Revision Letter

Editor: Jaehoon Paul Jeong
Date: March 31, 2021

OLD: draft-ietf-i2nsf-nsf-monitoring-data-model-06
NEW: draft-ietf-i2nsf-nsf-monitoring-data-model-07

Dear Andy Bierman,

I sincerely appreciate your detailed comments to improve the YANG module of our I2NSF NSF Monitoring YANG Data Model Draft.
I have addressed the pyang reporting errors. I use a bold font for your comments and use a regular font for my responses with a prefix “=> [PAUL]”.

Assignment Reviewer: Andy Bierman
State Completed
Review
review-ietf-i2nsf-nsf-monitoring-data-model-06-yangdoctors-lc-bierman-2021-03-23
Posted at
https://mailarchive.ietf.org/arch/msg/yang-doctors/KsSYxOnUcIY8CjHtm9xhAbOMeSc
Reviewed rev. 06
Review result Ready with Issues
Review completed: 2021-03-23

Status: Ready with Issues

Most of the issues raised in the review of draft-04 have been addressed.

Major Issues:

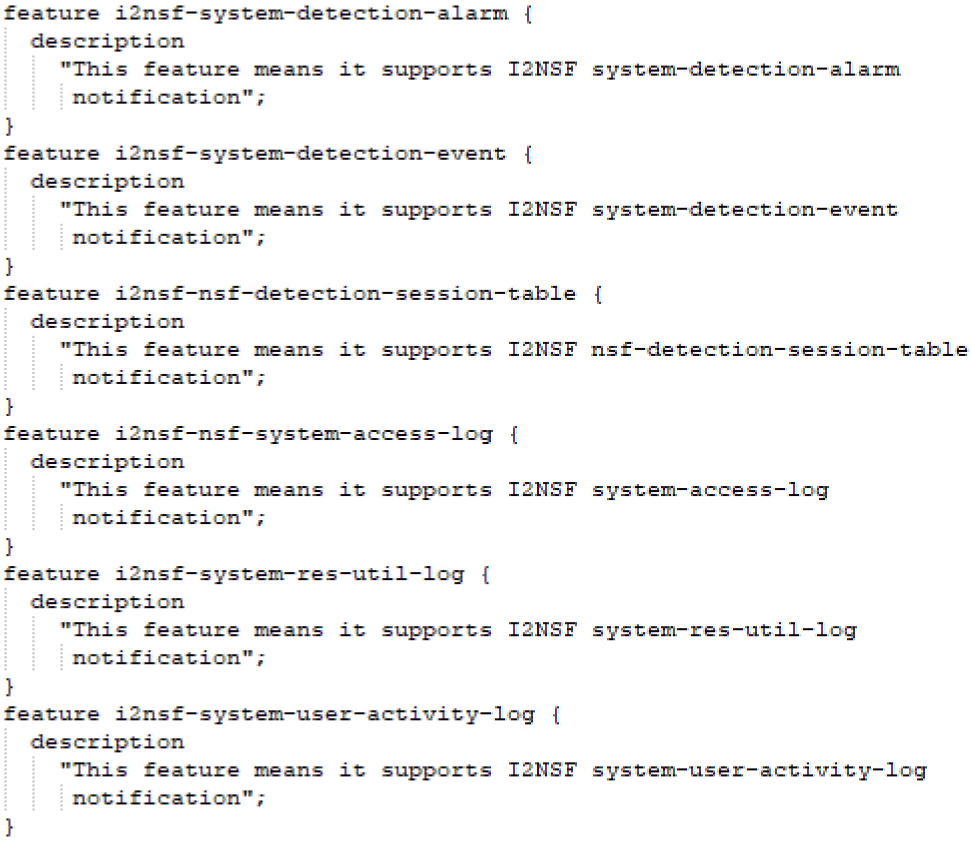
- None

Moderate Issues:

