I2NSF
Internet-Draft
Intended status: Standards Track
Expires: April 8, 2017

S. Hares
Huawei
R. Moskowitz
HTT Consulting
Xia
Huawei

October 5, 2016

I2NSF Capability Yang Model
draft-hares-i2nsf-capability-yang-01.txt

Abstract

This document defines a yang model that enables a I2NSF controller to control various network security functions in Network security devices.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 8, 2017.

Copyright Notice

Table of Contents

1. Introduction 2
2. High-level Yang 3
2.1. capability per NSF 3
2.2. Network Security Control 4
2.3. Security Content Capabilities 6
2.4. Attack Mitigation Capabilities 8
2.5. IT Resources linked to Capabilities 10
2.6. actions 10
3. Use of filter-based RIBS 10
4. YANG Modules 11
5. IANA Considerations 23
6. Security Considerations 23
7. References 23
7.1. Normative References 23
7.2. Informative References 23
Authors' Addresses 25

1. Introduction

[I-D.ietf-i2nsf-problem-and-use-cases] proposes two different types of interfaces:

- North-bound interface (NBI) provided by the network security functions (NSFs)

- Interface between I2NSF user/client with network controller:

This document provides a yang models that define the capabilities for security devices that can be utilized by I2NSF NBI between the I2RS network controller and the NSF devices to express the NSF devices capabilities. It can also be used by the IN2SF user application (or I2NSF client) to network controller to provide a complete list of the I2NSF capabilities the Network controller can control.

This document defines a yang data models based on the [I-D.xia-i2nsf-capability-interface-im], and initial work done in [I-D.xia-i2nsf-service-interface-dm]. Terms used in document are defined in [I-D.ietf-i2nsf-terminology].

This model is an attempt to merge draft-jeong-i2nsf-capability-interface-yang-02.txt, but it has not bene reviewed by this draft's authors. Hopefully, this is a good start for a merge. The Yang

module has not been changed to match the high-level-yang. This seemed prudent until we agreed upon the merge.

[I-D.xia-i2nsf-capability-interface-im] defines the following type of functionality in NSFs.

- network security control
- content security control, and
- attack mitigation control

This document contains high-level yang for each type of control. The features in each section have been built up from the following sources:

open-source: firewalls, IDS, IPS. This includes ECA policy for

basic-firewalls: in router, switches, firewalls,

firewall products commercial level

specialized devices IDS, IPS

2. High-level Yang

This section provides an overview of the high level yang.

2.1. capability per NSF

The high level yang capabilities per NSF device, controller, or application is the following:

```
ietf-i2nsf-capability
 +--rw nsf-capabilities
    +--rw capability* [name]
```

```
            +--rw nsf-name  string
            +--rw cfg-net-secctl-capabilities
          |  uses pkt-eca-policy:pkt-eca-policy-set
          +--rw cfg-net-sec-content-capabilities
              |  uses i2nsf-content-caps
              |  uses i2nsf-content-sec-actions
              +--rw cfg-attack-mitigate-capabilities*
              |  uses i2nsf-mitigate-caps
              +--rw ITResource [ITresource-name]
              |  uses cfg-ITResources
```

Figure 1

Each of these section mirror sections in: [I-D.xia-i2nsf-capability-interface-im]. The high level yang for cfg-net-secctl-capabilities, cfg-net-sec-content-capabilities, and cfg-attack-mitigate-capabilities. This draft is also utilizes the concepts originated in Basile, Lioy,Pitscheider, and Zhao[2015] concerning conflict resolution, use of external data, and ITResources. The authors are grateful to Cataldo for pointing out this excellent work.

2.2. Network Security Control

This section defines the network security control capabilites for each NSF entity (device, controller, APP). The portion of the top level model that this explains is the following:

```
            +--rw cfg-net-secctl-capabilities
          |  uses pkt-eca-policy:pkt-eca-policy-set
```

