Authors: J. Jeong   P. Lingga
         Sungkyunkwan University   Sungkyunkwan University
         J. Park
         ETRI

An Extension of I2NSF Framework for Security Management Automation in Cloud-Based Security Services

Abstract

This document describes an extension of the framework of Interface to Network Security Functions (I2NSF) for Security Management Automation (SMA) in cloud-based security services. The security management automation in this document deals with a security polity translation and a feedback-based security service enforcement. To support these two features in SMA, this document specifies an augmented architecture of the I2NSF framework with a new system component and a new interface.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 25 August 2022.

Copyright Notice

Table of Contents

1. Introduction

Interface to Network Security Functions (I2NSF) defines a framework
and interfaces for interacting with Network Security Functions
(NSFs) [RFC8192][RFC8329]. Note that an NSF is defined as software
that provides a set of security-related services, such as (i)
detecting unwanted activity, (ii) blocking or mitigating the effect
of such unwanted activity in order to fulfill service requirements,
and (iii) supporting communication stream integrity and
confidentiality [RFC8329]. The NSF can be implemented as a Virtual
Network Function (VNF) in a Network Functions Virtualization (NFV)
environment [ETSI-NFV][I-D.ietf-i2nsf-applicability].

This document describes an extension of the framework of Interface
to Network Security Functions (I2NSF) for Security Management
Automation (SMA) in cloud-based security services. The security
management automation includes a security polity translation and a
feedback-based security service enforcement. This document specifies
an augmented architecture of the I2NSF framework for the SMA
services with a new system component and a new interface.

For reliable management for networked security services, this document proposes a network management and verification facility using a decentralized audit system (e.g., blockchain [Bitcoin]). This audit system can facilitate the non-repudiation of configuration commands and monitoring data generated in the I2NSF framework.

Therefore, with the security service automation, this document facilitates the foundation of Intent-Based Networking (IBN) for intelligent security services [I-D.irtf-nmrg-ibn-concepts-definitions].

2. Terminology

This document uses the terminology described in [RFC8329] and [I-D.ietf-i2nsf-applicability]. In addition, the following terms are defined below:

  *Security Management Automation (SMA): It means that a high-level security policy from a user (or administrator) is well-enforced in a target I2NSF system. The high-level security policy can be translated into the corresponding low-level security policy by a security policy translator and dispatched to appropriate NSFs. Through the monitoring of the NSFs, the activity and performace of the NSFs is monitored and analyzed. If needed, the security rules of the low-level security policy are augmented or new security rules are generated and configured to appropriate NSFs.

  *Security Policy Translation (SPT): It means that a high-level security policy is translated to a low-level security policy that can be understood and configured by an NSF for a specific security service, such as firewall, web filter, deep packet inspection, DDoS-attack mitigation, and anti-virus.

