Guidelines for Security Policy Translation in Interface to Network Security Functions

draft-yang-i2nsf-security-policy-translation-11

IETF 114, Philadelphia
July 26, 2022

Jaehoon (Paul) Jeong, Patrick Lingga, Jinhyuk Yang, and Jeonghyeon (Joshua) Kim
Sungkyunkwan University
Security Policy Translation in I2NSF Framework (From Consumer-Facing Interface to NSF-Facing Interface)

- **Security Client**
- **I2NSF User**
- **Security Management System**
- **Security Controller**
- **Security Network**
- **NSF A**
- **NSF B**
- **NSF C**
- **NSF D**
- **Server**

**NSF: Network Security Function** (e.g., Firewall, Web Filter, Deep Packet Inspection, Antivirus, etc.)

**Consumer-Facing Interface (for High-level Security Policy)**

**NSF-Facing Interface (for Low-level Security Policy)**

**Security Policy Translator:**
CFI Policy -> NFI Policy
• Consumer-Facing Interface (CFI):
  - CFI is required because the web applications developed by each vendor need to have a standard interface specifying the data types used when the I2NSF User and Security Controller communicate with each other using this interface.
  - Therefore, the CFI document specifies the required information, their data types, and encoding schemes so that high-level security policies (or configuration information for security policies) can be transferred to Security Controller through CFI.
  - These high-level policies can be translated into low-level security policies by the Security Controller.
NSF-Facing Interface (NFI):

- NFI focuses on providing security policy configuration for NSFs as a low-level policy to deploy security services.
- Security Controller delivers the translated low-level policies to NSFs according to their respective security capabilities.
- The data model provides a generic NSF (i.e., Access Control Lists (ACLs)), which operates on packet headers for L2, L3, and L4, and an advanced NSF (e.g., IPS, URL-Filtering, anti-DDoS, Antivirus, and VoIP or Voice over Cellular Network (VoCN) Filter).
- The ACLs provided in the NFI YANG data model is imported from RFC 8519 (YANG Data Model for Network Access Control Lists (ACLs)).
Top-Level YANG Tree Comparison

Consumer-Facing Interface (CFI):
module: ietf-i2nsf-cons-facing-interface
   +--rw i2nsf-cfi-policy* [name]
      |  +--rw name string
      |  +--rw language? string
      |  +--rw resolution-strategy? identityref
      |  +--rw rules* [name]
      |  |   ...  
      |  +--rw endpoint-groups
      |  |   ...  
      |  +--rw threat-prevention
      |  |   ...
      ...

   NSF-Facing Interface (NFI):
module: ietf-i2nsf-nsf-facing-interface
   +--rw i2nsf-security-policy* [name]
      |  +--rw name string
      |  +--rw language? string
      |  +--rw resolution-strategy? identityref
      |  +--rw priority-usage? identityref
      |  +--rw default-action? identityref
      |  +--rw rules* [name]
      |  |   ...
      |  +--rw rule-group
      |  |   ...

- The top-level CFI and NFI YANG data model provide the language-tag and resolution-strategy.
- default action and priority usage are not provided in CFI YANG data model.
  - Reason: The Security Policy Translator can set these both default action and priority usage to the low-level security policy.
  - Philosophy of CFI: To make CFI as simple as possible.

- In CFI, endpoint groups and threat prevention are used to register information (e.g., mapping a user to an IP address) with the database for high-level configuration.
  - endpoint groups: user-group, device-group, location-group, and url-group
  - threat prevention: threat-feed-list and payload-content
Rule-Level YANG Tree Comparison

**Consumer-Facing Interface (CFI):**

```Yang
  +--rw rules* [name]
    |   +--rw name       string
    |   +--rw priority?  uint8
    |   +--rw event
    |   |   ...
    |   +--rw condition
    |   |   ...
    |   +--rw action
    |   ...
```

**NSF-Facing Interface (NFI):**

```Yang
  +--rw rules* [name]
    |   +--rw name               string
    |   +--rw description?       string
    |   +--rw priority?          uint8
    |   +--rw enable?            boolean
    |   +--rw long-connection
    |   |   +--rw enable?     boolean
    |   |   +--rw duration?   uint32
    |   +--rw event
    |   |   ...
    |   +--rw condition
    |   |   ...
    |   +--rw action
    |   ...
```

- The CFI and NFI data model use the **Event-Condition-Action (ECA) policy rule** with priority for the rule is provided in both YANG data model.