Note that yang simply uses the ietf-pkt-eca-policy-cfg from [I-D.ietf-i2rs-pkt-eca-data-model].

```
module ietf-pkt-eca-policy
  +--rw pkt-eca-policy-cfg
  |  +--rw pkt-eca-policy-set
  |     +--rw policies* [policy-name]
  |     |  +--rw policy-name string
  |     |  +--rw vrf-name string
  |     |  +--rw address-family
```

```
|     |  +--rw rule-list* [rule-name]
|     |  |  +--rw rule-name
|     |  |  +--rw rule-order-id uint16
|     |  |  +--rw default-action-id integer
|     |  |  +--rw default-resolution-strategy-id integer
|     |  +--rw rules* [order-id rule-name]
|     |     +--rw order-id uint16
|     |     +--rw rule-name string
|     |     +--rw policy-name string
|     |     +--rw cfg-rule-conditions [rule-cnd-id]
|     |     |  +--rw rule-cnd-id uint32
|     |     |  +--rw support
|     |     |  |  +--rw event-matches boolean
|     |     |  |  +--rw pkt-matches boolean
|     |     |  |  +--rw usr-context-matches boolean
|     |     |  +--rw eca-events-match*  [rule-event-id]
|     |     |  |  +--rw rule-event-it uint16
|     |     |  |  | ... time-event match (see below)
|     |     |  +--rw eca-condition-match
```



```
|     |        |  |  +--rw eca-pkt-matches* [pkt-match-id]
|     |        |  |  | ...(see packet matches below)
|     |        |  |  | ... (address, packet header, packet payload)
|     |        |  |  +--rw eca-user-context-matches* [usr-match-id]
|     |        |  |  | ... (see user context match below)
|     |        |  +--rw cfg-rule-actions [cfgr-action-id]
|     |        |  |  +--rw cfgr-action-id
|     |        |  |  +--rw eca-actions* [action-id]
|     |        |  |  |  +--rw action-id uint32
|     |        |  |  |  +--rw eca-ingress-actions*
|     |        |  |  |  | ... (permit, deny, mirror)
|     |        |  |  |  +--rw eca-fwd-actions*
|     |        |  |  |  | ...  (invoke, tunnel encap, fwd)
|     |        |  |  |  +--rw eca-egress-acttions*
|     |        |  |  |  | .. .
|     |        |  |  |  +--rw eca-qos-actions*
|     |        |  |  |  | ...
|     |        |  |  |  +--rw eca-security-actions*
|     |        |  +--rw policy-conflict-resolution* [strategy-id]
|     |        |  |  +--rw strategy-id integer
|     |        |  |  +--rw filter-strategy identityref
|     |        |  |  |  .. FMR, ADTP, Longest-match
```

```
|        |  +--rw global-strategy identityref
|        |  +--rw mandatory-strategy identityref
|        |  +--rw local-strategy identityref
|        |  +--rw resolution-fcn uint32
|        |  +--rw resolution-value uint32
|        |  +--rw resolution-info  string
|        |  +--rw associated-ext-data*
|        |  |  +--rw ext-data-id integer
|        |  +--rw cfg-external-data* [cfg-ext-data-id]
|        |  +--rw cfg-ext-data-id integer
|        |  +--rw data-type integer
|        |  +--rw priority uint64
|        |  |  uses external-data-forms
|        |  ... (other external data)
|     +--rw pkt-eca-policy-opstate
|        +--rw pkt-eca-opstate
|           +--rw policies-opstat* [policy-name]
|           |  +--rw rules-installed;
|           |  +--rw rules_opstat* [rule-name]
|                    |  +--rw strategy-used [strategy-id]
|           +--rw rules_opstate* [rule-order rule-name]
|           |  +--rw status
|           |  +--rw rule-inactive-reason
|           |  +--rw rule-install-reason
|           |  +--rw rule-installer
|           |  +--rw refcnt
```



```
           +--rw rules_pktstats* [rule-order rule-name]
           |  +--rw pkts-matched
           |  +--rw pkts-modified
           |  +--rw pkts-forward
                    +--rw op-external-data [op-ext-data-id]
                    |  +--rw op-ext-data-id integer
                    |  +--rw type identityref
                    |  +--rw installed-priority integer
                    |  |  (other details on external data )
```

figure 2

2.3. Security Content Capabilities

This section expands the

```
        +--rw cfg-net-sec-content-capabilities
            |  uses i2nsf-content-caps
            |  uses i2nsf-content-sec-actions
```