  *Feedback-Based Security Management (FSM): It means that a security service is evolved by updating a security policy (having security rules) and adding new security rules for detected security attacks by processing and analzing the monitoring data of NSFs.

```
    +------------+
    | I2NSF User |
    +------------+
          ^
          | Consumer-Facing Interface
          v
+-------------------+    Registration    +-----------------------+
|Security Controller|<------------------>|Developer's Mgmt System|
+-------------------+    Interface       +-----------------------+
         ^       ^
         |       |
         |       |  Application Interface  +-----------------------+
         |       +------------------------>|    I2NSF Analyzer     |
         |                                 +-----------------------+
         | NSF-Facing Interface               ^       ^       ^
         |                                     |       |       |
         |                                     |       |       |
         |    +------------------------------+         |       |
         |    |              +----------------------+  |       |
         |    |              |   Monitoring Interface  |       |
         v    v              v                         |       v
    +----------------+ +---------------+  +-----------------------+
    |    NSF-1       |-|    NSF-2      |...|        NSF-n          |
    |   (Firewall)   | | (Web Filter)  |  |(DDoS-Attack Mitigator)|
    +----------------+ +---------------+  +-----------------------+
```
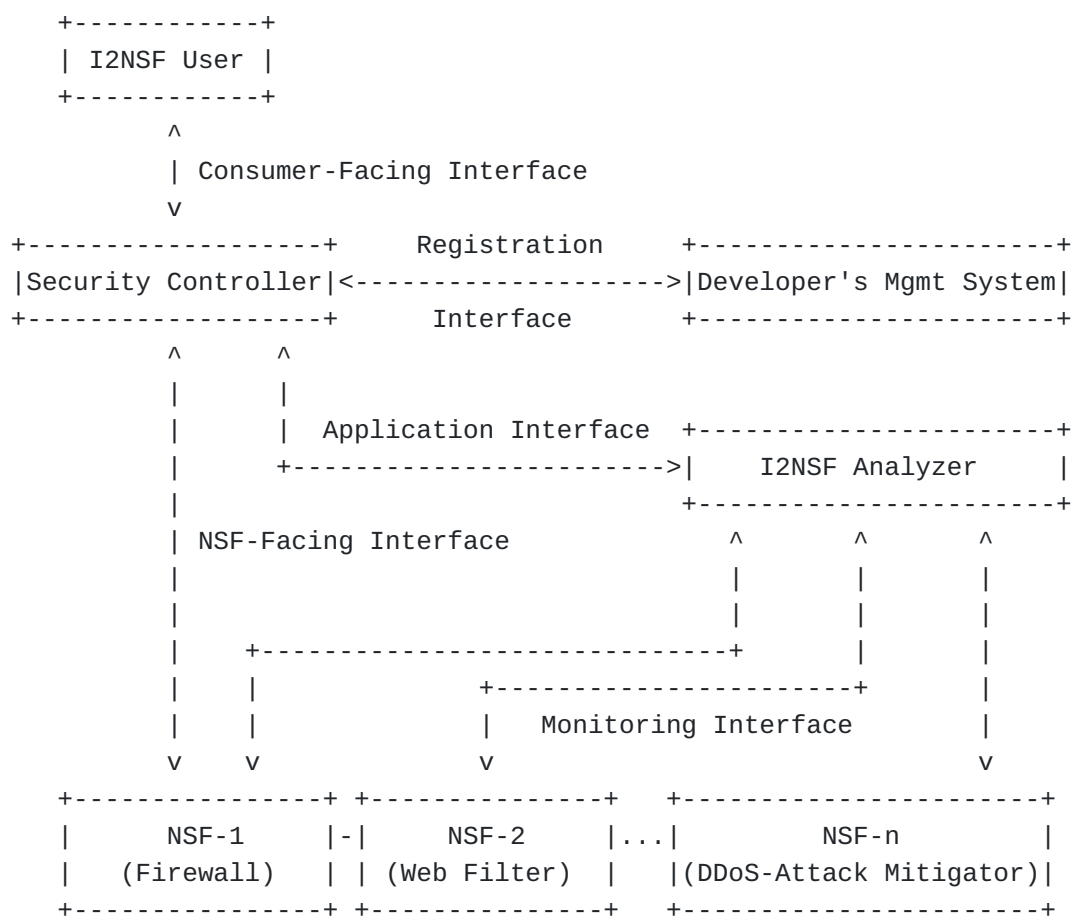
Figure 1: I2NSF Framework for Security Management Automation

3. I2NSF Framework for Security Management Automation

This section summarizes the I2NSF framework as defined in [RFC8329].
As shown in Figure 1, an I2NSF User can use security functions by
delivering high-level security policies, which specify security
requirements that the I2NSF user wants to enforce, to the Security
Controller via the Consumer-Facing Interface (CFI) [I-D.ietf-i2nsf-
consumer-facing-interface-dm].

3.1. Components with I2NSF Framework for Security Management Automation

The following are the system components for the SMA-based I2NSF
framework.

  *I2NSF User: An entity that delivers a high-level security policy
   to Security Controller.

