Revision Letter

Editor: Jaehoon Paul Jeong
Date: January 22, 2022

OLD: draft-ietf-i2nsf-nsf-facing-interface-dm-16
NEW: draft-ietf-i2nsf-nsf-facing-interface-dm-17

Dear Joe Clarke, Yoshifumi Nishida, Kyle Rose, and Tom Petch,

I sincerely appreciate your detailed comments to improve our NSF-Facing Interface YANG Data Model. I have addressed all your comments one by one. I use bold font for your comments and use a regular font for my responses with the prefix "=> [PAUL]".

Reviewer: Joe Clarke

Review result: Has Issues

I have been asked to review draft-ietf-i2nsf-nsf-facing-interface-dm on behalf of the Ops Directorate. While this draft represents an info model for the NSF-facing I2NSF interface, it seemed more practical from a configuration standpoint, and I was left wanting more fleshed out element descriptions. I found the model overall readable but was left wondering what I as an operator that might be configuring exactly in certain cases. I also found some perhaps YANG-ish things that I think should be fixed (e.g., leaf naming in parts). Below are some specific instances of these issues I had when reading:

Section 3.1

"The system policy provides for multiple system policies "

This sentence doesn't make much sense. Are you saying that the top-level system policy provides for multiple sub-policies?

=> [PAUL] We updated the sentence by replacing "system policy" with "security policy" as follows:

OLD:
The system policy provides for multiple system policies in one NSF, and each system policy is used by one virtual instance of the NSF/device. The system policy includes system policy name, priority usage, resolution strategy, default action, and rules.

NEW:
A security policy is used by one virtual instance of an NSF/device as a set of security rules to protect assets from major risk factors that threaten the system. There can be multiple security policies in a single NSF to provide the necessary protection. The

security policy includes its name, priority usage, resolution strategy, default action, and rules.

===

YANG module

```
identity system-event {
  description
    "Identity for system events";
}
```

I'm not crazy about descriptions that are just restatements of the type and the name. While you have a reference here, can you make these identity descriptions more useful without one needing to jump to other documents?

There are many other identities like this where I'd prefer to see more descriptive text to help me as a reader/implementer/operator understand more without having to jump between documents all the time.

=> [PAUL] We updated the descriptions for the identities to explain the identities in more detail. The name of "target-device" has been updated to "device-type" because the name of "target" can emphasize it being a "targeted" or "attacked" system.
The changes are as follows:

OLD:

```
  identity system-event {
    base event;
    description
      "Identity for system events";
    reference
      "draft-ietf-i2nsf-nsf-monitoring-data-model-11: I2NSF NSF
       Monitoring YANG Data Model - System event";
  }

  identity system-alarm {
    base event;
    description
      "Identity for system alarms";
    reference
      "draft-ietf-i2nsf-nsf-monitoring-data-model-11: I2NSF NSF
       Monitoring YANG Data Model - System alarm";
  }

  identity access-violation {
    base system-event;
    description
      "Identity for access violation
       system events";
    reference
      "draft-ietf-i2nsf-nsf-monitoring-data-model-11: I2NSF NSF
       Monitoring YANG Data Model - System event for access
```

```
     violation";
}

identity configuration-change {
  base system-event;
  description
    "Identity for configuration change
     system events";
  reference
    "draft-ietf-i2nsf-nsf-monitoring-data-model-11: I2NSF NSF
     Monitoring YANG Data Model - System event for configuration
     change";
}

identity memory-alarm {
  base system-alarm;
  description
    "Identity for memory alarm
     system alarms";
  reference
    "draft-ietf-i2nsf-nsf-monitoring-data-model-11: I2NSF NSF
     Monitoring YANG Data Model - System alarm for memory";
}

identity cpu-alarm {
  base system-alarm;
  description
    "Identity for CPU alarm
     system alarms";
  reference
    "draft-ietf-i2nsf-nsf-monitoring-data-model-11: I2NSF NSF
     Monitoring YANG Data Model - System alarm for CPU";
}

identity disk-alarm {
  base system-alarm;
  description
    "Identity for disk alarm
     system alarms";
  reference
    "draft-ietf-i2nsf-nsf-monitoring-data-model-11: I2NSF NSF
     Monitoring YANG Data Model - System alarm for disk";
}

identity hardware-alarm {
  base system-alarm;
  description
    "Identity for hardware alarm
     system alarms";
  reference
    "draft-ietf-i2nsf-nsf-monitoring-data-model-11: I2NSF NSF
     Monitoring YANG Data Model - System alarm for hardware";
}

identity interface-alarm {
  base system-alarm;
  description
```

```
      "Identity for interface alarm
       system alarms";
    reference
      "draft-ietf-i2nsf-nsf-monitoring-data-model-11: I2NSF NSF
       Monitoring YANG Data Model - System alarm for interface";
  }

  identity target-device {
    description
      "Base identity for target devices";
    reference
      "draft-ietf-i2nsf-capability-data-model-21:
       I2NSF Capability YANG Data Model";
  }
```

