Revision Letter

Editor: Jaehoon Paul Jeong
Date: August 14, 2021

OLD: draft-ietf-i2nsf-capability-data-model-16

NEW: draft-ietf-i2nsf-capability-data-model-17

Dear Tom Petch, Paul Wouters, and Roman Danyliw,

I sincerely appreciate your detailed comments to improve our Capability YANG Data Model.
I have addressed all your comments one by one. I use a black colored bold font for your comments and use a blue colored regular font for my responses with a prefix "=> [PAUL]".

This is one of six I2NSF I-D with YANG modules and there is a lot of overlap between them but the functionality offered, the approach taken, the structuring, the terminology used, varies between them which is likely to create confusion (and errors) in the user. This I-D is mostly a list of some 200 YANG identity which then appear in a YANG list to show what is supported and by implication what is not. (I have extracted the identity and list them at the end of this e-mail since much of what I say is clearer when the list is visible in its entirety). There are similar but different lists of YANG identity in draft-ietf-i2nsf-customer-facing, draft-ietf-i2nsf-nsf-facing, draft-ietf-i2nsf-nsf capability.

I find the choice of identifiers here poor. Some have the suffix -capability, others do not. Given the context in which the module uses them e.g leaf-list sctp-capability, then I find that suffix largely redundant. Ditto the suffix -action in the list of anti-ddos.

=> [Paul] We remove the suffix "capability" in the identity as the module has already specified it as capability.

OLD:

```
  identity system-event-capability {
    base event;
    description
      "Identity for system event";
    reference
      "draft-ietf-i2nsf-nsf-monitoring-data-model-04: I2NSF NSF
       Monitoring YANG Data Model - System event";
  }

.
.
.

  leaf-list system-event-capability {
    type identityref {
      base system-event-capability;
    }
    description
      "System event capabilities";
  }
```

NEW (YELLOW HIGHLIGHT = UPDATED):

```
  identity system-event {
    base event;
    description
      "Identity for system event";
    reference
      "draft-ietf-i2nsf-nsf-monitoring-data-model-09: I2NSF NSF
       Monitoring YANG Data Model - System event";
  }

.
.
.

  leaf-list system-event-capability {
    type identityref {
      base system-event;
    }
    description
      "System event capabilities";
  }
```

List of changed identities by removing “-capability” suffix:

- system-event-capability --> system-event
- system-alarm-capability --> system-alarm
- directional-capability --> directional
- ipv4-capability --> ipv4
- ipv6-capability --> ipv6
- tcp-capability --> tcp
- udp-capability --> udp
- sctp-capability --> sctp
- dccp-capability --> dccp
- icmp-capability --> icmpv4
- icmpv6-capability --> icmpv6
- url-capability --> url-filtering
- log-action-capability --> log-action
- ingress-action-capability --> ingress-action
- egress-action-capability --> egress-action
- default-action-capability --> default-action
- resolution-strategy-capability --> resolution-strategy
- anti-virus-capability --> anti-virus
- anti-ddos-capability --> anti-ddos
- ips-capability --> ips
- voip-volte-capability --> voip-volte-filter

Most identifier are multi-element, e.g., identity prefix-ipv6-address-flow-direction and the element order means that they will collate with all the prefix, range, exact together while ipv6-next-header will be somewhere different. A user likely needs all the ipv4 together, all the ipv6 etc so the identifier should be identity ipv6-address-prefix-flow-direction so that all the ipv6, sctp, udp etc are together. With YANG, as with many languages, the distinction between exact and range is moot; exact is usually modelled with a range with 'start' equal to 'end' so you could almost assume that both variants are supported and it is certainly the least important aspect of the identifier so should come last in the identifier if it is included at all (I would not).

=> [Paul] As you mention the exact value is mostly done with a “start” value equal to an “end” value. We remove the prefixes such as exact- and range- in the identity of the module, as these capabilities can be put together as one identity. We also combine identities that have the same value, e.g., port-number.

An example changes for the exact and range for the identity is as the following:

OLD:

```
  identity exact-tcp-port-num {
    base tcp-capability;
    description
      "Identity for exact-match TCP source or destination port
       number condition capability";
    reference
      "RFC 793: Transmission Control Protocol - Port Number
       draft-ietf-tcpm-rfc793bis: Transmission Control Protocol
       (TCP) Specification";
  }

  identity range-tcp-port-num {
    base tcp-capability;
    description
      "Identity for range-match TCP source or destination port
       number condition capability. The port numbers are
       specified by a pair of a start port number and an end
       port number.";

    reference
      "RFC 793: Transmission Control Protocol - Port Number
       draft-ietf-tcpm-rfc793bis: Transmission Control Protocol
       (TCP) Specification";
  }

  identity exact-udp-port-num {
    base udp-capability;
    description
      "Identity for exact-match UDP source or destination
       port number condition capability";
    reference
      "RFC 768: User Datagram Protocol - Port Number";
  }

  identity range-udp-port-num {
    base udp-capability;
    description
      "Identity for range-match UDP source or destination
       port number condition capability. The port numbers
       are specified by a pair of a start port number and
       an end port number.";
    reference
      "RFC 768: User Datagram Protocol - Port Number";
  }


...

  identity prefix-ipv4-address-flow-direction {
    base ipv4-capability;
    description
      "Identity for flow direction of prefix-match IPv4 source
       or destination address(es) condition capability where flow
       direction is either unidirectional or bidirectional";
    reference
      "RFC 4340: Datagram Congestion Control Protocol";
  }
```

```
  identity prefix-ipv6-address-flow-direction {
    base ipv6-capability;
    description
      "Identity for flow direction of prefix-match IPv6 source
       or destination address(es) condition capability where flow
       direction is either unidirectional or bidirectional";
    reference
      "RFC 8200: Internet Protocol, Version 6 (IPv6)
       Specification - Address";
  }
```

NEW (YELLOW HIGHLIGHT = UPDATED):

```
  identity source-port-number {
    base tcp;
    base udp;
    base sctp;
    base dccp;
    description
      "Identity for matching TCP, UDP, SCTP, and DCCP source port
       number condition capability";
    reference
      "RFC 793: Transmission Control Protocol - Port Number
       draft-ietf-tcpm-rfc793bis: Transmission Control Protocol
       (TCP) Specification
       RFC 768: User Datagram Protocol
       RFC 4960: Stream Control Transmission Protocol
       RFC 4340: Datagram Congestion Control Protocol";
  }

  identity destination-port-number {
    base tcp;
    base udp;
    base sctp;
    base dccp;
    description
      "Identity for matching TCP, UDP, SCTP, and DCCP destination port
       number condition capability";
    reference
      "RFC 793: Transmission Control Protocol - Port Number
       draft-ietf-tcpm-rfc793bis: Transmission Control Protocol
       (TCP) Specification
       RFC 768: User Datagram Protocol
       RFC 4960: Stream Control Transmission Protocol
       RFC 4340: Datagram Congestion Control Protocol";
  }

...

  identity flow-direction {
    base ipv4;
    base ipv6;
    description
      "Identity for flow direction of matching IPv4/IPv6 source
       or destination address(es) condition capability where a flow's
       direction is either unidirectional or bidirectional";
    reference
      "RFC 791: Internet Protocol
       RFC 8200: Internet Protocol, Version 6 (IPv6)
       Specification";
  }
```

Functionally there are fewer protocols than other I2NSF I-D; http only appears as a potential flood while draft-ietf-i2nsf-consumer-facing has ssh, ftp, http, pop3, nat etc and draft-ietf-i2nsf-nsf-facing has ospf, l2tp, eigrp, skip etc. Are these not some of these capabilities?

=> [Paul] The "identity protocol" in draft-ietf-i2nsf-nsf-facing is removed. This identity was supposed to be used for IPv4 Protocol Header Field or equal to IPv6 Next Header Field. It is better to use uint8 type instead of the identity, since each number is represented as a protocol according to the IANA's protocol number assignment (https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml). Hence we remove the identity protocol and next-header, respectively.

Also, the names of the container packet-security-ipv4-condition and packet-security-ipv6-condition are changed to ipv4 and ipv6, respectively.

We change the data model in **draft-ietf-i2nsf-nsf-facing-dm-13** as follows:

OLD (RED COLOR is REMOVED):

```
   identity protocol {
    description
      "Base identity for protocol of IPv4";
    reference
      "IANA: Assigned Internet Protocol Numbers
      RFC 791: Internet Protocol - Protocol";
  }

  identity next-header {
    description
      "Base identity for IPv6 next header";
    reference
      "RFC 8200: Internet Protocol, Version 6 (IPv6)
      Specification - Next Header";
  }

  identity icmp {
    base protocol;
    base next-header;
  .
  .
  .

container packet-security-ipv4-condition {
...
  leaf-list pkt-sec-ipv4-protocol {
    type identityref {
      base protocol;
    }
    description
      "The security policy rule according to
       IPv4 protocol.";
    reference
      "RFC 791: Internet Protocol - Protocol";
  }
}

container packet-security-ipv6-condition {
...
  leaf-list pkt-sec-ipv6-next-header {
    type identityref {
      base next-header;
    }
    description
      "The security policy rule according to
       IPv6 next header.";
```

```
    reference
      "RFC 8200: Internet Protocol, Version 6 (IPv6)
       Specification - Next header";
  }
}
```

NEW (YELLOW HIGHLIGHT = UPDATED):

```
container ipv4 {
  .
  .
  .
  leaf-list protocol {
    type uint8;
    description
      "The security policy rule according to
       IPv4 protocol header field.";
    reference
      "RFC 791: Internet Protocol - Protocol
       IANA: Assigned Internet Protocol Numbers";
  }
}

container ipv6 {
  .
  .
  .
  leaf-list next-header {
    type uint8;
    description
      "The security policy rule according to
       IPv6 next header.";
    reference
      "RFC 8200: Internet Protocol, Version 6 (IPv6)
       Specification - Next header
       IANA: Assigned Internet Protocol Numbers";
  }
}
```

We also change the identity protocol in the three data model drafts, i.e., ietf-i2nsf-nsf-monitoring, ietf-i2nsf-capability, and ietf-i2nsf-consumer-facing-interface. The drafts have the same identity for the protocol as the following:

NEW:

```
base protocol;
  identity ip {
  identity transport-protocol {
  identity application-protocol {

base ip;
  identity ipv4 {
  identity ipv6 {
  identity icmp {

base icmp;
  identity icmpv4 {
  identity icmpv6 {

base transport-protocol;
  identity tcp {
  identity udp {
  identity sctp {
```

```
  identity dccp {

base application-protocol;
  identity http {
  identity https {
  identity ftp {
  identity ssh {
  identity telnet {
  identity smtp {
  identity sftp {
  identity pop3 {
```

YANG allows multiple identity to be derived from a common base which makes it possible easily to e.g. reference just ipv4, or ipv4 and ipv6, or all such protocols. Not here. ipv4-capability is derived from 'identity condition' along with many other quite different capabilities such as

```
    identity context-capability {
```

which renders that YANG capability(!) of little use. draft-nsf-monitoring by contrast derives tcp, udp, icmp from a common base of ipv4 or ipv6, both of which are in turn derived from 'identity ip' which is derived from 'protocol-type' OK I disagree about the suffix -type but the structure there is what other IETF YANG modules widely use and I think right. In this I-D there is no concept of protocol - the identities for protocol are all derived from 'identity condition' (along with much else, unrelated concepts IMHO)

=> [Paul] We have derived the protocol identities with a common base called “protocol” to describe them better.