  *Security Controller: An entity that controls and manages other
   system components in the I2NSF framework. It translates a high-
   level security policy into the corresponding low-level security

policy and selects appropriate NSFs to execute the security rules of the low-level security policy.

  *Developer's Management System (DMS): An entity that provides an image of of a virtualized NSF for a security service to the I2NSF framework, and registers the capability and access information of an NSF with Security Controller.

  *Network Security Function (NSF): An entity that is a Virtual Network Function (VNF) for a specific network security service such as firewall, web filter, deep packet inspection, DDoS-attack mitigation, and anti-virus.

  *I2NSF Analyzer: An entity that collects monitoring data from NSFs and analyzes such data for checking the activity and performance of the NSFs using machine learning techniques (e.g., Deep Learning [Deep-Learning]). If there is a suspicious attack activity for the target network or NSF, I2NSF Analyzer delivers a report of the augmentation or generation of security rules to Security Controller.

For SMA-based security services with Feedback-Based Security Management (FSM), I2NSF Analyzer as a new I2NSF component is required for the legacy I2NSF framework [RFC8329] to collect monitoring data of NSFs and analyzing them. The actual implementation of monitoring data analysis is out of the scope of this document.

3.2. Interfaces with SMA-Based I2NSF Framework

The following are the interfaces for the SMA-based I2NSF framework. Note that the interfaces are modeled with YANG [RFC6020] and security policies are delivered through either RESTCONF [RFC8040] or NETCONF [RFC6241].

  *Consumer-Facing Interface: An interface between I2NSF User and Security Controller for the delivery of a high-level security policy [I-D.ietf-i2nsf-consumer-facing-interface-dm].

  *NSF-Facing Interface: An interface between Security Controller and an NSF for the delivery of a low-level security policy [I-D.ietf-i2nsf-nsf-facing-interface-dm].

  *Registration Interface: An interface between a DMS and Security Controller for the registration of an NSF's capability and access information with the Security Controller or the query of an NSF for a required low-level security policy [I-D.ietf-i2nsf-registration-interface-dm].

*Monitoring Interface: An interface between an NSF and I2NSF
 Analyzer for collecting monitoring data from an NSF to check the
 activity and performance of an NSF for a possible malicious
 traffic [I-D.ietf-i2nsf-nsf-monitoring-data-model].

*Application Interface: An interface between I2NSF Analyzer and
 Security Controller for the delivery of a report of the
 augmentation or generation of security rules to Security
 Controller, which lets Security Controller apply the report for
 security rules to its security policy management.

For SMA-based security services with FSM, Application Interface as a
new I2NSF interface is required for the legacy I2NSF framework
[RFC8329] to deliver a report of the augmentation or generation of
security rules to Security Controller on the basis of the analyzed
monitoring data of NSFs.

4. Inter-Interface Automatic Policy Mapping

To facilitate Security Policy Translation (SPT), Security Controller
needs to have a security policy translator that performs the
translation of a high-level security policy into the corresponding
low-level security policy. For the automatic SPT services, the I2NSF
framework needs to bridge a high-level YANG data model and a low-
level YANG data model in an automatic manner [I-D.ietf-i2nsf-
applicability][I-D.yang-i2nsf-security-policy-translation]. Note
that a high-level YANG data model is for the I2NSF Consumer-Facing
Interface [I-D.ietf-i2nsf-consumer-facing-interface-dm], and a low-
level YANG data model is for the I2NSF NSF-Facing Interface [I-
D.ietf-i2nsf-nsf-facing-interface-dm].

Figure 2 shows automatic mapping of high-level and low-level data
models. Automatic Data Model Mapper takes a high-level YANG data
module for the Consumer-Facing Inteface and a low-level YANG data
module for the NSF-Facing Interface. It then constructs a mapping
table associating the data attributes (or variables) of the high-
level YANG data module with the corresponding data attributes (or
variables) of the low-level YANG data module. Also, it generates a
set of production rules of the grammar for the construction of an
XML file of low-level security policy rules.