NEW:

```
  identity system-event {
    base event;
    description
      "Identity for system events. A system event (called alert) is
       defined as a warning about any changes of configuration, any
       access violation, the information of sessions and traffic
       flows.";
    reference
      "draft-ietf-i2nsf-nsf-monitoring-data-model-13: I2NSF NSF
       Monitoring YANG Data Model - System event";
  }

  identity system-alarm {
    base event;
    description
      "Identity for system alarms. A system alarm is defined as a
       warning related to service degradation in system hardware.";
    reference
      "draft-ietf-i2nsf-nsf-monitoring-data-model-13: I2NSF NSF
       Monitoring YANG Data Model - System alarm";
  }

  identity access-violation {
    base system-event;
    description
      "Identity for access-violation. An access-violation system
       event is an event when a user tries to access (read, write,
       create, or delete) any information or execute commands above
       their privilege.";
    reference
      "draft-ietf-i2nsf-nsf-monitoring-data-model-13: I2NSF NSF
       Monitoring YANG Data Model - System event for access
       violation";
  }

  identity configuration-change {
    base system-event;
```

```
  description
    "Identity for configuration change. A configuration change is
     a system event when a new configuration is added or an
     existing configuration is modified.";
     erence
    "draft-ietf-i2nsf-nsf-monitoring-data-model-13: I2NSF NSF
     Monitoring YANG Data Model - System event for configuration
     change";
}

identity memory-alarm {
  base system-alarm;
  description
    "Identity for memory alarm. Memory is the hardware to store
     information temporarily or for a short period, i.e., Random
     Access Memory (RAM). A memory alarm is emitted when the RAM
     usage exceeds the threshold.";
  reference
    "draft-ietf-i2nsf-nsf-monitoring-data-model-13: I2NSF NSF
     Monitoring YANG Data Model - System alarm for memory";
}

identity cpu-alarm {
  base system-alarm;
  description
    "Identity for CPU alarm. CPU is the Central Processing Unit
     that executes basic operations of the system. A CPU alarm
     is emitted when the CPU usage exceeds the threshold.";
  reference
    "draft-ietf-i2nsf-nsf-monitoring-data-model-13: I2NSF NSF
     Monitoring YANG Data Model - System alarm for CPU";
}

identity disk-alarm {
  base system-alarm;
  description
    "Identity for disk alarm. Disk is the hardware to store
     information for a long period, i.e., Hard Disk and Solid-State
     Drive. A disk alarm is emitted when the Disk usage exceeds
     the threshold.";
  reference
    "draft-ietf-i2nsf-nsf-monitoring-data-model-13: I2NSF NSF
     Monitoring YANG Data Model - System alarm for disk";
}

identity hardware-alarm {
  base system-alarm;
  description
    "Identity for hardware alarm. A hardware alarm is emitted
     when a problem of hardware, e.g., CPU, memory, disk, and interface,
     is detected.";
  reference
    "draft-ietf-i2nsf-nsf-monitoring-data-model-13: I2NSF NSF
     Monitoring YANG Data Model - System alarm for hardware";
}

identity interface-alarm {
```

```
    base system-alarm;
    description
      "Identity for interface alarm. Interface is the network
       interface for connecting a device with the network. An
       interface alarm is emitted when the state of the interface
       is changed.";
    reference
      "draft-ietf-i2nsf-nsf-monitoring-data-model-13: I2NSF NSF
       Monitoring YANG Data Model - System alarm for interface";
  }

  identity device-type {
    description
      "Base identity for types of device. This identity is used for
       type of the device for the destination of a packet or traffic flow.";
    reference
      "draft-ietf-i2nsf-capability-data-model-22:
       I2NSF Capability YANG Data Model";
  }
```

===

YANG module

identity anti-ddos { ... }

This, and other actions seem to be missing descriptive detail about what exactly is expected from the NSF if this is configured. Maybe this is left up to implementations, but in that case I'd expect some references to potential DDoS mitigation approaches to take.

=> [PAUL] We updated the descriptions to add more detailed explanation on the identities. We also provide a reference to the “anti-ddos” as follows:

OLD:

```
  identity content-security-control {
    base advanced-nsf;
    description
      "Base identity for content security control";
    reference
      "draft-ietf-i2nsf-capability-data-model-21:
       I2NSF Capability YANG Data Model";
  }

  identity attack-mitigation-control {
    base advanced-nsf;
    description
      "Base identity for attack mitigation control";
    reference
      "draft-ietf-i2nsf-capability-data-model-21:
       I2NSF Capability YANG Data Model";
  }
```

```
  identity anti-ddos {
    base attack-mitigation-control;
    description
      "Identity for advanced NSF Anti-DDoS or DDoS Mitigator
       capability.";
  }
```

NEW:

```
  identity content-security-control {
    base advanced-nsf;
    description
      "Base identity for content security control. Content security
       control is an NSF that evaluates the payload of a packet,
       such as Intrusion Prevention System (IPS), URL Filter,
       Antivirus, and VoIP/VoLTE Filter.";
    reference
      "draft-ietf-i2nsf-capability-data-model-22:
       I2NSF Capability YANG Data Model";
  }

  identity attack-mitigation-control {
    base advanced-nsf;
    description
      "Base identity for attack mitigation control. Attack mitigation
       control is an NSF that mitigates an attack such as
       anti-DDoS (i.e., DDoS-mitigator).";
    reference
      "draft-ietf-i2nsf-capability-data-model-22:
       I2NSF Capability YANG Data Model";
  }

  identity anti-ddos {
    base attack-mitigation-control;
    description
      "Identity for advanced NSF Anti-DDoS or DDoS Mitigator to
       protect a server or network from a DDoS attack. The mitigation
       approach is up to the implementation.";
    reference
      "RFC 4732: Internet Denial-of-Service Considerations - DoS
       Mitigation Strategies
       RFC 4987: TCP SYN Flooding Attacks and Common Mitigations -
       Common Defenses";
  }
```

===

YANG module

```
identity drop {
       base ingress-action;
       base egress-action;
```

```
      base default-action;
      description
        "Identity for drop";
      reference
        "draft-ietf-i2nsf-capability-data-model-21:
         I2NSF Capability YANG Data Model - Actions and
         Default Action";
...
}
```

Just as above, I was expecting more details about these actions actually mean and exactly the behavior one could expect. For example, how is a drop to be done? Does it matter if it's a silent drop vs. a drop/unreachable?

=> [PAUL] We added a new action called “reject” to indicate a drop with response. The drop mechanism should drop the packet without sending any response. We updated the data model as follows:

OLD:

```
  identity ingress-action {
    base action;
    description
      "Base identity for ingress action";
    reference
      "draft-ietf-i2nsf-capability-data-model-21:
       I2NSF Capability YANG Data Model - Ingress Action";
  }

  identity egress-action {
    base action;
    description
      "Base identity for egress action";
    reference
      "draft-ietf-i2nsf-capability-data-model-21:
       I2NSF Capability YANG Data Model - Egress Action";
  }

  identity default-action {
    base action;
    description
      "Base identity for default action";
    reference
      "draft-ietf-i2nsf-capability-data-model-21:
       I2NSF Capability YANG Data Model - Default Action";
  }

  identity pass {
    base ingress-action;
    base egress-action;
    base default-action;
    description
      "Identity for pass";
    reference
      "draft-ietf-i2nsf-capability-data-model-21:
       I2NSF Capability YANG Data Model - Actions and
```

```
      Default Action";
  }

  identity drop {
    base ingress-action;
    base egress-action;
    base default-action;
    description
      "Identity for drop";
    reference
      "draft-ietf-i2nsf-capability-data-model-21:
       I2NSF Capability YANG Data Model - Actions and
       Default Action";
  }

  identity mirror {
    base ingress-action;
    base egress-action;
    base default-action;
    description
      "Identity for mirror";
    reference
      "draft-ietf-i2nsf-capability-data-model-21:
       I2NSF Capability YANG Data Model - Actions and
       Default Action";
  }

  identity rate-limit {
    base ingress-action;
    base egress-action;
    base default-action;
    description
      "Identity for rate limiting action";
    reference
      "draft-ietf-i2nsf-capability-data-model-21:
       I2NSF Capability YANG Data Model - Actions and
       Default Action";
  }
```

