_zone: The destination security zone of the packet where the
   virus is found

o file_type: The type of the file where the virus is hided within

o file_name: The name of the file where the virus is hided within

o virus_info: The brief introduction of the virus

o raw_info: The information describing the packet triggering the
   event.

o rule_id: The ID of the rule being triggered

o rule_name: The name of the rule being triggered

o profile: Security profile that traffic matches.

7.5.4. Intrusion Event

   The following information should be included in an Intrustion Event:

   o event_name: The name of event. e.g., SEC_EVENT_Intrusion

   o sub_attack_type: Attack type, e.g., brutal force and buffer
      overflow

   o src_ip: The source IP address of the packet

   o dst_ip: The destination IP address of the packet

   o src_port:The source port number of the packet

   o dst_port: The destination port number of the packet

   o src_zone: The source security zone of the packet

   o dst_zone: The destination security zone of the packet

   o protocol: The employed transport layer protocol. e.g.,TCP and UDP

   o app: The employed application layer protocol. e.g.,HTTP and FTP

o rule_id: The ID of the rule being triggered

o rule_name: The name of the rule being triggered

o profile: Security profile that traffic matches

o intrusion_info: Simple description of intrusion

o raw_info: The information describing the packet triggering the
   event

7.5.5. Botnet Event

The following information should be included in a Botnet Event:

o event_name: The name of event. e.g., SEC_EVENT_Botnet

o botnet_name: The name of the detected botnet

o src_ip: The source IP address of the packet

o dst_ip: The destination IP address of the packet

o src_port: The source port number of the packet

o dst_port: The destination port number of the packet

o src_zone: The source security zone of the packet

o dst_zone: The destination security zone of the packet

o protocol: The employed transport layer protocol. e.g.,TCP and UDP

o app: The employed application layer protocol. e.g.,HTTP and FTP

o role: The role of the communicating parties within the botnet:

   1. The packet from the zombie host to the attacker

   2. The packet from the attacker to the zombie host

   3. The packet from the IRC/WEB server to the zombie host

   4. The packet from the zombie host to the IRC/WEB server

   5. The packet from the attacker to the IRC/WEB server

   6. The packet from the IRC/WEB server to the attacker

       7. The packet from the zombie host to the victim

   o botnet_info: Simple description of Botnet

   o rule_id: The ID of the rule being triggered

   o rule_name: The name of the rule being triggered

   o profile: Security profile that traffic matches

   o raw_info: The information describing the packet triggering the
      event.

7.5.6. Web Attack Event

   The following information should be included in a Web Attack Alarm:

   o event_name: The name of event. e.g., SEC_EVENT_WebAttack

   o sub_attack_type: Concret web attack type. e.g., SQL injection,
      command injection, XSS, CSRF

   o src_ip: The source IP address of the packet

   o dst_ip: The destination IP address of the packet

   o src_port: The source port number of the packet

   o dst_port: The destination port number of the packet

   o src_zone: The source security zone of the packet

   o dst_zone: The destination security zone of the packet

   o req_method: The method of requirement. For instance, "PUT" and
      "GET" in HTTP

   o req_url: Requested URL

   o url_category: Matched URL category

   o filtering_type: URL filtering type. e.g., Blacklist, Whitelist,
      User-Defined, Predefined, Malicious Category, and Unknown

   o rule_id: The ID of the rule being triggered

   o rule_name: The name of the rule being triggered

o profile: Security profile that traffic matches

7.6. NSF Logs

Characteristics:

o acquisition_method: subscription

o emission_type: on-change

o dampening_type: on_repetition

7.6.1. DDoS Logs

Besides the fields in a DDoS Alarm, the following information should be included in a DDoS Logs:

o attack_type: DDoS

o attack_ave_rate: The average pps of the attack traffic within the recorded time

o attack_ave_speed: The average bps of the attack traffic within the recorded time

o attack_pkt_num: The number of attack packets within the recorded time

o attack_src_ip: The source IP addresses of attack traffics. If there are a large number of IP addresses, then pick a certain number of resources according to different rules.

o action: Actions against DDoS attacks. e.g., Allow, Alert, Block, Discard, Declare, Block-ip, and Block-service.

7.6.2. Virus Logs

Besides the fields in a Virus Alarm, the following information should be included in a Virus Logs:

o attack_type: Virus

o protocol: The transport layer protocol

o app: The name of the application layer protocol

o times: The time of detecting the virus

o action: The actions dealing with the virus. e.g., alert and block

o os: The OS that the virus will affect. e.g., all, android, ios, unix, and windows

7.6.3. Intrusion Logs

Besides the fields in an Intrusion Alarm, the following information should be included in an Intrusion Logs:

o attack_type: Intrusion

o times: The times of intrusions happened in the recorded time

o os: The OS that the intrusion will affect. e.g., all, android, ios, unix, and windows

o action: The actions dealing with the intrusions. e.g., Allow, Alert, Block, Discard, Declare, Block-ip, and Block-service

o attack_rate: NUM the pps of attack traffic

o attack_speed: NUM the bps of attack traffic

7.6.4. Botnet Logs

Besides the fields in a Botnet Alarm, the following information should be included in a Botnet Logs:

o attack_type: Botnet

o botnet_pkt_num:The number of the packets sent to or from the detected botnet

o action: The actions dealing with the detected packets. e.g., Allow, Alert, Block, Discard, Declare, Block-ip, and Block-service.

o os: The OS that the attack aims at. e.g., all, android, ios, unix, and windows.

7.6.5. DPI Logs

DPI Logs provide statistics on uploaded and downloaded files and data, sent and received emails, and alert and block records on websites. It is helpful to learn risky user behaviors and why access to some URLs is blocked or allowed with an alert record.

   o type: DPI action types. e.g., File Blocking, Data Filtering, and
      Application Behavior Control

   o file_name: The file name

   o file_type: The file type

   o src_zone: Source security zone of traffic

   o dst_zone: Destination security zone of traffic

   o src_region: Source region of traffic

   o dst_region: Destination region of traffic

   o src_ip: Source IP address of traffic

   o src_user: User who generates traffic

   o dst_ip: Destination IP address of traffic

   o src_port: Source port of traffic

   o dst_port: Destination port of traffic

   o protocol: Protocol type of traffic

   o app: Application type of traffic

   o policy_id: Security policy id that traffic matches

   o policy_name: Security policy name that traffic matches

   o action: Action defined in the file blocking rule, data filtering
      rule, or application behavior control rule that traffic matches.

