Revision Letter

Applicability of Interfaces to Network Security Functions to Network-Based Security Services

(Old Draft Name: draft-ietf-i2nsf-applicability-09 and New Draft Name: draft-ietf-i2nsf-applicability-10)

Jaehoon Paul Jeong

05/02/2019

Hi!

I'm picking up where ekr left off (https://mailarchive.ietf.org/arch/msg/i2nsf/bVTGfSXR70UcFkwfkV4FsNHg8uo) with an AD review of draft-ietf-i2nsf-applicability-09.

(1) Section 1. Please do not expand NSF again the second sentence. The acronym NSF was defined in the first sentence.
=> Yes, NSF is not expanded again in the second sentence.

In Section 1.

OLD	NEW
Interface to Network Security Functions (I2NSF) defines a framework and interfaces for interacting with Network Security Functions (NSFs). Note that Network Security Function (NSF) is defined as a funcional block for a security service within an I2NSF framework that has well-defined I2NSF NSF-facing interface and other external interfaces and well-defined functional behavior [NFV-Terminology].	Interface to Network Security Functions (I2NSF) defines a framework and interfaces for interacting with Network Security Functions (NSFs). Note that 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 fulfil service requirements, and (iii) supporting communication stream integrity and confidentiality [i2nsf-terminology].

[i2nsf-terminology] Hares, S., Strassner, J., Lopez, D., Xia, L., and H. Birkholz, "Interface to Network Security Functions (I2NSF) Terminology", draft-ietf-i2nsf-terminology-07 (work in progress), January 2019.

(2) Global Typo - s/funcional/functional/
=> As shown in the above table for (1), “functional” is not used for the definition of NSF any more.

(3) Section 1. I had difficulty understanding the sentence:
"Note that Network Security Function (NSF) is defined as a functional block for a security service within an I2NSF framework that has well-defined I2NSF NSF-facing interface and other external interfaces and well-defined functional behavior [NFV-Terminology]."
=> I rephrased the definition of NSF with one defined in I2NSF Terminology Draft [i2nsf-terminology].

In Section 1.

OLD	NEW
Note that Network Security Function (NSF) is defined as a funcional block for a security service	Note that Network Security Function (NSF) is defined as software that provides a set of

within an I2NSF framework that has well-defined I2NSF NSF-facing interface and other external interfaces and well-defined functional behavior [NFV-Terminology].	security-related services, such as (i) detecting unwanted activity, (ii) blocking or mitigating the effect of such unwanted activity in order to fulfil service requirements, and (iii) supporting communication stream integrity and confidentiality [i2nsf-terminology].

** I'm not clear on what a functional block is and [NFV-Terminology] doesn't define it (although it does define other things also using this terminology)

=> In the new text for the definition of Network Security Function does not use a functional block any more.

** Why is this NSF definition different than the one provided in RFC8329 - "Network Security Functions (NSFs) are packet-processing engines that inspect and optionally modify packets traversing networks, either directly or in the context of sessions to which the packet is associated" or RFC8192, "An NSF is a function that is used to ensure integrity, confidentiality, or availability of network communication; to detect unwanted network activity; or to block, or at least mitigate, the effects of unwanted activity."

=> The definition of Network Security Function (NSF) is from I2NSF Terminology Draft [i2nsf-terminology] rather than NFV Terminology Standard [NFV-Terminology].

**Why use the [NVF-Terminology] citation? It does not appear to have an entry for NSF in the terms/definitions.
=> The definition of Network Security Function (NSF) is from I2NSF Terminology Draft [i2nsf-terminology] rather than NFV Terminology Standard [NFV-Terminology]. Thus, [NFV-Terminology] is cited for NSF.

(4) Section 1. Per "The I2NSF framework allows ... by utilizing the capabilities of such products and the virtualization of security functions in the NFV platform", I don't understand what the second clause ("by utilizing ...") is adding. It seems to simply restate that the products have capabilities and will be virtualized (which is implicit in the NFV).

=> We remove the redundant restatement and clarify how heterogeneous NSFs can be used in the I2NSF framework.

In Section 1.