Figure 3 shows high-to-low security policy translation. A security
policy translator is a component of Security Controller. The
translator consists of three components such as Policy Data
Extractor, Policy Attribute Mapper, and Policy Constructor.
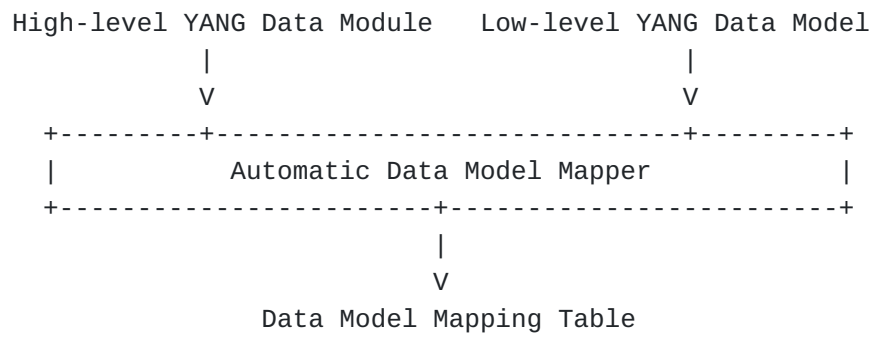
```
     High-level YANG Data Module   Low-level YANG Data Model
                 |                              |
                 V                              V
      +---------+----------------------------+---------+
      |             Automatic Data Model Mapper        |
      +------------------------+-----------------------+
                               |
                               V
                   Data Model Mapping Table


   Figure 2: Automatic Mapping of High-level and Low-level Data Models
```

```
+------------------------------------------------------+
|                                                      |
|                  I2NSF User                          |
|                                                      |
+------------------------+-----------------------------+
                         | Consumer-Facing Interface
                         |
              High-level Security Policy
   Security              |
   Controller            V
+------------------------+-----------------------------+
|  Security Policy       |                             |
|  Translator            V                             |