7.6.6. Vulnerabillity Scanning Logs

   Vulnerability scanning logs record the victim host and its related
   vulnerability information that should to be fixed. The following
   information should be included in the report:

   o victim_ip: IP address of the victim host which has vulnerabilities

   o vulnerability_id: The vulnerability id

   o vulnerability_level: The vulnerability level. e.g., high, middle,
      and low

   o OS: The operating system of the victim host

   o service: The service which has vulnerabillity in the victim host

   o protocol: The protocol type. e.g., TCP and UDP

   o port: The port number

   o vulnerability_info: The information about the vulnerability

   o fix_suggestion: The fix suggestion to the vulnerability.

7.6.7. Web Attack Logs

   Besides the fields in an Web Attack Alarm, the following information
   should be included in a Web Attack Report:

   o attack_type: Web Attack

   o rsp_code: Response code

   o req_clientapp: The client application

   o req_cookies: Cookies

   o req_host: The domain name of the requested host

   o raw_info: The information describing the packet triggering the
      event.

7.7. NSF Counters

   Characteristics:

   o acquisition_method: subscription or query

   o emission_type: periodical

   o dampening_type: none

7.7.1. Firewall counters

   Firewall counters provide visibility into traffic signatures,
   bandwidth usage, and how the configured security and bandwidth
   policies have been applied.

   o src_zone: Source security zone of traffic

   o dst_zone: Destination security zone of traffic

   o src_region: Source region of traffic

   o dst_region: Destination region of traffic

   o src_ip: Source IP address of traffic

   o src_user: User who generates traffic

   o dst_ip: Destination IP address of traffic

   o src_port: Source port of traffic

   o dst_port: Destination port of traffic

   o protocol: Protocol type of traffic

   o app: Application type of traffic

   o policy_id: Security policy id that traffic matches

   o policy_name: Security policy name that traffic matches

   o in_interface: Inbound interface of traffic

   o out_interface: Outbound interface of traffic

   o total_traffic: Total traffic volume

   o in_traffic_ave_rate: Inbound traffic average rate in pps

   o in_traffic_peak_rate: Inbound traffic peak rate in pps

   o in_traffic_ave_speed: Inbound traffic average speed in bps

   o in_traffic_peak_speed: Inbound traffic peak speed in bps

   o out_traffic_ave_rate: Outbound traffic average rate in pps

   o out_traffic_peak_rate: Outbound traffic peak rate in pps

   o out_traffic_ave_speed: Outbound traffic average speed in bps

   o out_traffic_peak_speed: Outbound traffic peak speed in bps.

7.7.2. Policy Hit Counters

   Policy Hit Counters record the security policy that traffic matches
   and its hit count. It can check if policy configurations are
   correct.

   o src_zone: Source security zone of traffic

   o dst_zone: Destination security zone of traffic

   o src_region: Source region of the traffic

   o dst_region: Destination region of the traffic

   o src_ip: Source IP address of traffic

   o src_user: User who generates traffic

   o dst_ip: Destination IP address of traffic

   o src_port: Source port of traffic

   o dst_port: Destination port of traffic

   o protocol: Protocol type of traffic

   o app: Application type of traffic

   o policy_id: Security policy id that traffic matches

   o policy_name: Security policy name that traffic matches

   o hit_times: The hit times that the security policy matches the
      specified traffic.

8. NSF Monitoring Management in I2NSF

   A standard model for monitoring data is required for an administrator
   to check the monitoring data generated by an NSF. The administrator
   can check the monitoring data through the following process. When
   the NSF monitoring data that is under the standard format is
   generated, the NSF forwards it to the security controller. The
   security controller delivers it to I2NSF Consumer or Developer's
   Management System (DMS) so that the administrator can know the state
   of the I2NSF framework.

In order to communicate with other components, an I2NSF framework
[RFC8329] requires the interfaces. The three main interfaces in
I2NSF framwork are used for sending monitoring data as follows:

o I2NSF Consumer-Facing Interface
   [I-D.ietf-i2nsf-consumer-facing-interface-dm]: When an I2NSF User
   makes a security policy and forwards it to the Security Controller
   via Consumer-Facing Interface, it can specify the threat-feed for
   threat prevention, the custom list, the malicious code scan group,
   and the event map group. They can be used as an event to be
   monitored by an NSF.

o I2NSF Registration Interface
   [I-D.ietf-i2nsf-registration-interface-dm]: The Network Functions
   Virtualization (NFV) architecture provides the lifecycle
   management of a Virtual Network Function (VNF) via the Ve-Vnfm
   interface. The role of Ve-Vnfm is to request VNF lifecycle
   management (e.g., the instantiation and de-instantiation of an
   NSF, and load balancing among NSFs), exchange configuration
   information, and exchange status information for a network
   service. In the I2NSF framework, the DMS manages data about
   resource states and network traffic for the lifecycle management
   of an NSF. Therefore, the generated monitoring data from NSFs are
   delivered from the Security Controller to the DMS via Registration
   Interface. These data are delivered from the DMS to the VNF
   Manager in the Management and Orchestration (MANO) in the NFV
   system [I-D.yang-i2nsf-nfv-architecture].

o I2NSF NSF-Facing Interface
   [I-D.ietf-i2nsf-nsf-facing-interface-dm]: After a high-level
   security policy from I2NSF User is translated by security policy
   translator [I-D.yang-i2nsf-security-policy-translation] in the
   Security Controller, the translated security policy (i.e., low-
   level policy) is applied to an NSF via NSF-Facing Interface. The
   monitoring data model specifies the list of events that can
   trigger Event-Condition-Action (ECA) policies via NSF-Facing
   Interface.