NEW:

```
  identity ingress-action {
    base action;
    description
      "Base identity for ingress action. The action to handle the
       network traffic that is entering the secured network.";
    reference
      "draft-ietf-i2nsf-capability-data-model-22:
       I2NSF Capability YANG Data Model - Ingress Action";
  }

  identity egress-action {
    base action;
    description
      "Base identity for egress action. The action to handle the
```

```
      network traffic that is exiting the secured network.";
  reference
    "draft-ietf-i2nsf-capability-data-model-22:
     I2NSF Capability YANG Data Model - Egress Action";
}

identity default-action {
  base action;
  description
    "Base identity for default action. The default action of the
     NSF when no rule matches the packet or flow.";
  reference
    "draft-ietf-i2nsf-capability-data-model-22:
     I2NSF Capability YANG Data Model - Default Action";
}

identity pass {
  base ingress-action;
  base egress-action;
  base default-action;
  description
    "Identity for pass. The pass action allows traffic that matches
     the rule to proceed through the NSF to reach the
     destination.";
  reference
    "draft-ietf-i2nsf-capability-data-model-22:
     I2NSF Capability YANG Data Model - Actions and
     Default Action";
}

identity drop {
  base ingress-action;
  base egress-action;
  base default-action;
  description
    "Identity for drop. The drop action denies the traffic that
     matches the rule. The drop action should do a silent drop,
     which does not give any response to the source.";
  reference
    "draft-ietf-i2nsf-capability-data-model-22:
     I2NSF Capability YANG Data Model - Actions and
     Default Action";
}

identity reject {
  base ingress-action;
  base egress-action;
  base default-action;
  description
    "Identity for reject action capability. The reject action
     denies packet to go through the NSF entering or exiting the
     internal network and send a response back to the source.
     The response depends on the packet and implementation.
     For example, a TCP packet is rejected with TCP RST response
     or a UDP packet may be rejected with an ICMP response message
     with Type 3 and Code 3, i.e., Destination Unreachable: Destination
     port unreachable.";
```

```
  }

  identity mirror {
    base ingress-action;
    base egress-action;
    base default-action;
    description
      "Identity for mirror. The mirror action copies a packet and sends
       the packet's copy to the monitoring entity while still allowing
       the packet or flow to go through the NSF.";
    reference
      "draft-ietf-i2nsf-capability-data-model-22:
       I2NSF Capability YANG Data Model - Actions and
       Default Action";
  }

  identity rate-limit {
    base ingress-action;
    base egress-action;
    base default-action;
    description
      "Identity for rate limiting action. The rate limit action
       limits the number of packets or flows that can go through the
       NSF by dropping packets or flows (randomly or
       systematically). The drop mechanism, e.g., silent drop and
       unreachable drop (i.e., reject), is up to the implementation";
    reference
      "draft-ietf-i2nsf-capability-data-model-22:
       I2NSF Capability YANG Data Model - Actions and
       Default Action";
  }
```

===

YANG module

```
identity day {
       description
         "This represents the base for days.";
     }
```

Maybe more of a YANG Doctors thing, but why not make days an enumeration where you can have day values? I'd think that would be more useful as I can't foresee someone adding new identities of base day.

=> [PAUL] We change the structure for "day" from an "identity" type into an "enumeration" type as follows:

OLD:

```
  identity day {
    description
      "This represents the base for days.";
  }
```

```
  identity monday {
    base day;
    description
      "This represents Monday.";
  }

  identity tuesday {
    base day;
    description
      "This represents Tuesday.";
  }

  identity wednesday {
    base day;
    description
      "This represents Wednesday.";
  }

  identity thursday {
    base day;
    description
      "This represents Thursday.";
  }

  identity friday {
    base day;
    description
      "This represents Friday.";
  }

  identity saturday {
    base day;
    description
      "This represents Saturday.";
  }

  identity sunday {
    base day;
    description
      "This represents Sunday.";
  }

...

            leaf-list day {
              when
                "../../frequency='weekly'";
              type identityref{
                base day;
              }
              min-elements 1;
              description
                "This represents the repeated day of every week
                 (e.g., Monday and Tuesday).  More than one day can
                 be specified.";
            }
```

NEW:

```
  typedef day {
    type enumeration {
      enum monday {
        description
          "This represents Monday.";
      }
      enum tuesday {
        description
          "This represents Tuesday.";
      }
      enum wednesday {
        description
          "This represents Wednesday";
      }
      enum thursday {
        description
          "This represents Thursday.";
      }
      enum friday {
        description
          "This represents Friday.";
      }
      enum saturday {
        description
          "This represents Saturday.";
      }
      enum sunday {
        description
          "This represents Sunday.";
      }
    }
    description
      "The type for representing the day of the week.";
  }

...

           leaf-list day {
              when
                "../../frequency='weekly'";
              type day;
              min-elements 1;
              description
                "This represents the repeated day of every week
                 (e.g., Monday and Tuesday).  More than one day can
                 be specified.";
            }
```

===

YANG module

```
leaf rule-name {
           type string;
           description
             "The name of the rule.";
         }
```

I wouldn’t prefix each leaf with “rule” since you’re already in the rules list. Moreover, you’re not doing this consistently here or in other lists (e.g., ethernet vs. ipv4).

