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

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of
the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

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 been 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-sectl-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-seccctl-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 capabilities for
each NSF entity (device, controller, APP). The portion of the top
level model that this explains is the following:

```
+--rw cfg-net-seccctl-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 action-id uint32
   +---rw eca-actions* [action-id]
   +---rw action-id uint32
   +---rw eca-ingress-actions* [action-id]
   |   ... (permit, deny, mirror)
   +---rw eca-fwd-actions* [action-id]
   |   ... (invoke, tunnel encap, fwd)
   +---rw eca-egress-actions* [action-id]
   |   ...
   +---rw eca-qos-actions* [action-id]
   |   ...

+---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* [action-id]
   +---rw ext-data-id integer

+---rw cfg-external-data* [cfg-ext-data-id]
   +---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
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 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
```
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-amp)
      |        +--rw dns-amp-fcn-supported boolean
```
2.5. IT Resources linked to Capabilities

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

```yaml
+--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.
4. YANG Modules

<CODE BEGINS> file "ietf-i2nsf-capability@2016-10-01.yang"
module 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.";

} groupings cfg-sec-content-caps {
    list cfg-fcn-groups {
        key "group-name";
        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-virus functionality";
  }
  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-virus functionality";
  }
  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
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 functionality";
    }
    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
calculated ";
}

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 function supported.";
  }
  description "List of port scanning mitigation functions configured.";
}
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 capability {
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-capabilities {
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


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.


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