9. Tree Structure

   The tree structure of the NSF monitoring YANG module is provided
   below:

   module: ietf-i2nsf-monitor
      +--rw counters
         +--rw system-interface
            | +--rw acquisition-method?   identityref
            | +--rw emission-type?   identityref

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

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

   notifications:
     +---n system-detection-alarm
     │  +--ro alarm-catagory?       identityref
     │  +--ro acquisition-method?   identityref
     │  +--ro emission-type?        identityref
     │  +--ro dampening-type?       identityref
     │  +--ro usage?                uint8
     │  +--ro threshold?            uint8
     │  +--ro message?              string
     │  +--ro time-stamp?           yang:date-and-time
     │  +--ro vendor-name?          string
     │  +--ro nsf-name?             string
     │  +--ro module-name?          string
     │  +--ro severity?             severity
     +---n system-detection-event
     │  +--ro event-catagory?       identityref
     │  +--ro acquisition-method?   identityref
     │  +--ro emission-type?        identityref
     │  +--ro dampening-type?       identityref
     │  +--ro user                  string
```

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

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

```
          │  +--ro vendor-name?    string
          │  +--ro nsf-name?       string
          │  +--ro module-name?    string
          │  +--ro severity?       severity
          +---n nsf-detection-web-attack
          │  +--ro src-ip?             inet:ipv4-address
          │  +--ro dst-ip?             inet:ipv4-address
          │  +--ro src-port?           inet:port-number
          │  +--ro dst-port?           inet:port-number
          │  +--ro src-zone?           string
          │  +--ro dst-zone?           string
          │  +--ro rule-id            uint8
          │  +--ro rule-name          string
          │  +--ro profile?           string
          │  +--ro raw-info?          string
          │  +--ro sub-attack-type?   identityref
          │  +--ro request-method?    identityref
          │  +--ro req-uri?           string
          │  +--ro uri-category?      string
          │  +--ro filtering-type*    identityref
          │  +--ro message?          string
          │  +--ro time-stamp?       yang:date-and-time
          │  +--ro vendor-name?      string
          │  +--ro nsf-name?         string
          │  +--ro module-name?      string
          │  +--ro severity?         severity
          +---n system-access-log
          │  +--ro login-ip             inet:ipv4-address
          │  +--ro administrator?       string
          │  +--ro login-mode?          login-mode
          │  +--ro operation-type?      operation-type
          │  +--ro result?             string
          │  +--ro content?            string
          │  +--ro acquisition-method?  identityref
          │  +--ro emission-type?       identityref
          │  +--ro dampening-type?      identityref
          +---n system-res-util-log
          │  +--ro system-status?       string
          │  +--ro cpu-usage?          uint8
          │  +--ro memory-usage?       uint8
          │  +--ro disk-usage?         uint8
          │  +--ro disk-left?          uint8
          │  +--ro session-num?        uint8
          │  +--ro process-num?        uint8
          │  +--ro in-traffic-rate?     uint32
          │  +--ro out-traffic-rate?    uint32
          │  +--ro in-traffic-speed?    uint32
          │  +--ro out-traffic-speed?   uint32
```

```
         │  +--ro acquisition-method?   identityref
         │  +--ro emission-type?        identityref
         │  +--ro dampening-type?       identityref
         +---n system-user-activity-log
         │  +--ro acquisition-method?   identityref
         │  +--ro emission-type?        identityref
         │  +--ro dampening-type?       identityref
         │  +--ro user                  string
         │  +--ro group                 string
         │  +--ro login-ip-addr         inet:ipv4-address
         │  +--ro authentication?       identityref
         │  +--ro access?               identityref
         │  +--ro online-duration?      string
         │  +--ro logout-duration?      string
         │  +--ro addtional-info?       string
         +---n nsf-log-ddos
         │  +--ro attack-type?          identityref
         │  +--ro attack-ave-rate?      uint32
         │  +--ro attack-ave-speed?     uint32
         │  +--ro attack-pkt-num?       uint32
         │  +--ro attack-src-ip?        inet:ipv4-address
         │  +--ro action?               log-action
         │  +--ro acquisition-method?   identityref
         │  +--ro emission-type?        identityref
         │  +--ro dampening-type?       identityref
         │  +--ro message?              string
         │  +--ro time-stamp?           yang:date-and-time
         │  +--ro vendor-name?          string
         │  +--ro nsf-name?             string
         │  +--ro module-name?          string
         │  +--ro severity?             severity
         +---n nsf-log-virus
         │  +--ro attack-type?          identityref
         │  +--ro action?               log-action
         │  +--ro os?                   string
         │  +--ro time                  yang:date-and-time
         │  +--ro acquisition-method?   identityref
         │  +--ro emission-type?        identityref
         │  +--ro dampening-type?       identityref
         │  +--ro message?              string
         │  +--ro time-stamp?           yang:date-and-time
         │  +--ro vendor-name?          string
         │  +--ro nsf-name?             string
         │  +--ro module-name?          string
         │  +--ro severity?             severity
         +---n nsf-log-intrusion
         │  +--ro attack-type?          identityref
         │  +--ro action?               log-action
```

```
     │  +--ro time                 yang:date-and-time
     │  +--ro attack-rate?         uint32
     │  +--ro attack-speed?        uint32
     │  +--ro acquisition-method?  identityref
     │  +--ro emission-type?       identityref
     │  +--ro dampening-type?      identityref
     │  +--ro message?             string
     │  +--ro time-stamp?          yang:date-and-time
     │  +--ro vendor-name?         string
     │  +--ro nsf-name?            string
     │  +--ro module-name?         string
     │  +--ro severity?            severity
     +---n nsf-log-botnet
     │  +--ro attack-type?         identityref
     │  +--ro action?              log-action
     │  +--ro botnet-pkt-num?      uint8
     │  +--ro os?                  string
     │  +--ro acquisition-method?  identityref
     │  +--ro emission-type?       identityref
     │  +--ro dampening-type?      identityref
     │  +--ro message?             string
     │  +--ro time-stamp?          yang:date-and-time
     │  +--ro vendor-name?         string
     │  +--ro nsf-name?            string
     │  +--ro module-name?         string
     │  +--ro severity?            severity
     +---n nsf-log-dpi
     │  +--ro attack-type?         dpi-type
     │  +--ro acquisition-method?  identityref
     │  +--ro emission-type?       identityref
     │  +--ro dampening-type?      identityref
     │  +--ro src-ip?              inet:ipv4-address
     │  +--ro dst-ip?              inet:ipv4-address
     │  +--ro src-port?            inet:port-number
     │  +--ro dst-port?            inet:port-number
     │  +--ro src-zone?            string
     │  +--ro dst-zone?            string
     │  +--ro src-region?          string
     │  +--ro dst-region?          string
     │  +--ro policy-id?           uint8
     │  +--ro policy-name?         string
     │  +--ro src-user?            string
     │  +--ro protocol?            identityref
     │  +--ro app?                 string
     │  +--ro message?             string
     │  +--ro time-stamp?          yang:date-and-time
     │  +--ro vendor-name?         string
     │  +--ro nsf-name?            string
```