The changes are as follows:

OLD:

```
  identity condition {
    description
      "Base identity for I2NSF conditions";
  }

  identity ipv4-capability {
    base condition;
    description
      "Base identity for IPv4 condition capability";
    reference
      "RFC 791: Internet Protocol";
  }

  identity ipv6-capability {
    base condition;
    description
      "Base identity for IPv6 condition capabilities";
    reference
      "RFC 8200: Internet Protocol, Version 6 (IPv6)
      Specification";
  }

  identity tcp-capability {
    base condition;
    description
      "Base identity for TCP condition capabilities";
    reference
```

```
    "RFC 793: Transmission Control Protocol
     draft-ietf-tcpm-rfc793bis: Transmission Control Protocol
     (TCP) Specification";
}

identity udp-capability {
  base condition;
  description
    "Base identity for UDP condition capabilities";
  reference
    "RFC 768: User Datagram Protocol";
}

identity sctp-capability {
 description
    "Identity for SCTP condition capabilities";
  reference
    "RFC 4960: Stream Control Transmission Protocol";
}

identity dccp-capability {
 description
    "Identity for DCCP condition capabilities";
  reference
    "RFC 4340: Datagram Congestion Control Protocol";
}
```

NEW:

```
identity protocol {
  description
    "Base identity for Internet Protocols";
}

identity ip {
  base protocol;
  description
    "Base identity for internet/network layer protocol,
     e.g., IPv4, IPv6, and ICMP.";
}

identity ipv4 {
  base ip;
  description
    "Base identity for IPv4 condition capability";
  reference
    "RFC 791: Internet Protocol";
}

identity ipv6 {
  base ip;
  description
    "Base identity for IPv6 condition capabilities";
  reference
    "RFC 8200: Internet Protocol, Version 6 (IPv6)
    Specification";
}

identity icmp {
  base protocol;
  description
    "Base identity for ICMPv4 and ICMPv6 condition capability";
  reference
    "RFC 792: Internet Control Message Protocol
     RFC 4443: Internet Control Message Protocol (ICMPv6)
```

```
      for the Internet Protocol Version 6 (IPv6) Specification
      - ICMPv6";
  }

  identity icmpv4 {
    base icmp;
    description
      "Base identity for ICMPv4 condition capability";
    reference
      "RFC 792: Internet Control Message Protocol";
  }

  identity icmpv6 {
    base icmp;
    description
      "Base identity for ICMPv6 condition capability";
    reference
      "RFC 4443: Internet Control Message Protocol (ICMPv6)
       for the Internet Protocol Version 6 (IPv6) Specification
       - ICMPv6";
  }

  identity transport-protocol {
    base protocol;
    description
      "Base identity for Layer 4 protocol condition capabilities, e.g.,
       TCP, UDP, SCTP, DCCP, and ICMP";
  }

  identity tcp {
    base transport-protocol;
      "Base identity for TCP condition capabilities";
    description
    reference
      "RFC 793: Transmission Control Protocol
       draft-ietf-tcpm-rfc793bis: Transmission Control Protocol
       (TCP) Specification";
  }

  identity udp {
    base transport-protocol;
    description
      "Base identity for UDP condition capabilities";
    reference
      "RFC 768: User Datagram Protocol";
  }

  identity sctp {
    base transport-protocol;
    description
      "Identity for SCTP condition capabilities";
    reference
      "RFC 4960: Stream Control Transmission Protocol";
  }

  identity dccp {
    base transport-protocol;
    description
      "Identity for DCCP condition capabilities";
    reference
      "RFC 4340: Datagram Congestion Control Protocol";
  }
```

identity icmp-capability and identity icmpv6-capability are likewise both derived from 'identity condition' when I would expect them to have a common icmp base given how much

overlap there is between icmp for v4 and icmp for v6 and a likely need to refer the two combined.

I-D nsf-monitoring is quite different here, deriving icmp from a base of either ipv4 or ipv6, while deriving icmpv4 from ipv4 and icmpv6 from ipv6. I think that approach much superior both in the choice of identifiers, icmpv4 and icmpv6 as opposed to icmp and icmpv6, and in allowing a simple reference to icmp in all its forms along with ipv4 or ipv6 specific when needed.

=> [Paul] We change the identity by adding the general “identity icmp” as the common base for icmpv4 and icmpv6. The changes are as follows:

OLD:

```
identity icmp-capability {
  base condition;
  description
    "Base identity for ICMP condition capability";
  reference
    "RFC 792: Internet Control Message Protocol";
}

identity icmp-type {
  base icmp-capability;
  description
    "Identity for ICMP type condition capability";
  reference
    "RFC 792: Internet Control Message Protocol";
}

identity icmp-code {
  base icmp-capability;
  description
    "Identity for ICMP code condition capability";
  reference
    "RFC 792: Internet Control Message Protocol";
}

identity icmpv6-capability {
  base condition;
  description
    "Base identity for ICMPv6 condition capability";
  reference
    "RFC 4443: Internet Control Message Protocol (ICMPv6)
     for the Internet Protocol Version 6 (IPv6) Specification
     - ICMPv6";
}

identity icmpv6-type {
  base icmpv6-capability;
  description
    "Identity for ICMPv6 type condition capability";
  reference
    "RFC 4443: Internet Control Message Protocol (ICMPv6)
     for the Internet Protocol Version 6 (IPv6) Specification
     - ICMPv6";
}

identity icmpv6-code {
  base icmpv6-capability;
  description
    "Identity for ICMPv6 code condition capability";
  reference
    "RFC 4443: Internet Control Message Protocol (ICMPv6)
     for the Internet Protocol Version 6 (IPv6) Specification
     - ICMPv6";
```

```
}
```

NEW (YELLOW HIGHLIGHT = UPDATED):

```
identity icmp {
  base protocol;
  description
    "Base identity for ICMPv4 and ICMPv6 condition capability";
  reference
    "RFC 792: Internet Control Message Protocol
     RFC 4443: Internet Control Message Protocol (ICMPv6)
     for the Internet Protocol Version 6 (IPv6) Specification
     - ICMPv6";
}

identity icmpv4 {
  base icmp;
  description
    "Base identity for ICMPv4 condition capability";
  reference
    "RFC 792: Internet Control Message Protocol";
}

identity icmpv6 {
  base icmp;
  description
    "Base identity for ICMPv6 condition capability";
  reference
    "RFC 4443: Internet Control Message Protocol (ICMPv6)
     for the Internet Protocol Version 6 (IPv6) Specification
     - ICMPv6";
}

identity type {
  base icmpv4;
  base icmpv6;
  description
    "Identity for ICMPv4 and ICMPv6 type condition capability";
  reference
    "RFC 792: Internet Control Message Protocol
     RFC 4443: Internet Control Message Protocol (ICMPv6)
     for the Internet Protocol Version 6 (IPv6) Specification
     - ICMPv6";
}

     "Base identity for ICMPv4 and ICMPv6 condition capability";
  reference
     "RFC 792: Internet Control Message Protocol";
      RFC 4443: Internet Control Message Protocol (ICMPv6)
identity code {
  base icmpv4;
  base icmpv6;
  description
    "Identity for ICMPv4 and ICMPv6 code condition capability";
  reference
    "RFC 792: Internet Control Message Protocol
     RFC 4443: Internet Control Message Protocol (ICMPv6)
     for the Internet Protocol Version 6 (IPv6) Specification
     - ICMPv6";
}
```

This I-D models identity ipv4-tos-dscp and identity ipv6-traffic-class-dscp/identity exact-ipv6-flow-label; nsf-facing has choices such as minimize-cost,

maximize-reliability derived from a choice of either identity type-of-service or identity traffic-class which seems to be more in line with current thinking.

=> [Paul] The identity minimize-cost, maximize-reliability, maximize-throughput, minimize-delay, and maximize-security are removed. These identities are not used in the module as the leaf-list is changed into “inet:dscp” in NSF-Facing Interface.

Below changes are made in the draft-ietf-i2nsf-nsf-facing-interface-dm-13:

OLD:

```
container packet-security-ipv4-condition {
  ...
  leaf-list pkt-sec-ipv4-tos {
    type identityref {
      base type-of-service;
    }
    description
      "The security policy rule according to
       IPv4 type of service.";
    reference
      "RFC 791: Internet Protocol - Type of service";
  }
}

container packet-security-ipv6-condition {
  ...
  leaf-list pkt-sec-ipv6-traffic-class {
    type identityref {
      base traffic-class;
    }
    description
      "The security policy rule according to
       IPv6 traffic class.";
    reference
      "RFC 8200: Internet Protocol, Version 6 (IPv6)
       Specification - Traffic class";
  }
}
```

NEW (YELLOW HIGHLIGHT = UPDATED):

```
container ipv4 {
  ...
  leaf-list dscp {
    type inet:dscp;
    description
      "The security policy rule according to
       IPv4 type of service for DSCP.";
    reference
      "RFC 791: Internet Protocol - Type of service
       RFC 2474: Definition of the Differentiated
       Services Field (DS Field) in the IPv4 and
       IPv6 Headers.";
  }
}

container ipv6 {
  ...
  leaf-list dscp {
    type inet:dscp;
    description
      "The security policy rule according to
```

```
       IPv6 traffic class for DSCP.";
    reference
      "RFC 8200: Internet Protocol, Version 6 (IPv6)
       Specification - Traffic class
       RFC 2474: Definition of the Differentiated
       Services Field (DS Field) in the IPv4 and
       IPv6 Headers.";
  }
}
```

With alarms, this I-D is more in line with others but with different terminology, here it is memory-alarm, cpu-alarm, disk-alarm, hardware-alarm, interface-alarm. I-D nsf-facing is the same but I-D nsf-monitoring has mem-usage-alarm, cpu-usage-alarm, disk-usage-alarm, hw-failure-alarm, ifnet-state-alarm. Not wrong but confusingly different.

=> [Paul] The names of nsf-monitoring data model identities are changed to synchronize with the names of the capability data model identities.
The below changes are in the draft-ietf-i2nsf-nsf-monitoring-data-model-09.

OLD:

```
  identity mem-usage-alarm {
    base alarm-type;
    description
      "A memory alarm is alerted.";
  }
  identity cpu-usage-alarm {
    base alarm-type;
    description
      "A CPU alarm is alerted.";
  }
  identity disk-usage-alarm {
    base alarm-type;
    description
      "A disk alarm is alerted.";
  }
  identity hw-failure-alarm {
    base alarm-type;
    description
      "A hardware alarm is alerted.";
  }
  identity ifnet-state-alarm {
    base alarm-type;
    description
      "An interface alarm is alerted.";
  }
```

NEW (YELLOW HIGHLIGHT = UPDATED):

```
  identity event {
    description
      "Base identity for I2NSF events.";
  }

  identity system-alarm {
    base event;
    description
      "Base identity for detectable system alarm types";
  }
  identity memory-alarm {
    base system-alarm;
    description
      "A memory alarm is alerted.";
```

```
  }
  identity cpu-alarm {
    base system-alarm;
    description
      "A CPU alarm is alerted.";
  }
  identity disk-alarm {
    base system-alarm;
    description
      "A disk alarm is alerted.";
  }
  identity hardware-alarm {
    base system-alarm;
    description
      "A hardware alarm (i.e., hardware failure) is alerted.";
  }
  identity interface-alarm {
    base system-alarm;
    description
      "An interface alarm is alerted.";
  }
```

This I-D has context-capability e.g. user, geography, ACL which I struggle to relate to the other I-D. (In passing, I would have thought ACL a sine qua non for any I2NSF work).

=> [Paul] The capability of ACL in an NSF is essential and discussed in the Security Consideration of the ietf-i2nsf-capability-data-model document. We removed the identity ACL from the NSF capability data model because the NACM YANG module can specify an access control list for an I2NSF interface such as Consumer-Facing Interface and NSF-Facing Interface. The other identity shows the context capability for the NSF. The NSF with context capability can check the information of a packet or flow such as a packet's source geographical location and a packet's target device type.

We change the identities into the following:

OLD:	NEW:
base context; identity access-control-list; identity application-layer-filter; identity target; identity user; identity group; identity geography;	base target-device; identity computer { identity mobile-phone { identity voip-volte-phone identity tablet { identity network-infrastructure-device { identity iot { identity vehicle { base user-condition; identity user { identity group { identity geographical-location; identity source-location { identity destination-location {

This I-D derives identity rule-log, identity session-log from log-action-capability. nsf-facing is the same but consumer-facing has identity log, identity syslog, identity session-log derived from secondary-action;

=> [Paul] The consumer-facing data model is changed to match with the NSF-facing and capability data model in terms of log identities.

The below changes are done in the draft-ietf-i2nsf-consumer-facing-interface-14.

OLD:

```
identity secondary-action {
   description
     "This field identifies additional actions if a rule is
      matched. This could be one of 'LOG', 'SYSLOG',
      'SESSION-LOG', etc.";
 }

 identity log {
   base secondary-action;
   description
     "The identity for logging.";
 }

 identity syslog {
   base secondary-action;
   description
     "The identity for system logging.";
 }

 identity session-log {
   base secondary-action;
   description
     "The identity for session logging.";
 }
```

NEW (YELLOW HIGHLIGHT = UPDATED):

```
 identity log-action {
   description
     "Base identity for log actions, such as rule-log and
     session-log action.";
 }

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

 identity session-log {
   base log-action;
   description
     "Identity for session log-action capability.
      Log the received packet based on the session.";
 }
```

Here ingress-action/egress-action/default-action are a common base for identity pass, identity drop, identity alert, identity mirror while I-D consumer-facing has primary-action as a base for identities pass, drop, alert, rate-limit, mirror and I-D nsf-facing has pass, drop, reject, alert, mirror.

=> [Paul] We change the action identity to properly follow the RFC8329 as it should be the reference for the I2NSF Data Model. In RFC8329, the actions are pass, drop, rate limiting, mirroring, invoke signaling, tunnel encapsulation, packet forwarding, and/or transformation. In the case of “alert”, it is similar to a mirror action for monitoring the traffic. We remove the action “alert” from all of the I2NSF data models. We also add the action “rate limiting” action to the data models.

