Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2018

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

March 5, 2018

YANG Data Model for Monitoring I2NSF Network Security Functions
draft-hong-i2nsf-nsf-monitoring-data-model-02

Abstract

This document proposes a YANG data model for monitoring Network Security Functions (NSFs) in the Interface to Network Security Functions (I2NSF) system. If the monitoring of NSFs is performed in a comrehensive 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

Copyright Notice

Table of Contents

1. Introduction

This document defines a YANG [RFC6020] data model for monitoring Network Security Functions (NSFs). This monitoring means the aquisition of vital information about NSFs via notifications, events, 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].

- 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.

- 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:

- Brackets "[" and "]" enclose list keys.

- Abbreviations before data node names: "rw" means configuration (read-write) and "ro" state data (read-only).

- Symbols after data node names: "?" means an optional node and "*" denotes a "list" and "leaf-list".

- Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").

- Ellipsis ("...") stands for contents of subtrees that are not shown.

4. 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 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 nsf-firewall
         |  +--rw src-ip?                   inet:ipv4-address
         |  +--rw dst-ip?                   inet:ipv4-address
         |  +--rw src-port?                 inet:port-number
         |  +--rw dst-port?                 inet:port-number
         |  +--rw src-zone?                 string
         |  +--rw dst-zone?                 string
         |  +--rw src-region?               string
         |  +--rw dst-region?               string
         |  +--rw policy-id                 uint8
         |  +--rw policy-name               string
         |  +--rw src-user?                 string
         |  +--rw protocol
         |  |  +--rw tcp?      boolean
         |  |  +--rw udp?      boolean
         |  |  +--rw icmp?     boolean
         |  |  +--rw icmpv6?   boolean
         |  |  +--rw ip?       boolean
         |  |  +--rw http?     boolean
         |  |  +--rw ftp?      boolean
         |  +--rw total-traffic?            uint32
         |  +--rw in-traffic-ave-rate?      uint32
         |  +--rw in-traffic-peak-rate?     uint32
         |  +--rw in-traffic-ave-speed?     uint32
         |  +--rw in-traffic-peak-speed?    uint32
         |  +--rw out-traffic-ave-rate?     uint32
         |  +--rw out-traffic-peak-rate?    uint32
         |  +--rw out-traffic-ave-speed?    uint32
         |  +--rw out-traffic-peak-speed?   uint32
         |  +--rw diretcions
         |     +--rw in-interface?    boolean
         |     +--rw out-interface?   boolean
         +--rw nsf-policy-hits
            +--rw src-ip?                   inet:ipv4-address
            +--rw dst-ip?                     inet:ipv4-address
            +--rw src-port?                   inet:port-number
            +--rw dst-port?                   inet:port-number
            +--rw src-zone?                   string
            +--rw dst-zone?                   string
            +--rw src-region?                 string
            +--rw dst-region?                 string
            +--rw policy-id                   uint8
            +--rw policy-name                 string
            +--rw src-user?                   string
            +--rw protocol
            |  +--rw tcp?      boolean
            |  +--rw udp?      boolean
            |  +--rw icmp?     boolean
            |  +--rw icmpv6?   boolean
            |  +--rw ip?       boolean
            |  +--rw http?     boolean
            |  +--rw ftp?      boolean
            +--rw total-traffic?              uint32
            +--rw in-traffic-ave-rate?        uint32
            +--rw in-traffic-peak-rate?       uint32
            +--rw in-traffic-ave-speed?       uint32
            +--rw in-traffic-peak-speed?      uint32
            +--rw out-traffic-ave-rate?       uint32
            +--rw out-traffic-peak-rate?      uint32
            +--rw out-traffic-ave-speed?      uint32
            +--rw out-traffic-peak-speed?     uint32
            +--rw hit-times?                  uint32
   notifications:
      +---n system-detection-alarm
      |  +--ro alarm-catagory?   identityref
      |  +--ro usage?            uint8
      |  +--ro threshold?        uint8
      |  +--ro message           string
      |  +--ro time-stamp        yang:date-and-time
      |  +--ro severity          severity
      +---n system-detection-access-violation
      |  +--ro group             string
      |  +--ro login-ip-addr     inet:ipv4-address
      |  +--ro authentication?   identityref
      |  +--ro message           string
      |  +--ro time-stamp        yang:date-and-time