```
         │   +--ro module-name?         string
         │   +--ro severity?            severity
         +---n nsf-log-vuln-scan
         │   +--ro vulnerability-id?    uint8
         │   +--ro victim-ip?           inet:ipv4-address
         │   +--ro protocol?            identityref
         │   +--ro port-num?            inet:port-number
         │   +--ro level?               severity
         │   +--ro os?                  string
         │   +--ro vulnerability-info?  string
         │   +--ro fix-suggestion?      string
         │   +--ro service?             string
         │   +--ro acquisition-method?  identityref
         │   +--ro emission-type?       identityref
         │   +--ro dampening-type?      identityref
         │   +--ro message?             string
         │   +--ro time-stamp?          yang:date-and-time
         │   +--ro vendor-name?         string
         │   +--ro nsf-name?            string
         │   +--ro module-name?         string
         │   +--ro severity?            severity
         +---n nsf-log-web-attack
             +--ro attack-type?         identityref
             +--ro rsp-code?            string
             +--ro req-clientapp?       string
             +--ro req-cookies?         string
             +--ro req-host?            string
             +--ro raw-info?            string
             +--ro acquisition-method?  identityref
             +--ro emission-type?       identityref
             +--ro dampening-type?      identityref
             +--ro message?             string
             +--ro time-stamp?          yang:date-and-time
             +--ro vendor-name?         string
             +--ro nsf-name?            string
             +--ro module-name?         string
             +--ro severity?            severity
```

Figure 1: Information Model for NSF Monitoring

10. YANG Data Model

   This section introduces a YANG data model for the information model
   of the NSF monitoring inforamtion model.

   <CODE BEGINS> file "ietf-i2nsf-monitor@2019-03-11.yang"
   module ietf-i2nsf-monitor {
       yang-version 1.1;

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

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

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

   Editor: Chaehong Chung
   <mailto:darkhong@skku.edu>";

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

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

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

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

revision "2019-03-11" {
  description "First revision";
```

```
      reference
        "RFC XXXX: I2NSF NSF Monitoring YANG Data Model";
    }

    typedef severity {
      type enumeration {
        enum high {
          description
            "high-level";
        }
        enum middle {
          description
            "middle-level";
        }
        enum low {
          description
            "low-level";
        }
      }
      description
        "An indicator representing severity";
    }
    typedef log-action {
      type enumeration {
        enum allow {
          description
            "If action is allow";
        }
        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
            "If action is block-ip";
        }
```

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

```
              "Root login-mode";
          }
          enum user{
            description
              "User login-mode";
          }
          enum guest{
            description
              "Guest login-mode";
          }
        }
        description
          "An indicater representing login-mode";
      }

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

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

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

```
          }
          identity sso-authentication {
            base authentication-mode;
            description
            "If authentication-mode is
            sso-authentication";
          }

          identity alarm-type {
            description
              "Base identity for detectable alarm types";
          }
          identity MEM-USAGE-ALARM {
            base alarm-type;
            description
            "A memory alarm is alerted";
          }
          identity CPU-USAGE-ALARM {
            base alarm-type;
            description
            "A cpu alarm is alerted";
          }
          identity DISK-USAGE-ALARM {
            base alarm-type;
            description
            "A disk alarm is alerted";
          }
          identity HW-FAILURE-ALARM {
            base alarm-type;
            description
            "A hardware alarm is alerted";
          }
          identity IFNET-STATE-ALARM {
            base alarm-type;
            description
            "An interface alarm is alerted";
          }
          identity event-type {
            description
              "Base identity for detectable event types";
          }
          identity ACCESS-DENIED {
            base event-type;
            description
            "The system event is access-denied.";
          }
          identity CONFIG-CHANGE {
            base event-type;
```

```
      description
      "The system event is config-change.";
    }

    identity flood-type {
      description
        "Base identity for detectable flood types";
    }
    identity syn-flood {
      base flood-type;
      description
        "A SYN flood is detected";
    }
    identity ack-flood {
      base flood-type;
      description
        "An ACK flood is detected";
    }
    identity syn-ack-flood {
      base flood-type;
      description
        "An SYN-ACK flood is detected";
    }
    identity fin-rst-flood {
      base flood-type;
      description
        "A FIN-RST flood is detected";
    }
    identity tcp-con-flood {
      base flood-type;
      description
        "A TCP connection flood is detected";
    }
    identity udp-flood {
      base flood-type;
      description
        "A UDP flood is detected";
    }
    identity icmp-flood {
      base flood-type;
      description
        "An ICMP flood is detected";
    }
    identity https-flood {
      base flood-type;
      description
        "A HTTPS flood is detected";
    }
```

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

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

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

```
        }
        identity macro {
          base virus-type;
          description
            "The detected virus type is macro";
        }
        identity intrusion-attack-type {
          base nsf-attack-type;
          description
            "The attack type is associatied with
            a detectedsystem-log intrusion";
        }
        identity brute-force {
          base intrusion-attack-type;
          description
            "The intrusion type is brute-force";
        }
        identity buffer-overflow {
          base intrusion-attack-type;
          description
            "The intrusion type is buffer-overflow";
        }
        identity web-attack-type {
          base nsf-attack-type;
          description
            "The attack type associated with
            a detected system-log web-attack";
        }
        identity command-injection {
          base web-attack-type;
          description
            "The detected web attack type is command injection";
        }
        identity xss {
          base web-attack-type;
          description
            "The detected web attack type is XSS";
        }
        identity csrf {
          base web-attack-type;
          description
            "The detected web attack type is CSRF";
        }
        identity ddos-attack-type {
          base nsf-attack-type;
          description
            "The attack type is associated with a detected
            nsf-log event";
```

```
        }

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

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

```
        }

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

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