=> [PAUL] The naming of the leaves has been updated by removing the prefix “rule-” to remove the redundancy of the naming. The changes are as follows:

OLD:

```
list rules {
  key "rule-name";
  description
    "This is a rule for network security functions.";

  leaf rule-name {
    type string;
    description
      "The name of the rule.";
  }

  leaf rule-description {
    type string;
    description
      "This description gives more information about
       rules.";
  }

  leaf rule-priority {
    type uint8 {
      range "1..255";
    }
    description
      "The priority keyword comes with a mandatory
       numeric value which can range from 1 up to 255.
       Note that a higher number means a higher priority";
  }

  leaf rule-enable {
    type boolean;
    description
      "True is enable.
       False is not enable.";
  }

...

  container ethernet {

    leaf ethernet-description {
```

```
      type string;
      description
        "The MAC Condition description";
    }

...

  container payload {
    description
      "Condition for packet payload";
    leaf packet-payload-description {
      type string;
      description
       "This is description for payload condition.";
    }
    leaf-list payload-content {
      type string;
      description
        "This is a condition for packet payload content.";
    }
  }

  container context {
    description
      "Condition for context";
    leaf context-description {
      type string;
      description
        "This is description for context condition.";
    }

...

  container users {
    description
      "Condition for users";
    leaf users-description {
      type string;
      description
        "This is the description for users' condition.";
    }

...

  container action {

    leaf action-clause-description {
      type string;
      description
        "Description for an action clause.";
    }


...

  list groups {
    key "group-name";
```

```
    description
      "This is a group for rules";

    leaf group-name {
      type string;
      description
        "This is a group for rules";
    }
```

NEW:

```
list rules {
  key "name";
  description
    "This is a rule for network security functions.";

  leaf name {
    type string;
    description
      "The name of the rule.";
  }

  leaf description {
    type string;
    description
      "This description gives more information about
       rules.";
  }

  leaf priority {
    type uint8 {
      range "1..255";
    }
    description
      "The priority for the rule comes with a mandatory
       numeric value which can range from 1 up to 255.
       Note that a higher number means a higher priority";
  }

  leaf enable {
    type boolean;
    description
      "If true, the rule is enabled and enforced.
       If false, the rule is configured but disabled
       and not enforced.";
  }

...

  container ethernet {

    leaf description {
      type string;
      description
```

```
        "The ethernet condition description";
    }

...

  container payload {
    description
      "Condition for packet payload";
    leaf description {
      type string;
      description
       "This is description for payload condition.";
    }
    leaf-list content {
      type string;
      description
        "This is a condition for packet payload content.";
    }
  }

  container context {
    description
      "Condition for context";
    leaf description {
      type string;
      description
        "This is description for context condition.";
    }

...

  container users {
    description
      "Condition for users";
    leaf description {
      type string;
      description
        "This is the description for users' condition.";
    }

...

  container action {

    leaf description {
      type string;
      description
        "Description for an action clause.";
    }

...

  list groups {
    key "name";
    description
      "This is a group for rules";
```

```
    leaf name {
      type string;
      description
        "This is the name of the group for rules";
    }
```

===

YANG module

```
leaf rule-enable {
           type boolean;
           description
             "True is enable.
              False is not enable.";
         }
```

You have a few "enable" leafs in this module, and I would flesh these out a bit more to add clarity. Maybe, . “If true, the rule is enabled and enforced. If false, the rule is configured but disabled and not enforced.”

Something like that.

=> [PAUL] The description of each -enable leaf is augmented with your guideline as follows:

OLD:

```
list rules {
  ...

  leaf rule-enable {
    type boolean;
    description
      "True is enable.
       False is not enable.";
  }

...

      container long-connection {
        description
          "A container for long connection. A long connection is a
           connection that is maintained after the socket connection
           is established, regardless of whether it is used for data
           traffic or not.";

        leaf enable {
          type boolean;
          description
            "True is enabled.
```

```
             False is not enabled.";
        }

        leaf duration {
          type uint16;
          units "second";
          description
            "This is the duration of the long-connection.";
        }
      }

...

    container rule-group {
      description
        "This is rule group";

      list groups {
        key "name";
        description
          "This is a group for rules";

        leaf name {
          type string;
          description
            "This is the name of the group for rules";
        }

        leaf-list rule-name {
          type leafref {
            path
              "../../../rules/rule-name";
          }
          description
            "The names of the rules to be grouped.";
        }

        leaf enable {
          type boolean;
          description
            "True is enabled, and False is not enabled.";
        }

        leaf description {
          type string;
          description
            "This is a description for rule-group";
        }
      }
    }
```

```
NEW:
```

```
list rules {
  ...

  leaf enable {
    type boolean;
    description
      "If true, the rule is enabled and enforced.
       If false, the rule is configured but disabled
       and not enforced.";
  }

...

      container long-connection {
        description
          "A container for long connection. A long connection is a
           connection that is maintained after the socket connection
           is established, regardless of whether it is used for data
           traffic or not.";

        leaf enable {
          type boolean;
          description
            "If true, the rule is enabled and enforced.
             If false, the rule is configured but disabled
             and not enforced.";
        }

        leaf duration {
          type uint16;
          units "second";
          description
            "This is the duration of the long-connection.";
        }
      }


...


    container rule-group {
      description
        "This is rule group";

      list groups {
        key "name";
        description
          "This is a group for rules";

        leaf name {
          type string;
          description
            "This is the name of the group for rules";
        }

        leaf-list rule-name {
          type leafref {
```

```
            path
              "../../../rules/rule-name";
          }
          description
            "The names of the rules to be grouped.";
        }

        leaf enable {
          type boolean;
          description
            "If true, the rule is enabled and enforced.
             If false, the rule is configured but disabled
             and not enforced.";
        }

        leaf description {
          type string;
          description
            "This is a description for rule-group";
        }
      }
    }
```

===

YANG module

```
leaf-list date {
                  when
                    "../../frequency='monthly'";
                  type int32{
                    range "1..31";
                  }
                  min-elements 1;
                  description
                    "This represents the repeated date of every month.
                     More than one date can be specified.";
                }
```

Does this need to be a 32-bit integer? Given the range, int8 should do.

=> [PAUL] We updated the type for date from “int32” to “int8”.