      |  +--ro severity          severity
      +---n system-detection-config-change
      |  +--ro group             string
      |  +--ro login-ip-addr     inet:ipv4-address
      |  +--ro authentication?   identityref
      |  +--ro message           string
      |  +--ro time-stamp        yang:date-and-time
      |  +--ro severity          severity
      +---n nsf-detection-flood
      |  +--ro event-message?    string
      |  +--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 flood-catagory?   identityref
      |  +--ro start-time        yang:date-and-time
      |  +--ro end-time          yang:date-and-time
      |  +--ro attack-rate?      uint32
      |  +--ro attack-speed?     uint32
      |  +--ro vendor-name?      string
      |  +--ro nsf-name?         string
      |  +--ro message           string
      |  +--ro time-stamp        yang:date-and-time
      |  +--ro severity          severity
      +---n nsf-detection-session-table
      |  +--ro current-session?     uint8
      |  +--ro maximum-session?     uint8
      |  +--ro threshold?           uint8
      |  +--ro table-indentifier?   string
      |  +--ro vendor-name?         string
      |  +--ro nsf-name?            string
      |  +--ro message              string
      |  +--ro time-stamp           yang:date-and-time
      |  +--ro severity             severity
      +---n nsf-detection-virus
      |  +--ro event-message?   string
      |  +--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 vendor-name?     string
      |  +--ro nsf-name?        string
      |  +--ro message          string
      |  +--ro time-stamp       yang:date-and-time
      |  +--ro severity         severity
      +---n nsf-detection-intrusion
      |  +--ro event-message?   string
      |  +--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
      |  |  +--ro tcp?      boolean
      |  |  +--ro udp?      boolean
      |  |  +--ro icmp?     boolean
      |  |  +--ro icmpv6?   boolean
      |  |  +--ro ip?       boolean
      |  |  +--ro http?     boolean
      |  |  +--ro ftp?      boolean
      |  +--ro intrusion?       identityref
      |  +--ro vendor-name?     string
      |  +--ro nsf-name?        string
      |  +--ro message          string
      |  +--ro time-stamp       yang:date-and-time
      |  +--ro severity         severity
      +---n nsf-detection-botnet
      |  +--ro event-message?   string
      |  +--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
      |  |  +--ro tcp?      boolean
      |  |  +--ro udp?      boolean
      |  |  +--ro icmp?     boolean
      |  |  +--ro icmpv6?   boolean
      |  |  +--ro ip?       boolean
      |  |  +--ro http?     boolean
      |  |  +--ro ftp?      boolean
      |  +--ro botnet-name?     string
      |  +--ro role?            string
      |  +--ro vendor-name?     string
      |  +--ro nsf-name?        string
      |  +--ro message          string
      |  +--ro time-stamp       yang:date-and-time
      |  +--ro severity         severity
      +---n nsf-detection-web-attack
      |  +--ro event-message?   string
      |  +--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 web-attack?      identityref
      |  +--ro protocol
      |  |  +--ro tcp?      boolean
      |  |  +--ro udp?      boolean
      |  |  +--ro icmp?     boolean
      |  |  +--ro icmpv6?   boolean
      |  |  +--ro ip?       boolean
      |  |  +--ro http?     boolean
      |  |  +--ro ftp?      boolean
      |  +--ro request?         identityref
      |  +--ro req-uri?         string
      |  +--ro uri-category?    string
      |  +--ro filter*          identityref
      |  +--ro vendor-name?     string
      |  +--ro nsf-name?        string
      |  +--ro message          string
      |  +--ro time-stamp       yang:date-and-time
      |  +--ro severity         severity
      +---n system-log-access-event
      |  +--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 vendor-name?      string
     |  +--ro nsf-name?         string
     +---n system-log-res-util-report
     |  +--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 vendor-name?         string
     |  +--ro nsf-name?            string
     +---n system-log-user-activity-event
     |  +--ro user                string
     |  +--ro group               string
     |  +--ro login-ip            inet:ipv4-address
     |  +--ro authentication?     identityref
     |  +--ro accese?             identityref
     |  +--ro online-duration?    string
     |  +--ro logout-duration?    string
     |  +--ro addtional-info?     string
     |  +--ro vendor-name?        string
     |  +--ro nsf-name?           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 os?                  string
     |  +--ro vendor-name?         string
     |  +--ro nsf-name?            string
     |  +--ro message              string
     |  +--ro time-stamp           yang:date-and-time
     |  +--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 vendor-name?    string
```

