Introduction

- Objective
  - This document describes an architecture that integrates service function chaining (SFC) into the I2NSF framework to support packet forwarding between NSFs.

- Motivation
  - To support an advanced security action in the I2NSF framework that allows an NSF to call another type of NSF
  - To enable composite inspection of packets through various types of NSFs
  - To enable load balancing over multiple NSF instances combined with dynamic NSF instantiation
To trigger an advanced security action, NSF₁ appends the capability name required for the advanced security action in NSH.

NSH includes:
- Service Path Identifier (e.g., SPI=1)
- Service Index (e.g., SI=0)
- Capability name required for an advanced security action (e.g., DPI)

 SPI 1: NSF₁
 SPI 2: NSF₁ → NSF₂
SFC-based Packet Forwarding in I2NSF

- Identify the particular NSF for DPI (NSF$_2$ 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 ($SPI=2$, $SI=1$).

The classifier may be co-resident with the NSFs.

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

NSH includes:
- Service Path Identifier (e.g., $SPI=1$)
- Service Index (e.g., $SI=0$)
- NSF name required for an advanced security action (e.g., DPI)
SFC-based Packet Forwarding in I2NSF

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

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

<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>
The Security Controller configures the classifier with service function chain/path information.

The Security Controller generates the forwarding information table of NSFs and configures the SFF with it.
Tunneling-based Forwarding

- Tunneling protocols can be utilized to support packet forwarding between SFF and NSF.
- We implemented network tunneling based on GRE (Generic Routing Encapsulation).

**Packet format**

<table>
<thead>
<tr>
<th>L2 Header</th>
<th>L3 Header (outer IP) Protocol=47</th>
<th>GRE header PT=0x894F</th>
<th>NSH NP=0x1 SPI=1 SI=1</th>
<th>Original packet</th>
</tr>
</thead>
</table>
Discussion

• SFC is suitable for enforcing the default (pre-determined) NSF path.

• Re-classification is required to support an advanced security action that the next NSF is determined in the I2NSF framework.
  – Introducing some overhead particularly when the classifier exists separately from an NSF

• Identifying a particular NSF for the given capability name (e.g., DPI) is required to fit into the I2NSF framework.
  – Interface between the Security Controller and SFC component (e.g., classifier, SFF) is required. → I2NSF-SFC Interface?
Update from -03 Version

- The following changes have been made from draft-hyun-i2nsf-nsf-triggered-steering-03.
  - Section 7 has been added to discuss implementation considerations of the SFC-enabled I2NSF architecture.
Next Step

• We will specify more details of what kind of information should be included in the NSH header to support packet forwarding between NSF-SFs and the formats.

• Design of I2NSF-SFC Interface
  – Information model & data model
Thank you!
Any questions or comments?