OLD:

```
identity pass {
  base ingress-action-capability;
  base egress-action-capability;
  base default-action-capability;
  description
    "Identity for pass action capability";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Ingress, egress, and pass actions.";
}

identity drop {
  base ingress-action-capability;
  base egress-action-capability;
  base default-action-capability;
  description
    "Identity for drop action capability";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Ingress, egress, and drop actions.";
}

identity alert {
  base ingress-action-capability;
  base egress-action-capability;
  base default-action-capability;
  description
    "Identity for alert action capability";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Ingress, egress, and alert actions.
     draft-ietf-i2nsf-nsf-monitoring-data-model-04: I2NSF
     NSF Monitoring YANG Data Model - Alarm (i.e., alert).";
}

identity mirror {
  base ingress-action-capability;
  base egress-action-capability;
  base default-action-capability;
  description
    "Identity for mirror action capability";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Ingress, egress, and mirror actions.";
}
```

NEW (YELLOW HIGHLIGHT = UPDATED):

```
identity pass {
  base ingress-action;
  base egress-action;
  base default-action;
  description
    "Identity for pass action capability";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Ingress, egress, and pass actions.";
}

identity drop {
  base ingress-action;
  base egress-action;
  base default-action;
  description
    "Identity for drop action capability";
  reference
    "RFC 8329: Framework for Interface to Network Security
```

```
      Functions - Ingress, egress, and drop actions.";
  }

  identity rate-limit {
    base ingress-action;
    base egress-action;
    base default-action;
    description
      "Identity for rate limiting action capability.";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Ingress, egress, and rate limiting actions.";
  }

  identity mirror {
    base ingress-action;
    base egress-action;
    base default-action;
    description
      "Identity for mirror action capability.";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Ingress, egress, and mirror actions.";
  }
```

resolution-strategy-capability is the base for identity fmr, identity lmr, identity pmr, identity pmre, identity pmrn and this is the same as I-D nsf-facing(!)

=> [Paul] We change the identity resolution-strategy-capability into resolution-strategy without the suffix “-capability” which is the same between the ietf-i2nsf-capability and the ietf-i2nsf-nsf-facing-interface.

OLD:

```
  identity resolution-strategy-capability {
    description
      "Base identity for resolution strategy capability";
  }

  identity fmr {
    base resolution-strategy-capability;
    description
      "Identity for First Matching Rule (FMR) resolution
       strategy capability";
  }

  identity lmr {
    base resolution-strategy-capability;
    description
      "Identity for Last Matching Rule (LMR) resolution
       strategy capability";
  }

  identity pmr {
    base resolution-strategy-capability;
    description
      "Identity for Prioritized Matching Rule (PMR) resolution
       strategy capability";
  }

  identity pmre {
    base resolution-strategy-capability;
    description
      "Identity for Prioritized Matching Rule with Errors (PMRE)
```

```
      resolution strategy capability";
  }

  identity pmrn {
    base resolution-strategy-capability;
    description
      "Identity for Prioritized Matching Rule with No Errors (PMRN)
       resolution strategy capability";
  }
```

NEW:

```
  identity resolution-strategy {
    description
      "Base identity for resolution strategy capability";
  }

  identity fmr {
    base resolution-strategy;
    description
      "Identity for First Matching Rule (FMR) resolution
       strategy capability";
  }

  identity lmr {
    base resolution-strategy;
    description
      "Identity for Last Matching Rule (LMR) resolution
       strategy capability";
  }

  identity pmr {
    base resolution-strategy;
    description
      "Identity for Prioritized Matching Rule (PMR) resolution
       strategy capability";
  }

  identity pmre {
    base resolution-strategy;
    description
      "Identity for Prioritized Matching Rule with Errors (PMRE)
       resolution strategy capability";
  }

  identity pmrn {
    base resolution-strategy;
    description
      "Identity for Prioritized Matching Rule with No Errors (PMRN)
       resolution strategy capability";
  }
```

This I-D uses advanced-nsf-capability as the base for anti-virus-capability, anti-ddos-capability, ips-capability, voip-volte-capability. I-D consumer-facing has security-event-type as a base for ddos, spyware, trojan, ransomware while I-D nsf-facing uses content-security-control as a base for firewall, antivirus, ips, ids, url-filtering, voip-volte, mail-filtering, file-blocking, pkt-capture, application-control and then I-D nsf-monitoring has nsf-attack-type as a base for botnet-attack-type and virus-type, and virus-type is then the base for trojan, worm, macro. Here, antivirus is the base for identity detect and identity allow-list which for me is a completely different set of concepts.

=> [Paul] We change the information model for the capability such that the advanced-nsf defines two new categories for content-security-control and attack-mitigation-control. This document only defines content-security-control as IPS, URL-filtering, antivirus, and VoIP/VoLTE filter, while it defines attack-mitigator-control as anti-ddos. This can be extended with other Network Security Functions in the future. All other data model documents are synchronized to follow this instruction. For the antivirus, in this data model, it shows the antivirus capability for security attack defense, while in the monitoring data model, the identity of the type of virus shows a type of security attack rather than defense.

The information model explains it as follows:

OLD:
In this document, two kinds of condition capabilities are used to classify different capabilities of NSFs such as generic-nsf-capabilities for generic NSFs and advanced-nsf-capabilities for advanced NSFs. First, the generic-nsf-capabilities define the common capabilities of NSFs such as IPv4 capability, IPv6 capability, TCP capability, UDP capability, SCTP capability, DCCP capability, ICMP capability, and ICMPv6 capability. Second, the advanced-nsf-capabilities define advanced capabilities of NSFs such as anti-virus capability, anti-DDoS capability, Intrusion Prevention System (IPS) capability, HTTP capability, and VoIP/VoLTE capability.

NEW:
In this document, two kinds of condition capabilities are used to classify different capabilities of NSFs such as generic-nsf-capabilities and advanced-nsf-capabilities. First, the generic-nsf-capabilities define NSFs that operate on packet header for layer 2 (i.e., Ethernet capability), layer 3 (i.e., IPv4 capability, IPv6 capability, ICMPv4 capability, and ICMPv6 capability.), and layer 4 (i.e., TCP capability, UDP capability, SCTP capability, and DCCP capability). Second, the advanced-nsf-capabilities define NSFs that operate on features different from the generic-nsf-capabilities, e.g., the payload, cross flow state, application layer, traffic statistics, network behavior, etc. This document defines the advanced-nsf into two categories such as content-security-control and attack-mitigation-control. • Content security control is an NSF that evaluates the payload of a packet, such as Intrusion Prevention System (IPS), URL-Filtering, Antivirus, and VoIP/VoLTE Filter. • Attack mitigation control is an NSF that mitigates an attack such as anti-DDoS (DDoS-mitigator). The advanced-nsf can be extended with other types of NSFs. This document only provides five advanced-nsf capabilities, i.e., IPS capability, URL-Filtering capability, Antivirus capability, VoIP/VoLTE Filter capability, and Anti-DDoS capability. Note that VoIP and VoLTE are merged into a single capability in this document because VoIP and VoLTE use the Session Initiation Protocol (SIP) [RFC3261] for a call setup. See Section 3.1 for more information about the condition in the ECA policy model.

We change the names of the identities in the ietf-i2nsf-capability data model to be the same with those of the identities in the ietf-i2nsf-nsf-facing-interface data model as follows:

OLD (ietf-i2nsf-capability):

```
  identity advanced-nsf-capability {
    description
      "Base identity for advanced Network Security Function (NSF)
       capability.  This can be used for advanced NSFs such as
       Anti-Virus, Anti-DDoS Attack, IPS, and VoIP/VoLTE Security
       Service.";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF capability";
```

```
}

identity anti-virus-capability {
  base advanced-nsf-capability;
  description
    "Identity for advanced NSF Anti-Virus capability.
     This can be used for an extension point for Anti-Virus
     as an advanced NSF.";
     reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Advanced NSF Anti-Virus capability";
}

identity anti-ddos-capability {
  base advanced-nsf-capability;
  description
    "Identity for advanced NSF Anti-DDoS Attack capability.
     This can be used for an extension point for Anti-DDoS
     Attack as an advanced NSF.";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Advanced NSF Anti-DDoS Attack capability";
}

identity ips-capability {
  base advanced-nsf-capability;
  description
    "Identity for advanced NSF IPS capabilities.  This can be
     used for an extension point for IPS as an advanced NSF.";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Advanced NSF IPS capability";
}

identity voip-volte-capability {
  base advanced-nsf-capability;
  description
    "Identity for advanced NSF VoIP/VoLTE Security Service
     capability.  This can be used for an extension point
     for VoIP/VoLTE Security Service as an advanced NSF.";
  reference
    "RFC 3261: SIP: Session Initiation Protocol";
}
```

NEW (ietf-i2nsf-capability) :

```
identity advanced-nsf {
  description
    "Base identity for advanced Network Security Function (NSF)
     capability.";
}

identity content-security-control {
  base advanced-nsf;
  description
    "Base identity for content security control. Content security
     control is an NSF that evaluates a packet's payload such as
     Intrusion Prevention System (IPS), URL-Filtering, Antivirus,
     and VoIP/VoLTE Filter.";
}

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
```

```
      or DDoS-mitigator.";
}

identity ips {
  base content-security-control;
  description
    "Base identity for IPS (Intrusion Prevention System) capability
     that prevents malicious activity within a network";
}

identity url-filtering {
  base content-security-control;
  description
    "Base identity for url filtering capability that limits access by
     comparing the web traffic's URL with the URLs for web filtering
     in a database";
}

identity anti-virus {
  base content-security-control;
  description
    "Base identity for antivirus capability to protect the network
     by detecting and removing viruses or malwares.";
}

identity voip-volte-filtering {
  base content-security-control;
  description
    "Base identity for advanced NSF VoIP/VoLTE Security Service
     capability to filter the VoIP/VoLTE packets or flows.";
  reference
    "RFC 3261: SIP: Session Initiation Protocol";
}

identity anti-ddos {
  base content-security-control;
  description
    "Base identity for advanced NSF Anti-DDoS Attack or DDoS Mitigator
     capability.";
}
```

OLD (ietf-i2nsf-nsf-facing-interface):

```
identity content-security-control {
  description
    "Base identity for content security control";
  reference
    "RFC 8329: Framework for Interface to
     Network Security Functions - Flow-Based
     NSF Capability Characterization
     draft-ietf-i2nsf-capability-data-model-15:
     I2NSF Capability YANG Data Model";
}

identity firewall {
  base content-security-control;
  description
    "Identity for firewall that monitors
     incoming and outgoing network traffic
     and permits or blocks data packets based
     on a set of security rules.";
}

identity antivirus {
```

```
  base content-security-control;
  description
    "Identity for antivirus that prevents,
     scans, detects and deletes viruses
     from a computer";
}

identity ips {
  base content-security-control;
  description
    "Identity for IPS (Intrusion Prevention System)
    that prevents malicious activity within a network";
}

...

identity attack-mitigation-control {
  description
    "Base identity for attack mitigation control";
  reference
    "RFC 8329: Framework for Interface to
     Network Security Functions - Flow-Based
     NSF Capability Characterization
     draft-ietf-i2nsf-capability-data-model-15:
     I2NSF Capability YANG Data Model";
}

identity syn-flood {
  base attack-mitigation-control;
  description
    "Identity for syn flood
     that weakens the SYN flood attack";
}

identity udp-flood {
  base attack-mitigation-control;
  description
    "Identity for udp flood
     that weakens the UDP flood attack";
}

...
```

NEW (ietf-i2nsf-nsf-facing-interface):

```
identity advanced-nsf {
  description
    "Base identity for advanced Network Security Function (NSF)
     capability.  This can be used for advanced NSFs such as
     Anti-DDoS Attack, IPS, URL-Filtering, Antivirus
     and VoIP/VoLTE Filter.";
  reference
    "draft-ietf-i2nsf-capability-data-model-17:
     I2NSF Capability YANG Data Model";
}

identity content-security-control {
  base advanced-nsf;
  description
    "Base identity for content security control";
  reference
    "draft-ietf-i2nsf-capability-data-model-17:
     I2NSF Capability YANG Data Model";
}

identity ips {
```

```
    base content-security-control;
    description
      "Identity for IPS (Intrusion Prevention System)
       that prevents malicious activity within a network";
  }

  identity url-filtering {
    base content-security-control;
    description
      "Identity for url filtering that limits access by comparing the
       web traffic's URL with the URLs for web filtering in a
       database";
  }

  identity anti-virus {
    base content-security-control;
    description
      "Identity for antivirus to protect the network by detecting and
       removing viruses or malwares.";
  }

  identity voip-volte-filter {
    base content-security-control;
    description
      "Identity for VoIP/VoLTE security service that filters out the
       packets or flows of malicious users with a deny list of
       malicious users in a database";
  }

  identity attack-mitigation-control {
    base advanced-nsf;
    description
      "Base identity for attack mitigation control";
    reference
      "draft-ietf-i2nsf-capability-data-model-17:
       I2NSF Capability YANG Data Model";
  }

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

Returning to ddos, here the I-D diverge widely. This I-D has 10 derived identity, with the only appearance of http in the I-D but which it separates from https; likewise it splits icmp-flood from icmpv6 flood (without a common icmp base) and dns-request from dns-reply, again the only appearance of the dns protocol and, again, with no common base.

=> [Paul] We change the identity for anti-ddos-capability. Http-flood, https-flood, etc. does not represent a capability of an NSF, instead it is representing the attack. We change it into packet-rate, flow-rate, and byte-rate, which is a capability to detect the rate of packets coming into the network based on the number of packets, flows, or bytes.

The following are changed in the ietf-i2nsf-capability data model:

OLD:

```
  identity anti-ddos-capability {
    base advanced-nsf-capability;
    description
```