OLD	NEW
The I2NSF framework allows heterogeneous NSFs developed by different security solution vendors to be used in the Network Functions Virtualization (NFV) environment [ETSI-NFV] by utilizing the capabilities of such products and the virtualization of security functions in the NFV platform.	The I2NSF framework allows heterogeneous NSFs developed by different security solution vendors to be used in the Network Functions Virtualization (NFV) environment [ETSI-NFV] by utilizing the capabilities of such NSFs through I2NSF interfaces such as Customer-Facing Interface [consumer-facing-inf-dm] and NSF-Facing Interface [nsf-facing-inf-dm]. In the I2NSF framework, each NSF initially registers the profile of its own capabilities into the Security Controller (i.e., network operator management system [RFC8329]) in the I2NSF system via Registration Interface [registration-inf-dm] so that each NSF can be selected and used to enforce a given security policy from I2NSF User

	(i.e., network security administrator). Note that Developer's Management System (DMS) is management software that provides a vendor's security service software as a Virtual Network Function (VNF) in an NFV environment (or middlebox in the legacy network) as an NSF, and registers the capabilities of an NSF into Security Controller via Registration Interface for a security service [RFC8329].

(5) Section 1. Per "In the I2NSF framework, each NSF initially registers the profile of its own capabilities into the system in order for themselves to be available in the system", this sentence doesn't parse for me.

=> We clarify the registration of an NSF into the I2NSF framework with its capabilities along with Registration Interface and Developer's Management System.

In Section 1.

OLD	NEW
In the I2NSF framework, each NSF initially registers the profile of its own capabilities into the system in order for themselves to be available in the system.	The I2NSF framework allows heterogeneous NSFs developed by different security solution vendors to be used in the Network Functions Virtualization (NFV) environment [ETSI-NFV] by utilizing the capabilities of such NSFs through I2NSF interfaces such as Customer-Facing Interface [consumer-facing-inf-dm] and NSF-Facing Interface [nsf-facing-inf-dm]. In the I2NSF framework, each NSF initially registers the profile of its own capabilities into the Security Controller (i.e., network operator management system [RFC8329]) in the I2NSF system via Registration Interface [registration-inf-dm] so that each NSF can be selected and used to enforce a given security policy from I2NSF User (i.e., network security administrator). Note that Developer's Management System (DMS) is management software that provides a vendor's security service software as a Virtual Network Function (VNF) in an NFV environment (or middlebox in the legacy network) as an NSF, and registers the capabilities of an NSF into Security Controller via Registration Interface for a security service [RFC8329].

Do you mean, "In the I2NSF framework, each NSF initially registers a profile of its capabilities in the system"? If so, I think clarity of what system (I think "I2NSF system") is being referenced is needed.

=> In the above table, "Security Controller in the I2NSF system" replaced "the system" to clarify which system component will take charge of the registration of an NSF with some capabilities. We define Developer's Management System and Security Controller from RFC 8329 and I2NSF Terminology Draft [i2nsf-terminology], respectively.

In Section 1.

OLD	NEW
In the I2NSF framework, each NSF initially registers the profile of its own capabilities into the system in order for themselves to be available in the system.	The I2NSF framework allows heterogeneous NSFs developed by different security solution vendors to be used in the Network Functions Virtualization (NFV) environment [ETSI-NFV] by utilizing the capabilities of such NSFs through I2NSF interfaces such as Customer-Facing Interface [consumer-facing-inf-dm] and NSF-Facing Interface [nsf-facing-inf-dm]. In the I2NSF framework, each NSF initially registers the profile of its own capabilities into the Security Controller (i.e., network operator management system [RFC8329]) in the I2NSF system via Registration Interface [registration-inf-dm] so that each NSF can be selected and used to enforce a given security policy from I2NSF User (i.e., network security administrator). Note that Developer's Management System (DMS) is management software that provides a vendor's security service software as a Virtual Network Function (VNF) in an NFV environment (or middlebox in the legacy network) as an NSF, and registers the capabilities of an NSF into Security Controller via Registration Interface for a security service [RFC8329]. Security Controller is defined as a management component that contains control plane functions to manage NSFs and facilitate information sharing among other components (e.g., NSFs and I2NSF User) in an I2NSF system [i2nsf-terminology]. Security Controller maintains the mapping between a capability and an NSF, so it can perform to translate a high-level security policy received from I2NSF User to a low-level security policy configured and enforced in an NSF [policy-translation]. Security Controller can monitor the states and security attacks in NSFs through NSF monitoring [nsf-monitoring-dm].

