Applicability of Interfaces to Network Security Functions to Networked Security Services (draft-ietf-i2nsf-applicability-01)

IETF 100, Singapore
November 14, 2017

Jaehoon (Paul) Jeong, Sangwon Hyun, Tae-Jin Ahn, Susan Hares, and Diego Lopez
Motivation of This Draft

 I2NSF Applicability
  – I2NSF Chartered Working Item

  – This draft explains how I2NSF framework and interfaces can be used for real network security services.

 Contents
  – Security service procedure in I2NSF framework
  – Use Case: Time–dependent web access control
    • Firewall & Web filter
I2NSF Framework

I2NSF Framework for user-friendly and efficient security service provisioning

1. **I2NSF User** asks for security services with high-level security policies to **Security Controller** via **Consumer-Facing Interface**.

2. **Security Controller** delivers low-level security policies to **NSFs** via **NSF-Facing Interface**.

3. **Network Security Function** configures such low-level security policies into its local system.

4. **Developer’s Mgmt System** registers the capabilities of NSFs via **Registration Interface**.
Information and Data Models for I2NSF

- **Consumer-Facing Interface**
  - Information Model
    - draft-kumar-i2nsf-client-facing-interface-im-04
  - Data Model
    - draft-jeong-i2nsf-consumer-facing-interface-dm-04

- **NSF-Facing Interface**
  - Information Model
    - draft-ietf-i2nsf-capability-00
  - Data Model
    - draft-kim-i2nsf-nsf-facing-interface-data-model-04

- **Registration Interface**
  - Information Model
    - draft-hyun-i2nsf-registration-interface-im-03
  - Data Model
    - draft-hyun-i2nsf-registration-interface-dm-02
Security Service Procedure in I2NSF Framework

1. High-level Security Policy
2. Identify security capabilities required for high-level security policy
3. Identify NSFs having the required security capabilities
4. Generate low-level security policies for the identified NSFs
5. Low-level Security Policy
6. Incoming packets
7. Outgoing packets

- NSF1
- NSF2

- Valid packets
- Invalid packets
Use Case: Time-dependent Web Access Control

1. Block the access of staff to Facebook from 9am to 6pm

Security Controller

2. Required security capabilities and NSFs
   - IP address and port number inspection ➔ Firewall
   - URL inspection and time check ➔ Web filter

3. Generate low-level security policies for Firewall and Web filter
Use Case: Time-dependent Web Access Control

1. **I2NSF User**
2. **Security Controller**
3. **Firewall**
4. **Web Filter**

- **Staff**
- **facebook.com**
- **Low-level Security Policy**
- **Staff’s IP and port 80?**
- **To facebook.com during working hours?**
- **SFC**
Combination of I2NSF and SDN

- **Accelerated Security Service**
  - Simple packet filtering rules by **SDN switch**
  - Complicated security inspection by **NSFs**

---

**I2NSF User**

1. Consumer-Facing Interface (RESTCONF/YANG)

2. NSF-Facing Interface (NETCONF/YANG)

**Clients**

**Switch Controller**

- Install new rules (e.g., drop packets with security policy)
- 3. Southbound Interface (Openflow or I2RS)

**Security Controller**

- High-level Policy
- Low-level Policy

**Switches**

1. Switch 1
2. Switch 2
3. Switch 3

**Outgoing packets**

**NSFs**

**Firewall**

**Web Filter**

**Valid packets**

**Invalid packets**

**Traffic Generation**

- 4. Traffic Generation
- Incoming packets
Update from -00 Version

The following changes have been made from draft-ietf-i2nsf-applicability-00.

- In Section 3, time-based web access control service is added as a general use case in the I2NSF framework.
- In Section 4, we explained the I2NSF framework with SDN to support the divided security policy enforcement for efficient security service.
- In Section 4.2, the centralized VoIP/VoLTE security system is clarified as a use case to explain the security service chaining using SFC.
Next Steps

- **Reflection of new use cases of I2NSF**
  - This draft will be extended with new use cases.
  - This draft will be described in more detail with the experience and lessons from IETF I2NSF Hackathon Project.

- **Plan: WGLC at IETF 101?**

- **Welcome your Feedback!**