```
        reference
          "RFC 792: Internet Control Message Protocol";
      }
      identity icmpv4 {
        base ipv4;
        description
          "ICMPv4 protocol type.";
      }
      identity icmpv6 {
        base ipv6;
        description
          "ICMPv6 protocol type.";
      }
      identity ip {
        base protocol-type;
        description
          "General IP protocol type.";
          reference
            "RFC 791: Internet Protocol
            RFC 2460: Internet Protocol, Version 6 (IPv6)";
      }
      identity ipv4 {
        base ip;
        description
          "IPv4 protocol type.";
          reference
            "RFC 791: Internet Protocol";
      }
      identity ipv6 {
        base ip;
        description
          "IPv6 protocol type.";
          reference
            "RFC 2460: Internet Protocol, Version 6 (IPv6)";
      }
      identity http {
        base tcp;
        description
          "HTPP protocol type.";
          reference
            "RFC 2616: Hypertext Transfer Protocol";
      }
      identity ftp {
        base tcp;
        description
         "FTP protocol type.";
         reference
           "RFC 959: File Transfer Protocol";
```

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

```
        leaf emission-type {
          type identityref {
            base emission-type;
          }
          description
            "The emission-type for characteristics";
        }
        leaf dampening-type {
          type identityref {
            base dampening-type;
          }
          description
            "The dampening-type for characteristics";
        }
      }
      grouping i2nsf-system-alarm-type-content {
        description
          "A set of system alarm type contents";
        leaf usage {
          type uint8;
          description
            "specifies the amount of usage";
        }
        leaf threshold {
          type uint8;
          description
            "The threshold triggering the alarm or the event";
        }
      }
      grouping i2nsf-system-event-type-content {
        description
          "System event metadata associated
           with system events caused by user activity.";
        leaf user {
          type string;
          mandatory true;
          description
            "Name of a user";
        }
        leaf group {
          type string;
          mandatory true;
          description
            "Group to which a user belongs.";
        }
        leaf login-ip-addr {
          type inet:ipv4-address;
          mandatory true;
```

```
            description
              "Login IP address of a user.";
          }
          leaf authentication {
            type identityref {
              base authentication-mode;
            }
            description
              "The authentication-mode for authentication";
          }
        }
        grouping i2nsf-nsf-event-type-content-extend {
          description
            "A set of common IPv4-related NSF event
            content elements";
          leaf src-ip {
            type inet:ipv4-address;
            description
              "The source IP address of the packet";
          }
          leaf dst-ip {
            type inet:ipv4-address;
            description
              "The destination IP address of the packet";
          }
          leaf src-port {
            type inet:port-number;
            description
              "The source port of the packet";
          }
          leaf dst-port {
            type inet:port-number;
            description
              "The destination port of the packet";
          }
          leaf src-zone {
            type string;
            description
              "The source security zone of the packet";
          }
          leaf dst-zone {
            type string;
            description
              "The destination security zone of the packet";
          }
          leaf rule-id {
            type uint8;
            mandatory true;
```

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

```
            type string;
            description
              "Security profile that traffic matches.";
          }
          leaf raw-info {
            type string;
            description
              "The information describing the packet
              triggering the event.";
          }
        }
        grouping traffic-rates {
          description
            "A set of traffic rates
            for statistics data";
          leaf total-traffic {
            type uint32;
            description
              "Total traffic";
          }
          leaf in-traffic-ave-rate {
            type uint32;
            description
              "Inbound traffic average rate in pps";
          }
          leaf in-traffic-peak-rate {
            type uint32;
            description
              "Inbound traffic peak rate in pps";
          }
          leaf in-traffic-ave-speed {
            type uint32;
            description
              "Inbound traffic average speed in bps";
          }
          leaf in-traffic-peak-speed {
            type uint32;
            description
              "Inbound traffic peak speed in bps";
          }
          leaf out-traffic-ave-rate {
            type uint32;
            description
              "Outbound traffic average rate in pps";
          }
          leaf out-traffic-peak-rate {
            type uint32;
            description
```

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

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

```
        leaf src-region {
          type string;
          description
            "Source region of the traffic";
        }
        leaf dst-region{
          type string;
          description
            "Destination region of the traffic";
        }
        leaf policy-id {
          type uint8;
          description
            "The ID of the policy being triggered";
        }
        leaf policy-name {
          type string;
          description
            "The name of the policy being triggered";
        }
        leaf src-user{
          type string;
          description
            "User who generates traffic";
        }
        leaf protocol {
          type identityref {
            base protocol-type;
          }
          description
            "Protocol type of traffic";
        }
        leaf app {
          type string;
          description
            "Application type of traffic";
        }
      }

    notification system-detection-alarm {
      description
        "This notification is sent, when a system alarm
        is detected.";
      leaf alarm-catagory {
        type identityref {
          base alarm-type;
        }
        description
```

```
            "The alarm catagory for
            system-detection-alarm notification";
        }
      uses characteristics;
      uses i2nsf-system-alarm-type-content;
      uses common-monitoring-data;
    }
    notification system-detection-event {
      description
        "This notification is sent, when a security-sensitive
         authentication action fails.";
      leaf event-catagory {
        type identityref {
          base event-type;
        }
        description
          "The event catagory for system-detection-event";
      }
      uses characteristics;
      uses i2nsf-system-event-type-content;
      uses common-monitoring-data;
    }
    notification nsf-detection-flood {
      description
        "This notification is sent,
        when a specific flood type is detected";
      leaf event-name {
        type identityref {
         base SEC-EVENT-DDOS;
        }
        description
        "The event name for nsf-detection-flood";
      }
      uses i2nsf-nsf-event-type-content;
      leaf sub-attack-type {
        type identityref {
          base flood-type;
        }
        description
          "Any one of Syn flood, ACK flood, SYN-ACK flood,
           FIN/RST flood, TCP Connection flood, UDP flood,
           Icmp flood, HTTPS flood, HTTP flood, DNS query flood,
           DNS reply flood, SIP flood, and etc.";
      }
      leaf start-time {
        type yang:date-and-time;
        mandatory true;
        description
```