1) too many YANG features

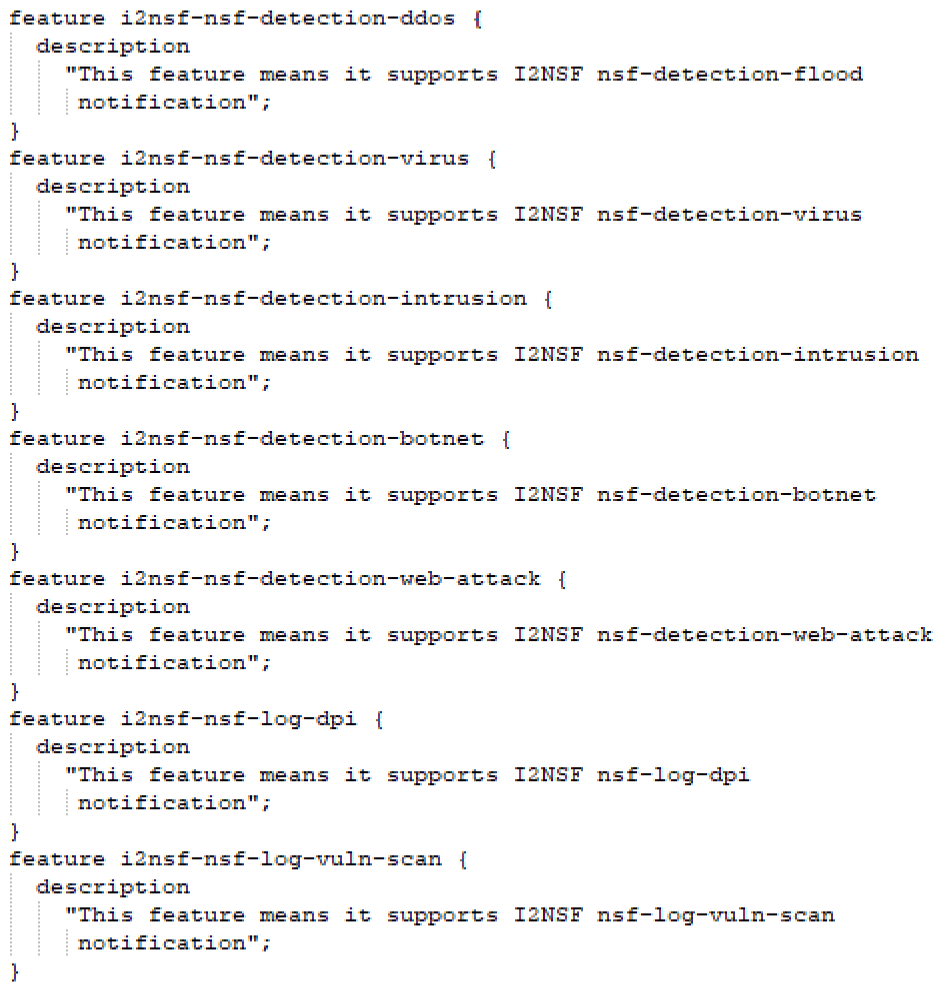
There are 13 YANG features, one for each of the 13 notification-stmts defined. There should be as few YANG features defined as possible. They should only be used if it is an unreasonable burden (compared to the feature value) for all servers to support the functionality.

=> [PAUL] We reduced the number of YANG features in the module from 13 features to 7 features.
The removed features are supposed to be supported on every NSF (server) as it holds basic information for monitoring the server. The removed features are as the following:



```
feature i2nsf-system-detection-alarm {
  description
    "This feature means it supports I2NSF system-detection-alarm
     notification";
}
feature i2nsf-system-detection-event {
  description
    "This feature means it supports I2NSF system-detection-event
     notification";
}
feature i2nsf-nsf-detection-session-table {
  description
    "This feature means it supports I2NSF nsf-detection-session-table
     notification";
}
feature i2nsf-nsf-system-access-log {
  description
    "This feature means it supports I2NSF system-access-log
     notification";
}
feature i2nsf-system-res-util-log {
  description
    "This feature means it supports I2NSF system-res-util-log
     notification";
}
feature i2nsf-system-user-activity-log {
  description
    "This feature means it supports I2NSF system-user-activity-log
     notification";
}
```