```
    "Identity for advanced NSF Anti-DDoS Attack capability.
     This can be used for an extension point for Anti-DDoS
     Attack as an advanced NSF.";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Advanced NSF Anti-DDoS Attack capability";
}

identity syn-flood-action {
  base anti-ddos-capability;
  description
    "Identity for advanced NSF Anti-DDoS SYN Flood Action
     capability.  This can be used for an extension point for
     Anti-DDoS SYN Flood Action as an advanced NSF.";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Advanced NSF Anti-DDoS SYN Flood Action
     capability";
}

identity udp-flood-action {
  base anti-ddos-capability;
  description
    "Identity for advanced NSF Anti-DDoS UDP Flood Action
     capability.  This can be used for an extension point for
     Anti-DDoS UDP Flood Action as an advanced NSF.";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Advanced NSF Anti-DDoS UDP Flood Action
     capability";
}

identity http-flood-action {
  base anti-ddos-capability;
  description
    "Identity for advanced NSF Anti-DDoS HTTP Flood Action
     capability.  This can be used for an extension point for
     Anti-DDoS HTTP Flood Action as an advanced NSF.";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Advanced NSF Anti-DDoS HTTP Flood Action
     capability";
}

identity https-flood-action {
  base anti-ddos-capability;
  description
    "Identity for advanced NSF Anti-DDoS HTTPS Flood Action
     capability.  This can be used for an extension point for
     Anti-DDoS HTTPS Flood Action as an advanced NSF.";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Advanced NSF Anti-DDoS HTTPS Flood Action
     capability";
}

identity dns-request-flood-action {
  base anti-ddos-capability;
  description
    "Identity for advanced NSF Anti-DDoS DNS Request Flood
     Action capability.  This can be used for an extension
     point for Anti-DDoS DNS Request Flood Action as an
     advanced NSF.";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Advanced NSF Anti-DDoS DNS Request Flood
     Action capability";
}
```

```
identity dns-reply-flood-action {
  base anti-ddos-capability;
  description
    "Identity for advanced NSF Anti-DDoS DNS Reply Flood
     Action capability.  This can be used for an extension
     point for Anti-DDoS DNS Reply Flood Action as an
     advanced NSF.";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Advanced NSF Anti-DDoS DNS Reply Flood
     Action capability";
}

identity icmp-flood-action {
  base anti-ddos-capability;
  description
    "Identity for advanced NSF Anti-DDoS ICMP Flood Action
     capability.  This can be used for an extension point
     for Anti-DDoS ICMP Flood Action as an advanced NSF.";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Advanced NSF Anti-DDoS ICMP Flood Action
     capability";
}

identity icmpv6-flood-action {
  base anti-ddos-capability;
  description
    "Identity for advanced NSF Anti-DDoS ICMPv6 Flood Action
     capability.  This can be used for an extension point
     for Anti-DDoS ICMPv6 Flood Action as an advanced NSF.";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Advanced NSF Anti-DDoS ICMPv6 Flood Action
     capability";
}

identity sip-flood-action {
  base anti-ddos-capability;
  description
    "Identity for advanced NSF Anti-DDoS SIP Flood Action
     capability.  This can be used for an extension point
     for Anti-DDoS SIP Flood Action as an advanced NSF.";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Advanced NSF Anti-DDoS SIP Flood Action
     capability";
}

identity detect-mode {
  base anti-ddos-capability;
  description
    "Identity for advanced NSF Anti-DDoS Detection Mode
     capability.  This can be used for an extension point
     for Anti-DDoS Detection Mode as an advanced NSF.";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Advanced NSF Anti-DDoS Detection Mode
     capability";
}

identity baseline-learning {
  base anti-ddos-capability;
  description
    "Identity for advanced NSF Anti-DDoS Baseline Learning
     capability.  This can be used for an extension point
     for Anti-DDoS Baseline Learning as an advanced NSF.";
```

```
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF Anti-DDoS Baseline Learning
       capability";
  }
```

NEW:

```
  identity anti-ddos {
    base advanced-nsf;
    description
      "Identity for advanced NSF Anti-DDoS Attack capability.
       This can be used for an extension point for Anti-DDoS
       Attack as an advanced NSF.";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF Anti-DDoS Attack capability";
  }

  identity packet-rate {
    base anti-ddos;
    description
      "Identity for advanced NSF Anti-DDoS detecting Packet Rate
       Capability where a packet rate is defined as the arrival rate of
       packets toward a victim destination node.  The NSF with this
       capability can detect the incoming packet rate and create an
       alert if the rate exceeds the threshold.";
  }

  identity flow-rate {
    base anti-ddos;
    description
      "Identity for advanced NSF Anti-DDoS detecting Flow Rate
       Capability where a flow rate is defined as the arrival rate of
       flows towards a victim destination node.  The NSF with this
       capability can detect the incoming flow rate and create an
       alert if the rate exceeds the threshold.";
  }

  identity byte-rate {
    base anti-ddos;
    description
      "Identity for advanced NSF Anti-DDoS detecting Byte Rate
       Capability where a byte rate is defined as the arrival rate of
       Bytes toward a victim destination node.  The NSF with this
       capability can detect the incoming byte rate and create an
       alert if the rate exceeds the threshold.";
  }
```

I-D nsf-monitoring uses a base of flood type from which it derives 14 identity adding syn-ack, fin-rst, tcp-con. It keeps the dns-request, dns-reply but gives a choice of icmp, icmpv4, icmpv6 but does not derive the last two from the first. (What is meant by 'icmp'? beware, it all depends on the author, could be v4 could be v4 and v6, you cannot tell).

=> [Paul] The “identity icmp-flood” is removed to prevent confusion when using the data model by keeping “identity icmpv4-flood” and “identity icmpv6-flood”.

The below change is done in the draft-ietf-i2nsf-nsf-monitoring-data-model-09:

Removed:

```
  identity icmp-flood {
    base flood-type;
```

```
    description
      "Either an ICMPv4 or ICMPv6 flood is detected.";
  }
```

I-D nsf-facing is quite close to I-D nsf-monitoring for ddos but uses a base of attack-mitigation-control; joins http and https in identity http-and-https-flood, splits dns into dns and dns-amp rather than query and reply, adds oversized-icmp (icmp-... please) and then tacks on ip-sweep, port-scanning, ping-of-death, teardrop, tracert which may be attacks but are not ddos and would seem to make no appearance in this capabilities I-D.

=> [Paul] We define the attack-mitigation-control in the capability information model. The I2NSF data model documents only discuss DDoS mitigator, that is, anti-DDoS. All of the floods can be simplified as anti-DDoS, while the other attack-mitigations are not discussed. Hence, we removed the other attack-mitigations to synchronize the data models.

We change the identity of attack-mitigation-control in ietf-i2nsf-nsf-facing-interface YANG module to follow the capability as the following:

OLD:

```
  identity attack-mitigation-control {
    description
      "Base identity for attack mitigation control";
    reference
      "RFC 8329: Framework for Interface to
       Network Security Functions - Flow-Based
       NSF Capability Characterization
       draft-ietf-i2nsf-capability-data-model-15:
       I2NSF Capability YANG Data Model";
  }

  identity syn-flood {
    base attack-mitigation-control;
    description
      "Identity for syn flood
       that weakens the SYN flood attack";
  }

  identity udp-flood {
    base attack-mitigation-control;
    description
      "Identity for udp flood
       that weakens the UDP flood attack";
  }

.
.
.

  leaf-list attack-mitigation-control {
      type identityref {
        base attack-mitigation-control;
      }
      description
        "Attack-mitigation-control is the NSFs that weaken
         the attacks related to a denial of service
         and reconnaissance.
         The Profile is divided into content security
         control and attack-mitigation-control.
         Attack mitigation control: syn flood, udp flood,
         icmp flood, ip frag flood, ipv6 related, http flood,
         https flood, dns flood, dns amp flood, ssl ddos,
         ip sweep, port scanning, ping of death, teardrop,
```

```
        oversized icmp, tracert.";
  }
```

NEW:

```
  identity attack-mitigation-control {
    base advanced-nsf;
    description
      "Base identity for attack mitigation control";
    reference
      "draft-ietf-i2nsf-capability-data-model-17:
       I2NSF Capability YANG Data Model";
  }

  identity anti-ddos {
    base attack-mitigation-control;
    description
      "Identity for advanced NSF Anti-DDoS or DDoS Mitigator
       capability.";
  }
.
.
.
container advanced-action {
  description
    "If the packet needs to be additionally inspected,
     the packet is passed to advanced network
     security functions according to the profile.
     The profile means the types of NSFs where the packet
     will be forwarded in order to additionally
     inspect the packet.
     The advanced action activates Service Function
     Chaining (SFC) for further inspection of a packet.";

 leaf-list content-security-control {
    type identityref {
      base content-security-control;
    }
    description
      "Content-security-control is the NSFs that
       inspect the payload of the packet.
       The profile is divided into content security
       control and attack-mitigation-control.
       Content security control: ips, url filtering, anti-virus,
       and voip-volte-filter. This can be extended according
       to the provided NSFs.";
    reference
      "draft-ietf-i2nsf-capability-data-model-17:
       I2NSF Capability YANG Data Model - YANG Tree Diagram";
  }

  leaf-list attack-mitigation-control {
    type identityref {
      base attack-mitigation-control;
    }
    description
      "Attack-mitigation-control is the NSFs that weaken
       the attacks related to a denial of service
       and reconnaissance.
       The profile for the types of NSFs for mitigation is
       divided into content security control and
       attack-mitigation-control.
       Attack mitigation control: Anti-DDoS or DDoS mitigator.
       This can be extended according to the provided NSFs such
       as mitigators for ip sweep, port scanning, ping of death,
       teardrop, oversized icmp, and tracert.";
```

```
    reference
      "draft-ietf-i2nsf-capability-data-model-17:
       I2NSF Capability YANG Data Model - YANG Tree Diagram";
  }
```

This I-D uses voip-volte-capability as a base for voip-volte-call-id and identity user-agent. I-D nsf-facing derives identity voip-volte from content-security-control (not a concept I see in the capabilities I-D) while draft-ietf-i2nsf-registration-interface-dm would appear to expect there to be an identity 'voice-id'.

=> [Paul] We changed the name of the identity of voip-volte into voip-volte-filtering to describe it as an NSF instead of VoIP or VoLTE in pages 19 and 21 in this revision letter. The draft-ietf-i2nsf-registration-interface-dm's examples were not updated to follow the current Capability Data Model. We have updated its examples accordingly as follows:

OLD:

```
<nsf-registrations
  xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-reg-interface"
  xmlns:cap="urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability">
  <nsf-information>
    <capability-name>voip_volte_filter</capability-name>
    <nsf-capability-info>
      <security-capability>
        <condition-capabilities>
          <advanced-nsf-capabilities>
            <voip-volte-capability>cap:voice-id</voip-volte-capability>
          </advanced-nsf-capabilities>
        </condition-capabilities>
        <action-capabilities>
          <ingress-action-capability>cap:pass</ingress-action-capability>
          <ingress-action-capability>cap:drop</ingress-action-capability>
          <ingress-action-capability>cap:alert</ingress-action-capability>
          <egress-action-capability>cap:pass</egress-action-capability>
          <egress-action-capability>cap:drop</egress-action-capability>
          <egress-action-capability>cap:alert</egress-action-capability>
        </action-capabilities>
        <ipsec-method>cap:ikeless</ipsec-method>
      </security-capability>
      <performance-capability>
        <processing>
          <processing-average>1000</processing-average>
          <processing-peak>5000</processing-peak>
        </processing>
        <bandwidth>
          <outbound>
            <outbound-average>1000</outbound-average>
            <outbound-peak>5000</outbound-peak>
          </outbound>
          <inbound>
            <inbound-average>1000</inbound-average>
            <inbound-peak>5000</inbound-peak>
          </inbound>
        </bandwidth>
      </performance-capability>
    </nsf-capability-info>
    <nsf-access-info>
      <capability-name>voip_volte_filter</capability-name>
      <ip>192.0.2.11</ip>
      <port>3000</port>
    </nsf-access-info>
  </nsf-information>
</nsf-registrations>
```

NEW:

```
<nsf-registrations
  xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-reg-interface"
  xmlns:cap="urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability">
  <nsf-information>
    <capability-name>voip_volte_filter</capability-name>
    <nsf-capability-info>
      <security-capability>
        <condition-capabilities>
          <advanced-nsf-capabilities>
            <voip-volte-capability>cap:call-id</voip-volte-capability>
          </advanced-nsf-capabilities>
        </condition-capabilities>
        <action-capabilities>
          <ingress-action-capability>cap:pass</ingress-action-capability>
          <ingress-action-capability>cap:drop</ingress-action-capability>
          <ingress-action-capability>cap:mirror</ingress-action-capability>
          <egress-action-capability>cap:pass</egress-action-capability>
          <egress-action-capability>cap:drop</egress-action-capability>
          <egress-action-capability>cap:mirror</egress-action-capability>
        </action-capabilities>
      </security-capability>
      <performance-capability>
        <processing>
          <processing-average>1000</processing-average>
          <processing-peak>5000</processing-peak>
        </processing>
        <bandwidth>
          <outbound>
            <outbound-average>1000</outbound-average>
            <outbound-peak>5000</outbound-peak>
          </outbound>
          <inbound>
            <inbound-average>1000</inbound-average>
            <inbound-peak>5000</inbound-peak>
          </inbound>
        </bandwidth>
      </performance-capability>
    </nsf-capability-info>
    <nsf-access-info>
      <capability-name>voip_volte_filter</capability-name>
      <ip>192.0.2.11</ip>
      <port>3000</port>
    </nsf-access-info>
  </nsf-information>
</nsf-registrations>
```

No one I-D on its own is wrong and every I-D has parts that I think just right - it is just when they are put together it all gets confusing. What I lack is an underlying information model which separates out the different concepts and gives them identifiers for these YANG I-D to use both in nomenclature and it is going into more detail. A monitoring I-D may specify more detailed flood prevention e.g but I should be able to refer it back to a capability in this I-D. RFC8329 is not that model.

=> [Paul] The identities in each draft have different purposes. We added some explanations for clarifying the purpose of each identity on each draft as follows:

NEW (ietf-i2nsf-capability-data-model - Section 5):

The data model in this document provides identities for the capabilities of NSFs. Every identity in the data model represents the capability of an NSF. Each identity is explained in the description of the identity.

NEW (ietf-i2nsf-consumer-facing-interface - Section 7.1):
This document provides identities in the data model to be used for configuration of an NSF. Each identity is used for a different type of configuration. The details are explained on the description of each identity.

NEW (ietf-i2nsf-nsf-facing-interface - Section 4.1):
This document provides identities in the data model for the configuration of an NSF. The identity has the same concept with the corresponding identity in [I-D.ietf-i2nsf-consumer-facing-interface-dm].

NEW (ietf-i2nsf-nsf-monitoring - Section 10):
The data model provided in this document uses identities to be used to get information of an NSF's monitoring data. Every identity used in the document gives information or status about the current situation of an NSF.

I append below the list of identity I have extracted from the capabilities I-D. I think that seeing just the list of identifiers highlights the challenges a user will have. I have similar lists for customer-facing, nsf-facing, nsf-monitoring. In YANG the 'base' is specified after the 'identity' statement. For the purposes of this review I find it clearer to place the 'base' first and then list the identity derived from that base. The order of the identity statements is the same as in the I-D.

=> [Paul] Below we edit the identities according to your comments. Important changes:

- Remove the prefix "-capability".
- Create a new base identity, e.g., ip for IPv4 and IPv6, and icmp for ICMPv4 and ICMPv6, and transport-protocol for TCP, UDP, SCTP, and DCCP. All of which are based on identity protocol.
- Combine "range-match" and "exact-match" into "range-match".
- Combine common header fields into one field to reduce the number of identities (e.g., "source-port-number" is a common header field for TCP, UDP, SCTP, and DCCP).

Tom Petch

```
/*
* Identities changes from draft-ietf-i2nsf-capability-data-model-16 to 17.
*/
```

OLD (RED COLOR = REMOVED)	NEW (YELLOW HIGHLIGHT = UPDATED)
base event; identity system-event-capability { identity system-alarm-capability { base system-event-capability; identity access-violation { identity configuration-change { base system-alarm-capability; identity memory-alarm { identity cpu-alarm {	base event; identity system-event { identity system-alarm { identity time { base system-event; identity access-violation { identity configuration-change { base system-alarm; identity memory-alarm { identity cpu-alarm {

```
  identity disk-alarm {
  identity hardware-alarm {
  identity interface-alarm {




base condition;
  identity context-capability {

base context-capability;
  identity access-control-list {
  identity application-layer-filter {
  identity target {
  identity user {
  identity group {
  identity geography {









base directional-capability;
  identity unidirectional {
  identity bidirectional {

```

```
  identity disk-alarm {
  identity hardware-alarm {
  identity interface-alarm {

base time;
  identity absolute-time {
  identity periodic-time {



base target-device;
  identity computer {
  identity mobile-phone {
  identity voip-volte-phone {
  identity tablet {
  identity network-infrastructure-device {
  identity iot {
  identity vehicle {

base user-condition;
  identity user {
  identity group {

base geography-location;
  identity source-location {
  identity destination-location {

base directional;
  identity unidirectional {
  identity bidirectional {

base protocol;
  identity ethernet {

base ethernet;
  identity source-mac-address {
  identity destination-mac-address {
  identity ether-type {

base protocol;
  identity ip {

base ip; //Internet Layer Protocol
  identity ipv4 {
  identity ipv6 {


//Common IPv4 and IPv6 Header Field Matching
//Capabilities
  base ipv4;
  base ipv6;
    identity dscp {
    identity length {
    identity ttl {
    identity next-header {
    identity source-address {
    identity destination-address {
    identity flow-direction {

  //IPv4 Header Field Matching Capabilities
  base ipv4;
    identity header-length {
    identity identification {
    identity fragment-flags {
    identity fragment-offset {
    identity ipv4-options; //ipv4- prefix is
                           //used for
```

```
base condition;
  identity ipv4-capability {



base ipv4-capability;
  identity exact-ipv4-header-length {
  identity range-ipv4-header-length {
  identity ipv4-tos-dscp {
  identity exact-ipv4-total-length {
  identity range-ipv4-total-length {
  identity ipv4-id {
  identity ipv4-fragment-flags {
  identity exact-ipv4-fragment-offset {
  identity range-ipv4-fragment-offset {
  identity exact-ipv4-ttl {
  identity range-ipv4-ttl {
  identity ipv4-protocol {
  identity prefix-ipv4-address-flow-direction {
  identity prefix-ipv4-address {
  identity prefix-ipv4-src-address {
  identity prefix-ipv4-dst-address {
  identity range-ipv4-address-flow-direction {
```

```
  identity range-ipv4-address {
  identity range-ipv4-src-address {
  identity range-ipv4-dst-address {
  identity ipv4-ip-opts {
  identity ipv4-geo-ip {

base condition;
  identity ipv6-capability {

base ipv6-capability;
  identity ipv6-traffic-class-dscp {
  identity exact-ipv6-flow-label {
  identity range-ipv6-flow-label {
  identity exact-ipv6-payload-length {
  identity range-ipv6-payload-length {
  identity ipv6-next-header {
  identity exact-ipv6-hop-limit {
  identity range-ipv6-hop-limit {
  identity prefix-ipv6-address-flow-direction {
  identity prefix-ipv6-address {
  identity prefix-ipv6-src-address {
  identity prefix-ipv6-dst-address {
  identity range-ipv6-address-flow-direction {
  identity range-ipv6-address {
  identity range-ipv6-src-address {
  identity range-ipv6-dst-address {
  identity ipv6-header-order {
  identity ipv6-options {
  identity ipv6-hop-by-hop {
  identity ipv6-routing-header {
  identity ipv6-fragment-header {
  identity ipv6-destination-options {
  identity ipv6-geo-ip {

base condition;
  identity icmp-capability {

base icmp-capability;
  identity icmp-type {
  identity icmp-code {

base condition;
  identity icmpv6-capability {

base icmpv6;
  identity icmpv6-type {
  identity icmpv6-code {

base condition;
  identity tcp-capability {

base tcp-capability;
  identity exact-tcp-port-num-flow-direction {
  identity exact-tcp-port-num {
  identity exact-tcp-src-port-num {
  identity exact-tcp-dst-port-num {
  identity range-tcp-port-num-flow-direction {
  identity range-tcp-port-num {
  identity range-tcp-src-port-num {
  identity range-tcp-dst-port-num {
  identity tcp-flags {
  identity tcp-options {


base condition;
  identity udp-capability {

base udp-capability;
```

```
                                //distinguishing it
                                //with
                                //tcp-options



//IPv6 Header Field Matching Capabilities
base ipv6;
  identity flow-label {
  identity header-order {
  identity hop-by-hop {
  identity routing-header {
  identity fragment-header {
  identity destination-options {









base protocol;
  identity icmp {

base icmp;
  identity icmpv4 {
  identity icmpv6 {

base icmpv4;
base icmpv6;
  identity type {
  identity code {



base protocol;
  identity transport-protocol;

base transport-protocol;
  identity tcp {
  identity udp {
  identity sctp {
  identity dccp {

base tcp;
base udp;
base sctp;
base dccp;
  identity source-port-number {
  identity destination-port-number {


base tcp;
  identity flags {
  identity tcp-options { //tcp- prefix is used
```

```
  identity exact-udp-port-num-flow-direction {
  identity exact-udp-port-num {
  identity exact-udp-src-port-num {
  identity exact-udp-dst-port-num {
  identity range-udp-port-num-flow-direction {
  identity range-udp-port-num {
  identity range-udp-src-port-num {
  identity range-udp-dst-port-num {
  identity exact-udp-total-length {
  identity range-udp-total-length {

base sctp-capability;  [**yes really]
  identity exact-sctp-port-num-flow-direction {
  identity exact-sctp-port-num {
  identity exact-sctp-src-port-num {
  identity exact-sctp-dst-port-num {
  identity range-sctp-port-num-flow-direction {
  identity range-sctp-port-num {
  identity range-sctp-src-port-num {
  identity range-sctp-dst-port-num {
  identity sctp-verification-tag {
  identity sctp-chunk-type {

base dccp-capability;
  identity exact-dccp-port-num-flow-direction {
  identity exact-dccp-port-num {
  identity exact-dccp-src-port-num {
  identity exact-dccp-dst-port-num {
  identity range-dccp-port-num-flow-direction {
  identity range-dccp-port-num {
  identity range-dccp-src-port-num {
  identity range-dccp-dst-port-num {
  identity dccp-service-code {

base log-action-capability;
  identity rule-log {
  identity session-log {

base ingress-action-capability;
base egress-action-capability;
base default-action-capability;
  identity pass {
  identity drop {
  identity alert {
  identity mirror {

base egress-action-capability;
  identity invoke-signaling {
```

```
                          //for
                          //distinguishing it
                          //with ipv4-options
  base udp;
  identity total-length {

base sctp;
  identity verification-tag {
  identity chunk-type {

base dccp;
  identity service-code {

  base protocol;
    identity application-protocol {

  base application-protocol;
    identity http {
    identity https {
    identity ftp {
    identity ssh {
    identity telnet {
    identity smtp {
    identity sftp {
    identity pop3 {

base action;
  identity log-action {
  identity ingress-action {
  identity egress-action {
  identity default-action {

base log-action;
  identity rule-log {
  identity session-log {

base ingress-action;
base egress-action;
base default-action;
  identity pass {
  identity drop {
  identity rate-limit {
  identity mirror {