```
            "The time stamp indicating when the attack started";
        }
        leaf end-time {
          type yang:date-and-time;
          mandatory true;
          description
            "The time stamp indicating when the attack ended";
        }
        leaf attack-rate {
          type uint32;
          description
            "The PPS rate of attack traffic";
        }
        leaf attack-speed {
          type uint32;
          description
            "The BPS speed of attack traffic";
        }
        uses common-monitoring-data;
      }
      notification nsf-detection-session-table {
        description
          "This notification is sent, when an a session table
          event is deteced";
        leaf current-session {
          type uint8;
          description
            "The number of concurrent sessions";
        }
        leaf maximum-session {
          type uint8;
          description
            "The maximum number of sessions that the session
             table can support";
        }
        leaf threshold {
          type uint8;
          description
            "The threshold triggering the event";
        }
        uses common-monitoring-data;
      }
      notification nsf-detection-virus {
        description
          "This notification is sent, when a virus is detected";
        uses i2nsf-nsf-event-type-content-extend;
        leaf virus {
          type identityref {
```

```
            base virus-type;
          }
          description
            "The virus type for nsf-detection-virus notification";
        }
        leaf virus-name {
          type string;
          description
          "The name of the detected virus";
        }

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

```
              "The sub attack type for intrusion attack";
          }
          uses common-monitoring-data;
        }
        notification nsf-detection-botnet {
          description
            "This notification is send, when a botnet event is
             detected";
          uses i2nsf-nsf-event-type-content-extend;
          leaf attack-type {
            type identityref {
              base botnet-attack-type;
            }
            description
              "The attack type for botnet attack";
          }
          leaf protocol {
            type identityref {
              base protocol-type;
            }
            description
              "The protocol type for nsf-detection-botnet notification";
          }
          leaf botnet-name {
            type string;
            description
              "The name of the detected botnet";
          }
          leaf role {
            type string;
            description
              "The role of the communicating
               parties within the botnet";
          }
          uses common-monitoring-data;
        }
        notification nsf-detection-web-attack {
          description
            "This notification is send, when an attack event is
             detected";
          uses i2nsf-nsf-event-type-content-extend;
          leaf sub-attack-type {
            type identityref {
              base web-attack-type;
            }
            description
              "Concret web attack type, e.g., sql injection,
               command injection, XSS, CSRF";
```

```
         }
         leaf request-method {
           type identityref {
             base req-method;
           }
           description
             "The method of requirement. For instance, PUT or
             GET in HTTP";
         }
         leaf req-uri {
           type string;
           description
           "Requested URI";
         }
         leaf uri-category {
           type string;
           description
           "Matched URI category";
         }
         leaf-list filtering-type {
           type identityref {
             base filter-type;
           }
           description
             "URL filtering type, e.g., Blacklist, Whitelist,
              User-Defined, Predefined, Malicious Category,
              Unknown";
         }
         uses common-monitoring-data;
       }
       notification system-access-log {
         description
           "The notification is send, if there is
           a new system log entry about
            a system access event";
         leaf login-ip {
           type inet:ipv4-address;
           mandatory true;
           description
             "Login IP address of a user";
         }
         leaf administrator {
           type string;
           description
             "Administrator that maintains the device";
         }
         leaf login-mode {
           type login-mode;
```

```
            description
              "Specifies the administrator log-in mode";
          }
          leaf operation-type {
            type operation-type;
            description
              "The operation type that the administrator execute";
          }
          leaf result {
            type string;
            description
              "Command execution result";
          }
          leaf content {
            type string;
            description
              "The Operation performed by an administrator
              after login";
          }
          uses characteristics;
        }
        notification system-res-util-log {
          description
            "This notification is send, if there is
            a new log entry representing ressource
             utiliztation updates.";
          leaf system-status {
             type string;
             description
               "The current systems
               running status";
          }
          leaf cpu-usage {
            type uint8;
            description
              "Specifies the relative amount of
              cpu usage wrt plattform ressources";
          }
          leaf memory-usage {
             type uint8;
             description
               "Specifies the amount of memory usage";
          }
          leaf disk-usage {
            type uint8;
            description
              "Specifies the amount of disk usage";
          }
```

```
          leaf disk-left {
            type uint8;
            description
              "Specifies the amount of disk left";
          }
          leaf session-num {
            type uint8;
            description
              "The total number of sessions";
          }
          leaf process-num {
            type uint8;
            description
              "The total number of process";
          }
          leaf in-traffic-rate {
            type uint32;
            description
              "The total inbound traffic rate in pps";
          }
          leaf out-traffic-rate {
            type uint32;
            description
              "The total outbount traffic rate in pps";
          }
          leaf in-traffic-speed {
            type uint32;
            description
              "The total inbound traffic speed in bps";
          }
          leaf out-traffic-speed {
            type uint32;
            description
              "The total outbound traffic speed in bps";
          }
          uses characteristics;
        }
      notification system-user-activity-log {
        description
          "This notification is send, if there is
          a new user activity log entry";
        uses characteristics;
        uses i2nsf-system-event-type-content;
        leaf access {
          type identityref {
            base access-mode;
          }
          description
```

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

```
              description
                "the number of attack packets";
            }
            leaf attack-src-ip {
              type inet:ipv4-address;
              description
                "The source IP addresses of attack
                 traffics. If there are a large
                 amount of IP addresses, then
                 pick a certain number of resources
                 according to different rules.";
            }
            leaf action {
              type log-action;
              description
                "Action type: allow, alert,
                block, discard, declare,
                block-ip, block-service";
            }
            uses characteristics;
            uses common-monitoring-data;
          }
          notification nsf-log-virus {
            description
              "This notification is send, If there is
              a new virus event log enry in the nsf log";
            leaf attack-type {
              type identityref {
                base virus-type;
              }
              description
                "The virus type for nsf-log-virus notification";
            }
            leaf action {
              type log-action;
              description
                "Action type: allow, alert,
                block, discard, declare,
                block-ip, block-service";
            }
            leaf os{
              type string;
              description
                "simple os information";
            }
            leaf time {
              type yang:date-and-time;
              mandatory true;
```

