IETF-116
Update Report for I2NSF Consumer-Facing Interface and I2NSF Registration Interface

March 30th, 2023
Jaehoon (Paul) Jeong and Patrick Lingga
Sungkyunkwan University
Email: {pauljeong, patricklink}@skku.edu

draft-ietf-i2nsf-consumer-facing-interface-dm-27
draft-ietf-i2nsf-registration-interface-dm-23
I2NSF Consumer-Facing Interface YANG Data Model

draft-ietf-i2nsf-consumer-facing-interface-dm-27
The YANG data model adds “profile” to follow the NSF-Facing Interface YANG Data Model, where two leaves for antivirus can be configured, i.e., “profile” and “exception-files”.

The “profile” is used to configure the “deny-list”, while the “exception-files” is to configure the “allow-list”.

OLD:

```
| ---rw anti-virus
|   | ---rw exception-files* string
```

NEW:

```
| ---rw anti-virus
|   | ---rw profile* string
|   | ---rw exception-files* string
```
case offset {
    leaf offset {
        type int32;
        units "bytes";
        description "The field specifies where to start searching for the specified content pattern within the payload. For example, an offset of 5 means to start looking for the specified content pattern after the first 5 bytes of the payload. A negative value means to start from the last bytes of the payload. For example, an offset of -5 means to start looking for the specified content pattern from the last 5 bytes of the payload.";
    }
}

• The latest YANG data model explains the handling of negative values for offset and distance in payload information.

case distance {
    leaf distance {
        type int32;
        units "bytes";
        description "The field specifies how far a payload should be ignored before starting to search for the specified content pattern relative to the end of the previous specified content pattern match. This can be thought of as exactly the same thing as offset, except it is relative to the end of the last pattern match instead of the beginning of the packet. For example, a distance of 5 means to start looking for the specified content pattern 5 bytes after the last byte of the matched pattern. A negative value means to start looking before the last byte of the previous matched pattern. For example, a distance of -5 means to start looking for the specified content pattern 5 bytes before the last byte of the previous matched pattern.

Note that this field cannot be used if the content is the first order of the list.";
    }
}
Comment from Transport Area:
It is not entirely clear from the rest of the context of this document, but if this filtering occurs anywhere other than the destination IP address of these packets, then ICMP messages from routers should be used, not those from hosts.

I.e., if the issue is packets to/from a NFV service, then host errors are appropriate, but if the issue is packets relayed through an NFV service, then router errors should be used instead.
I2NSF Registration Interface YANG Data Model for NSF Capability Registration

draft-ietf-i2nsf-registration-interface-dm-23
Latest Updates of Registration Interface (1/4)

OLD Version:

Registration and Query are separated.

NEW Version:

Registration and Query are combined.
**OLD Capability Registration YANG Tree:**

```
augment /i2nsfcap:nsf:
  +++rw nsf-specification
    | +++rw cpu
    |   | +++rw model? string
    |   | +++rw clock-speed? uint16
    |   | +++rw cores? uint8
    |   | +++rw threads? uint16
    | +++rw memory
    |   | +++rw capacity? uint32
    |   | +++rw speed? uint32
    | +++rw disk
    |   | +++rw capacity? uint32
    | +++rw bandwidth
    |   | +++rw outbound? uint64
    |   | +++rw inbound? uint64
+++rw nsf-access-info
  +++rw ip? union
  +++rw port? inet:port-number
  +++rw management-protocol? enumeration
  +++rw name? string
  +++rw password? ianach:crypt-hash
```

**OLD Capability Query YANG Tree:**

```
I2NSF Capability Query
rpcs:
  +++x nsf-capability-query
    +++w input
      | +++w query-nsf-capability
      |   | +++uses ietf-i2nsf-capability
    +++ro output
      +++ro nsf-access-info
        +++ro nsf-name? string
        +++ro ip? union
        +++ro port? inet:port-number
        +++ro management-protocol? enumeration
        +++ro name? string
        +++ro password? ianach:crypt-hash
```

**Figure 5. YANG Tree of NSF Capability Registration Module**

**Figure 6. YANG Tree of NSF Capability Query Module**
Latest Updates of Registration Interface (3/4)

NEW Capability Registration and Query Combined YANG Tree:

NSF Capability Registration
rpcs:
  +---x nsf-capability-registration
    |  +---w input
    |     |  +---w query-nsf-capability
    |     |      +--uses ietf-i2nsf-capability
    |  +--ro output
    |     +--ro nsf* [nsf-name]
    |        +--ro nsf-name string
    |        +--ro version? string
    |        +--uses ietf-i2nsf-capability
    |        +--ro nsf-specification
    |           +--ro cpu
    |           |  +--ro model? string
    |           |  +--ro clock-speed? uint16
    |           |  +--ro cores? uint8
    |           |  +--ro threads? uint16
    |           +--ro memory
    |           |  +--ro capacity? uint32
    |           |  +--ro speed? uint32
    |           +--ro disk
    |           |  +--ro capacity? uint32
    |           +--ro bandwidth
    |           |  +--ro outbound? uint64
    |           |  +--ro inbound? uint64
    |           +--ro nsf-access-info
    |             +--ro ip? union
    |             +--ro port? inet:port-number
    |             +--ro management-protocol? enumeration

• In the old version, the Capability Registration and Capability Query are separated. It complicates the architecture of Security Controller and Developer Management’s System (DMS).

• In this version, the new YANG data model combines Capability Registration and Capability Query with the same YANG data model which simplifies the architecture (i.e., Security Controller as a NETCONF Client and DMS as a NETCONF Server).

• The YANG data model uses an RPC statement:
  • The input utilizes Capability YANG data model to query the requested capabilities.
  • The output allows multiple NSFs (in the case one where an NSF is unable to provide the requested capabilities) to be registered with Security Controller along with their capabilities, specification, and access information.
NEW Capability Update YANG Tree:

I2NSF Capability Update
rpcs:
  +---x nsf-capability-update
    +---w input
      | +---w nsf-name? string
      | +---w version? string
    +--ro output
      +--ro nsf
        +--ro nsf-name? string
        +--ro version? string
        +--uses ietf-i2nsf-capability
        +--ro nsf-specification
          | +--ro cpu
          |   | +--ro model? string
          |   | +--ro clock-speed? uint16
          |   | +--ro cores? uint8
          |   | +--ro threads? uint16
          | +--ro memory
          |   | +--ro capacity? uint32
          |   | +--ro speed? uint32
          | +--ro disk
          |   | +--ro capacity? uint32
          |   +--ro bandwidth
          |     +--ro outbound? uint64
          |     +--ro inbound? uint64
        +--ro nsf-access-info
          +--ro ip? union
          +--ro port? inet:port-number
          +--ro management-protocol? enumeration

Figure 6. YANG Tree of NSF Capability Update Module

- In this version, a YANG data model to update the NSFs is provided.

- The YANG data model uses an RPC statement:
  - The inputs are the “name” and “version” of the NSF.
  - The output returns the latest capabilities, specification, and access information.

- In the case where no updates existed, the DMS can reply with a negative response (i.e., rpc-error with a message).
Next Step

- Currently both drafts are under IESG Evaluation.

- After IESG Evaluation, both drafts can be concluded, and all five I2NSF YANG Data Model Drafts can be published as RFCs.
  - Capability, NSF-Facing Interface, and Monitoring Interface
  - Consumer-Facing Interface and Registration Interface.