(6) Section 1. Per "In addition, the Security Controller ...", this sentence introduces the concept of a Security Controller but doesn't define it. Also, this seems like a level of detail not needed in the introduction.

=> We define Security Controller from I2NSF Terminology Draft as follows.

In Section 1.

OLD	NEW
	Security Controller is defined as a management component that contains control plane functions to manage NSFs and facilitate information sharing among other components (e.g., NSFs and I2NSF User) in an I2NSF system [i2nsf-terminology]. Security Controller maintains the

	mapping between a capability and an NSF, so it can perform to translate a high-level security policy received from I2NSF User to a low-level security policy configured and enforced in an NSF [policy-translation]. Security Controller can monitor the states and security attacks in NSFs through NSF monitoring [nsf-monitoring-dm].

(7) Section 2,
This document uses the terminology described in [RFC7665], [RFC7149],
[ITU-T.Y.3300], [ONF-OpenFlow], [ONF-SDN-Architecture],
[ITU-T.X.1252], [ITU-T.X.800], [NFV-Terminology], [RFC8329],
[i2nsf-terminology], [consumer-facing-inf-dm], [i2nsf-nsf-cap-im],
[nsf-facing-inf-dm], [registration-inf-dm], and
[nsf-triggered-steering].

This sentence has 15 references covering hundreds of pages as having the relevant terminology. Are all of them needed? That's seems like a lot background reading.

=> I remove some references except important references as follows.

In Section 2.

OLD	NEW
This document uses the terminology described in [RFC7665], [RFC7149], [ITU-T.Y.3300], [ONF-OpenFlow], [ONF-SDN-Architecture], [ITU-T.X.1252], [ITU-T.X.800], [NFV-Terminology], [RFC8329], [i2nsf-terminology], [consumer-facing-inf-dm], [i2nsf-nsf-cap-im], [nsf-facing-inf-dm], [registration-inf-dm], and [nsf-triggered-steering].	This document uses the terminology described in [RFC7665], [RFC7149], [ITU-T.Y.3300], [ONF-SDN-Architecture], [ITU-T.X.800], [NFV-Terminology], [RFC8329], and [i2nsf-terminology].

(8) Figure 1. I found it confusing that this Figure 1 diagram didn't use the same names as Figure 1 of [RFC8329]. Specifically, why did the "Network Operator Management System" get renamed a "Security Controller"?

=> Security Controller is more frequently used in I2NSF WG and I2NSF drafts than Network Operator Management System. Security Controller is specified as Network Operator Management System in RFC 8329 in the text before Figure 1 as follows.

In Section 1.

OLD	NEW
In the I2NSF framework, each NSF initially registers the profile of its own capabilities into the system in order for themselves to be available in the system.	In the I2NSF framework, each NSF initially registers the profile of its own capabilities into the Security Controller (i.e., network operator management system [RFC8329]) in the I2NSF system via Registration Interface [registration-inf-dm] so that each NSF can be selected and used to enforce a given security policy from I2NSF User (i.e., network security administrator).

(9) Section 3. Recommend the following editorial change since none of the third paragraph has anything to do with the NSF-Facing Interface:
OLD: Finally, the Security Controller sends the generated low-level security policies to the NSFs [i2nsf-nsf-cap-im][nsf-facing-inf-dm].

NEW: Finally, the Security Controller sends the generated low-level security policies to the NSFs [i2nsf-nsf-cap-im][nsf-facing-inf-dm] via the NSF Facing Interface.

DROP: The Security Controller requests NSFs to perform low-level security services via the NSF-Facing Interface.

=> We reflect the requested replacement as follows:

In Section 3.