OLD:

```
            leaf-list date {
              when
                "../../frequency='monthly'";
              type int32{
                range "1..31";
```

```
              }
              min-elements 1;
              description
                "This represents the repeated date of every month.
                 More than one date can be specified.";
            }
```

NEW:

```
            leaf-list date {
              when
                "../../frequency='monthly'";
              type int8{
                range "1..31";
              }
              min-elements 1;
              description
                "This represents the repeated date of every month.
                 More than one date can be specified.";
            }
```

===

YANG module

```
leaf alert-packet-rate {
              type uint32;
              units "pps";
              description
                "The alert rate of flood detection for
                 packets per second (PPS) of an IP address.";
            }
```

As I understand it, these are thresholds before an alert will be generated? If so, can you make that more explicit in this and other threshold descriptions?
=> [PAUL] We updated the descriptions to explain the details as follows:

OLD:

```
        container ddos {
          description
            "Condition for DDoS attack.";

          leaf description {
            type string;
            description
             "This is description for ddos condition.";
          }

          leaf alert-packet-rate {
            type uint32;
```

```
            units "pps";
            description
              "The alert rate of flood detection for
               packets per second (PPS) of an IP address.";
          }

          leaf alert-flow-rate {
            type uint32;
            description
              "The alert rate of flood detection for
               flows per second of an IP address.";
          }

          leaf alert-byte-rate {
            type uint32;
            units "BPS";
            description
              "The alert rate of flood detection for
               bytes per second of an IP address.";
          }
        }
```

NEW:

```
        container ddos {
          description
            "Condition for DDoS attack.";

          leaf description {
            type string;
            description
             "This is description for ddos condition.";
          }

          leaf alert-packet-rate {
            type uint32;
            units "pps";
            description
              "The alert rate of flood detection for
               packets per second (PPS) of an IP address.
               If the PPS of an IP address exceeds
               the alert rate threshold, an alert
               will be generated.";
          }

          leaf alert-flow-rate {
            type uint32;
            description
              "The alert rate of flood detection for
               flows per second of an IP address.
               If the flows per second of an IP address
               exceeds the alert rate threshold, an alert
               will be generated.";
```

```
          }

          leaf alert-byte-rate {
            type uint32;
            units "Bps";
            description
              "The alert rate of flood detection for
               bytes per second (Bps) of an IP address.
               If the bytes per second of an IP address
               exceeds the alert rate threshold, an alert
               will be generated.";
          }
        }
```

===

YANG module

In your application container, I'm not sure what application object, group, label, and category are. More description text and references would be helpful.
=> [PAUL] We updated the data model for the “application” container as an “application-protocol” identity by referring to the I2NSF Capability YANG Data Model Draft (draft-ietf-i2nsf-capability-data-model-22) as follows:

OLD:

```
          container application {
            description
              "Condition for application";
            leaf description {
              type string;
              description
               "This is description for application condition.";
            }
            leaf-list object {
              type string;
              description
                "This is application object.";
            }
            leaf-list group {
              type string;
              description
                "This is application group.";
            }
            leaf-list label {
              type string;
              description
                "This is application label.";
            }
            container category {
              description
```

```
            "This is application category";
          list application-category {
            key "name subcategory";
            description
              "This is application category list";

            leaf name {
              type string;
              description
               "This is name for application category.";
            }
            leaf subcategory {
              type string;
              description
               "This is application subcategory.";
            }
          }
        }
      }
```

NEW:

```
  identity application-protocol {
    description
      "Base identity for Application protocol";
  }

  identity http {
    base application-protocol;
    description
      "The identity for Hypertext Transfer Protocol.";
    reference
      "RFC 7230: Hypertext Transfer Protocol (HTTP/1.1): Message
       Syntax and Routing
       RFC 7231: Hypertext Transfer Protocol (HTTP/1.1): Semantics
       and Content";
  }

  identity https {
    base application-protocol;
    description
      "The identity for Hypertext Transfer Protocol Secure.";
    reference
      "RFC 2818: HTTP over TLS (HTTPS)
       RFC 7230: Hypertext Transfer Protocol (HTTP/1.1): Message
       Syntax and Routing
       RFC 7231: Hypertext Transfer Protocol (HTTP/1.1): Semantics
       and Content";
  }

  identity ftp {
    base application-protocol;
    description
      "The identity for File Transfer Protocol.";
    reference
```

```
      "RFC 959: File Transfer Protocol (FTP)";
  }

  identity ssh {
    base application-protocol;
    description
      "The identity for Secure Shell (SSH) protocol.";
    reference
      "RFC 4250: The Secure Shell (SSH) Protocol";
  }

  identity telnet {
    base application-protocol;
    description
      "The identity for telnet.";
    reference
      "RFC 854: Telnet Protocol";
  }

  identity smtp {
    base application-protocol;
    description
      "The identity for Simple Mail Transfer Protocol.";
    reference
      "RFC 5321: Simple Mail Transfer Protocol (SMTP)";
  }

  identity pop3 {
    base application-protocol;
    description
      "The identity for Post Office Protocol 3. This include
       POP3 over TLS";
    reference
      "RFC 1939: Post Office Protocol - Version 3 (POP3)";
  }

  identity imap {
    base application-protocol;
    description
      "The identity for Internet Message Access Protocol. This
       include IMAP over TLS";
    reference
      "RFC 9051: Internet Message Access Protocol (IMAP) - Version
       4rev2";
  }

...

          container application {
            description
              "Condition for application";
            leaf description {
              type string;
              description
               "This is description for application condition.";
            }
            leaf-list protocol {
```

```
            type identityref {
              base application-protocol;
            }
            description
              "The condition based on the application layer
               protocol";
          }
        }
```

===

YANG module

container geography-location { ... }

"geographic" reads better to me than "geography"

Speaking of which, why not reference
https://datatracker.ietf.org/doc/draft-ietf-netmod-geo-location/ for geo-location?
=> [PAUL] We updated the name from “geography” to “geographic” and added the reference of draft-ietf-netmod-geo-location for the container. The updates are as follows:

OLD:

```
  container geography-location {
    description
      "The location which network traffic flow is associated
       with. The region can be the geographical location
       such as country, province, and city,
       as well as the logical network location such as
       IP address, network section, and network domain.";

    leaf description {
      type string;
      description
        "This is description for generic context condition.
         Vendors can write instructions for generic context
         condition that vendor made";
    }

    leaf-list source {
      type string;
      description
        "The src-geography-location is a geographical
         location mapped into an IP address. It matches the
         mapped IP address to the source IP address of the
         traffic flow.";
      reference
        "ISO 3166: Codes for the representation of
         names of countries and their subdivisions";
    }
```

```
    leaf-list destination {
      type string;
      description
        "The dest-geography-location is a geographical
         location mapped into an IP address. It matches the
         mapped IP address to the destination IP address of
         the traffic flow.";
      reference
        "ISO 3166: Codes for the representation of
         names of countries and their subdivisions";
    }
  }
```