The remaining features are only implemented on a specific NSF (server) that supports such security-specific features. Hence, we leave the remaining features on the data model. The remaining features are shown below:



```
feature i2nsf-nsf-detection-ddos {
  description
    "This feature means it supports I2NSF nsf-detection-flood
     notification";
}
feature i2nsf-nsf-detection-virus {
  description
    "This feature means it supports I2NSF nsf-detection-virus
     notification";
}
feature i2nsf-nsf-detection-intrusion {
  description
    "This feature means it supports I2NSF nsf-detection-intrusion
     notification";
}
feature i2nsf-nsf-detection-botnet {
  description
    "This feature means it supports I2NSF nsf-detection-botnet
     notification";
}
feature i2nsf-nsf-detection-web-attack {
  description
    "This feature means it supports I2NSF nsf-detection-web-attack
     notification";
}
feature i2nsf-nsf-log-dpi {
  description
    "This feature means it supports I2NSF nsf-log-dpi
     notification";
}
feature i2nsf-nsf-log-vuln-scan {
  description
    "This feature means it supports I2NSF nsf-log-vuln-scan
     notification";
}
```

2) list /i2nsf-monitoring-configuration/system-alarm

**This is yet another alarm management system created in the IETF.
I guess the WG decided that RFC 8632 was not suitable.**

It is not clear how this system prevents excessive notifications sent to a client.

What happens when the CPU, memory, or disk usage crosses back and forth over the threshold? Seems like an alarm is generated for each upward crossing of the threshold leaf.

The precise behavior for triggering and then re-arming an alarm needs to be specified in the YANG module.

RMON Alarms (RFC 2819) defines one way to prevent bursts of SNMP notifications, using an alarm reset threshold.

YANG Push (RFC 8641) uses a dampening-period approach to prevent flooding the receiver with notifications.

Also, it is not clear what use-case is served by "threshold = 0".

The range is 0..100 instead of 1. .100.

=> [PAUL] In the document, a dampening type is defined to mitigate the impact of repetitive notifications. We added a new data model for configuring the dampening period.

NEW

```
  grouping dampening {
    description
      "A grouping for dampening period of notification.";
    leaf dampening-period {
      type uint32;
      units "centiseconds";
      default "0";
      description
        "Specifies the minimum interval between the assembly of
         successive update records for a single receiver of a
         subscription.  Whenever subscribed objects change and
         a dampening-period interval (which may be zero) has
         elapsed since the previous update record creation for
         a receiver, any subscribed objects and properties
         that have changed since the previous update record
         will have their current values marshalled and placed
         in a new update record.";
      reference
        "RFC 8641:  Subscription to YANG Notifications for
         Datastore Updates - Section 5.";
    }
  }

container i2nsf-monitoring-configuration {
    description
      "The container for configuring I2NSF monitoring.";
    container i2nsf-system-detection-alarm-configuration {
      if-feature "i2nsf-system-detection-alarm";
      description
        "The container for configuring I2NSF system-detection-alarm
         notification";
      uses enable-notification;
      list system-alarm {
        key alarm-type;
        description
          "Configuration for system alarm (i.e., CPU, Memory,
           and Disk Usage)";
        uses dampening;
      }
```

We also updated the range from 0..100 to 1..100 in the data model.

OLD

```
leaf threshold {
  type uint8 {
    range "0..100";
  }
  units "percent";
  description
    "The configuration for threshold percentage to trigger
     the alarm.";
}
```

NEW

```
leaf threshold {
  type uint8 {
    range "1..100";
  }
  units "percent";
  description
    "The configuration for threshold percentage to trigger
     the alarm.";
}
```

Minor Issues:

3) too many notifications

This module creates a lot of notifications to manage, and they are all optional to implement. This increases complexity in both the client implementation and operations.

If you really need all 13 notifications then OK, but 13 notification events is a lot for one YANG module, especially if this set will get even larger over time.

Here is one way to reduce the number of event definitions. The example below has 1 event and 13 sub-event types, but it could also apply to N event types each with some sub-event types.