base egress-action;
  identity invoke-signaling {
  identity tunnel-encapsulation {
```

identity forwarding { identity redirection { base resolution-strategy-capability; identity fmr { identity lmr { identity pmr { identity pmre { identity pmrn { base advanced-nsf-capability; identity anti-virus-capability { identity anti-ddos-capability { identity ips-capability { identity voip-volte-capability {	identity forwarding { identity transformation { base resolution-strategy; identity fmr { identity lmr { identity pmr { identity pmre { identity pmrn { base advanced-nsf; identity content-security-control { identity attack-mitigation-control { base content-security-control; identity ips { identity url-filtering { Identity anti-virus { identity voip-volte-filtering { base attack-mitigation-control; identity anti-ddos {
base anti-virus-capability; identity detect { identity allow-list { base anti-ddos-capability; identity syn-flood-action { identity udp-flood-action { identity http-flood-action { identity https-flood-action { identity dns-request-flood-action { identity dns-reply-flood-action { identity icmp-flood-action { identity icmpv6-flood-action { identity sip-flood-action { identity detect-mode { identity baseline-learning { base ips-capability; identity signature-set { identity ips-exception-signature { base condition; identity url-capability { base url-capability; identity pre-defined { identity user-defined { base voip-volte-capability; identity voip-volte-call-id { identity user-agent { base ipsec-capability; identity ike { identity ikeless {	base anti-virus; identity detect { identity exception-files { base anti-ddos; identity packet-rate { identity byte-rate { identity flow-rate { base ips; identity signature-set { identity ips-exception-signature { base url-filtering; identity pre-defined { identity user-defined { base voip-volte; identity call-id { identity user-agent { //IPSec is removed because it is removed in the //NSF-Facing Interface Data Model

---

Reviewer: Paul Wouters

Review result: Has Nits

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.

The summary of the review Has Nits

The issues that Michael Scharf raised regarding TOS have been addressed. Thank you. I have no items that are serious issues, just some comments that you may take into consideration for a minor update.

Nits:

The privacy section talks about a trade-off between privacy and security. But I do not understand what trade-off is meant. The document does not seem to make any trade-off. It just defines capabilities that can be used, some of which might process private material. But the trade-offs of that are really at the protocol level (like they did use TLS or IPsec or why not). I don't think describing technical capabilities is a trade-off of security vs privacy. Perhaps the section could talk about the discovery and/or usage of capabilities and that those capabilities handling private information should attempt to report their usage/findings/events underst conditions that preserve the privacy (e.g., require TLS or IPsec between SG and NSF ?)

=> [PAUL] We have updated the Privacy Considerations as follows:

OLD:
This YANG module specified in this document make a trade-off between privacy and security. Some part of the YANG data model specified in this document might use highly sensitive private data of the client. The data used in this YANG data model can be used for the NSFs to improve the security of the network.

NEW:
This YANG module specifies the capabilities for NSFs. Some of the capabilities in this document MAY require highly sensitive private data to operate properly. The usage of such capability MUST be reported to the users and permitted before using the private information related to the capability. Using any of the capabilities that require private data MUST preserve the privacy by preventing any leakage or unauthorized disclosure of the private data.

The Security section talks about layers that "can use" SSH or TLS for security. I'm not sure why it does not say SHOULD or MUST ? I would rewrite "need to be tightly secured and monitored" to "MUST be tightly secured, monitored and audited".

=> [PAUL] We have rewritten the security considerations following your comment.

NEW:
The lowest layer of NETCONF protocol layers MUST use Secure Shell (SSH) [RFC4254][RFC6242] as a secure transport layer. The lowest layer of RESTCONF protocol layers MUST use HTTP over Transport Layer Security (TLS), that is, HTTPS [RFC7230][RFC8446] as a secure transport layer. Some of the features that this document defines capability indicators for are highly sensitive and/or privileged operations that inherently require access to individuals' private data. These are subtress and data nodes that are considered privacy sensitive: • voip-volte-filtering-capability: The NSF that is able to filter VoIP/VoLTE calls might need to identify certain individual identification. • user-condition-capabilities: The capability uses a set of IP addresses mapped to users. • geography-capabilities: The IP address used in this capability can identify a user's geographical location. It is noted that private information is made accessible in this manner. Thus, the nodes/entities given access to this data MUST be tightly secured, monitored, and audited to prevent leakage or other unauthorized disclosure of private data.

Section 3.1 states:

These capabilities MAY have their access control restricted by a policy; In light of the other recommendations in the Security Section, I think this MAY should really be a SHOULD or even MUST. Alternatively, perhaps say "Some of these capabilities SHOULD" ?

=> [PAUL] We believe MUST is appropriate for this context. We have rewritten it following your suggestion.

NEW:
Based on the above principles, this document defines a capability model that enables an NSF to register (and hence advertise) its set of capabilities that other I2NSF Components can use. These capabilities MUST have their access control restricted by a policy; this is out of scope for this document.

Reviewer: Roman Danyliw

Hi!

I conducted a second AD review on -16 of draft-ietf-i2nsf-capability-data-model. Thanks for the work to merge the text from the previous stand-alone information model document and the preliminary IESG feedback that came before the document was removed from the telechat. My feedback is below:

**** Section 1.**

> **As the industry becomes more sophisticated and network devices (e.g., Internet-of-Things (IoT) devices, autonomous vehicles, and smartphones using Voice over IP (VoIP) and Voice over LTE (VoLTE)) require advanced security protection in various scenario, service providers have a lot of problems described in [RFC8192].**

There seems to be a slight change in framing between RFC8192 and this sentence. RFC8192 discusses the problem as protecting infrastructures and networks, this text frames it as "devices". This isn't necessarily a problem, I just wanted to ask if that drift was intentional.

=> [PAUL] We clarify the problem of cloud-based security services in network infrastructure for providing network devices with efficient and reliable security services as follows.

NEW:
As the industry becomes more sophisticated and network devices (e.g., Internet-of-Things (IoT) devices, autonomous vehicles, and smartphones using Voice over IP (VoIP) and Voice over LTE (VoLTE)) require advanced security protection in various scenarios, security service providers have a lot of problems described in [RFC8192] to provide such network devices with efficient and reliable security services in network infrastructure, that is, cloud-based security services.

**** Section 1. Editorial.**

OLD

> **Security Capabilities describe the functions that Network Security Functions (NSFs) are available to provide for security policy enforcement purposes.**

NEW

> **Security Capabilities describe the functions that Network Security Functions (NSFs) can provide for security policy enforcement.**

=> [PAUL] The sentence has been updated according to your comment while Network Security Functions (NSFs) are replaced with NSFs. Note that the acronym of NSF is defined before.

NEW:

Security Capabilities describe the functions that NSFs can provide for security policy enforcement.

**** Section 1.**
Security Capabilities are independent of the actual security control mechanisms that will implement them.
Can you clarify the intent of this statement? There is a distinction being made between a "security control mechanism" and a "policy" and the "NSF functionality" that I don't follow.
=> [PAUL] We updated the sentence to clarify the relationship between security capabilities and the functionality of an NSF as follows:

OLD:
Security Capabilities are independent of the actual security control mechanisms that will implement them.

NEW:
Security Capabilities are independent of the implementation of the functionality of an NSF for those capabilities.

**** Section 1. Editorial. The sentence beginning with "That is, it is not needed ..." seem repetitive to the text before and after it. I recommend removing it.**
=> [PAUL] The sentence is removed according to your suggestion.

**** Section 1. Per "Note that this YANG data model outlines ...", can you clarify what "outlines" means?**
=> [PAUL] The word “outlines” describes that this YANG data model “structures” both the NSF Monitoring YANG data model and the NSF-Facing Interface YANG data model. We also change the Software-Defined Networking (SDN)-based IPsec flow protection to NSF-Facing Interface YANG Data Model as it is more accurate to say that the NSF Monitoring YANG data model and NSF-Facing YANG data model follow the structure of the I2NSF Capability YANG data model.

OLD:
Note that this YANG data model outlines an NSF monitoring YANG data model [I-D.ietf-i2nsf-nsf-monitoring-data-model] and a YANG data model for Software-Defined Networking (SDN)-based IPsec flow protection [I-D.ietf-i2nsf-sdn-ipsec-flow-protection].

NEW:
Note that this YANG data model structures both the NSF Monitoring Interface YANG data model [I-D.ietf-i2nsf-nsf-monitoring-data-model] and the NSF-Facing Interface YANG data model [I-D.ietf-i2nsf-nsf-facing-interface-dm].

**** Section 1. In the paragraph beginning with "This document provides an information model ...", is there are reason that "NSF" and "Security devices" is being used interchangeably. I thought architecturally, the unit of capability was a NSF.**

=> [PAUL] We change the “security devices” into “NSFs” as follows:

OLD:
This document provides an information model and the corresponding YANG data model [RFC6020] [RFC7950] that defines the capabilities of NSFs to centrally manage the capabilities of those security devices.

NEW:
This document provides an information model and the corresponding YANG data model [RFC6020] [RFC7950] that defines the capabilities of NSFs to centrally manage the capabilities of those NSFs.

**** Section 1. Per the bulleted list of starting with the text of "The 'ietf-i2nsf-capability' YANG module defined in this document ...", there is distinction made between "advanced network security functions" and "generic network security functions". Where is the difference between those two explained? (There is another comment on this below and Eric Vynke also mentioned it in his initial ballot)**

=> [PAUL] We removed the distinctions between the generic and advanced NSFs in Section 1. We introduce generic and advanced NSFs in Section 5.1. The changes are as follows:

OLD: (Red color font is removed)
Section 1 The "ietf-i2nsf-capability" YANG module defined in this document provides the following features: • Definition for time capabilities of network security functions. • Definition for event capabilities of generic network security functions. • Definition for condition capabilities of generic network security functions. • Definition for condition capabilities of advanced network security functions. • Definition for action capabilities of generic network security functions. • Definition for resolution strategy capabilities of generic network security functions. • Definition for default action capabilities of generic network security functions. --- Section 5.1 Condition capabilities are used to specify capabilities of a set of attributes, features, and/or values that are to be compared with a set of

known attributes, features, and/or values in order to determine whether a set of actions needs to be executed or not so that an imperative I2NSF policy rule can be executed. In this document, two kinds of condition capabilities are used to classify different capabilities of NSFs such as generic-nsf-capabilities for generic NSFs and advanced-nsf-capabilities for advanced NSFs. First, the generic-nsf-capabilities define the common capabilities of NSFs such as IPv4 capability, IPv6 capability, TCP capability, UDP capability, SCTP capability, DCCP capability, ICMP capability, and ICMPv6 capability. Second, the advanced-nsf-capabilities define advanced capabilities of NSFs such as anti-virus capability, anti- DDoS capability, Intrusion Prevention System (IPS) capability, HTTP capability, and VoIP/VoLTE capability. Note that VoIP and VoLTE are merged into a single capability in this document because VoIP and VoLTE use the Session Initiation Protocol (SIP) [RFC3261] for a call setup. See Section 3.1 for more information about the condition in the ECA policy model.

NEW:

Section 1

The "ietf-i2nsf-capability" YANG module defined in this document provides the following features:

- Definition for event capabilities of network security functions.
- Definition for condition capabilities of network security functions.
- Definition for action capabilities of network security functions.
- Definition for resolution strategy capabilities of network security functions.
- Definition for default action capabilities of network security functions.

Section 5.1

Condition capabilities are used to specify capabilities of a set of attributes, features, and/or values that are to be compared with a set of known attributes, features, and/or values in order to determine whether a set of actions needs to be executed or not so that an imperative I2NSF policy rule can be executed. In this document, two kinds of condition capabilities are used to classify different capabilities of NSFs such as generic-nsf-capabilities and advanced-nsf-capabilities. First, the generic-nsf-capabilities define NSFs that operate on packet header for layer 2 (i.e., Ethernet capability), layer 3 (i.e., IPv4 capability, IPv6 capability, ICMPv4 capability, and ICMPv6 capability.), and layer 4 (i.e., TCP capability, UDP capability, SCTP capability, and DCCP capability). Second, the advanced-nsf-capabilities define NSFs that operate on features different from the generic-nsf-capabilities, e.g., the payload, cross flow state, application layer, traffic statistics, network behavior, etc. This document defines the advanced-nsf into two categories such as content-security-control and attack-mitigation-control.

- Content security control is an NSF that evaluates the payload of a packet, such as Intrusion Prevention System (IPS), URL-Filtering, Antivirus, and VoIP/VoLTE Filter.
- Attack mitigation control is an NSF that mitigates an attack such as anti-DDoS (DDoS-mitigator).

The advanced-nsf can be extended with other types of NSFs. This document only provides five advanced-nsf capabilities, i.e., IPS capability, URL-Filtering capability, Antivirus capability, VoIP/VoLTE Filter capability, and Anti-DDoS capability. Note that VoIP and VoLTE are merged into a single capability in this document because VoIP and VoLTE use the Session Initiation Protocol (SIP) [RFC3261] for a call setup. See Section 3.1 for more information about the condition in the ECA policy model.

**** Section 3. I'm having trouble finding the information model (CapIM). Section 4 has a data model. Section 3.1. describes the properties of the information model. Is the ECA text in Section 3.1 - 3.3 the CapIM?**

=> [PAUL] The information model is described in Section 3 of the document. We explain Design Principles and ECA Policy Model in Section 3.1. We added the explanation at the beginning of the paragraph as follows:

OLD:
A Capability Information Model (CapIM) is a formalization of the functionality that an NSF advertises. This enables the precise specification of what an NSF can do in terms of security policy enforcement, so that computer-based tasks can unambiguously refer to, use, configure, and manage NSFs. Capabilities MUST be defined in a vendor- and technology-independent manner (e.g., regardless of the differences among vendors and individual products).

NEW:
This section provides the I2NSF Capability Information Model (CapIM). A CapIM is a formalization of the functionality that an NSF advertises. This enables the precise specification of what an NSF can do in terms of security policy enforcement, so that computer-based tasks can unambiguously refer to, use, configure, and manage NSFs. Capabilities MUST be defined in a vendor- and technology-independent manner (e.g., regardless of the differences among vendors and individual products).

**** Section 3. Per the paragraph starting with "Analogous considerations can be applied for channel protection protocols ...", this text seems rather broad in scope. The data model appears to let you configure IPSec.**

=> [PAUL] The configuration for IPsec has been removed from the capability data model as the IPSec configuration in NSF-Facing Interface data model has been removed.

**** Section 3. Editorial. s/[RFC8329] , /[RFC8329], / (i.e., remove the extra space between the reference and the comma.**

=> [PAUL] The extra space between the reference and the comma has been removed.

**** Section 3.1. Editorial. s/-po This document/This document/**

=> [PAUL] The “-po” has been removed.

**** Section 3.1. Editorial. s/resepectively/ respectively/ (multiple places)**

=> [PAUL] The typos for the multiple places have been fixed.

**** Section 3.1. A few of these requirements are generically written, and I wondering if it needs to be so in the I2NSF context. For example:**
-- For the Advertisement requirement, is this "dedicated, well-known" interface anything but the registration interface?

=> [PAUL] The description for advertisement is updated with Registration Interface as follows:

OLD:
Advertisement: A dedicated, well-known interface MUST be used to advertise and register the capabilities of each NSF. This same interface MUST be used by other I2NSF Components to determine what Capabilities are currently available to them.

NEW:
Advertisement: Registration interface [I-D.ietf-i2nsf-registration-interface-dm] MUST be used to advertise and register the capabilities of each NSF. This same interface MUST be used by other I2NSF Components to determine what Capabilities are currently available to them.

-- For the Execution requirement, is this "monitoring" capability more than the monitoring module"?

=> [PAUL] The description for advertisement is updated as follows:

OLD:
Execution: Dedicated, well-known interfaces MUST be used to configure and monitor the use of a capability, resepectively. These provide a standardized ability to describe its functionality, and report its processing results, resepectively. These facilitate multi-vendor interoperability.

NEW:
Execution: NSF-Facing Interface [I-D.ietf-i2nsf-nsf-facing-interface-dm] and NSF Monitoring Interface [I-D.ietf-i2nsf-nsf-monitoring-data-model] MUST be used to configure the use of a capability into an NSF and monitor the NSF, respectively. These provide a standardized ability to describe its functionality, and report its processing results, respectively. These facilitate multi-vendor interoperability.

**** Section 3.1. Per the requirement for automation and scalability, no I2NSF document I can find provides guidance on how to realize this design. As there a normative MUSTs/SHOULDs here, the bounds of these need more details.**

-- Are these implementation details out of scope for I2NSF?

=> [PAUL] The automation of security capabilities of NSFs is explained with Registration Interface as follows:

OLD:
Automation: The system MUST have the ability to auto-discover, auto-negotiate, and auto-update its security capabilities (i.e., without human intervention). These features are especially useful for the management of a large number of NSFs. They are essential for adding smart services (e.g., refinement, analysis, capability reasoning, and optimization) to the security scheme employed. These features are supported by many design patterns, including the Observer Pattern [OODOP], the Mediator Pattern [OODMP], and a set of Message Exchange Patterns [Hohpe].

NEW:
Automation: The system MUST have the ability to auto-discover, auto-negotiate, and auto-update its security capabilities (i.e., without human intervention). These features are especially useful for the management of a large number of NSFs. They are essential for adding smart services (e.g., refinement, analysis, capability reasoning, and optimization) to the security scheme employed. These features are supported by many design patterns, including the Observer Pattern [OODOP], the Mediator Pattern [OODMP], and a set of Message Exchange Patterns [Hohpe]. Registration Interface [I-D.ietf-i2nsf-registration-interface-dm] can register the capabilities of NSFs with the security controller from the request of Developer's Management System providing NSFs and the corresponding security capabilities. Also, this interface can send a query to Developer's Management System in order to find an NSF to satisfy the requested security capability from the security controller that receives a security policy.

-- How much "scale up/down or scale in/out" is needed?

=> [PAUL] We describe the handling of scalability changes of NSFs with NSF Monitoring Interface as follows:

OLD:
Scalability: The management system SHOULD have the capability to scale up/down or scale in/out. Thus, it can meet various performance requirements derived from changeable network traffic or service requests. In addition, security capabilities that are affected by scalability changes SHOULD support reporting statistics to the security controller to assist its decision on whether it needs to invoke scaling or not.

NEW:
Scalability: The management system SHOULD have the capability to scale up/down or scale in/out. Thus, it can meet various performance requirements derived from changeable network traffic or service requests. In addition, security capabilities that are affected by scalability changes SHOULD support reporting statistics to the security controller to assist its decision on whether it needs to invoke scaling or

not. NSF Monitoring Interface [I-D.ietf-i2nsf-nsf-monitoring-data-model] can observe the performance of NSFs to let the security controller decide scalability changes of the NSFs.

** Section 3.1.

Furthermore, when an unknown threat (e.g., zero-day exploits and unknown malware) is reported by an NSF, new capabilities may be created, and/or existing capabilities may be updated (e.g., by updating its signature and algorithm). This results in enhancing the existing NSFs (and/or creating new NSFs) to address the new threats. New capabilities may be sent to and stored in a centralized repository, or stored separately in a vendor's local repository. In either case, a standard interface facilitates this update process.

I understand the general update mechanism of security tools with new signatures of algorithms described here, but cannot link it to the abstract nature of the capability model described in this document. The granularity of the capability model appears to be "has ips capability" not "has ips capability to mitigate exploit-X".

=> [PAUL] The paragraph is updated to accommodate capability updates for newly found security attacks or threats by the capability model.

OLD:
Furthermore, when an unknown threat (e.g., zero-day exploits and unknown malware) is reported by an NSF, new capabilities may be created, and/or existing capabilities may be updated (e.g., by updating its signature and algorithm). This results in enhancing the existing NSFs (and/or creating new NSFs) to address the new threats. New capabilities may be sent to and stored in a centralized repository, or stored separately in a vendor's local repository. In either case, a standard interface facilitates this update process.

NEW:
Furthermore, NSFs are subject to the updates of security capabilities and software to cope with newly found security attacks or threats, hence new capabilities may be created, and/or existing capabilities may be updated (e.g., by updating its signature and algorithm). New capabilities may be sent to and stored in a centralized repository, or stored separately in a vendor's local repository. In either case, Registration Interface can facilitate this update process for Developer's Management System to let the security controller update its repository for NSFs and their security capabilities.

** Section 3.1. Per "definitions of all I2NSF policy-related terms are also defined in [RFC8329]", the only defines in RFC8329 on ECA is in Section 7.0. The definitions in this section appears to be a super-set of those. Is this reference needed?

=> [PAUL] The sentence is removed.

**** Section 3.1. The definitions of the ECA elements in this section don't entirely agree with the definitions in Section 7 of RFC8329. For example, for action, is it flow or packet+flow specific?**
-- Here: "An action is used to control and monitor aspects of flow-based NSFs"
-- RFC8329: "defines the type of operations that may be performed on this packet or flow"
=> [PAUL] We removed the "flow-based" to generalize the sentence as follows:

OLD: (Red color font is removed)
Action: An action is used to control and monitor aspects of flow-based NSFs when the event and condition clauses are satisfied.

NEW:
Action: An action is used to control and monitor aspects of NSFs to handle packets or flows when the event and condition clauses are satisfied.

**** Section 3.2. I don't follow the intent of this section. It defines the concept of a "matched policy rule" and terms like "Ac" and "Ec" which aren't used anywhere else in the document. The title suggested (to me) that there would be some guidance on how to match rules, but there is no guidance there beyond what's already stated in Section 3.1. I would recommend removing it.**
=> [PAUL] Section 3.2 is removed as the content is already covered enough by Section 3.1.

**** Section 3.2. Recommend removing the sentence "To precisely describe ..." as it could be read a redefinition of the ECA terms.**
=> [PAUL] Section 3.2 is removed.

**** Section 3.2. Editorial. s/the properties to characterize/the properties that characterize/**
=> [PAUL] Section 3.2 is removed.

**** Section 3.3. R1 and R2 are presented to show rules that don't conflict. Based on their descriptions their action clause affecting the same object in different ways isn't clear because I don't know what "conduct anti-malware investigation" entails. Please also expand "FW" to be firewall.**
=> [PAUL] We change the description for the two policy rules R1 and R2 as follows:

OLD:
R1: During 8am-6pm, if traffic is external, then run through FW R2: During 7am-8pm, conduct anti-malware investigation

NEW:
R1: During 8am-6pm, if traffic is external, then run through firewall R2: During 7am-8pm, run anti-virus

**** Section 3.3. I appreciate that R1 - R4 are high level rules that that will get translated into more specific guidance and are intended to demonstrate the parts of ECA. However, I'm having difficulty matching those rules with the capabilities of the YANG module described in this document. In particular, R3 and R4 don't appear to be security related unless there is something assumed by virtue of being "GoldService" or "BronzeServer". What capabilities expressed in the YANG module would one use to encode these rules?**

=> [PAUL] The description for R3 and R4 is updated in terms of security services (e.g., web-filtering) as follows:

OLD:
R3: During 8am-6pm, John gets GoldService R4: During 10am-4pm, FTP from all users gets BronzeService

NEW:
R3: During 9am-6pm, allow John to access social networking service websites R4: During 9am-6pm, disallow all users to access social networking service websites

**** Section 3.3.**

> **Conflicts theoretically compromise the correct functioning of devices (as happened for routers several year ago). However, NSFs have been designed to cope with these issues. Since conflicts are originated by simultaneously matching rules, an additional process decides the action to be applied, e.g., among the ones which the matching rule would have enforced. This process is described by means of a resolution strategy for conflicts.**

-- Per "(as happened for routers several years ago)", can this event be referenced or explained

=> [PAUL] We remove this example.

OLD: (Red color font is removed)
Conflicts theoretically compromise the correct functioning of devices (as happened for routers several year ago). However, NSFs have been designed to cope with these issues. Since conflicts are originated by simultaneously matching rules, an additional process decides the action to be applied, e.g., among the ones which the matching rule would have enforced. This process is described by means of a resolution strategy for conflicts.

NEW:
Conflicts theoretically compromise the correct functioning of devices.

However, NSFs have been designed to cope with these issues. Since conflicts are originated by simultaneously matching rules, an additional process decides the action to be applied, e.g., among the actions which the matching rule would have enforced. This process is described by means of a resolution strategy for conflicts.

-- This text appears to be making assumptions about the internal implementation of NSF (i.e., "conflicts are originated by simultaneously matching rules"). Is that a safe assumption? Should this matching strategy be more clearly stated an underlying requirement of the NSFs that I2NSF can handle

=> [PAUL] The finding and handling of conflicted matching rules is performed by resolution strategies in the security controller. The implementation of such resolution strategies is out of scope for I2NSF.

OLD:
Conflicts theoretically compromise the correct functioning of devices (as happened for routers several year ago). However, NSFs have been designed to cope with these issues. Since conflicts are originated by simultaneously matching rules, an additional process decides the action to be applied, e.g., among the ones which the matching rule would have enforced. This process is described by means of a resolution strategy for conflicts.

NEW:
Conflicts theoretically compromise the correct functioning of devices. However, NSFs have been designed to cope with these issues. Since conflicts are originated by simultaneously matching rules, an additional process decides the action to be applied, e.g., among the actions which the matching rule would have enforced. This process is described by means of a resolution strategy for conflicts. The finding and handling of conflicted matching rules is performed by resolution strategies in the security controller. The implementation of such resolution strategies is out of scope for I2NSF.

**** Section 3.3.**

> **On the other hand, it may happen that, if an event is caught, none of the policy rules matches the event.**

How can an event be caught if there is no event clause in any rule to match it? The subsequent logic about a firewall doesn't follow for me because the default rule still is a rule.

=> [PAUL] The paragraph is updated with the matching of event and condition as follows:

OLD:
On the other hand, it may happen that, if an event is caught, none of the policy rules matches the event. As a simple case, no rules may match a packet arriving at border firewall. In this case, the packet is usually dropped,

that is, the firewall has a default behavior to manage the cases that are not covered by specific rules.

NEW:
On the other hand, it may happen that, if an event is caught, none of the policy rules matches the condition. Note that a packet or flow is handled only when it matches both the event and condition of a policy rule according to the ECA policy model. As a simple case, no condition in the rules may match a packet arriving at the border firewall. In this case, the packet is usually dropped, that is, the firewall has a default behavior of packet dropping in order to manage the cases that are not covered by specific rules.

**** Section 3.3. As noted for Section 3.2, this section introduces RSc and Dc, but this notation is not used elsewhere in the document. Why is this needed?**

=> [PAUL] We changed the descriptions as follows:

OLD:
RSc is the set of Resolution Strategies that can be used to specify how to resolve conflicts that occur between the actions of the same or different policy rules that are matched and contained in this particular NSF; Dc defines the notion of a Default action. This action can be either an explicit action or a set of actions.

NEW:
Resolution Strategies: They that can be used to specify how to resolve conflicts that occur between the actions of the same or different policy rules that are matched and contained in this particular NSF; Default action: It provides the default behavior to be executed when there are no other alternatives. This action can be either an explicit action or a set of actions.

**** Section 4. Editorial. s/is used for the Security Controller/is used by the Security Controller/**

=> [PAUL] The sentence has been updated following your suggestion.

**** Section 4 notes that the primary use of this YANG model is for the DMs to populate (via the registration interface) the capabilities of various NSFs. Given that scope, it is a bit striking that the narrative describing Figure 1 primarily discusses only the byproduct of the database on the controller created by the YANG module in this document.**

=> [PAUL] We change the narrative to discuss the Registration Interface and the Capability data model instead of the database on the Security Controller. The changes are as follows:

OLD:
That is, this Registration Interface uses the YANG module described in this document to describe the capabilities of an NSF that is registered with the Security Controller. With the database of the capabilities of the NSFs that are maintained centrally, the NSFs can be more easily managed, which can resolve many of the problems described in [RFC8192].

NEW:
That is, this Registration Interface uses the YANG module described in this document to describe the capabilities of an NSF that is registered with the Security Controller. As described in [RFC8192], with the usage of Registration Interface and the YANG module in this document, the NSFs manufactured by multiple vendors can be managed together by the Security Controller in a centralized way and be updated dynamically by each vendor as the NSF has software or hardware updates.

**** Section 5.1. Is it possible define generic-nsf and advanced-nsf capabilities with a more principled definition. Perhaps that the generic-nsf operates on layer 3 and 4 headers only; and the advanced in application layer or those requiring cross flow state?**

=> [PAUL] The definition has been updated as follows:

OLD:
Section 5.1 Condition capabilities are used to specify capabilities of a set of attributes, features, and/or values that are to be compared with a set of known attributes, features, and/or values in order to determine whether a set of actions needs to be executed or not so that an imperative I2NSF policy rule can be executed. In this document, two kinds of condition capabilities are used to classify different capabilities of NSFs such as generic-nsf-capabilities for generic NSFs and advanced-nsf-capabilities for advanced NSFs. First, the generic-nsf-capabilities define the common capabilities of NSFs such as IPv4 capability, IPv6 capability, TCP capability, UDP capability, SCTP capability, DCCP capability, ICMP capability, and ICMPv6 capability. Second, the advanced-nsf-capabilities define advanced capabilities of NSFs such as anti-virus capability, anti- DDoS capability, Intrusion Prevention System (IPS) capability, HTTP capability, and VoIP/VoLTE capability. Note that VoIP and VoLTE are merged into a single capability in this document because VoIP and VoLTE use the Session Initiation Protocol (SIP) [RFC3261] for a call setup. See Section 3.1 for more information about the condition in the ECA policy model.

NEW:
Section 5.1 Condition capabilities are used to specify capabilities of a set of attributes, features, and/or values that are to be compared with a set of

known attributes, features, and/or values in order to determine whether a set of actions needs to be executed or not so that an imperative I2NSF policy rule can be executed. In this document, two kinds of condition capabilities are used to classify different capabilities of NSFs such as generic-nsf-capabilities and advanced-nsf-capabilities. First, the generic-nsf-capabilities define NSFs that operate on packet header for layer 2 (i.e., Ethernet capability), layer 3 (i.e., IPv4 capability, IPv6 capability, ICMPv4 capability, and ICMPv6 capability.), and layer 4 (i.e., TCP capability, UDP capability, SCTP capability, and DCCP capability). Second, the advanced-nsf-capabilities define NSFs that operate on features different from the generic-nsf-capabilities, e.g., the payload, cross flow state, application layer, traffic statistics, network behavior, etc. This document defines the advanced-nsf into two categories such as content-security-control and attack-mitigation-control.

- Content security control is an NSF that evaluates the payload of a packet, such as Intrusion Prevention System (IPS), URL-Filtering, Antivirus, and VoIP/VoLTE Filter.
- Attack mitigation control is an NSF that mitigates an attack such as anti-DDoS (DDoS-mitigator).

The advanced-nsf can be extended with other types of NSFs. This document only provides five advanced-nsf capabilities, i.e., IPS capability, URL-Filtering capability, Antivirus capability, VoIP/VoLTE Filter capability, and Anti-DDoS capability. Note that VoIP and VoLTE are merged into a single capability in this document because VoIP and VoLTE use the Session Initiation Protocol (SIP) [RFC3261] for a call setup. See Section 3.1 for more information about the condition in the ECA policy model.

**** Yang module. "This can be used for an extension point ..." is used in a few places. Can the proposed approach for making extensions be further explained?**

=> [PAUL] The sentence “This can be used for an extension point …” means the base of the identity that can be extended. We removed the sentence and added “Base” in the descriptions. An example change is as follows:

OLD:

```
identity anti-virus-capability {
  base advanced-nsf-capability;
  description
    "Identity for advanced NSF Anti-Virus capability.
     This can be used for an extension point for Anti-Virus
     as an advanced NSF.";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Advanced NSF Anti-Virus capability";
}
```

NEW:

```
identity anti-virus {
  base content-security-control;
  description
    "Base identity for anti-virus capability to protect the network
```

```
      by detecting and removing viruses.";
  }
```

**** YANG module. There is a notation being used in the reference section which is not clear to me. It is of the form: "RFC XXX: <title of RFC - <text>". For example, in leaf-list default action-capabilities, the reference reads "RFC 8329: Framework for Interface to Network Security Functions - Ingress and egress actions." What part of RFC8329 am I supposed to be reading. There is not Section with the title "Ingress and egress" and that exact phrase appears only in Section 9.2 which doesn't appear germane. It would be much clearer if the references were of the form "RFC XXX: <title of RFC - <Section #>" instead.**

=> [PAUL] We change the reference to have a section number to make it clearer as follows:

OLD:

```
identity ingress-action-capability {
  description
    "Base identity for ingress-action capability";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Ingress action";
}

identity egress-action-capability {
  description
    "Base identity for egress-action capability";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Egress action";
}
```

NEW:

```
identity ingress-action {
  base action;
  description
    "Base identity for ingress-action capability";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Section 7.2";
}

identity egress-action {
  base action;
  description
    "Base identity for egress-action capability";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Section 7.2";
}
```

**** Identity application-layer-filter. The references seem to suggest that this is HTTP only. Is the intent for generic capability or for an HTTP-only focus?**

=> [PAUL] We change the data model for the application filter. We created a new base of application-protocol for the application-layer-filter to list the capability of detecting the possible application layer protocols (e.g., HTTP, HTTPS, FTP, POP3, IMAP, etc.).

OLD:

```
identity application-layer-filter {
   base context-capability;
   description
     "Identity for application-layer-filter condition capability.
      application-layer-filter capability can examine the contents
      of a packet (e.g., a URL contained in an HTTP message).";
   reference
     "RFC7230: Hypertext Transfer Protocol (HTTP/1.1): Message
      Syntax and Routing
      RFC7231: Hypertext Transfer Protocol (HTTP/1.1): Semantics
      and Content";
 }
.
.
.

 leaf-list context-capabilities {
   type identityref {
     base context-capability;
   }
   description
     "Security context capabilities";
 }
```

NEW:

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

 identity http {
   base application-protocol;
   description
     "The identity for HTTP protocol.";
   reference
     "RFC 2616: Hypertext Transfer Protocol (HTTP)
      RFC7230: Hypertext Transfer Protocol (HTTP/1.1): Message
      Syntax and Routing
      RFC7231: Hypertext Transfer Protocol (HTTP/1.1): Semantics
      and Content";
```

```
}

identity https {
  base application-protocol;
  description
    "The identity for HTTPS protocol.";
  reference
    "RFC 2818: HTTP over TLS (HTTPS)
     RFC7230: Hypertext Transfer Protocol (HTTP/1.1): Message
     Syntax and Routing
     RFC7231: Hypertext Transfer Protocol (HTTP/1.1): Semantics
     and Content";
}

identity ftp {
  base application-protocol;
  description
    "The identity for ftp protocol.";
  reference
    "RFC 959: File Transfer Protocol (FTP)";
}

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

identity sftp {
  base application-protocol;
  description
    "The identity for sftp.";
  reference
    "RFC 913: Simple File Transfer Protocol (SFTP)";
}

identity pop3 {
  base application-protocol;
```

```
    description
      "The identity for pop3.";
    reference
      "RFC 1081: Post Office Protocol - Version 3 (POP3)";
  }

  identity imap {
    base application-protocol;
    description
      "The identity for Internet Message Access Protocol (IMAP).";
    reference
      "RFC 3501: INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1";
  }

.
.
.

  container context-capabilities {
    description
      "Security context capabilities";
    leaf-list application-filter-capabilities{
      type identityref {
        base application-protocol;
      }
      description
        "Context capabilities based on the application protocol";
    }
    ...
  }
```

**** Identity baseline-learning, signature-set, ips-exception-signature. RFC8329 makes no reference to these types of capabilities. What are these?**

=> [PAUL] The baseline-learning is removed as we change the anti-DDoS capability to packet-rate, byte-rate, and flow-rate detection. We updated the descriptions and removed the RFC8329 from the reference of the identities. The changes are as follows:

OLD: (Red color is removed)

```
  identity baseline-learning {
    base anti-ddos-capability;
    description
      "Identity for advanced NSF Anti-DDoS Baseline Learning
       capability.  This can be used for an extension point
       for Anti-DDoS Baseline Learning as an advanced NSF.";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF Anti-DDoS Baseline Learning
       capability";
  }
```

```
identity signature-set {
  base ips-capability;
  description
    "Identity for advanced NSF IPS Signature Set capability.
     This can be used for an extension point for IPS Signature
     Set as an advanced NSF.";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Advanced NSF IPS Signature Set capability";
}

identity ips-exception-signature {
  base ips-capability;
  description
    "Identity for advanced NSF IPS Exception Signature
     capability.  This can be used for an extension point for
     IPS Exception Signature as an advanced NSF.";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Advanced NSF IPS Exception Signature Set
     capability";
}
```

NEW:

```
identity signature-set {
  base ips;
  description
    "Identity for the capability of IPS to set the signature.
     Signature is a set of rules to detect an intrusive activity.";
  reference
    "RFC 4766:  Intrusion Detection Message Exchange Requirements -
     Section 2.2.13";

}

identity exception-signature {
  base ips;
  description
    "Identity for the capability of IPS to exclude signatures from
     detecting the intrusion.";
  reference
    "RFC 4766:  Intrusion Detection Message Exchange Requirements -
     Section 2.2.13";

}
```

**** leaf-list anti-virus-capability, anti-ddos-capability, url-capability, voip-volte-capability. All of these refer to RFC8329 but I can't find the reference Section in that document which describes these capabilities**

=> [PAUL] We remove the reference to RFC 8329 for the advanced-nsf-capabilities as the RFC 8329 does not mention such capability.

**** Section 8. I appreciate the inclusion of this new section in response to the original IESG telechat. I don't follow how it informs awareness on privacy issues - no insight is being provide on how the trade-off is being made and even what privacy issues are arising beyond simply stating there are some. I would suggest reframing this section to emphasize that this module is intended to describe the capabilities of a diverse set of network security function already in common use in enterprise security operations. The specificity of the privacy issues can be addressed with reference as is already done with further fidelity as noted in the next comment in Section 9. Ben Kaduk made a few comments on privacy language in his initial ballot too.**

=> [PAUL] The privacy considerations section, which was suggested by Ben Kaduk, is updated according to the comments from Paul Wouters who is a member of Security Directorate as follows:

OLD:
This YANG module specified in this document make a trade-off between privacy and security. Some part of the YANG data model specified in this document might use highly sensitive private data of the client. The data used in this YANG data model can be used for the NSFs to improve the security of the network. In regards to the privacy data used, the security for accessibility of the data should be tightly secured and monitored. The Security Considerations are discussed in Section 9.

NEW:
This YANG module specifies the capabilities for NSFs. Some of the capabilities in this document MAY require highly sensitive private data to operate properly. The usage of such capability MUST be reported to the users and permitted before using the private information related to the capability. Using any of the capabilities that require private data MUST preserve the privacy by preventing any leakage or unauthorized disclosure of the private data. In regards to the privacy data used, the security for accessibility of the data should be tightly secured and monitored. The Security Considerations are discussed in Section 9.

**** Section 9. Thanks for using the YANG Security Considerations template (https://trac.ietf.org/trac/ops/wiki/yang-security-guidelines) as a starting point. Please include the other elements of the template:**

-- Discuss the readable nodes noting the consequences from the perspective of the attacker (e.g., reading nodes will reveal the specific configuration of security controls to an attacker (a) craft an attack path that avoids observation or mitigations; may revealing topology information to inform additional targets or enable lateral movement; enabling the construction of an attack path that avoids observation or mitigations; or (c) provide an indication that the operator has discovered the attack). The scope of this is likely the entire data model.

-- Discuss the specifics of which readable nodes might be considered privacy sensitive

=> [PAUL] We added your comment to the Security Considerations as follows:

NEW:
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 with their sensitivity/vulnerability: • list nsf: The leak of this node to an attacker could reveal the specific configuration of security controls to an attacker. An attacker can craft an attack path that avoids observation or mitigations; one may reveal topology information to inform additional targets or enable lateral movement; one enables the construction of an attack path that avoids observation or mitigations; one provides an indication that the operator has discovered the attack. Some of the features that this document defines capability indicators for are highly sensitive and/or privileged operations that inherently require access to individuals' private data. These are subtrees and data nodes that are considered privacy sensitive: • voip-volte-filtering-capability: The NSF that is able to filter VoIP/VoLTE calls might identify certain individual identification. • user-condition-capabilities: The capability uses a set of IP addresses mapped to users. • geography-capabilities: The IP address used in this capability can identify a user's geographical location. It is noted that some private information is made accessible in this manner. Thus, the nodes/entities given access to this data MUST be tightly secured, monitored, and audited to prevent leakage or other unauthorized disclosure of private data. Refer to [RFC6973] for the description of privacy aspects that protocol designers (including YANG data model designers) should consider along with regular security and privacy analysis.

**** References**

-- RFC8805 should be a normative reference

=> [PAUL] RFC 8805 has been moved to normative reference.

-- Can the shepherd write-up please be updated to reflect that there are several downrefs. From idnits:

**** Downref: Normative reference to an Informational RFC: RFC 6691**

**** Downref: Normative reference to an Informational RFC: RFC 8192**

**** Downref: Normative reference to an Informational RFC: RFC 8329**

**** and also RFC8805 as noted above**

=> [PAUL] The above references except RFC8805 have been moved to informative references.

-- Per "Unused Reference: 'RFC2119' is defined on line 2884, but no explicit reference was found in the text", this means that the boiler plate such as:

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",

"SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Is not present but the text is still citing RFC2119. Please add the above text to Section 2 since I believe that the "MUSTs" and "SHOULDs" present in the document are in fact intended to be normative?

=> [PAUL] The text for RFC2119 has been added to Section 2.

Regards,

Roman

Thanks for your valuable comments.

Best Regards,
Jaehoon (Paul) Jeong