NEW:

```
  container geographic-location {
    description
      "The location which network traffic flow is associated
       with. The region can be the geographical location
       such as country, province, and city,
       as well as the logical network location such as
       IP address, network section, and network domain.";
    reference
      "draft-ietf-netmod-geo-location-11: A YANG Grouping for
       Geographic Locations";

    leaf description {
      type string;
      description
        "This is the description for the geographic location
         condition. It is used to describe the conditions and
         instructions that should be implemented.";
    }

    leaf-list source {
      type string;
      description
        "The source is a geographic location mapped into an
         IP address. It matches the mapped IP address to the
         source IP address of the traffic flow.";
      reference
        "ISO 3166: Codes for the representation of
         names of countries and their subdivisions
         draft-ietf-netmod-geo-location-11: A YANG Grouping
         for Geographic Locations";
    }

    leaf-list destination {
      type string;
      description
        "The destination is a geographic location mapped into
         an IP address. It matches the mapped IP address to
         the destination IP address of the traffic flow.";
```

```
      reference
        "ISO 3166: Codes for the representation of
         names of countries and their subdivisions
         draft-ietf-netmod-geo-location-11: A YANG Grouping
         for Geographic Locations";
    }
  }
```

Reviewer: Yoshifumi Nishida

Review result: Almost Ready

This document has been reviewed as part of the transport area review team's ongoing effort to review key IETF documents. These comments were written primarily for the transport area directors, but are copied to the document's authors and WG to allow them to address any issues raised and also to the IETF discussion list for information.

When done at the time of IETF Last Call, the authors should consider this review as part of the last-call comments they receive. Please always CC tsv-art@ietf.org if you reply to or forward this review.

Summary: I think this document is almost ready for publication, but it will be better to check the following minor points.

1: Page 48

We don't need to support header length for TCP while supporting total length for UDP? I am wondering if we want to support TCP option type here.

=> [PAUL] We updated the data model to import a data model from RFC 8519 (YANG Data Model for Network Access Control Lists (ACLs)) according to the comments of a Security directorate member (Kyle Rose) in page 37 in this revision letter. The data model provides an access control list by matching the packet header (e.g., Ethernet, IPv4, IPv6, ICMPv4, ICMPv6, TCP, and UDP). The data model in RFC 8519 provides the TCP header fields such as data-offset and TCP options. Our data model includes these source port number and destination port number in the “tcp” container as follows.

NEW:

```
  container tcp {
    description
      "The purpose of this container is to represent
       TCP packet header information to determine
       if the set of policy actions in this ECA policy
       rule should be executed or not.";
    reference
      "draft-ietf-tcpm-rfc793bis-25: Transmission Control
       Protocol (TCP) Specification";
```

```
    leaf description {
      type string;
      description
       "This is description for tcp condition.";
    }

    container source-port-number {
      choice source-port {
        case range-or-operator {
          uses packet-fields:port-range-or-operator;
          description
            "Source port definition from range or operator.
             Can be used when a single port range to be
             specified.";
        }
        case port-list {
          list port-numbers {
            key "start";
            uses port-range;
            description
              "List of source port numbers.";
          }
          description
            "Source port definition from list of port numbers.
             In the case of multiple port ranges needed to be
             specified.";
        }
        description
          "The choice of source port definition using
           range/operator or a choice to use list of port
           numbers.";
      }
      description
        "The security policy rule according to
         tcp source port number.";
      reference
        "draft-ietf-tcpm-rfc793bis-25: Transmission Control
         Protocol (TCP) Specification - Port Number";
    }

    container destination-port-number {
      choice destination-port {
        case range-or-operator {
          uses packet-fields:port-range-or-operator;
          description
            "Destination port definition from range or
             operator.
             Can be used when a single port range to be
             specified.";
        }
        case port-list {
          list port-numbers {
            key "start";
            uses port-range;
            description
              "List of destination port numbers.";
```

```
          }
          description
            "Destination port definition from list of port
             numbers.
             In the case of multiple port ranges needed to be
             specified.";
        }
        description
          "The choice of destination port definition using
           range/operator or a choice to use list of port
           numbers.";
      }
      description
        "The security policy rule according to
         tcp destination port number.";
      reference
        "draft-ietf-tcpm-rfc793bis-25: Transmission Control
         Protocol (TCP) Specification - Port Number";
    }

    uses packet-fields:acl-tcp-header-fields;

  }
```

2: Page 50:

```
    list total-length {
              key "start end";
              leaf start {
                type uint32;
                description
                  "Start udp total length for a range match.";
              }
              leaf end {
                type uint32;
                must '. >= ../start' {
                  error-message
                    "The end hop limit MUST be equal or greater than
                     the start hop limit.";
                }
                description
                  "End udp total length for a range match.";
              }
```

-> is this error message correct?

=> [PAUL] The data model is updated to follow RFC 8519 YANG Data Model, so the “total-length” list is not defined any more in our draft. The updates are as follows:

OLD:
container udp {

```
  ...
  list total-length {
    key "start end";
    leaf start {
      type uint32;
      description
        "Start udp total length for a range match.";
    }
    leaf end {
      type uint32;
      must '. >= ../start' {
        error-message
          "The end hop limit MUST be equal or greater than
           the start hop limit.";
      }
      description
        "End udp total length for a range match.";
    }
    description
      "The security policy rule according to
       udp total length.  If only one value is needed,
       then set both start and end to the same value";
    reference
      "RFC 768: User Datagram Protocol - Total Length";
  }
}
```

NEW:

```
  container udp {
    ...

    uses packet-fields:acl-udp-header-fields;
  }
```

The “packet-fields:acl-udp-header-fields” above is defined in RFC 8519 (YANG Data Model for Network Access Control Lists (ACLs)). Total-length of a UDP datagram is covered by the “length” leaf in the “packet-fields:acl-udp-header-fields” grouping as follows.

```
   grouping acl-udp-header-fields {
    description
      "Collection of UDP header fields that can be used
       to set up a match filter.";
    leaf length {
      type uint16;
      description
        "A field that specifies the length in bytes of
         the UDP header and UDP data.  The minimum
         length is 8 bytes because that is the length of
         the header.  The field size sets a theoretical
         limit of 65,535 bytes (8-byte header plus 65,527
         bytes of data) for a UDP datagram.  However, the
         actual limit for the data length, which is
         imposed by the underlying IPv4 protocol, is
```

```
        65,507 bytes (65,535 minus 8-byte UDP header
        minus 20-byte IP header).
        In IPv6 jumbograms, it is possible to have
        UDP packets of a size greater than 65,535 bytes.
        RFC 2675 specifies that the Length field is set
        to zero if the length of the UDP header plus
        UDP data is greater than 65,535.";
    }
  }
```

3: Page 51

```
            leaf-list verification-tag {
              type uint32;
              description
                "The security policy rule according to
                 udp total length.";
              reference
                "RFC 4960: Stream Control Transmission Protocol
                           - Verification Tag";
            }
```

-> Is this description correct?
-> In my understanding, verification tag would be random values.
I am wondering how we utilize it.