|  +-------------------+---------------------+  |
|  |                                         |  |
|  |        +-------------------------+      |  |
|  |        | Policy Data Extractor   |      |  |
|  |        +-------------------------+      |  |
|  |                                         |  |
|  |        +-------------------------+      |  |
|  |        | Policy Attribute Mapper |      |  |
|  |        +-------------------------+      |  |
|  |                                         |  |
|  |        +-------------------------+      |  |
|  |        |    Policy Constructor   |      |  |
|  |        +-------------------------+      |  |
|  |                                         |  |
|  +-------------------+---------------------+  |
|                      |                        |
|                      V                        |
+------------------------+-----------------------------+
                         |  NSF-Facing Interface
                         |
              Low-level Security Policy
                         |
                         V
+------------------------+-----------------------------+
|                                                      |
|                    NSF(s)                            |
|                                                      |
+------------------------------------------------------+
```
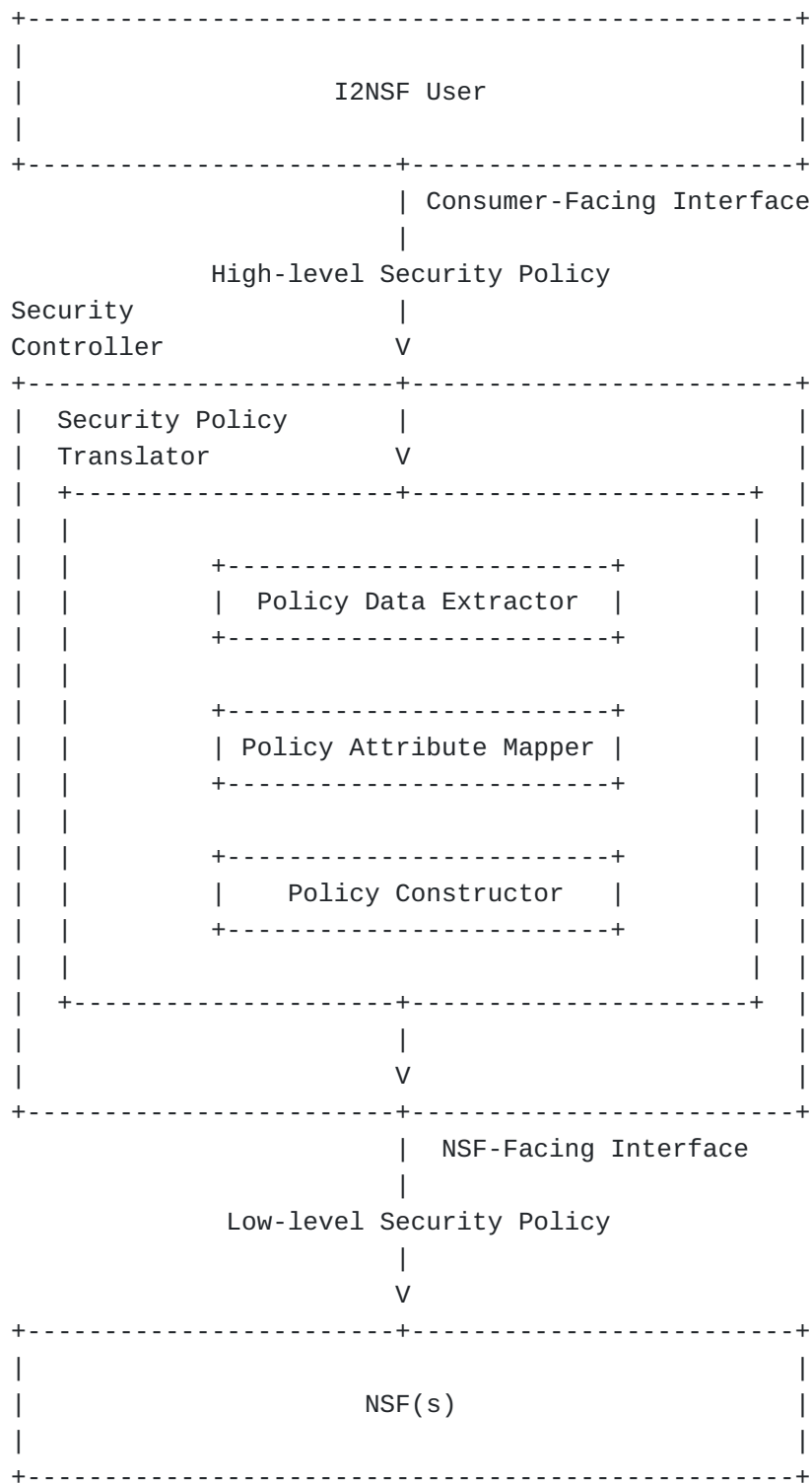
        Figure 3: High-to-Low Security Policy Translation

