Revision Letter

I2NSF Network Security Function-Facing Interface YANG Data Model

(Old Draft Name: draft-ietf-i2nsf-nsf-facing-interface-dm-06 and New Draft Name: draft-ietf-i2nsf-nsf-facing-interface-dm-07)

Jaehoon Paul Jeong

07/25/2019

Hi Acee,

I sincerely appreciate your valuable comments. My answers start with the mark “[PAUL]”

.

Document: draft-ietf-i2nsf-nsf-facing-interface-dm-06

Reviewer: Acee Lindem

Review Date: June 22, 2019

Review Type: Working Group Last Call

Intended Status: Standards Track

Summary: Needs to go back to Working Group for rework and another WGLC

Modules: "ietf-i2nsf-policy-rule-for-nsf@2019-06-12.yang"

Tech Summary: The model defines different types of I2NSF security policy. Each is comprised of an event, a condition, and an action. There is significant overlap with other IETF models. Within I2NSF, there is repetition of definitions which needs to go into a common I2NSF types module. Additionally, the data descriptions were done quickly and never reviewed or edited. I believe it needs to go back to the working group for another revision and working group last call.

Major Comments:

1. Why don't you leverage the definitions in RFC 8519 for packet matching? We don't need all this defined again.

=> [PAUL] Due to the time limitation, this revision cannot reflect the usage of definitions In RFC 8519 for packet matching. I will reflect your comments on the next revision.

2. Date and time are defined in RFC 6991. Why don't those suffice?

=> [PAUL] I revised the date-and-time according to your comments.

OLD:

```
typedef start-time-type {
      type union {
          type string {
             pattern '\d{2}:\d{2}:\d{2}(\.\d+)?'
                      + '(Z|[\+\-]\d{2}:\d{2})';
          }

          type enumeration {
             enum right-away {
                description
                   "Immediate rule execution
                    in the system.";
             }
          }
      }

      description
         "Start time when the rules are applied.";
   }

   typedef end-time-type {
      type union {
          type string {
             pattern '\d{2}:\d{2}:\d{2}(\.\d+)?'
                      + '(Z|[\+\-]\d{2}:\d{2})';
          }

          type enumeration {
             enum infinitely {
                description
```

```
                "Infinite rule execution
                 in the system.";
            }
          }
        }
        description
          "End time when the rules are applied.";
    }
```

NEW:

```
typedef start-time-type {
        type union {
            type ynag:date-and-time;

          type enumeration {
             enum right-away {
                description
                  "Immediate rule execution
                   in the system.";
            }
          }
        }

        description
          "Start time when the rules are applied.";
    }

    typedef end-time-type {
        type union {
            type ynag:date-and-time;

          type enumeration {
             enum infinitely {
```

```
            description
              "Infinite rule execution
               in the system.";
          }
        }
      }
      description
        "End time when the rules are applied.";
  }
```

3. Refer to the intervals as "time-intervals" rather than "time-zones". The term "time-zone" has a completely different connotation.

=> [PAUL] I changed the time-zones into time-intervals.

OLD:

```
|       |   +--rw time-zones
  |       |   |   +--rw absolute-time-zone
  |       |   |   |   +--rw start-time?     start-time-type
  |       |   |   |   +--rw end-time?       end-time-type
  |       |   |   +--rw periodic-time-zone
  |       |   |       +--rw day
  |       |   |       |   +--rw every-day?          boolean
  |       |   |       |   +--rw specific-day*     day-type
  |       |   |       +--rw month
  |       |   |           +--rw every-month?          boolean
  |       |   |           +--rw specific-month*     month-type
```

NEW:

```
|       |   +--rw time-intervals
  |       |   |   +--rw absolute-time-interval
  |       |   |   |   +--rw start-time?     start-time-type
  |       |   |   |   +--rw end-time?       end-time-type
  |       |   |   +--rw periodic-time-interval
  |       |   |       +--rw day
  |       |   |       |   +--rw every-day?          boolean
  |       |   |       |   +--rw specific-day*     day-type
  |       |   |       +--rw month
```

```
|       |  |          +--rw every-month?          boolean
|       |  |          +--rw specific-month*       month-type
```

4. What the "acl-number"? Also, ACLs are named (RFC 8519). Also, why define all the packet matching and then reference an ACL.

=> [PAUL] We delete acl-number. And, in that case of packet matching, due to the time limitation, this revision cannot reflect the usage of definitions In RFC 8519 for packet matching. In the next revision, I will reflect your comment.

5. The descriptions are very awkwardly worded and in many cases simply repeat the data node or identify description without hyphens. I started trying to fix this but it was too much. I'll pass for on for some examples. There are enough co-authors and contributors that one would expect much better.