```
      |  +--ro nsf-name?      string
      |  +--ro message        string
      |  +--ro time-stamp     yang:date-and-time
      |  +--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 vendor-name?    string
      |  +--ro nsf-name?       string
      |  +--ro message         string
      |  +--ro time-stamp      yang:date-and-time
      |  +--ro severity        severity
      +---n nsf-log-botnet
      |  +--ro attack-type?      identityref
      |  +--ro action?           log-action
      |  +--ro botnet-pkt-num?   uint8
      |  +--ro os?               string
      |  +--ro vendor-name?      string
      |  +--ro nsf-name?         string
      |  +--ro message           string
      |  +--ro time-stamp        yang:date-and-time
      |  +--ro severity          severity
      +---n nsf-log-dpi
      |  +--ro attack-type?   dpi-type
      |  +--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 vendor-name?   string
      |  +--ro nsf-name?      string
      |  +--ro message        string
      |  +--ro time-stamp     yang:date-and-time
      |  +--ro severity       severity
      +---n nsf-log-vuln-scan
      |  +--ro vulnerability-id?   uint8
      |  +--ro victim-ip?          inet:ipv4-address
      |  +--ro protocol
      |  |  +--ro tcp?      boolean
      |  |  +--ro udp?      boolean
      |  |  +--ro icmp?     boolean
      |  |  +--ro icmpv6?   boolean
      |  |  +--ro ip?       boolean
      |  |  +--ro http?     boolean
      |  |  +--ro ftp?      boolean
      |  +--ro port-num?           inet:port-number
      |  +--ro level?              severity
      |  +--ro os?                 string
      |  +--ro addtional-info?     string
      |  +--ro vendor-name?        string
      |  +--ro nsf-name?           string
      |  +--ro message             string
      |  +--ro time-stamp          yang:date-and-time
      |  +--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 vendor-name?     string
         +--ro nsf-name?        string
         +--ro message          string
         +--ro time-stamp       yang:date-and-time
         +--ro severity         severity
```

Figure 1: Information Model for NSF Monitoring

5. YANG Data Model

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

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

module ietf-i2nsf-nsf-monitoring-dm {
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-i2nsf-nsf-monitoring-dm";
  prefix
    monitoring-information;
  import ietf-inet-types{
    prefix inet;
  }
  import ietf-yang-types {
    prefix yang;
  }
  organization
    "IETF I2NSF (Interface to Network Security Functions)
     Working Group";

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

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

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

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

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

  revision "2018-03-05" {
    description "Third revision";
    reference
      "draft-zhang-i2nsf-info-model-monitoring-05";
  }

  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 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 memory-alarm {
       base alarm-type;
       description
         "A memory alarm is alerted";
     }
     identity cpu-alarm {
       base alarm-type;
       description
         "A cpu alarm is alerted";
     }
     identity disk-alarm {
       base alarm-type;
       description
         "A disk alarm is alerted";
     }
     identity hardware-alarm {
       base alarm-type;
       description
         "A hardware alarm is alerted";
     }
     identity interface-alarm {
       base alarm-type;
       description
         "An interface alarm is alerted";
     }
     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 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/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
         "TBD";
     }
     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";
     }

     grouping protocol {
       description
         "A set of protocols";
       container protocol {
         description
         "Protocol types:
          TCP, UDP, ICMP, ICMPv6, IP, HTTP, FTP and etc.";
         leaf tcp {
           type boolean;
           description
             "TCP protocol type.";
         }
         leaf udp {
           type boolean;
           description
             "UDP protocol type.";
         }
         leaf icmp {
           type boolean;
           description
```

```
             "ICMP protocol type.";
         }
         leaf icmpv6 {
           type boolean;
           description
             "ICMPv6 protocol type.";
         }
         leaf ip {
           type boolean;
           description
             "IP protocol type.";
         }
         leaf http {
           type boolean;
           description
             "HTTP protocol type.";
         }
         leaf ftp {
           type boolean;
           description
             "ftp protocol type.";
         }
       }
     }
     grouping common-notification-content {
       description
       "TBD";
       leaf message {
         type string;
         mandatory true;
         description
           "This is a freetext annotation of
           monitoring notification content";
       }
       leaf time-stamp {
         type yang:date-and-time;
         mandatory true;
         description
           "Indicates the time of message generation";
       }
       leaf severity {
         type severity;
         mandatory true;
         description
           "The severity of the alarm such
           asvcritical, high, middle, low.";
       }
     }
     grouping common-nsf-notification-content {
