

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 25, 2019

D. Hong
J. Jeong
J. Kim
Sungkyunkwan University
S. Hares
L. Xia
Huawei
H. Birkholz
Fraunhofer SIT
October 22, 2018

YANG Data Model for Monitoring I2NSF Network Security Functions
draft-hong-i2nsf-nsf-monitoring-data-model-05

Abstract

This document proposes a 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 a data tree to specify an information model for monitoring NSFs, but also the corresponding YANG data model for monitoring NSFs.

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 April 25, 2019.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Requirements Language	2
3. Terminology	3
3.1. Tree Diagrams	3
4. NSF Monitoring Management	3
5. Information Model Structure	4
6. YANG Data Model	12
7. Acknowledgments	46
8. References	46
8.1. Normative References	46
8.2. Informative References	46
Appendix A. Changes from draft-hong-i2nsf-nsf-monitoring-data-model-04	48
Authors' Addresses	48

1. Introduction

This document defines a YANG [RFC6020] data model for monitoring Network Security Functions (NSFs). The monitoring means the acquisition of vital information about NSFs via notifications, records or counters. The data model for the monitoring presented in this document is derived from the information model for monitoring NSFs through the NSF-Facing Interface specified in [i2nsf-monitoring-im].

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Terminology

This document uses the terminology described in [i2nsf-terminology][i2nsf-framework]. Especially, the following terms are from [i2nsf-monitoring-im].

- o Information Model: An information model is a representation of concepts of interest to an environment in a form that is independent of data repository, data definition language, query language, implementation language, and protocol.
- o Data Model: A data model is a representation of concepts of interest to an environment in a form that is dependent on data repository, data definition language, query language, implementation language, and protocol.

3.1. Tree Diagrams

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in these diagrams [i2rs-rib-data-model] is as follows:

- o Brackets "[" and "]" enclose list keys.
- o Abbreviations before data node names: "rw" means configuration (read-write) and "ro" state data (read-only).
- o Symbols after data node names: "?" means an optional node and "*" denotes a "list" and "leaf-list".
- o Parentheses enclose choice and case nodes, and case nodes are alsoconfigure marked with a colon (":").
- o Ellipsis ("...") stands for contents of subtrees that are not shown.

4. NSF Monitoring Management

A standard model for monitoring data is required for the administrator to check the monitoring data generated by 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 so that the administrator can know the state of the framework. In order to communicate with other component, it requires the interfaces. The three main interfaces in I2NSF framework are used for sending monitoring data like as follows:

- o I2NSF Consumer-Facing Interface : When an I2NSF user makes a security policy and forwards it to the security controller through the 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 NSF.
- o I2NSF Registration Interface : The NFV architecture provides VNF lifecycle management through the Ve-Vnfm interface. The role of Ve-Vnfm is to request VNF lifecycle management, exchange configuration information, and exchange status information for network service lifecycle management. For I2NSF framework case, the DMS manages data such as resource state and so on for lifecycle of NSF. Therefore the generated monitoring data need to be delivered to Vn-Vnfm via DMS in the environment that can support VNF.
- o I2NSF NSF-Facing Interface : After the policy is translated by security policy translator in security controller, the translated security policy(low-level) is applied to NSF via NSF-Facing Interface. The monitoring data model specifies the list of event that can trigger the ECA(Event-Condition-Action) policy at NSF-Facing Interface.

5. Information Model Structure

Figure 1 shows the overview of a structure tree of monitoring information based on the [i2nsf-monitoring-im].

```

module: ietf-i2nsf-nsf-monitoring-dm
  +--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 additional-info?          string
+---n nsf-log-ddos
|   +--ro attack-type?             identityref
|   +--ro attack-ave-rate?         uint32
|   +--ro attack-ave-speed?       uint32
|   +--ro attack-pkt-num?         uint32
|   +--ro attack-src-ip?          inet:ipv4-address
|   +--ro action?                  log-action
|   +--ro acquisition-method?     identityref
|   +--ro emission-type?          identityref
|   +--ro dampening-type?         identityref
|   +--ro message?                 string
|   +--ro time-stamp?             yang:date-and-time
|   +--ro vendor-name?            string
|   +--ro nsf-name?               string
|   +--ro module-name?            string
|   +--ro severity?               severity
+---n nsf-log-virus
|   +--ro attack-type?             identityref
|   +--ro action?                  log-action
|   +--ro os?                      string
|   +--ro time                     yang:date-and-time
|   +--ro acquisition-method?     identityref
|   +--ro emission-type?          identityref
|   +--ro dampening-type?         identityref
|   +--ro message?                 string
|   +--ro time-stamp?             yang:date-and-time
|   +--ro vendor-name?            string
|   +--ro nsf-name?               string
|   +--ro module-name?            string
|   +--ro severity?               severity
+---n nsf-log-intrusion
|   +--ro attack-type?             identityref
|   +--ro action?                  log-action
|   +--ro time                     yang:date-and-time
|   +--ro attack-rate?            uint32
|   +--ro attack-speed?           uint32
|   +--ro acquisition-method?     identityref
|   +--ro emission-type?          identityref
|   +--ro dampening-type?         identityref
|   +--ro message?                 string
|   +--ro time-stamp?             yang:date-and-time
|   +--ro vendor-name?            string
|   +--ro nsf-name?               string
|   +--ro module-name?            string
|   +--ro severity?               severity
+---n nsf-log-botnet
|   +--ro attack-type?             identityref
```