Content Security Control

```
   +--rw cfg-netsec-content-caps*
   |  +--rw cfg-groups* [group-name]
   |  |  +--rw group-name string
   |  |  +--rw group-rule-list* [rule-name]
   |  |  |  +--rw rule-name string
   |  |  |  +--rw rule-order-id integer
   |  |  |  +--rw default-action-id integer
   |  |  |  +--rw default-resolution-strategy-id integer|
   |  +--rw cfg-netsec-content-rules* [rule-order-id rule-name]
   |  |  +--rw cfg-netsec-content-rule
   |  |  |  +--rw rule-order-id integer
   |  |  |  +--rw rule-name string
   |  |  |  +--rw cfg-filter-rules
   |  |  |  |  +--rw cfg-anti-virus-rule
   |  |  |  |  |  +--rw antivirus-support?  Boolean
   |  |  |  |  |  +--rw source string
   |  |  |  +--rw cfg-IPS-rule
   |  |  |  |  +--rw ips-support?  boolean
   |  |  |  |  +--rw source string
   |  |  |  +--rw cfg-IDS-rule
   |  |  |  |  +--rw ids-support? boolean
   |  |  |  |  +--rw source string
   |  |  |  +--rw cfg-url-filter-rule
   |  |  |  |  +--rw url-filtering-support? boolean
```



```
   |  |  |  |  +--rw source string
   |  |  |  +--rw cfg-file-block-rule
   |  |  |  |  +--rw file-blocking-support? boolean
   |  |  |  |  +--rw source string
   |  |  |  +--rw cfg-data-filter-rule
   |  |  |  |  +--rw data-filtering-support? boolean
   |  |  |  |  +--rw source string
   |  |  |  |  | ... description
```

```
|  |  |  +--rw cfg-APP-behave-rule
|  |  |  |  +--rw app-control-support? boolean
|  |  |  |  +--rw source string
|  |  |  +--rw cfg-mail-filter-rule
|  |  |  |  +--rw mail-filter-support?  boolean
|  |  |  |  +--rw source string
|  |  |  +--rw cfg-pkt-capture-rule
|  |  |  |  +--rw pkt-capture-support? boolean
|  |  |  |  +--rw source string
|  |  |  +--rw cfg-file-isolate-rule
|  |  |  |  +--rw file-isolation-support? boolean
|  |  |  |  +--rw source string
|  |  |  +--rw voip-volte-rule
|  |  |  |  +--rw voip-volte-support? boolean
|  +--rw cfg-sec-content-actions
|  |  +--voip-volte-rules* [voip-volte-rule-id]
|  |  |  +--rw voip-volte-rule-id uint16
|  |  |  +--rw voip-volte-event
|  |  |  |  +--rw called-voip boolean
|  |  |  |  +--rw called-volte boolean
|  |  |  +--rw condition-match
|  |  |  |  +--rw sip-header* [sip-header-uri]
|  |  |  |  +--rw sip-header-uri string
|  |  |  |  +--rw sip-header-method string
|  |  |  |  +--rw expire-time yang:date-and-time
|  |  |  |  +--rw sip-header-user-agent uint32
|  |  |  |  +--rw cell-region* [cell-id-region]
|  |  |  |  |  +-rw cell-id-region uint32
|  |  |  +--rw action
|  |  |  |  +--rw action-type identityref
|  |  |  |  +--rw (action-type)?
|  |  |  |  |  +--: (ingress-action)
|  |  |  |  |  |  +--rw ingress-permit boolean
|  |  |  |  |  |  |  +--rw ingress-deny boolean
|  |  |  |  |  |  |  +--rw ingress-mirror boolean
|  |  |  |  |  +--: (egress-action)
|  |  |  |  |  |  +--rw egress-redirection boolean
```

figure 3

2.4. Attack Mitigation Capabilities

The high level yang below expands the following section of the top-level model:

```
           +--rw cfg-attack-mitigate-capabilities
           |  uses cfg-attack-mitigate-caps
```