       description
       "TBD";
       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";
       }
     }
     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 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
           "TBD";
       }
     }

     grouping i2nsf-nsf-event-type-content {
       description
         "A set of common IPv4-related NSF event
         content elements";
       leaf event-message {
         type string;
         description
           "The message for nsf events";
       }
       leaf src-ip {
         type inet:ipv4-address;
         description
           "The source IP address of the packet";
       }
       leaf dst-ip {
         type inet:ipv4-address;
         description
           "The destination IP address of the packet";
       }
       leaf src-port {
         type inet:port-number;
         description
           "The source port of the packet";
       }
       leaf dst-port {
         type inet:port-number;
         description
           "The destination port of the packet";
       }
       leaf src-zone {
         type string;
         description
           "The source security zone of the packet";
       }
       leaf dst-zone {
         type string;
         description
           "The destination security zone of the packet";
       }
       leaf rule-id {
         type uint8;
         mandatory true;
         description
           "The ID of the rule being triggered";
       }
       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";
       }
       uses protocol;
       uses traffic-rates;
     }

     notification system-detection-alarm {
       description
         "TBD";
       leaf alarm-catagory {
         type identityref {
           base alarm-type;
         }
         description
           "TBD";
       }
       uses i2nsf-system-alarm-type-content;
       uses common-notification-content;
     }
     notification system-detection-access-violation {
       description
         "This notification is sent, when a security-sensitive
          authentication action fails.";
       uses i2nsf-system-event-type-content;
       uses common-notification-content;
     }
     notification system-detection-config-change {
       description
         "This notification is sent,
         when an unauthorized confinguration
         change action is detected.";
       uses i2nsf-system-event-type-content;
       uses common-notification-content;
     }
     notification nsf-detection-flood {
       description
         "This notification is sent,
         when a specific flood type is detected";
       uses i2nsf-nsf-event-type-content;
       leaf flood-catagory {
         type identityref {
           base flood-type;
         }
         description
           "TBD";
       }
       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-nsf-notification-content;
       uses common-notification-content;
     }
     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";
       }
       leaf table-indentifier {
         type string;
         description
           "The number of session table exceeded the threshold";
       }
       uses common-nsf-notification-content;
       uses common-notification-content;
     }
     notification nsf-detection-virus {
       description
         "This notification is sent, when a virus is detected";
       uses i2nsf-nsf-event-type-content;
       leaf virus {
         type identityref {
           base virus-type;
         }
         description
           "TBD";
       }
       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-nsf-notification-content;
       uses common-notification-content;
     }
     notification nsf-detection-intrusion {
       description
         "This notification is send, when an intrusion event
          is detected.";
       uses i2nsf-nsf-event-type-content;
       uses protocol;
       leaf intrusion {
         type identityref {
           base intrusion-attack-type;
         }
         description
           "TBD";
       }
       uses common-nsf-notification-content;
       uses common-notification-content;
     }
     notification nsf-detection-botnet {
       description
         "This notification is send, when a botnet event is
          detected";
       uses i2nsf-nsf-event-type-content;
       leaf attack-type {
         type identityref {
           base botnet-attack-type;
         }
         description
           "TBD";
       }
       uses protocol;
       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-nsf-notification-content;
       uses common-notification-content;
     }
     notification nsf-detection-web-attack {
       description
         "This notification is send, when an attack event is
          detected";
       uses i2nsf-nsf-event-type-content;
       leaf web-attack {
         type identityref {
           base web-attack-type;
         }
         description
           "TBD";
       }
       uses protocol;
       leaf request {
         type identityref {
           base req-method;
         }
         description
           "TBD";
       }
       leaf req-uri {
         type string;
         description
         "Requested URI";
       }
       leaf uri-category {
         type string;
         description
         "Matched URI category";
       }
       leaf-list filter {
         type identityref {
           base filter-type;
         }
         description
           "TBD";
       }
       uses common-nsf-notification-content;
       uses common-notification-content;
     }
     notification system-log-access-event {
       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 common-nsf-notification-content;
     }