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

```
             "This noticiation is send, if there is
             a new botnet event log in the nsf log";
           leaf attack-type {
             type identityref {
               base botnet-attack-type;
             }
             description
               "The botnet attack type for
               nsf-log-botnet notification";
           }
           leaf action {
             type log-action;
             description
               "Action type: allow, alert,
               block, discard, declare,
               block-ip, block-service";
           }
           leaf botnet-pkt-num{
             type uint8;
             description
               "The number of the packets sent to
               or from the detected botnet";
           }
           leaf os{
             type string;
             description
               "simple os information";
           }
           uses characteristics;
           uses common-monitoring-data;
         }
         notification nsf-log-dpi {
           description
             "This notification is send, if there is
             a new dpi event in the nsf log";
           leaf attack-type {
             type dpi-type;
             description
               "The type of the dpi";
           }
           uses characteristics;
           uses i2nsf-nsf-counters-type-content;
           uses common-monitoring-data;
         }
         notification nsf-log-vuln-scan {
           description
             "This notification is send, if there is
             a new vulnerability-scan report in the nsf log";
```

```
        leaf vulnerability-id {
          type uint8;
          description
            "The vulnerability id";
        }
        leaf victim-ip {
          type inet:ipv4-address;
          description
            "IP address of the victim host
            which has vulnerabilities";
        }
        leaf protocol {
          type identityref {
            base protocol-type;
          }
          description
            "The protocol type for
            nsf-log-vuln-scan notification";
        }
        leaf port-num {
          type inet:port-number;
            description
              "The port number";
        }
        leaf level {
          type severity;
          description
            "The vulnerability severity";
        }
        leaf os {
          type string;
          description
            "simple os information";
        }
        leaf vulnerability-info {
          type string;
          description
            "The information about the vulnerability";
        }
        leaf fix-suggestion {
          type string;
          description
            "The fix suggestion to the vulnerability";
        }
        leaf service {
          type string;
          description
            "The service which has vulnerabillity in the victim host";
```

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

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

                    Figure 2: Data Model of Monitoring

11. IANA Considerations

   This document requests IANA to register the following URI in the
   "IETF XML Registry" [RFC3688]:

            URI: urn:ietf:params:xml:ns:yang:ietf-i2nsf-monitor
            Registrant Contact: The IESG.
            XML: N/A; the requested URI is an XML namespace.

   This document requests IANA to register the following YANG module in
   the "YANG Module Names" registry [RFC7950].

```
   name: ietf-i2nsf-monitor
         namespace: urn:ietf:params:xml:ns:yang:ietf-i2nsf-monitor
         prefix: iim
         reference: RFC XXXX
```

12. Security Considerations

   The YANG module described in this document defines a schema for data
   that is designed to be accessed via network management protocols such
   as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer
   is the secure transport layer, and the mandatory-to-implement secure
   transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer
   is HTTPS, and the mandatory-to-implement secure transport is TLS
   [RFC8446].

   The NETCONF access control model [RFC8341] provides the means to
   restrict access for particular NETCONF or RESTCONF users to a
   preconfigured subset of all available NETCONF or RESTCONF protocol
   operations and content.

   All data nodes defined in the YANG module which can be created,
   modified and deleted (i.e., config true, which is the default) are
   considered sensitive. Write operations (e.g., edit-config) applied
   to these data nodes without proper protection can negatively affect
   framework operations. The monitoring YANG module should be protected
   by the secure communication channel, to ensure its confidentiality
   and integrity. In another side, the NSF and security controller can
   all be faked, which lead to undesireable results, i.e., leakage of an
   NSF's important operational information, faked NSF sending false
   information to mislead security controller. The mutual
   authentication is essential to protected against this kind of attack.
   The current mainstream security technologies (i.e., TLS, DTLS, IPSEC,
   X.509 PKI) can be employed approriately to provide the above security
   functions.

   In addition, to defend against the DDoS attack caused by a lot of
   NSFs sending massive notifications to the security controller, the
   rate limiting or similar mechanisms should be considered in an NSF
   and security controller, whether in advance or just in the process of
   DDoS attack.

13. References

13.1. Normative References

   [I-D.ietf-netconf-subscribed-notifications]
             Voit, E., Clemm, A., Prieto, A., Nilsen-Nygaard, E., and
             A. Tripathy, "Subscription to YANG Event Notifications",
             draft-ietf-netconf-subscribed-notifications-23 (work in
             progress), February 2019.

   [I-D.ietf-netconf-yang-push]
             Clemm, A., Voit, E., Prieto, A., Tripathy, A., Nilsen-
             Nygaard, E., Bierman, A., and B. Lengyel, "Subscription to
             YANG Datastores", draft-ietf-netconf-yang-push-22 (work in
             progress), February 2019.

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
             Requirement Levels", BCP 14, RFC 2119,
             DOI 10.17487/RFC2119, March 1997,
             <https://www.rfc-editor.org/info/rfc2119>.

   [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688,
             DOI 10.17487/RFC3688, January 2004,
             <https://www.rfc-editor.org/info/rfc3688>.

   [RFC3877] Chisholm, S. and D. Romascanu, "Alarm Management
             Information Base (MIB)", RFC 3877, DOI 10.17487/RFC3877,
             September 2004, <https://www.rfc-editor.org/info/rfc3877>.

   [RFC4949] Shirey, R., "Internet Security Glossary, Version 2",
             FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007,
             <https://www.rfc-editor.org/info/rfc4949>.

   [RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424,
             DOI 10.17487/RFC5424, March 2009,
             <https://www.rfc-editor.org/info/rfc5424>.

   [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed.,
             and A. Bierman, Ed., "Network Configuration Protocol
             (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011,
             <https://www.rfc-editor.org/info/rfc6241>.

   [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure
             Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011,
             <https://www.rfc-editor.org/info/rfc6242>.

   [RFC6587] Gerhards, R. and C. Lonvick, "Transmission of Syslog
             Messages over TCP", RFC 6587, DOI 10.17487/RFC6587, April
             2012, <https://www.rfc-editor.org/info/rfc6587>.

   [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types",
              RFC 6991, DOI 10.17487/RFC6991, July 2013,
              <https://www.rfc-editor.org/info/rfc6991>.

   [RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken,
              "Specification of the IP Flow Information Export (IPFIX)
              Protocol for the Exchange of Flow Information", STD 77,
              RFC 7011, DOI 10.17487/RFC7011, September 2013,
              <https://www.rfc-editor.org/info/rfc7011>.

   [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language",
              RFC 7950, DOI 10.17487/RFC7950, August 2016,
              <https://www.rfc-editor.org/info/rfc7950>.