This design template adds one more layer in the notification message, but it is probably easier for the client and operator to manage. The deployment may require filters and access control rules that become more complex for a large number of notifications.

```
notification i2nsf-event {
  description
    "Wrapper for all I2NSF events";

  choice sub-event-type {
    description
      "This choice must be augmented with cases for each allowed
       sub-event.  Only 1 sub-event will be instantiated in each
       i2nsf-event message. Each case is expected to define one
       container with all the sub-event fields.";

     // could put sub-events inline
     case i2nsf-system-detection-alarm {
       if-feature "i2nsf-system-detection-alarm";
       container i2nsf-system-detection-alarm {
         // contents of i2nsf-system-detection-alarm data
       }
     }

   }
 }


 // could add sub-events via augments at any time
 augment "/i2nsf-event/sub-event-type" {
   case i2nsf-system-detection-event {
     if-feature "i2nsf-system-detection-event";
     container i2nsf-system-detection-event {
       // contents of i2nsf-system-detection-event data
     }
   }
 }
```

=> [PAUL] We have updated the data model according to your guide to reduce the number of notifications. We created 3 parent notifications (i.e., i2nsf-event, i2nsf-logs, and i2nsf-nsf-event). Each notification is used for different purposes. The i2nsf-event is used for general notifications that are triggered by an event and should be supported on all types of NSF. The i2nsf-log is used for notifications that are received from the logs of the NSF. The i2nsf-nsf-event is used for advanced notifications that are supported only on security-specific NSF (e.g., i2nsf-nsf-detection-ddos and i2nsf-nsf-detection-virus).

Parts of the changed data model are shown below, the full changes can be seen in the document.

OLD

```
notification i2nsf-system-detection-alarm {
    if-feature "i2nsf-system-detection-alarm";
    description
      "This notification is sent, when a system alarm
       is detected.";
    leaf alarm-category {
      type identityref {
       base alarm-type;
      }
      description
        "The alarm category for
         system-detection-alarm notification";
    }
    uses characteristics;
    uses i2nsf-system-alarm-type-content;
    uses common-monitoring-data;
  }
```

NEW

```
notification i2nsf-event {
    description
      "Notification for I2NSF Event.";
    choice sub-event-type {
      description
        "This choice must be augmented with cases for each allowed
         sub-event. Only 1 sub-event will be instantiated in each
         i2nsf-event message. Each case is expected to define one
         container with all the sub-event fields.";
      case i2nsf-system-detection-alarm {
        container i2nsf-system-detection-alarm{
          description
            "This notification is sent, when a system alarm
             is detected.";
          leaf alarm-category {
            type identityref {
             base alarm-type;
            }
            description
              "The alarm category for
               system-detection-alarm notification";
          …
        }
      }
     case …
  }
}
```

Nits:

4) underscore vs. hyphen

**There are many field names in sec. 7 that are incorrect
because they use an underscore instead of a hyphen char
(e.g. req_cookies but leaf name is req-cookies)**

=> [PAUL] We have updated the Section 7 of the document to follow the naming in the data model.

5) verbose SNMP-style names

**The term -configuration in the object names is unusual.
Repeating the parent name (like SMIv2) is not usually done in YANG.
(e.g., i2nsf-system-detection-event-configuration)**
=> [PAUL] We removed the "-configuration" in the data model except the parent container.

OLD

```
container i2nsf-monitoring-configuration {
  description
    "The container for configuring I2NSF monitoring.";
  container i2nsf-system-detection-alarm-configuration {
    description
      "The container for configuring I2NSF system-detection-alarm
       notification";
    …
```

NEW

```
container i2nsf-monitoring-configuration {
  description
    "The container for configuring I2NSF monitoring.";
  container i2nsf-system-detection-alarm {
    description
      "The container for configuring I2NSF system-detection-alarm
       notification";
    …
```

**6) identifiers should use well-known abbreviations or spell
out the word if not too long. E.g "ave" -> "average"**
=> [PAUL] We have spelled out the words or used well known abbreviations in the documents.

OLD

```
  leaf in-traffic-ave-rate {
    type uint32;
    units "pps";
    description
```

```
     "Inbound traffic average rate in packets per second (pps)";
 }
```

NEW

```
 leaf in-traffic-average-rate {
   type uint32;
   units "pps";
   description
     "Inbound traffic average rate in packets per second (pps)";
 }
```

7) Is there a reason some identities are ALL-CAPS and others are all-lower-case? This should be explained in the YANG module.