Attack mitigation

```
   +--rw cfg-attack-mitigate-caps
   |  +--rw cfg-groups* [group-name]
   |  |  +--rw group-name string
   |  |  +--rw group-rule-list* [rule-name]
   |  |  |  +--rw rule-name string
   |  |  |  +--rw rule-order-id integer
   |  |  |  +--rw default-action-id integer
   |  |  |  +--rw default-resolution-strategy-id integer|
   |  +--rw cfg-netsec-content-rules* [rule-order-id rule-name]
   |  |  +--rw rule-order-id integer
   |  |  +--rw attack-mitigation-type identityref
   |  |  +--:(network-attack-type)?
   |  |  |  +--:sync-flood
   |  |  |  +--rw syn-flood-support boolean
   |  |  |  +--rw sync-flood* [sync-flood-fcn]
   |  |  |  +--rw sync-flood-fcn uint16
   |  |  |  +--:(udp-flood)
   |  |  |  |  +--rw udp-flood-supported boolean
   |  |  |  |  +--rw udp-flood-fcn string   //std or vendor name
   |  |  |  +--:(icmp-flood)
   |  |  |  |  +--rw icmp-flood-supported boolean
   |  |  |  |  +--rw cfg-icmp-flood* [icmp-flood-fcn]
   |  |  |  |  +--rw icmp-flood-fcn string
   |  |  |  +--:(ip_frag_flood)
   |  |  |  |  +--rw ipfrag-flood-fcn-supported boolean
   |  |  |  +--rw cfg-ip-frag-flood* [ipfrag-flood-fcn]
   |  |  |  |  +--rw ipfrag-flood-fcn string //std/vendor name
   |  |  |  +--:(http_flood)
   |  |  |  |  +--rw http-flood-fcn-supported boolean
   |  |  |  |  +--rw cfg-http-flood* [http-flood-fcn]
   |  |  |  |  +--rw http-flood-fcn string
   |  |  |  +--:(dns-flood)
   |  |  |  |  +--rw dns-flood-fcn-supported boolean
   |  |  |  |  +--rw cfg-dns-flood* [dns-flood-fcn]
   |  |  |  |  +--rw dns-flood-fcn string  //std or vendor name
   |  |  |  +--:(dns-amplify)
   |  |  |  |  +--rw dns-amp-fcn-supported boolean
```

```
|  |  |  |  +--rw cfg-dns-amplify* [dns-amp-fcn]
|  |  |  |  +--rw dns-amp-fcn string  //std or vendor name
|  |  |  +--:(SSL-DDoS)
|  |  |  |  +--rw ssl-ddos-fcn-support boolean
|  |  |  |  +--rw cfg-ssl-ddos* [ssl-dos-fcn]
|  |  |  |  +--rw ssl-dos-fcn string
|  |  |  +--: (ip-sweep):
|  |  |  |  +--rw ipsweep-fcn-supported boolean
|  |  |  |  +--rw cfg-IP-Sweep* [ipsweep-fcn]
|  |  |  |  +--rw ipsweep-fcn string  //std or vendor name
|  |  |  +--: (port-scanning)
|  |  |  |  +--rw port-scan-fcn-supported boolean
|  |  |  |  +--rw cfg-Port-scanning [port-scan-fcn]
|  |  |  |  +--rw port-scan-fcn string  //std or vendor name
|  |  |  +--: (ping-of-death)
|  |  |  |  +--rw pingd-fcn-supported boolean
|  |  |  |  +--rw cfg-ping-of-death* [pingd-function]
|  |  |  |  +--rw pingd-fcn string  //std or vendor name
|  |  |  +--:(icmp-oversize)
|  |  |  |  +--rw o-icmp-fcn-supported boolean
|  |  |  +--rw cfg-oversize-ICMP* [o-icmp-fcn]
|  |  |  |  +--rw o-icmp-fcn string  //std or vendor name
|  |  +--:(single-packet-attack)?
|  |  |  +--rw single-packet-type? identityref
|  |  |  +--:(scan-and-sniff-attack)
|  |  |  |  +--scan-n-sniff-type identityref
|  |  |  |  +--(scan-n-sniff-type)?
|  |  |  |  |--:(ip-sweep-attack)
|  |  |  |  |  +--rw 1p-ip-sweep-attack-support boolean
|  |  |  |  |  +--rw 1p-ip-sweep-attack-fcn string
|  |  |  |  +--:(port-scanning-attack)
|  |  |  |  |  +--rw 1pk-port-scanning-support boolean
|  |  |  |  |  +--rw 1pk_port-sanning-fcn string
|  |  |  +--:(malformed-packet-attack)
|  |  |  |  +--1pk-malformed-packet-attack-type identityref
|  |  |  |  +--:(ping-of-death-attack)
|  |  |  |  |  +--rw 1pk-ping-of-death-support boolean
|  |  |  |  |  +--rw 1pk-ping-of-death-fcn string
|  |  |  |  +--:(teardrop-attack)
|  |  |  |  |  +--rw 1pk-teardrop-attack-support boolean
|  |  |  |  |  +--rw 1pk-teardrop-attack-fcn string
|  |  |  +--:(special-packet-attack)
|  |  |  |  +--rw special-packet-attack-type identityref
|  |  |  |  +--(special-packet-attack-type)?
|  |  |  |  |  +--:(oversized-icmp-attack)
|  |  |  |  |  |  +--rw oversized-icmp-attack-support boolean
```

```
|  |  |  |  |  |  +--rw oversized-icmp-attack-fcn string
|  |  |  |  |  +--:(tracert-attack)
```



```
|  |  |  |  |  |  +--rw tracert-attack-support boolean
|  |  |  |  |  |  +--rw tracert-attack-fcn string
```

figure 4

2.5. IT Resources linked to Capabilities

Tis section provides a link between capabilities and IT resources. This section has a lsit of IT Resources by name. Additional input is needed.

```
+--rw cfg-ITResources
|  +--ITResources* [ITresource-name]
|  |  +--rw ITresource-name string
|  | ..
```

2.6. actions

The following notifications indicate when rules are added or deleted.