Policy Data Extractor extracts attributes related to a security
policy from a high-level security policy XML file that is delivered
from an I2NSF User to a Security Controller [I-D.ietf-i2nsf-
consumer-facing-interface-dm].

Policy Attribute Mapper maps the attributes and their values of a high-level security policy to the corresponding attributes and their values of a low-level security policy. Note that the values of a high-level security policy may involve a human language and must be converted to an appropriate value for a low-level security policy (e.g., employees -> 192.0.1.0/24).

Policy Constructor constructs a low-level security policy XML file that is delivered from the Security Controller to an appropriate NSF [I-D.ietf-i2nsf-nsf-facing-interface-dm].

5. Security Audit System

The I2NSF framework is weak to both an inside attack and a supply chain attack since it trusts in NSFs provided by Developer's Management System (DMS) and assumes that NSFs work for their security services appropriately. [I-D.ietf-i2nsf-applicability].

To detect the malicious activity of either an insider attacker with its DMS or a supply chain attacker with its compromised DMS, a security audit system is required for the I2NSF framework. For this audit service in the I2NSF framework, a decentralized security audit system (e.g., blockchain [Bitcoin]) is required. This audit system can facilitate the non-repudiation of configuration commands and monitoring data generated in the I2NSF framework.

A security audit system has four main objectives such as follows:

  *To check the existence of a security policy, a management system
   and its procedures;

  *To identify and understand the existing vulnerabilities and risks
   of a supply chain attacker;

  *To review existing security controls on operational,
   administrative and managerial issues;

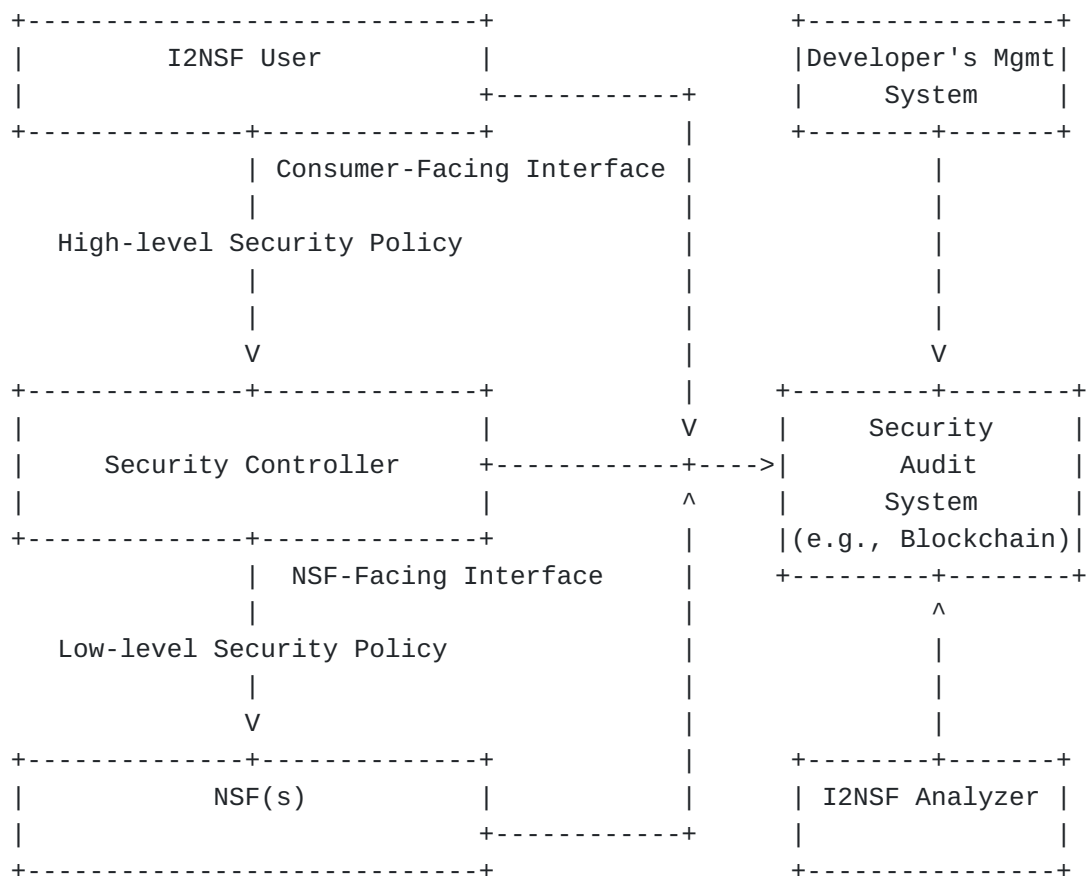 *To provide recommendations and corrective actions for further
   improvement.

```
+------------------------------+         +----------------+
|         I2NSF User           |         |Developer's Mgmt|
|                     +------------+      |    System      |
+--------------+--------------+   |       +--------+-------+
               | Consumer-Facing Interface |        |
               |                  |        |        |
     High-level Security Policy   |        |
               |                  |        |
               |                  |        |
               V                  |        V
+--------------+--------------+   |   +---------+--------+
|                     |        V  |   |   Security       |
|    Security Controller   +------------+---->|       Audit       |
|                     |        ^  |   |       System      |
+--------------+--------------+   |   |(e.g., Blockchain)|
               | NSF-Facing Interface |   +---------+--------+
               |                  |        ^
     Low-level Security Policy    |        |
               |                  |        |
               V                  |        |
+--------------+--------------+   |   +--------+-------+
|           NSF(s)            |   |   | I2NSF Analyzer |
|                     +------------+   |                |
+------------------------------+       +----------------+
```