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

```

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

```

Figure 1: Information Model for NSF Monitoring

6. YANG Data Model

This section introduces a YANG data model for the information model of monitoring information based on [i2nsf-monitoring-im].

```

<CODE BEGINS> file "ietf-i2nsf-nsf-monitoring-dm@2018-10-22.yang"
module ietf-i2nsf-nsf-monitoring-dm {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-i2nsf-nsf-monitoring-dm";
  prefix
    monitoring-information;
  import ietf-inet-types {
    prefix inet;
  }
  import ietf-yang-types {
    prefix yang;
  }
  organization
    "IETF I2NSF (Interface to Network Security Functions)

```

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

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

  Editor: Dongjin Hong
  <mailto:dong.jin@skku.edu>

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

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

revision "2018-07-02" {
  description "Fifth revision";
  reference
    "draft-zhang-i2nsf-info-model-monitoring-06";
}

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 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.";
}
identity udp {
    base ipv4;
```

```
    base ipv6;
    description
      "UDP protocol type.";
  }
  identity icmp {
    base ipv4;
    base ipv6;
    description
      "General ICMP protocol type.";
  }
  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.";
  }
  identity ipv4 {
    base ip;
    description
      "IPv4 protocol type.";
  }
  identity ipv6 {
    base ip;
    description
      "IPv6 protocol type.";
  }
  identity http {
    base tcp;
    description
      "HTTP protocol type.";
  }
  identity ftp {
    base tcp;
    description
      "FTP protocol type.";
  }
  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 applicable).";
    }
    leaf file-name {
      type string;
      description
        "The name of file virus code is found in (if applicable).";
    }
    uses common-monitoring-data;
  }
notification nsf-detection-intrusion {
  description
    "This notification is 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 outbound traffic rate in pps";
    }
    leaf in-traffic-speed {
        type uint32;
        description
            "The total inbound traffic speed in bps";
    }
    leaf out-traffic-speed {
        type uint32;
        description
            "The total outbound traffic speed in bps";
    }
    uses characteristics;
}
notification system-user-activity-log {
    description
        "This notification is 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 additional-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 notification 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

7. Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (R-20160222-002755, Cloud based Security Intelligence Technology Development for the Customized Security Service Provisioning).

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

8.2. Informative References

[i2nsf-cunsumer-facing-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-01 (work in
progress), July 2018.

[i2nsf-framework]

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

[i2nsf-monitoring-im]

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

[i2nsf-nsf-facing-dm]

Kim,, J., Jeong,, J., Park, J., Hares, S., and Q. Lin,
"I2NSF Network Security Function-Facing Interface YANG
Data Model", draft-ietf-i2nsf-nsf-facing-interface-dm-01
(work in progress), July 2018.

[i2nsf-registration-im]

Hyun,, S., Jeong,, J., Roh, T., Wi, S., and J. Park,
"Registration Interface Information Model", draft-hyun-
i2nsf-registration-interface-im-06 (work in progress),
July 2018.

[i2nsf-terminology]

Hares,, S., Strassner,, J., Lopez,, D., Xia,, L., and H.
Birkholz,, "Interface to Network Security Functions
(I2NSF) Terminology", draft-ietf-i2nsf-terminology-05
(work in progress), July 2018.

[i2rs-rib-data-model]

Wang, L., Chen, M., Dass, A., Ananthakrishnan, H., Kini,
S., and N. Bahadur, "A YANG Data Model for Routing
Information Base (RIB)", draft-ietf-i2rs-rib-data-model-10
(work in progress), February 2018.

Appendix A. Changes from draft-hong-i2nsf-nsf-monitoring-data-model-04

The following changes are made from draft-hong-i2nsf-nsf-monitoring-data-model-04:

1. The association with the three main interfaces of the I2NSF framework is described in Section 3.
2. Typos are corrected.

Authors' Addresses

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

Phone: +82 10 7630 5473
EMail: dong.jin@skku.edu

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>

Jinyong Tim Kim
Department of Computer Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon, Gyeonggi-Do 16419
Republic of Korea

Phone: +82 10 8273 0930
EMail: timkim@skku.edu

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
USA

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