(to be completed after discussion with Paul Jeong, Jin-Yong Kim, and Dae-Young Hyun, and Jung-Soo Park, and Taei-Jin Ahn.)

3. Use of filter-based RIBS

The packet-eca policy is kept for configuration, I2RS ephemeral state, and BGP stored policy state in filter-based RIBS. These RIBS have the high-level yang structures below and are described in [I-D.ietf-i2rs-fb-rib-data-model]. These filter-ribs may be leveraged in I2NSF storage devices for the policy storage.

```
     +--rw fb-ribs
       +--rw fb-rib* [rib-name]
       |  +--rw rib-name string
       |  |  rw fb-type identityref /config, i2rs, bgp
       |  +--rw rib-afi rt:address-family
       |  +--rw fb-rib-intf* [name]
       |  |  +--rw name string
       |  |  +--rw intf if:interface
       |  +--rw default-ribs
       |  |  +--rw rt-rib string            // routing kernel rib
       |  |  +--rw config-rib string;       // static rt-rib
       |  |  +--rw i2rs-rib string;         // ephemeral rt-rib
       |  |  +--rw bgp-instance-name string // bgp instance
       |  |  +--rw bgp-rib  string          // bgp rib
       |  +--rw fb-rib-refs
       |  |  +--rw fb-rib-update-ref uint32 //count of writes
       |  +--rw mounts-using*
       |  |  +--rw mount-name string     //
       |  +--use pkt-eca:pkt-eca-policy-set
```