OLD	NEW
Finally, the Security Controller sends the generated low-level security policies to the NSFs [i2nsf-nsf-cap-im][nsf-facing-inf-dm]. The Security Controller requests NSFs to perform low-level security services via the NSF-Facing Interface.	Finally, the Security Controller sends the generated low-level security policies to the NSFs via the NSF-Facing Interface [nsf-facing-inf-dm].

(10) Section 3. Per the final sentence of paragraph 2, why is [i2nsf-nsf-cap-im] appropriate? Doesn't the Security Controller use only the YANG module from [nsf-facing-inf-dm]?

=> Yes, we remove the reference to the information model draft [i2nsf-nsf-cap-im] because the reference to the data model [nsf-facing-inf-dm] is enough like the above box.

(11) Section 3. Per "Note that an inside attacker at the DMS can seriously weaken the I2NSF system ...", I concur with the assessment that a DMS can subvert the I2NSF system. Three related points:

** The boundary/scope of an I2NSF system wasn't clear to me. It appears to me that an I2NSF system is security controller + NSFs. There are several interfaces defined for the controller and NSFs. Everything else (e.g., DMS, I2NSF user) is outside the scope of the I2NSF system, correct? I draw attention to this distinction because identifying where this insider is located needs to be clearer.

=> DMS and I2NSF User are within the scope of the I2NSF system because DMS provides the Security Controller with the capability information of NSFs via Registration Interface, and I2NSF gives high-level security policies to the Security Controller via Consumer-Facing Interface. DMS can be compromised to attack the Security Controller by providing the Security Controller with malicious NSFs, and controlling those NSFs in real time. Similarly, I2NSF User can be compromised to attack the Security Controller by allowing hackers to intrude the I2NSF system or generating useless security policies to let the resources in the I2NSF system be exhausted.

In Section 3.

OLD	NEW
Note that an inside attacker at the DMS can seriously weaken the I2NSF system's security.	Note that an inside attacker at the DMS can seriously weaken the I2NSF system's security.

	That is, DMS can be compromised to attack the Security Controller by providing the Security Controller with malicious NSFs, and controlling those NSFs in real time.

=> We specify an inside attack at I2NSF User and also the counterattack as follows:

In Section 3.

OLD	NEW
Note that an inside attacker at the DMS can seriously weaken the I2NSF system's security.	The Consumer-Facing Interface between an I2NSF User and the Security Controller can be implemented using, for example, RESTCONF [RFC8040]. Data models specified by YANG [RFC6020] describe high-level security policies to be specified by an I2NSF User. The data model defined in [consumer-facing-inf-dm] can be used for the I2NSF Consumer-Facing Interface. Note that an inside attacker at the I2NSF User can misuse the I2NSF system so that the network system under the I2NSF system is vulnerable to security attacks. To handle this type of threat, the Security Controller needs to monitor the activities of all the I2NSF Users as well as the NSFs through the I2NSF NSF monitoring functionality [nsf-monitoring-dm]. Note that the monitoring of the I2NSF Users is out of scope for I2NSF.

** If the DMS can provide the software package for the NSF, I'm not sure how the insider threat is mitigated. The attacker can already run a software load of her choice on your network (that you have permitted).

=> Through the NSF monitoring, the Security Controller can monitor the activities and states of NSFs to diagnosis to see whether the NSFs are working in normal conditions or in abnormal conditions including the insider threat.

In Section 3.

OLD	NEW
	Note that an inside attacker at the DMS can seriously weaken the I2NSF system's security. That is, DMS can be compromised to attack the Security Controller by providing the Security Controller with malicious NSFs, and controlling those NSFs in real time. To deal with this type of threat, the role of the DMS should be restricted to providing an I2NSF system with the software package/image for NSF execution, and the DMS should never be able to access NSFs in online/activated status for the I2NSF system's security. On the other hand, an access to active NSFs should be allowed only to the Security Controller, not the DMS during the provisioning time of those NSFs to the I2NSF system.