=> [PAUL] I reflected the sentences that you revised on the revision.

6. There is overlap of definitions with the I2NSF capabilities draft. The common types and identities should be factored into a common I2NSF types module.

=> [PAUL] Due to the time limitation, this revision cannot reflect the factoring of the common types and identities. I will reflect your comments on the next revision.

7. The "Security Considerations" in section 8 do not conform to the recommended template in https://trac.ietf.org/trac/ops/wiki/yang-security-guidelines>

=> [PAUL] I revised “Security Considerations” Section according to the recommended template.

OLD:

The YANG module specified in this document defines a data schema designed to be accessed through network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040].

The lowest NETCONF layer is the secure transport layer, and the required transport secure transport is Secure Shell (SSH) [RFC6242].

The lowest RESTCONF layer is HTTPS, and the required transport secure transport is TLS [RFC8446].

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

NEW:

The YANG module specified in this document defines a data schema designed to be accessed through network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040].

The lowest NETCONF layer is the secure transport layer, and the required transport secure transport is Secure Shell (SSH) [RFC6242].

The lowest RESTCONF layer is HTTPS, and the required transport secure transport is TLS [RFC8446].

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

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

- ietf-i2nsf-policy-rule-for-nsf: The attacker may provide incorrect policy information of any target NSFs by illegally modifying this.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

- ietf-i2nsf-policy-rule-for-nsf: The attacker may gather the security policy information of any target NSFs and misuse the security policy information for subsequent attacks.

Minor Comments:

1. Section 3.1 should reference RFC8340 rather than attempting to include tree diagram formatting semantics.

=> [PAUL] I deleted it that attempts to include tree diagram formatting semantics.

OLD:

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 [RFC8340] 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.

NEW:

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 is referred from [RFC8340]

2. "iiprfn" is a poor choice for default model prefix - I suggest "nsfintf". It is only one character longer and actually is expands to something meaningful.

=> [PAUL] I agree with your comment on modifying iipfn into nsfintf.

3. RFC 2460 is obsoleted by RFC 8200.

=> [PAUL] I replaced RFC 2460 with RFC 8200 as a reference.

4. RFC 791 is the wrong reference for IPv4 TOS. It should be RFC 1394.

=> [PAUL] I replaced RFC 791 with RFC 1394 for a reference to IPv4 TOS.

5. What is the IGRP protocol? I'm familiar with EIGRP but not IGRP.

=> [PAUL] This version considers both IGRP protocol and EIGRP protocol.

(https://en.wikipedia.org/wiki/List_of_IP_protocol_numbers)

6. What is the skip protocol? Is this about skipping the check? If so, why is it needed.

=> [PAUL] Skip was one of the icmp type. However, it is an unsupported icmp type, so I removed it in this version.

7. Reference for IPv6 ICMP should be RFC 2463.

=> [PAUL] I revised that according to your comments.

8. Why do you include Photuris definitions? Nobody uses this.

=> [PAUL] I deleted the Photuris according to your comments.

9. Note that all the keys for all 'config true' lists must be unique so your specification in the description as well as'mandatory true' are redundant for the 'rules' list. This mistake is in other lists as well.

=> [PAUL] I deleted redundant mandatory true according to your comments.

10. What is 'during' time?

=> [PAUL] I revised the description for “leaf during”.

OLD:

```
leaf during {
        type uint16;
        description
          "This is during time.";
  }
```

NEW:

```
leaf during {
        type uint16;
        description
          "This has long-connection during a time.";
  }
```

11. What is a "security-grup"? Is this a security-group?

=> [PAUL] I changed security-grup into security-group.

12. The module prologue doesn't match the example in Appendix B of RFC 8407.

=> [PAUL] I think the module prologue conforms to the example in Appendix B of RFC 8407. If not, could you give me more detailed comments on it?

13. There needs to be a good definition of absolute and periodic time in the descriptions.

=> [PAUL] I added the definition of absolute and periodic time in the descriptions.

OLD:

absolute-time-zone: Rule execution according to absolute time.

periodic-time-zone: Rule execution according to periodic time.

NEW:

absolute-time-interval: Rule execution time according to absolute time. The absolute time intervals mean the exact time to start or end.

periodic-time-interval: Rule execution time according to periodic time. The periodic time intervals mean repeated time like day, week, or month.

14. The References do not include all the RFCs referenced by YANG model reference statements.

=> [PAUL] I added RFCs referenced by YANG model reference statements such as RFC 768, RFC 790,

RFC 791, RFC 792, RFC 793, RFC 3261, and RFC 8200.

Thanks for your valuable comments.

Best Regards,

Paul Jeong