figure 5

4. YANG Modules

```
<CODE BEGINS> file "ietf-i2nsf-capability@2016-10-01.yang"
  module ietf-i2nsf-capability {
    namespace "urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability";
   // replace with iana namespace when assigned
    prefix "i2nsf-capability";
    import ietf-pkt-eca-policy {
          prefix pkt-eca-policy;
     }
```

```
// meta

  organization "IETF I2NSF WG";

contact
   "email: Susan Hares: shares@ndzh.com
    email: Robert Moskowitz rgm@htt-consult.com;
        email: Frank Xia
        email: Aldo Basile cataldo.basile@polito.it";


description
  "This module describes a capability model
       for I2NSF devices .";

      revision "2016-10-01" {
```



```
            description "second revision";
            reference "draft-hares-i2nsf-capability-yang-01.txt";
          }

       grouping ITResources {
            list ITResource {
              key ITResource-id;
                  leaf ITResource-id {
                     type uint64;
                     description "ID for ITResource";
                  }
                  leaf ITResource-name {
                          type string;
                          description "ITResource name.";
                  }
                  description "list of IT Resources.";
            }
            description "IT Resource grouping.";
          }


          grouping cfg-sec-content-caps {
           list cfg-fcn-groups {     // functions in 2 lists:
             key "group-name";       // group and functions
                  leaf group-name {
```

```
                    type string;
                    description " name of function
                     group";
                   }
                   list group-fnc-list {
                     key "fcn-name";
                     leaf fcn-name {
                      type string;
                      description "security content
                       function name";
                     }
                     leaf fcn-order-id {
                      type uint64;
                      description "function order
                      in list of functions.";
                     }
                     leaf default-action-id {
                      type uint64;
                      description "default
                      extended action id";
                     }
                     leaf default-cr-resolve-id {
                      type uint32;
```



```
                      description "default
                      policy conflict resolution
                      policy identifier.";
                     }
                     description "list of
                     functions per group.
                     e.g. group A has
                     5 functions.";
                   }

              description "list of
              groups with associated
                   security content functions.";
       }

         list cfg-sec-content-fcns {
           key "fcn-order-id function-name";
           leaf fcn-order-id {
```

```
  type uint64;
  description "order id for rule";
}
leaf function-name {
   type string;
    description "rule name";
 }
 list anti-virus {
    key "anti-virus-name";
    leaf anti-virus-name {
      type string;
      description "name of
      anti-virtus functionalty";
     }
    leaf anti-virus-supported {
        type boolean;
        description "anti-virus
       feature supported";
    }
  description "anti-virus functions";
}
list IPS {
   key "IPS-name";
   leaf IPS-name {
           type string;
           description "name of
           anti-virtus functionalty";
          }
          leaf IPS-supported {
           type boolean;
```



```
                  description "IPS
                  capability
                   supported";
                 }
                 description "IPS capability";
               }

               list IDS  {
                 key "IDS-name";
                 leaf IDS-name {
                  type string;
```

```
                    description "name of IDS";
                  }
                  leaf IDS-supported {
                   type boolean;
                   description "anti-virus
                    feature supported";
                  }
                  description "IDS
                    capabilities";
                }

          list url-filter  {
                    key "url-filter-name";
                    leaf url-filter-name {
                     type string;
                     description "name of IDS";
                    }
                    leaf url-filter-supported {
                     type boolean;
                     description "url filter
                      feature supported";
                    }
                    description "URL filter
                    capabilities";
                  }

                  list file-block  {
                    key "fblock-name";
                    leaf fblock-name {
                     type string;
                     description "name of
                     file block function";
                    }
                    leaf fblock-supported {
                     type boolean;
                     description "anti-virus
```



```
                      feature supported";
                    }
                    description "file block
                    capabilities";
```

```
                    }

                    list data-filter  {
                      key "dfilter-name";
                      leaf dfilter-name {
                       type string;
                       description "name of
                        data filer";
                      }
                      leaf dfilter-supported {
                       type boolean;
                       description "anti-virus
                        feature supported";
                      }
                      description "data filter
                       capabilities";
                    }

                    list app-behave  {
                      key "app-behave-name";
                      leaf app-behave-name {
                       type string;
                       description "name of
                        application behavior
                            control function.";
                      }
                      leaf app-behave-supported {
                       type boolean;
                       description "application
                        behavior control
                            security capability
                            supported.";
                      }
                      description "Application
                        behavior control security
                       capabilities";
                    }

                    list mail-filter  {
                      key "mfilter-name";
                      leaf mfilter-name {
                       type string;
                       description "name of
                        data filer";
```

```
                    }
                    leaf mfilter-supported {
                     type boolean;
                     description "mail filter
                     supported";
                    }
                    description "mail filter";
                  }


                  list pkt-capture  {
                    key "pkt-capture-name";
                    leaf pkt-capture-name {
                     type string;
                     description "name of
                      data filer";
                    }
                    leaf pkt-capture-supported {
                     type boolean;
                     description "pkt capture
                      facility supported";
                    }
                    description "packet capture
                     facility supported ";
                  }

                  list file-isolate  {
                    key "f-isolate-name";
                    leaf f-isolate-name {
                     type string;
                     description "name of
                      file isolate capability";
                    }
                    leaf f-isolate-supported {
                     type boolean;
                     description "file isolate
                     capability supported ";
                    }
                    description "file isolate
                    capability ";
                  }
               description "list of
                   security content capabilities.";
              }
          description "configured
              security content capabilities";
       }
```

```
       grouping cfg-content-sec-actions {
        list content-sec-actions {
              key "action-name";
              leaf action-name {
               type string;
           description "name of extra
                    content security action
                    beyond function policy";
              }
         description "list
             of content security actions";
        }
            description "configure
            content security actions
            configured beyond capability
            function existance";
          }

           grouping cfg-attack-mitigate-caps {
            // group and then rules
            list cfg-mitigate-fncs-groups {
              key "group-name";
                   leaf group-name {
                    type string;
                    description " name of function
                     group";
                   }
                   list group-mitigate-fncs-list {
                     key "fcn-name";
                     leaf fcn-name {