	However, note that an inside attacker can access the active NSFs, which are being executed as either VNFs or middleboxes in the I2NSF system, through a back door (i.e., an IP address and a port number that are known to the DMS to control an NSF). However, the Security Controller can detect and prevent inside attacks by monitoring the activities of all the DMSs as well as the NSFs through the I2NSF NSF monitoring functionality [nsf-monitoring-dm]. Through the NSF monitoring, the Security Controller can monitor the activities and states of NSFs, and then can make a diagnosis to see whether the NSFs are working in normal conditions or in abnormal conditions including the insider threat.

** Per https://mailarchive.ietf.org/arch/msg/i2nsf/Xc92QkEPgRWC3FKuRvnaiNNFY2o, I concur with ekr that the text needs to be clearer on what the DMS can do to the I2NSF system.

=> The DMS provides the Security System with the capability information of NSFs so that the Security Controller matches the given security policies to the corresponding NSFs with the capability information, and also provide a vendor's security service software or middlebox for a network security function. The definition of DMS is provided in the text as follows:

In Section 1.

OLD	NEW
	Note that Developer's Management System (DMS) is management software that provides a vendor's security service software as a Virtual Network Function (VNF) in an NFV environment (or middlebox in the legacy network) as an NSF, and registers the capabilities of an NSF into Security Controller via Registration Interface for a security service [RFC8329].

(12) Section 3, Per "On the other hand, an access to running (online) NSFs should be allowed only to the Security Controller, not the DMS.", this sentence isn't clear to me.

** It doesn't parse so I don't know who is supposed to get what access -- specifically, "an access to running NSFs"

=> "An access to running NSFs" means "an access to active NSFs, which are being executed as either VNFs or middleboxes in the I2NSF system, through a back door (i.e., an IP address and a port number that are known to the DMS to control an NSF)". We specify the access to NSFs from an inside attacker at a DMS as follows:

In Section 3.

OLD	NEW
On the other hand, an access to running (online) NSFs should be allowed only to the Security Controller, not the DMS.	On the other hand, an access to active NSFs should be allowed only to the Security Controller, not the DMS during the provisioning time of those

	NSFs to the I2NSF system. However, note that an inside attacker can access the active NSFs, which are being executed as either VNFs or middleboxes in the I2NSF system, through a back door (i.e., an IP address and a port number that are known to the DMS to control an NSF). However, the Security Controller can detect and prevent inside attacks by monitoring the activities of all the DMSs as well as the NSFs through the I2NSF NSF monitoring functionality [nsf-monitoring-dm]. Through the NSF monitoring, the Security Controller can monitor the activities and states of NSFs, and then can make a diagnosis to see whether the NSFs are working in normal conditions or in abnormal conditions including the insider threat.

** "running (online) NSFs" is proposing an operational construct which is also not clear to me. It that the equivalent of saying in production? If it means production, is there a distinction being made between interacting with the DMS during the time of provisioning and then after the fact?

=> “Running (online) NSFs” means “Active NSFs which are being executed as either VNFs or middleboxes in the I2NSF system”. Thus, “running” means not “production”, but “provisioning” as follows:

In Section 3.

OLD	NEW
On the other hand, an access to running (online) NSFs should be allowed only to the Security Controller, not the DMS.	On the other hand, an access to active NSFs should be allowed only to the Security Controller, not the DMS during the provisioning time of those NSFs to the I2NSF system. However, note that an inside attacker can access the active NSFs, which are being executed as either VNFs or middleboxes in the I2NSF system, through a back door (i.e., an IP address and a port number that are known to the DMS to control an NSF). However, the Security Controller can detect and prevent inside attacks by monitoring the activities of all the DMSs as well as the NSFs through the I2NSF NSF monitoring functionality [nsf-monitoring-dm]. Through the NSF monitoring, the Security Controller can monitor the activities and states of NSFs, and then can make a diagnosis to see whether the NSFs are working in normal conditions or in abnormal conditions including the insider threat.

