Service Function Chaining-Enabled I2NSF Architecture
(draft-hyun-i2nsf-nsf-triggered-steering-05)

IETF 101, London
March 21, 2018

Sangwon Hyun, Jaehoon (Paul) Jeong [Presenter], Jung-Soo Park, and Susan Hare
Updates from the Previous Version

• Changes from draft-hyun-i2nsf-nsf-triggered-steering-04:
  - Section 7.4 has been added in order to discuss the implementation considerations of a Service Function Chaining (SFC)-enabled I2NSF Architecture.
    • This section discusses the implementation of traffic steering by using OpenDaylight Controller supporting SFC.
  - The references have been updated to reflect the latest documents.
SFC Consideration

- Security Controller configures the classifier with service function chain/path information.
- Security Controller generates the forwarding information table of NSFs and configures the SFF with it.

- NSF path information
- Mapping between capability names and NSFs

SPI 1: NSF₁
SPI 2: NSF₁ → NSF₂

<table>
<thead>
<tr>
<th>SPI</th>
<th>SI</th>
<th>NH</th>
<th>Transport protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>10.1.1.2</td>
<td>GRE</td>
</tr>
</tbody>
</table>

- Forwarding information table to identify the next NSF from a given SPI and SI
SFC Implementation Consideration

- I2NSF Security Controller Function
- SDN Switch Controller Function
- SDN Switch Traffic Steering Function

According to security policy rules, generates the policies of the identified NSF Chain.

- Operate to support Service Function Chain
- SFC element (e.g., SF, SFF, SFC, SFP...) is created, updated, deleted for NSF Chain

Data Plane Elements processes
- Forwarding traffic based on the traffic forwarding rules

Deliver NSF Chain

SDN Switch Controller

Deliver the generated traffic forwarding rule

SDN Switch (e.g., SFF)
Next Steps

- The Service Function Chaining (SFC) of NSFs chaining with capability names (e.g., firewall, DPI, and DDoS attack mitigation) is required to fit into the I2NSF framework.

- For this, we need to consider a new interface called I2NSF-SFC Interface to support the Service Function Chaining (SFC) of NSFs.

- Design of I2NSF-SFC Interface
  - We will design the Information Model & YANG Data Model of I2NSF-SFC Interface.
Appendix (1/3)

- **SFC-based Packet Forwarding in I2NSF**
  - To trigger an advanced security action, NSF$_1$ appends the capability name required for the advanced security action into NSH.

- SPI 1: NSF$_1$
- SPI 2: NSF$_1$ $\rightarrow$ NSF$_2$

**NSH includes**
- Service Path Identifier (e.g., SPI=1)
- Service Index (e.g., SI=255)
- Capability name required for an advanced security action (e.g., DPI)
Appendix (2/3)

- Identify the particular NSF for DPI (NSF₂ is a DPI.) specified in NSH and determine the new NSF path of the packet
- Re-classification to change the existing path into the new one (SPI=2, SI=1)

NSH includes
- Service Path Identifier (e.g., SPI=1)
- Service Index (e.g., SI=2)
- NSF name required for an advanced security action (e.g., DPI)

The classifier may be co-resident with the NSFs.

The classifier may be co-resident with the NSFs.

- SPI 1: NSF₁
- SPI 2: NSF₁ → NSF₂
Appendix (3/3)

- Interpret the NSF path information
- Identify the next NSF on the path
- Forward the packet to the next NSF

1. Re-classification request & response
2. SPI 1: NSF₁, SPI 2: NSF₁ → NSF₂
3. Packet forwarding

<table>
<thead>
<tr>
<th>SPI</th>
<th>SI</th>
<th>NH</th>
<th>Transport protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>10.1.1.2</td>
<td>GRE</td>
</tr>
</tbody>
</table>