                      type string;
                      description "security content
                       function name";
                     }
                     leaf fcn-order-id {
                      type uint64;
                      description "function order
                      in list of functions.";
                     }
                     leaf default-action-id {
```

```
                    type uint64;
                    description "default
                    extended action id";
                   }
                   leaf default-cr-resolve-id {
                    type uint32;
                    description "default
                    policy conflict resolution
```



```
                    policy identifier.";
                   }
                   description "list of
                   functions per group.
                   e.g. group A has
                   5 functions.";
                  }

              description "list of
              groups with associated
                   attack mitigate functions.";
             }


         list cfg-attack-mitigate-rule {
               key "rule-order-id rule-name";
               leaf rule-order-id {
                     type uint64;
                     description "order id for
                     configured mitigate
                     function";
           }
           leaf rule-name {
                    type string;
                    description "mitigate
             rule name";
                   }
                   list cfg-sync-flood  {
                     key sync-flood-fcn;
                     leaf sync-flood-fcn {
                      type string;
                      description "name of
                      sync flood functionalty";
```

```
                    }
                    leaf sync-flood-fcn-supported {
                     type boolean;
                     description "sync-flood
                      mitigation fcn supported";
                    }
                    description "list of
                    sync flood mitigation
                    functions ";
                  }
                  list cfg-udp-flood {
                    key "udp-flood-fcn";
                    leaf udp-flood-fcn {
                     type string;
                     description "name of
```



```
                      udp flood mitigation function ";
                    }
                    leaf udp-flood-fcn-supported {
                     type boolean;
                     description "udp flood
                     prevent function
                     capability supported";
                    }
                    description "list of
                    udp-flood mitigation
                    functions node
                    (configured capability).";
                  }

                  list cfg-icmp-flood  {
                    key "icmp-flood-fcn";
                    leaf icmp-flood-fcn {
                     type string;
                     description "name of
                     icmp flood prevention
                     function";
                    }
                    leaf icmp-flood-fcn-supported {
                     type boolean;
                     description "icmp
                      flood mitigation
```

```
                    feature supported";
                  }
                  description "list for
                  icmp flood prevention
                  functions part of
                  attack mitigation
                    capabilities.";
                }


          list cfg-http-flood  {
                    key "http-flood-fcn";
                    leaf http-flood-fcn {
                     type string;
                     description "name of
                     http flood
                     mitigation function";
                    }
                    leaf http-flood-fcn-supported {
                     type boolean;
                     description "support
                     for http flood function
```



```
                     capability is active.";
                    }
                    description "list of
                    http flood
                    mitigation functions
                    configured ";
                  }

                  list cfg-dns-flood  {
                    key "dns-flood-fcn";
                    leaf dns-flood-fcn {
                     type string;
                     description "name of
                     dns flood mitigation
                     function";
                    }
                    leaf dns-flood-fcn-supported {
                     type boolean;
                     description "dns flood
```

```
                    mitigation support is
                    active.";
                  }
                  description "list of
                  dns flood
                  mitigation functions
                  configured.";
                }

                list cfg-dns-amplify {
                  key "dns-amplify-fcn";
                  leaf dns-amplify-fcn {
                   type string;
                   description "name of
                    dns amplify mitigation
                        function.";
                  }
                  leaf dfilter-supported {
                   type boolean;
                   description "dns
                   amplification mitigation
                   function is active.";
                  }
                  description "list of
                  dns amplification
                  mitigation functions
                  configured.";
                }
```



```
                list SSL-DoS  {
                  key "ssl-dos-fcn";
                  leaf ssl-dos-fcn {
                   type string;
                   description "name of
                   SSL DoS mitigation
                   function";
                  }
                  leaf ssl-dos-supported {
                   type boolean;
                   description "SSL DoS
                   mitigation function is
```

```
                    active.";
                  }
                  description "List of
                  SSL DoS functions configured.";
                }

                list cfg-IP-Sweep  {
                  key "ipsweep-fcn";
                  leaf ipsweep-fcn {
                   type string;
                   description "name of
                   ip sweep mitigation
                   function.";
                  }
                  leaf ipsweep-fcn-supported {
                   type boolean;
                   description "IP Sweep
                   mitigation function
                   active.";
                  }
                  description "list of
                  IP Sweep mitigation
                  functions in NSF device.";
                }

                list cfg-Port-scanning  {
                  key "port-scan-fcn";
                  leaf port-scan-fcn {
                   type string;
                   description "name of
                    port-scan mitigation
                        function.";
                  }
                  leaf port-scan-fcn-supported {
                   type boolean;
                   description "port scanning
```



```
                    mitigation fcn supported.";
                  }
                  description "List of
                  port scanning mitigation
                  functions. ";
```

```
                }

                list cfg-ping-of-death  {
                  key "pingd-fcn";
                  leaf pingd-fcn {
                   type string;
                   description "name of
                    ping of death
                        mitigation function";
                  }
                  leaf pingd-fcn-supported{
                   type boolean;
                   description "active support
            for this ping of death
            mitigation function";
                  }
                  description "List of ping of
                  death mitigation
                  functions.";
                }
             description "attack
                 mitigation rule .";
          }  // rules
       description "configured
             attack mitigation functions.";

    }   // cfg-attack-mitigate-policy-set

  container i2nsf-capabilities {
      list capabilty {
            key "nsf-name";
       leaf nsf-name {
             type string;
             description "name of
             nsf or nsf group
             capabilities drawn from.";
        }
           container cfg-net-secctl-capabilities {
            uses pkt-eca-policy:pkt-eca-policy-set;
            description "network security
              control capabilities configured.";
           }
           container cfg-sec-content-capabilities {
```

```
              uses cfg-sec-content-caps;
              uses cfg-content-sec-actions;
                  description "security content
                  capabilities configured.";
            }
            container cfg-attack-mitigate-capabilites {
              uses cfg-attack-mitigate-caps;
                  description "attack mitigation capabilities";
            }
        container cfg-ITResources {
              uses ITResources;
                  description "IT Resources
                  associated with NSF.";
            }
           description "List of NSF
           capabilities per nsf, nsf group
           or nsf application.";
      }  //end of list

     description "I2NSF capabilities";
    }  // end of container
  }
   <CODE ENDS>
```