                Figure 4: Activity Auditing with Security Audit System

   [Figure 4](#) shows activity auditing with a security audit system in the
   I2NSF framework. All the components in the I2NSF framwork report its
   activities (such as configuration commands and monitoring data) to
   Security Audit System (e.g., Blockchain) as transactions. The
   security audit system can analyze the reported activities from the
   I2NSF components to detect malicious activities such as supply chain
   attack.

   In order to determine a minimum set of controls required to reduce
   the risks from a supply chain attacker, the security audit system
   should analyze the activities of all the components in the I2NSF
   framework periodically, evaluate possible risks, and take an action
   to such risks since vulnerabilities and threats may change in
   different environments over time.

6. Security Considerations

   The same security considerations for the I2NSF framework [RFC8329]
   are applicable to this document.

   The development and introduction of I2NSF Analyzer in the I2NSF
   Framework will create new security concerns that have to be

anticipated at the design and specification time. The usage of machine learning to analyze the data add a risk of its model to be attacked (e.g., adversarial attack) and can result in a bad security policy being deployed.

7. IANA Considerations

This document does not require any IANA actions.

8. References

8.1. Normative References

[RFC8192]   Hares, S., Lopez, D., Zarny, M., Jacquenet, C., Kumar, R., and J. Jeong, "Interface to Network Security Functions (I2NSF): Problem Statement and Use Cases", RFC 8192, DOI 10.17487/RFC8192, July 2017, <https://www.rfc-editor.org/info/rfc8192>.

[RFC8329]   Lopez, D., Lopez, E., Dunbar, L., Strassner, J., and R. Kumar, "Framework for Interface to Network Security Functions", RFC 8329, DOI 10.17487/RFC8329, February 2018, <https://www.rfc-editor.org/info/rfc8329>.

[RFC6020]   Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC8040]   Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC6241]   Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[I-D.ietf-i2nsf-consumer-facing-interface-dm]
            Jeong, J. (., Chung, C., Ahn, T., Kumar, R., and S. Hares, "I2NSF Consumer-Facing Interface YANG Data Model", Work in Progress, Internet-Draft, draft-ietf-i2nsf-consumer-facing-interface-dm-16, 28 January 2022, <https://www.ietf.org/archive/id/draft-ietf-i2nsf-consumer-facing-interface-dm-16.txt>.

[I-D.ietf-i2nsf-nsf-facing-interface-dm] Kim, J. (., Jeong, J. (., Park, J., Hares, S., and Q. Lin, "I2NSF Network Security Function-Facing Interface YANG Data Model", Work in Progress, Internet-Draft, draft-ietf-i2nsf-nsf-facing-

interface-dm-20, 31 January 2022, <https://www.ietf.org/archive/id/draft-ietf-i2nsf-nsf-facing-interface-dm-20.txt>.

[I-D.ietf-i2nsf-registration-interface-dm] Hyun, S., Jeong, J. (., Roh, T., Wi, S., and J. Park, "I2NSF Registration Interface YANG Data Model", Work in Progress, Internet-Draft, draft-ietf-i2nsf-registration-interface-dm-14, 28 January 2022, <https://www.ietf.org/archive/id/draft-ietf-i2nsf-registration-interface-dm-14.txt>.