   [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF
              Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017,
              <https://www.rfc-editor.org/info/rfc8040>.

   [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
              2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
              May 2017, <https://www.rfc-editor.org/info/rfc8174>.

   [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration
              Access Control Model", STD 91, RFC 8341,
              DOI 10.17487/RFC8341, March 2018,
              <https://www.rfc-editor.org/info/rfc8341>.

   [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol
              Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018,
              <https://www.rfc-editor.org/info/rfc8446>.

13.2. Informative References

   [I-D.ietf-i2nsf-capability]
              Xia, L., Strassner, J., Basile, C., and D. Lopez,
              "Information Model of NSFs Capabilities", draft-ietf-
              i2nsf-capability-04 (work in progress), October 2018.

   [I-D.ietf-i2nsf-consumer-facing-interface-dm]
              Jeong, J., Kim, E., Ahn, T., Kumar, R., and S. Hares,
              "I2NSF Consumer-Facing Interface YANG Data Model", draft-
              ietf-i2nsf-consumer-facing-interface-dm-02 (work in
              progress), November 2018.

   [I-D.ietf-i2nsf-nsf-facing-interface-dm]
             Kim, J., Jeong, J., J., J., PARK, P., Hares, S., and Q.
             Lin, "I2NSF Network Security Function-Facing Interface
             YANG Data Model", draft-ietf-i2nsf-nsf-facing-interface-
             dm-02 (work in progress), November 2018.

   [I-D.ietf-i2nsf-registration-interface-dm]
             Hyun, S., Jeong, J., Roh, T., Wi, S., J., J., and P. PARK,
             "I2NSF Registration Interface YANG Data Model", draft-
             ietf-i2nsf-registration-interface-dm-01 (work in
             progress), November 2018.

   [I-D.ietf-i2nsf-terminology]
             Hares, S., Strassner, J., Lopez, D., Xia, L., and H.
             Birkholz, "Interface to Network Security Functions (I2NSF)
             Terminology", draft-ietf-i2nsf-terminology-07 (work in
             progress), January 2019.

   [I-D.yang-i2nsf-nfv-architecture]
             Yang, H., Kim, Y., Jeong, J., and J. Kim, "I2NSF on the
             NFV Reference Architecture", draft-yang-i2nsf-nfv-
             architecture-04 (work in progress), November 2018.

   [I-D.yang-i2nsf-security-policy-translation]
             Yang, J., Jeong, J., and J. Kim, "Security Policy
             Translation in Interface to Network Security Functions",
             draft-yang-i2nsf-security-policy-translation-02 (work in
             progress), October 2018.

   [RFC3954] Claise, B., Ed., "Cisco Systems NetFlow Services Export
             Version 9", RFC 3954, DOI 10.17487/RFC3954, October 2004,
             <https://www.rfc-editor.org/info/rfc3954>.

   [RFC6087] Bierman, A., "Guidelines for Authors and Reviewers of YANG
             Data Model Documents", RFC 6087, DOI 10.17487/RFC6087,
             January 2011, <https://www.rfc-editor.org/info/rfc6087>.

   [RFC8329] Lopez, D., Lopez, E., Dunbar, L., Strassner, J., and R.
             Kumar, "Framework for Interface to Network Security
             Functions", RFC 8329, DOI 10.17487/RFC8329, February 2018,
             <https://www.rfc-editor.org/info/rfc8329>.

   [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams",
             BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018,
             <https://www.rfc-editor.org/info/rfc8340>.

Appendix A. Changes from draft-hong-i2nsf-nsf-monitoring-data-model-06

   The following changes are made from draft-hong-i2nsf-nsf-monitoring-
   data-model-06:

   o This version has reflected the comments from Tom Petch as follows.

   o In Editorial Note, RFC XXXX: I2NSF NSF Monitoring YANG Data Model
      is mentioned.

   o In Section 2, Requirements Language and Terminology are integrated
      and the explain for YANG Data Diagrams is moved to Terminology.

   o In Section 2.3, NMDA conformance is mentioned.

   o In Section 2.1, the reference [RFC8174] is added.

   o In Section 2.3, the reference [RFC8340] that specifies the format
      for tree diagrams is added for the tree diagrams.

   o In Section 10, the copyright of the YANG Module is added in
      description.

   o In Section 10, the YANG import statements includes reference
      statements.

   o In Section 10, the YANG Module includes RFC XXX to notify the RFC
      from which it comes.

   o In Section 10, the the identity for protocols includes reference
      statements.

   o In Section 11, for the YANG Module Names and URI in the IETF XML
      Registry, the section is added.

   o In Section 12,

Appendix B. Acknowledgments

Appendix C. Contributors

   This document is made by the group effort of I2NSF working group.
   Many people actively contributed to this document. The following are
   considered co-authors:

   o Jinyong Tim Kim (Sungkyunkwan University)

   o Dongjin Hong (Sungkyunkwan University)

   o Dacheng Zhang (Huawei)

   o Yi Wu (Aliababa Group)

   o Rakesh Kumar (Juniper Networks)

   o Anil Lohiya (Juniper Networks)

Authors' Addresses

   Jaehoon Paul Jeong
   Department of Software
   Sungkyunkwan University
   2066 Seobu-Ro, Jangan-Gu
   Suwon, Gyeonggi-Do 16419
   Republic of Korea

   Phone: +82 31 299 4957
   Fax: +82 31 290 7996
   EMail: pauljeong@skku.edu
   URI: http://iotlab.skku.edu/people-jaehoon-jeong.php

   Chaehong Chung
   Department of Computer Engineering
   Sungkyunkwan University
   2066 Seobu-Ro, Jangan-Gu
   Suwon, Gyeonggi-Do 16419
   Republic of Korea

   Phone: +82 10 8541 7158
   EMail: darkhong@skku.edu

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
USA

Phone: +1-734-604-0332
EMail: shares@ndzh.com

Liang Xia (Frank)
Huawei
101 Software Avenue, Yuhuatai District
Nanjing, Jiangsu
China

EMail: Frank.xialiang@huawei.com

Henk Birkholz
Fraunhofer Institute for Secure Information Technology
Rheinstrasse 75
Darmstadt 64295
Germany

EMail: henk.birkholz@sit.fraunhofer.de