5. IANA Considerations

No IANA considerations exist for this document at this time. URL will be added.

6. Security Considerations

Security of I2NSF is defined in (need reference here).

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

7.2. Informative References

[I-D.ietf-i2nsf-gap-analysis] Hares, S., Moskowitz, R., and D. Zhang, "Analysis of Existing work for I2NSF", draft-ietf-i2nsf-gap-analysis-00 (work in progress), February 2016.

[I-D.ietf-i2nsf-problem-and-use-cases]
Hares, S., Dunbar, L., Lopez, D., Zarny, M., and C. Jacquenet, "I2NSF Problem Statement and Use cases", draft-ietf-i2nsf-problem-and-use-cases-00 (work in progress), February 2016.

[I-D.ietf-i2nsf-terminology]
Hares, S., Strassner, J., Lopez, D., and L. Xia, "Interface to Network Security Functions (I2NSF) Terminology", draft-ietf-i2nsf-terminology-00 (work in progress), May 2016.

[I-D.ietf-i2rs-fb-rib-data-model]
Hares, S., Kini, S., Dunbar, L., Krishnan, R., Bogdanovic, D., and R. White, "Filter-Based RIB Data Model", draft-ietf-i2rs-fb-rib-data-model-00 (work in progress), June 2016.

[I-D.ietf-i2rs-pkt-eca-data-model]
Hares, S., Wu, Q., and R. White, "Filter-Based Packet Forwarding ECA Policy", draft-ietf-i2rs-pkt-eca-data-model-00 (work in progress), June 2016.

[I-D.ietf-netmod-acl-model]
Bogdanovic, D., Koushik, K., Huang, L., and D. Blair, "Network Access Control List (ACL) YANG Data Model", draft-ietf-netmod-acl-model-06 (work in progress), December 2015.

[I-D.ietf-opsawg-firewalls]
Baker, F. and P. Hoffman, "On Firewalls in Internet Security", draft-ietf-opsawg-firewalls-01 (work in progress), October 2012.

[I-D.xia-i2nsf-capability-interface-im]
Xia, L., Zhang, D., elopez@fortinet.com, e., Bouthors, N., and L. Fang, "Information Model of Interface to Network Security Functions Capability Interface", draft-xia-i2nsf-capability-interface-im-05 (work in progress), March 2016.

[I-D.xia-i2nsf-service-interface-dm]
Xia, L., Strassner, J., and D. Bogdanovic, "Data Model of

Interface to Network Security Functions Service Interface", draft-xia-i2nsf-service-interface-dm-00 (work in progress), February 2015.

[RFC2975] Aboba, B., Arkko, J., and D. Harrington, "Introduction to Accounting Management", RFC 2975, DOI 10.17487/RFC2975, October 2000, <http://www.rfc-editor.org/info/rfc2975>.

[RFC3198] Westerinen, A., Schnizlein, J., Strassner, J., Scherling, M., Quinn, B., Herzog, S., Huynh, A., Carlson, M., Perry, J., and S. Waldbusser, "Terminology for Policy-Based Management", RFC 3198, DOI 10.17487/RFC3198, November 2001, <http://www.rfc-editor.org/info/rfc3198>.

[RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, DOI 10.17487/RFC3234, February 2002, <http://www.rfc-editor.org/info/rfc3234>.

[RFC3539] Aboba, B. and J. Wood, "Authentication, Authorization and Accounting (AAA) Transport Profile", RFC 3539, DOI 10.17487/RFC3539, June 2003, <http://www.rfc-editor.org/info/rfc3539>.

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <http://www.rfc-editor.org/info/rfc4949>.

[RFC7277] Bjorklund, M., "A YANG Data Model for IP Management", RFC 7277, DOI 10.17487/RFC7277, June 2014, <http://www.rfc-editor.org/info/rfc7277>.

Authors' Addresses

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
USA

Phone: +1-734-604-0332
Email: shares@ndzh.com

Robert Moskowitz
HTT Consulting
Oak Park, MI
USA

Phone: +1-248-968-9809
Email: rgm@htt-consult.com

Liang Xia (Frank)
Huawei
101 Software Avenue, Yuhuatai District
Nanjing, Jiangsu
China

Email: Frank.xialiang@huawei.com