     notification system-log-res-util-report {
       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 common-nsf-notification-content;
     }
     notification system-log-user-activity-event {
       description
         "This notification is send, if there is
         a new user activity log entry";
       leaf user {
         type string;
         mandatory true;
         description
           "Name of a user";
       }
       leaf group {
         type string;
         mandatory true;
         description
           "Group to which a user belongs.";
       }
       leaf login-ip {
         type inet:ipv4-address;
         mandatory true;
         description
           "Login IP address of a user.";
       }
       leaf authentication {
         type identityref {
           base authentication-mode;
         }
         description
           "TBD";
       }
       leaf accese {
         type identityref {
           base access-mode;
         }
         description
           "TBD";
       }
       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";
       }
       uses common-nsf-notification-content;
     }
     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
           "TBD";
       }
       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";
        }
        leaf os {
          type string;
          description
            "simple os information";
        }
        uses common-nsf-notification-content;
        uses common-notification-content;
      }
      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
            "TBD";
        }
        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 common-nsf-notification-content;
       uses common-notification-content;
     }
     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
           "TBD";
       }
       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 common-nsf-notification-content;
       uses common-notification-content;
     }
     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
           "TBD";
       }
       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 common-nsf-notification-content;
       uses common-notification-content;
     }
     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";
       }
       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;
         mandatory true;
         description
           "The ID of the policy being triggered";
       }
       leaf policy-name {
         type string;
         mandatory true;
         description
           "The name of the policy being triggered";
       }
       leaf src-user{
         type string;
         description
           "User who generates traffic";
       }
       uses common-nsf-notification-content;
       uses common-notification-content;
     }
     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";
       }
       uses protocol;
       leaf port-num {
         type inet:port-number;
           description
             "The port number";
       }
       leaf level {
         type severity;
         description
           "The vulnerability severity";
       }
       leaf os {
         type string;
         description
           "simple os information";
       }
       leaf addtional-info {
         type string;
         description
           "TBD";
       }
       uses common-nsf-notification-content;
       uses common-notification-content;
     }
     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
           "TBD";
       }
       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 common-nsf-notification-content;
       uses common-notification-content;
     }
     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 i2nsf-system-counter-type-content;
       }
       container nsf-firewall {
         description
           "The nsf counter type is firewall counter";
         uses i2nsf-nsf-counters-type-content;
         container diretcions {
           description
```

```
             "Inbound or Outbound";
           leaf in-interface {
             type boolean;
             description
               "If the bound is inbound";
           }
           leaf out-interface {
             type boolean;
             description
               "If the bound is outbound";
           }
         }
       }
       container nsf-policy-hits {
         description
           "The counters of policy hit";
         uses i2nsf-nsf-counters-type-content;
         leaf hit-times {
           type uint32;
           description
             "The hit times for policy";
         }
       }
     }
   }
   <CODE ENDS>
```

Figure 2: Data Model of Monitoring

6. 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).

This document has greatly benefited from inputs by Daeyoung Hyun.

7. References

7.1. 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.

7.2. Informative References

[i2nsf-framework]
Lopez,, D., Lopez,, E., Dunbar, L., Strassner, J., and R. Kumar, "Framework for Interface to Network Security Functions", draft-ietf-i2nsf-framework-10 (work in progress), November 2017.

[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-05 (work in progress), October 2017.

[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), October 2017.

[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. draft-hong-i2nsf-nsf-monitoring-data-model-02

The following changes are made from draft-hong-i2nsf-nsf-monitoring-data-model-01:

1. The YANG data model is defined in more detail based on the information model for monitoring NSFs.

2. Some of descriptions for YANG data model are revised.

3. Typos and grammatical errors 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@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