(13) Section 3, Per "Also, the Security Controller can detect and prevent inside attacks by monitoring the activity of all the DMSs ... through the I2NSF NSF monitoring capability", is the monitoring interface also capable of observing the DMS? I ask because the monitoring interface is described in RFC8329 as part of I2NSF NSF-Facing Interface (Section 3.2). The Registration Interface description (Section 3.3 of RFC8329) makes no reference to any monitoring capability.
=> Yes, the NSF monitoring capability is performed over the NSF-Facing Interface [nsf-monitoring-dm]. The Security Controller can monitor the states and security attacks (including inside attacks) in NSFs

through NSF monitoring [nsf-monitoring-dm]. For monitoring DMSs, DMS monitoring capability is required. However, this DMS monitoring is out of scope for I2NSF.

In Section 3.

OLD	NEW
Also, the Security Controller can detect and prevent inside attacks by monitoring the activity of all the DMSs as well as the NSFs through the I2NSF NSF monitoring functionality [nsf-monitoring-dm].	However, the Security Controller can detect and prevent inside attacks by monitoring the activities of all the DMSs as well as the NSFs through the I2NSF NSF monitoring functionality [nsf-monitoring-dm]. Through the NSF monitoring, the Security Controller can monitor the activities and states of NSFs, and then can make a diagnosis to see whether the NSFs are working in normal conditions or in abnormal conditions including the insider threat. Note that the monitoring of the DMSs is out of scope for I2NSF.

(14) Section 3, I'm not clear on what is MTI or the alternatives. The text says "The Consumer-Facing Interface ... can be implemented ... by [consumer-facing-inf-dm]" and "the NSF facing interface ... can be implemented using NETCONF ... [with] ... the data model defined in [nsf-facing-inf-dm]". Why can? If not with those references then with what?
=> The Consumer-Facing Interface can be implemented as an XML file based on the Consumer-Facing Interface YANG data model [consumer-facing-inf-dm] along with RESTCONF, which befits a web-based user interface for an I2NSF User. In a similar way, the NSF-Facing Interface can be implemented as an XML file based on the NSF-Facing Interface YANG data model [nsf-facing-inf-dm] along with NETCONF, which befits a command-line-based remote-procedure call for a Security Controller.

In Section 3.

OLD	NEW
The Consumer-Facing Interface between an I2NSF User and the Security Controller can be implemented using, for example, RESTCONF [RFC8040]. Data models specified by YANG [RFC6020] describe high-level security policies to be specified by an I2NSF User. The data model defined in [consumer-facing-inf-dm] can be used for the I2NSF Consumer-Facing Interface.	The Consumer-Facing Interface can be implemented as an XML file based on the Consumer-Facing Interface data model [consumer-facing-inf-dm] along with RESTCONF [RFC8040], which befits a web-based user interface for an I2NSF User to send a Security Controller a high-level security policy. Data models specified by YANG [RFC6020] describe high-level security policies to be specified by an I2NSF User. The data model defined in [consumer-facing-inf-dm] can be used for the I2NSF Consumer-Facing Interface.

In Section 3.

OLD	NEW
The NSF-Facing Interface between the Security Controller and NSFs can be implemented using NETCONF [RFC6241]. YANG data models describe low-level security policies for the sake of NSFs, which are translated from the high-level security policies by the Security Controller. The data model defined in [nsf-facing-inf-dm] can be used for the I2NSF NSF-Facing Interface.	The NSF-Facing Interface can be implemented as an XML file based on the NSF-Facing Interface YANG data model [nsf-facing-inf-dm] along with NETCONF, which befits a command-line-based remote-procedure call for a Security Controller to configure an NSF with a low-level security policy. Data models specified by YANG [RFC6020] describe low-level security policies

	for the sake of NSFs, which are translated from the high-level security policies by the Security Controller. The data model defined in [nsf-facing-inf-dm] can be used for the I2NSF NSF-Facing Interface.

(15) Section 3. What it intentional to say that the Consumer interface can be RESTCONF+YANG (with a reference); the NSF-Facing Interface is NETCONF (but YANG with no reference); and the registration interface is RESTCONF (no reference to YANG)?

=> The Consumer-Facing Interface can be implemented as an XML file based on the Consumer-Facing Interface YANG data model, and the XML file can be sent from an I2NSF User to a Security Controller through the RESTCONF protocol for remote-procedure call.

In Section 3.