=> [PAUL] We have updated the data model to be lower-cased in all identities for keeping the consistency of the data model.

Example:

OLD

```
identity MEM-USAGE-ALARM {
  base alarm-type;
  description
    "A memory alarm is alerted.";
}
```

NEW

```
identity mem-usage-alarm {
  base alarm-type;
  description
    "A memory alarm is alerted.";
}
```

=> [PAUL] There are other changes in the current document.

1. We found some useful monitoring information that can help improve the module. To handle specific traffic flow statistics for data analysis (e.g., the detection of DoS or DDoS attacks), we added a new feature as the following:

NEW

```
notification i2nsf-event {
  description
    "Notification for I2NSF Event.";
  choice sub-event-type {
    ...
    case i2nsf-traffic-flows {
      container i2nsf-traffic-flows {
        description
          "This notification is sent to inform about the traffic
           flows.";
        leaf src-ip {
          type inet:ip-address;
          description
            "The source IPv4 (or IPv6) address of the packet";
        }
        leaf dst-ip {
          type inet:ip-address;
          description
            "The destination IPv4 (or IPv6) address of the packet";
        }
        leaf protocol {
          type identityref {
            base protocol-type;
          }
          description
            "The protocol type for nsf-detection-intrusion
             notification";
        }
        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 arrival-rate {
          type uint32;
          units "pps";
          description
            "The arrival rate of the packet in packets
            per second";
        }
        uses characteristics;
        uses common-monitoring-data;
      }
    }
  …
}
```

2. We also added a new field for ddos-attack event to give more information, i.e., attack destination IP address.

OLD

```
case i2nsf-nsf-detection-ddos {
  if-feature "i2nsf-nsf-detection-ddos";
  container i2nsf-nsf-detection-ddos {
    description
      "This notification is sent, when a specific flood type
       is detected.";
    uses i2nsf-nsf-event-type-content;
    leaf 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 (i.e., ICMPv4 or ICMPv6) flood, HTTP flood,
         HTTPS flood, DNS query flood, DNS reply flood, SIP
         flood, 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-src-ip {
      type inet:ip-address;
      description
        "The source IPv4 (or IPv6) addresses of attack
         traffic. If there are a large amount of IPv4
         (or IPv6) addresses, then pick a certain number
         of resources according to different rules.";
    }
    uses attack-rates;
    uses log-action;
    uses characteristics;
    uses common-monitoring-data;
  }
}
```

NEW

```
case i2nsf-nsf-detection-ddos {
  if-feature "i2nsf-nsf-detection-ddos";
  container i2nsf-nsf-detection-ddos {
    description
      "This notification is sent, when a specific flood type
       is detected.";
    uses i2nsf-nsf-event-type-content;
    leaf 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 (i.e., ICMPv4 or ICMPv6) flood, HTTP flood,
         HTTPS flood, DNS query flood, DNS reply flood, SIP
         flood, 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-src-ip {
      type inet:ip-address;
      description
        "The source IPv4 (or IPv6) addresses of attack
         traffic. If there are a large number of IPv4
         (or IPv6) addresses, then pick a certain number
         of resources according to different rules.";
    }
    leaf attack-dst-ip {
      type inet:ip-address;
      description
        "The destination IPv4 (or IPv6) addresses of attack
         traffic. If there are a large number of IPv4
         (or IPv6) addresses, then pick a certain number
         of resources according to different rules.";
```

```
          }
          uses attack-rates;
          uses log-action;
          uses characteristics;
          uses common-monitoring-data;
        }
      }
```

Thanks for your help and support.

Best Regards,
Paul
--
===========================
Mr. Jaehoon (Paul) Jeong, Ph.D.
Associate Professor
Department of Computer Science and Engineering
Sungkyunkwan University
Office: +82-31-299-4957
Email: jaehoon.paul@gmail.com, pauljeong@skku.edu
Personal Homepage: http://iotlab.skku.edu/people-jaehoon-jeong.php