- **long-connection** (i.e., a connection that is maintained after the socket connection is established) is provided in NFI to handle stateful network service.
  - **Reason:** The Security Policy Translator can set this **long-connection** to the low-level security policy.
  - **Philosophy of CFI:** To make CFI as simple as possible.

- The contents of the ECA is different for CFI and NFI data model as shown in the next slides.
Data Model Mapping between CFI and NFI for Policy Translation

• Relationship between CFI and NFI
  - CFI and NFI have almost one-to-one mapping relation.
  - The YANG trees of CFI and NFI have a similar structure, based on I2NSF Capability YANG Data Model.
  - With these characteristics, the attributes of CFI YANG module can be mapped to those of NFI YANG module by a deterministic mapping algorithm (e.g., Zhang-Shasha Algorithm).
  - The values of the attributes of a CFI Policy XML file can be converted to those of the attributes of an NFI Policy XML file by Deterministic Finite Automata (DFA).
  - The values of the attributes of the NFI Policy can be constructed into an NFI Policy XML file by a YANG-aware tool (e.g., PyangBind).
SPT has five components as follows:

- NSF Database
- Data Model Mapper
- Data Extractor
- Data Converter
- Policy Generator
An Exemplary High-level Security Policy

High-Level Policy for SNS Access Blocking

```xml
<i2nsf-cfi-policy
 xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-cfi-policy">
 <name>security_policy_for_blocking_sns</name>
 <rules>
  <name>block_access_to_sns_during_office_hours</name>
  <condition>
   <firewall>
    <source>employees</source>
   </firewall>
   <url>
    <url-name>sns-websites</url-name>
   </url>
  </condition>
  <actions>
   <primary-action>
    <action>drop</action>
   </primary-action>
  </actions>
 </rules>
</i2nsf-cfi-policy>
```
The Procedure of Policy Provisioning with NSFs

**Policy Data Conversion**

High-level Policy Data
- Policy name: sns_policy
- Rule name: block_sns_rule
- Source: employees
- URL-Name: sns-websites
- Action: drop

Low-level Policy Data
- Policy name: sns_policy
- Rule name: block_sns_rule
- Source-IPV4-Network: 192.0.2.0/24
- Action: drop

**NSF 1**
- IPv4 Firewall
  - IPv4-capability: Next-header, Source-address, Destination-address
  - TCP-capability: Source-port, Destination-port
  - Action-capability: drop, pass

**NSF 2**
- Low-level Policy Data for IPv4 Firewall
  - Policy name: sns_policy
  - Rule name: block_sns_rule
  - Source-IPV4-Network: 192.0.2.0/24
  - Advanced-action: web-filter

**NSF 3**
- Low-level Policy Data for Web Filter
  - Policy name: sns_policy
  - Rule name: block_sns_rule
  - URL-Category/User-defined: facebook, instagram
  - Action: drop
An Exemplary Low-level Security Policy

1. Low-Level Policy for Firewall

```xml
<i2nsf-security-policy
  <name>sns_access</name>
  <rules>
    <name>block_sns_access_during_operation_time_for_ipv4</name>
    <condition>
      <ipv4>
        <source-ipv4-network>192.0.2.0/24</source-ipv4-network>
      </ipv4>
    </condition>
    <action>
      <advanced-action>
        <content-security-control>
          url-filtering
        </content-security-control>
      </advanced-action>
    </action>
  </rules>
</i2nsf-security-policy>
```

employees translated to an IPv4 network subnet

2. Low-Level Policy for Web Filter

```xml
<i2nsf-security-policy
  <name>sns_access</name>
  <rules>
    <name>block_sns_access_during_operation_time</name>
    <condition>
      <url-category>
        <user-defined>Facebook</user-defined>
        <user-defined>Instagram</user-defined>
      </url-category>
    </condition>
    <action>
      <packet-action>
        <egress-action>drop</egress-action>
      </packet-action>
    </action>
  </rules>
</i2nsf-security-policy>
```
sns-websites translated into Facebook and Instagram
Next Step

• This draft explains the relationship between CFI and NFI as guidelines for security policy translation.

• It helps the audience to understand how to link the CFI data model and NFI data model.

• It suggests an exemplary architecture and procedure of Security Policy Translator (SPT).

• With this draft, an implementer can design and implement its own SPT that can work in the I2NSF ecosystem.

• This draft is proposed as a WG item in I2NSF Re-chartering.