=> [PAUL] We remove the verification tag from the data model as the verification tag value is a random value and would not be useful to match the parameter.

4: Page 52

We don't need packet type for DCCP while supporting chunk types for SCTP?

=> [PAUL] DCCP Packet type is added to the data model as follows:

NEW:

```
container dccp {
...
  leaf-list type {
    type uint8 {
      range "0..15";
    }
    description
      "The security policy rule according to the 4 bits of
       dccp type header field for dccp packet types such as
       DCCP-Request, DCCP-Response, DCCP-Data, DCCP-Ack, and
       DCCP-DataAck.";
    reference
      "RFC 4340: Datagram Congestion Control Protocol (DCCP)
                 - Packet Types";
  }
```

```
}
```

5: Page 70

```
     <tcp>
       <destination-port-number>
        <start>5060</start>
        <start>5061</end>
       </destination-port-number>
     </tcp>
```

   -> should be "<end>5061</end>" ?

=> [PAUL] We have updated the example to follow the new data model as follows:

OLD:

```
<i2nsf-security-policy
xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-policy-rule-for-nsf">
 <system-policy-name>voip_volte_inspection</system-policy-name>
 <rules>
  <rule-name>block_malicious_voice_id</rule-name>
  <condition>
   <ipv4>
    <destination-address>
     <ipv4-range>
      <start>192.0.2.11</start>
      <end>192.0.2.90</end>
     </ipv4-range>
    </destination-address>
   </ipv4>
   <tcp>
    <destination-port-number>
     <start>5060</start>
     <start>5061</end>
    </destination-port-number>
   </tcp>
  </condition>
  <action>
   <advanced-action>
    <content-security-control>
      voip-volte-filter
    </content-security-control>
   </advanced-action>
  </action>
 </rules>
</i2nsf-security-policy>
```

NEW:

```
<i2nsf-security-policy
xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-policy-rule-for-nsf">
 <system-policy-name>voip_volte_inspection</system-policy-name>
 <rules>
  <name>block_malicious_voice_id</name>
  <condition>
   <ipv4>
    <destination-ipv4-network>192.0.2.0/24</destination-ipv4-network>
   </ipv4>
   <tcp>
    <destination-port-number>
     <lower-port>5060</lower-port>
     <upper-port>5061</upper-port>
    </destination-port-number>
   </tcp>
  </condition>
  <action>
   <advanced-action>
    <content-security-control>
      voip-volte-filter
    </content-security-control>
   </advanced-action>
  </action>
 </rules>
</i2nsf-security-policy>
```

6: Page 72

```
      <tcp>
       <destination-port-number>
        <start>80</start>
        <end>80</end>
       </destination-port>
       <destination-port-number>
        <start>443</start>
        <end>443</end>
       </destination-port>
      </tcp>
```

-> should be "</destination-port-number>" instead of "</destination-port>" ?

=> [PAUL] We have updated the example to fix the typo.

OLD:

```
<i2nsf-security-policy
xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-policy-rule-for-nsf">
 <system-policy-name>flood_attack_mitigation</system-policy-name>
 <rules>
  <rule-name>mitigate_http_and_https_flood_attack</rule-name>
  <condition>
   <ipv4>
```

```
    <destination-address>
     <ipv4-range>
      <start>192.0.2.11</start>
      <end>192.0.2.11</end>
     </ipv4-range>
    </destination-address>
   </ipv4>
   <tcp>
    <destination-port-number>
     <start>80</start>
     <end>80</end>
    </destination-port>
    <destination-port-number>
     <start>443</start>
     <end>443</end>
    </destination-port>
   </tcp>
  </condition>
  <action>
   <advanced-action>
    <attack-mitigation-control>
      anti-ddos
    </attack-mitigation-control>
   </advanced-action>
  </action>
 </rules>
</i2nsf-security-policy>
```

NEW:

```
<i2nsf-security-policy
xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-policy-rule-for-nsf">
 <system-policy-name>flood_attack_mitigation</system-policy-name>
 <rules>
  <rule-name>mitigate_http_and_https_flood_attack</rule-name>
  <condition>
   <ipv4>
    <destination-ipv4-network>192.0.2.0/24</destination-ipv4-network>
   </ipv4>
   <tcp>
    <destination-port-number>
     <port-numbers>
      <start>80</start>
      <end>80</end>
     </port-numbers>
     <port-numbers>
      <start>443</start>
      <end>443</end>
     </port-numbers>
    </destination-port-number>
   </tcp>
  </condition>
  <action>
   <advanced-action>
```

```
    <attack-mitigation-control>
      anti-ddos
    </attack-mitigation-control>
   </advanced-action>
  </action>
 </rules>
</i2nsf-security-policy>
```

Reviewer: Kyle Rose

Review result: Has Issues

I have reviewed this document as part of the security directorate's ongoing effort to review all IETF documents being processed by the IESG. These comments were written primarily for the benefit of the security area directors. Document editors and WG chairs should treat these comments just like any other last call comments.

This document Has Issues.

I don't actually have a lot to say about this document from a security perspective: its purpose is to describe, using YANG, a data model intended as the basis for configuration schemas developed for implementations of the Interface to Network Security Functions (I2NSF) framework. Security considerations for the most part should be addressed in documents describing system architecture or normatively detailing how implementations are to make use of the data model described here. I'm not going to relitigate any such issues here.

The main issues I found in this document are ones that I, as someone not terribly familiar with YANG, found troubling from a general engineering perspective. These are best illustrated by example:

*** Why are `ipvX-prefix`, `-range`, and (the misleadingly-named) `-address` defined here? These concepts are generic enough that they should be in modules of their own to minimize variation among data models and the errors that will inevitably result from capturing the same concept in slightly different ways that are not obvious to the user.**

=> [PAUL] We have updated the data model by using RFC 8519 (YANG Data Model for Network Access Control Lists (ACLs)) to remove the above issues. RFC 8519 provides the YANG Data Model for (Ethernet, IPv4, IPv6, ICMP, TCP, and UDP).