[I-D.ietf-i2nsf-nsf-monitoring-data-model]
        Jeong, J. (., Lingga, P., Hares, S., Xia, L. (., and H. Birkholz, "I2NSF NSF Monitoring Interface YANG Data Model", Work in Progress, Internet-Draft, draft-ietf-i2nsf-nsf-monitoring-data-model-14, 28 January 2022, <https://www.ietf.org/archive/id/draft-ietf-i2nsf-nsf-monitoring-data-model-14.txt>.

8.2. Informative References

[I-D.ietf-i2nsf-applicability] Jeong, J. P., Hyun, S., Ahn, T., Hares, S., and D. R. Lopez, "Applicability of Interfaces to Network Security Functions to Network-Based Security Services", Work in Progress, Internet-Draft, draft-ietf-i2nsf-applicability-18, 16 September 2019, <https://www.ietf.org/archive/id/draft-ietf-i2nsf-applicability-18.txt>.

[I-D.irtf-nmrg-ibn-concepts-definitions]
        Clemm, A., Ciavaglia, L., Granville, L. Z., and J. Tantsura, "Intent-Based Networking - Concepts and Definitions", Work in Progress, Internet-Draft, draft-irtf-nmrg-ibn-concepts-definitions-06, 15 December 2021, <https://www.ietf.org/archive/id/draft-irtf-nmrg-ibn-concepts-definitions-06.txt>.

[I-D.yang-i2nsf-security-policy-translation] Jeong, J. (., Lingga, P., Yang, J., and C. Chung, "Security Policy Translation in Interface to Network Security Functions", Work in Progress, Internet-Draft, draft-yang-i2nsf-security-policy-translation-10, 21 February 2022, <https://www.ietf.org/archive/id/draft-yang-i2nsf-security-policy-translation-10.txt>.

[ETSI-NFV] "Network Functions Virtualisation (NFV); Architectural Framework", Available: https://www.etsi.org/deliver/etsi_gs/nfv/001_099/002/01.01.01_60/gs_nfv002v010101p.pdf, October 2013.

[Bitcoin]   Nakamoto, S., "Bitcoin: A Peer-to-Peer Electronic Cash
                System", URL: https://bitcoin.org/bitcoin.pdf, May 2009.

[Deep-Learning] Goodfellow, I., Bengio, Y., and A. Courville, "Deep
                Learning", Publisher: The MIT Press, URL: https://
                www.deeplearningbook.org/, November 2016.

Appendix A. Acknowledgments

Appendix B. Contributors

This document is made by the group effort of I2NSF working group.
Many people actively contributed to this document, such as Linda
Dunbar, Yoav Nir, and Qin Wu. The authors sincerely appreciate their
contributions.

The following are co-authors of this document:

Yunchul Choi Electronics and Telecommunications Research Institute
218 Gajeong-Ro, Yuseong-Gu Daejeon, 34129 Republic of Korea EMail:
cyc79@etri.re.kr

Younghan Kim School of Electronic Engineering Soongsil University
369, Sangdo-ro, Dongjak-gu Seoul 06978 Republic of Korea EMail:
younghak@ssu.ac.kr

Appendix C. Changes from draft-jeong-i2nsf-security-management-automation-02

The following changes are made from draft-jeong-i2nsf-security-
management-automation-02:

  *This version is updated to fix the typos and clarify the text
   explaining Policy Attribute Mapper.

  *Add a new Security Consideration for machine learning usage to
   produce a security policy.

Authors' Addresses

Jaehoon (Paul) Jeong
Department of Computer Science and Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea

Phone: +82 31 299 4957
Email: pauljeong@skku.edu
URI: http://iotlab.skku.edu/people-jaehoon-jeong.php

Patrick Lingga
Department of Electronic, Electrical and Computer Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea

Phone: +82 31 299 4957
Email: patricklink@skku.edu

Jung-Soo Park
Electronics and Telecommunications Research Institute
218 Gajeong-Ro, Yuseong-Gu
Daejeon
305-700
Republic of Korea

Phone: +82 42 860 6514
Email: pjs@etri.re.kr