*** Overall, I have to imagine that much of the `nsfintf` data model is generic enough that it should be captured externally. For instance, `tcp-flags`, `port-range`, `flow-label`, `dscp`, etc. are generally useful elements of an abstract transport data model that they shouldn't be defined here, but rather incorporated from an external data model that is maintained by those in (for example) the transport area.**

=> [PAUL] We reuse the fields of protocols as elements by importing an external data model in RFC 8519 (YANG Data Model for Network Access Control Lists (ACLs)).

Am I just commenting on the YANG ecosystem in general? If these are standard practices, then the overall ecosystem has major latent problems. Ideally, a particular YANG module seems like it should describe only those elements defined at a particular layer, in this case rules and actions, and use reference external modules for elements that are defined at lower layers.
=> [PAUL] We reuse the elements in RFC 8519 (YANG Data Model for Network Access Control Lists (ACLs)).

Also some nits:

*** `ipvX-address` describes a subspace of the global IPvX address space, not a single address. The name is going to cause problems.**
=> [PAUL] We removed "grouping ipvX-*" in the data model as we utilize RFC 8519 for the condition. The grouping is no longer needed in the data model.

*** The descriptions given are often (usually?) just restatements of the entity name. Example is `identity priority-by-order` described as "Identity for priority by order". The description should provide some value for the user beyond simply restating the name.**
=> [PAUL] The descriptions have been updated to provide more details as follows:

OLD:

```
  identity priority-usage {
    description
      "Base identity for priority usage type.";
  }

  identity priority-by-order {
    base priority-usage;
    description
      "Identity for priority by order";
  }

  identity priority-by-number {
    base priority-usage;
    description
      "Identity for priority by number";
  }

  identity rule-log {
    base log-action;
    description
      "Identity for rule log";
  }

  identity session-log {
    base log-action;
    description
      "Identity for session log";
  }

  identity redirection {
    base egress-action;
    description
```

```
      "Identity for redirection";
  }
```

NEW:

```
  identity priority-usage {
    description
      "Base identity for priority usage type to define the type of
       priority to be implemented in a security policy rule, such
       as priority by order and priority by number.";
  }

  identity priority-by-order {
    base priority-usage;
    description
      "Identity for priority by order. This indicates that the
       priority of a security policy rule follows the order of the
       configuration. The earlier the configuration is, the higher
       the priority is.";
  }

  identity priority-by-number {
    base priority-usage;
    description
      "Identity for priority by number. This indicates the priority
       of a security policy rule follows the number or value of the
       configuration. The higher the value is, the higher the
       priority is.";
  }

  identity rule-log {
    base log-action;
    description
      "Identity for rule log. Log the received packet or flow based
       on the rule.";
  }

  identity session-log {
    base log-action;
    description
      "Identity for session log. Log the tasks that is performed
       during a session.";
  }

  identity redirection {
    base egress-action;
    description
      "Identity for redirection. This action redirects the packet to
       another destination.";
  }
```

*** The headings in section 5 should be clearly labeled with the word "example", such as "Example Security Requirement 1".**

=> [PAUL] We updated the headings for example following your comment as follows:

OLD:

NEW:

*** IPv6 addresses in text MUST be represented in lowercase, according to RFC 5952 section 4.3.**

=> [PAUL] The IPv6 addresses in the example has been updated to use a lowercase instead of the uppercase as follows:

OLD:

```
<i2nsf-security-policy
xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-policy-rule-for-nsf">
 <system-policy-name>sns_access</system-policy-name>
 <rules>
  <rule-name>block_sns_access_during_operation_time</rule-name>
  <event>
    <time>
      <start-date-time>2021-03-11T09:00:00.00Z</start-date-time>
      <end-date-time>2021-12-31T18:00:00.00Z</end-date-time>
      <period>
        <start-time>09:00:00Z</start-time>
        <end-time>18:00:00Z</end-time>
        <day>monday</day>
        <day>tuesday</day>
        <day>wednesday</day>
        <day>thursday</day>
        <day>friday</day>
      </period>
      <frequency>weekly</frequency>
    </time>
  </event>
  <condition>
```

```
   <ipv6>
    <source-address>
     <ipv6-range>
      <start>2001:DB8:0:1::11</start>
      <end>2001:DB8:0:1::90</end>
     </ipv6-range>
    </source-address>
   </ipv6>
  </condition>
  <action>
   <advanced-action>
    <content-security-control>
      url-filtering
    </content-security-control>
   </advanced-action>
  </action>
 </rules>
</i2nsf-security-policy>
```

NEW:

```
<i2nsf-security-policy
xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-policy-rule-for-nsf">
 <system-policy-name>sns_access</system-policy-name>
 <rules>
  <name>block_sns_access_during_operation_time</name>
  <event>
    <time>
      <start-date-time>2021-03-11T09:00:00.00Z</start-date-time>
      <end-date-time>2021-12-31T18:00:00.00Z</end-date-time>
      <period>
        <start-time>09:00:00Z</start-time>
        <end-time>18:00:00Z</end-time>
        <day>monday</day>
        <day>tuesday</day>
        <day>wednesday</day>
        <day>thursday</day>
        <day>friday</day>
      </period>
    </time>
    <frequency>weekly</frequency>
  </event>
  <condition>
   <ipv6>
    <source-ipv6-network>2001:db8:0:1::0/120</source-ipv6-network>
   </ipv6>
  </condition>
  <action>
   <advanced-action>
    <content-security-control>
      url-filtering
    </content-security-control>
   </advanced-action>
  </action>
 </rules>
```

```
</i2nsf-security-policy>
```

I believe that two references in the YANG module need adding to the I-D references
RFC3168
draft-ietf-tcpm-accurate-ecn

=> [PAUL] We updated the data model by utilizing RFC 8519. RFC 3168 and draft-ietf-tcpm-accurate-ecn are no longer referenced in the YANG module. So, the references are not added. And several other references are removed as the references are no longer in use.

The removed references are:

- RFC 2474
- RFC 8344
- IANA-Protocol-Numbers
- IANA-TCP-Parameters
- IANA-ICMP-Parameters
- IANA-ICMPv6-Parameters

Tom Petch

Thanks for your valuable comments.

Best Regards,
Jaehoon (Paul) Jeong