OLD	NEW
The Consumer-Facing Interface between an I2NSF User and the Security Controller can be implemented using, for example, RESTCONF [RFC8040]. Data models specified by YANG [RFC6020] describe high-level security policies to be specified by an I2NSF User. The data model defined in [consumer-facing-inf-dm] can be used for the I2NSF Consumer-Facing Interface.	The Consumer-Facing Interface can be implemented as an XML file based on the Consumer-Facing Interface data model [consumer-facing-inf-dm] along with RESTCONF [RFC8040], which befits a web-based user interface for an I2NSF User to send a Security Controller a high-level security policy. Data models specified by YANG [RFC6020] describe high-level security policies to be specified by an I2NSF User. The data model defined in [consumer-facing-inf-dm] can be used for the I2NSF Consumer-Facing Interface.

=> The NSF-Facing Interface can be implemented as an XML file based on the NSF-Facing Interface YANG data model, and the XML file can be sent from a Security Controller to an NSF through the NETCONF protocol for remote-procedure call.

In Section 3.

OLD	NEW
The NSF-Facing Interface between the Security Controller and NSFs can be implemented using NETCONF [RFC6241]. YANG data models describe low-level security policies for the sake of NSFs, which are translated from the high-level security policies by the Security Controller. The data model defined in [nsf-facing-inf-dm] can be used for the I2NSF NSF-Facing Interface.	The NSF-Facing Interface can be implemented as an XML file based on the NSF-Facing Interface YANG data model [nsf-facing-inf-dm] along with NETCONF, which befits a command-line-based remote-procedure call for a Security Controller to configure an NSF with a low-level security policy. Data models specified by YANG [RFC6020] describe low-level security policies for the sake of NSFs, which are translated from the high-level security policies by the Security Controller. The data model defined in [nsf-facing-inf-dm] can be used for the I2NSF NSF-Facing Interface.

=> The Registration Interface can be implemented as an XML file based on the Registration Interface YANG data model, and the XML file can be sent from a DMS to a Security Controller through the NETCONF protocol for remote-procedure call. Since NETCONF is more appropriate for general

configuration for NSF capability information at a Security Controller than RESTCONF, NETCONF is used for the Registration Interface.

In Section 3.

OLD	NEW
The Registration Interface between the Security Controller and the Developer's Management System can be implemented by RESTCONF [RFC8040]. The data model defined in [registration-inf-dm] can be used for the I2NSF Registration Interface.	The Registration Interface can be implemented as an XML file based on the Registration Interface YANG data model [registration-inf-dm] along with NETCONF [RFC6241], which befits a command-line-based remote-procedure call for a DMS to send a Security Controller an NSF's capability information. Data models specified by YANG [RFC6020] describe the registration of an NSF's capabilities to enforce security services at the NSF. The data model defined in [registration-inf-dm] can be used for the I2NSF Registration Interface.

(16) Section 4. I found the term "an example XML code" vague given that this document is supposed to be an applicability statement highlighting I2NSF. To what schema does this XML conform? Is this a notional example or a complete instance? On what interface would this have been sent?

=> We show a real Consumer-Facing Interface XML code from the Consumer-Facing Interface data model draft [consumer-facing-inf-dm] with some modification as follows:

In Section 4.

OLD

```
<I2NSF>
   <name>block_website</name>
   <cond>
      <src>Staff_Member's_PC</src>
      <dest>Example.com</dest>
      <time-span-start>9:00AM</time-span-start>
      <time-span-end>-6:00PM</time-span-end>
   </cond>
   <action>block<action>
</I2NSF>
```

NEW

```
<?xml version="1.0" encoding="UTF-8" ?>
    <ietf-i2nsf-cfi-policy:policy>
      <policy-name>block_website</policy-name>
      <rule>
        <rule-name>block_website_during_working_hours</rule-name>
        <event>
          <time-information>
            <begin-time>09:00</begin-time>
            <end-time>18:00</end-time>
          </time-information>
        </event>
        <condition>
          <firewall-condition>
            <source-target>
              <src-target>Staff_Member's_PC</src-target>
            </source-target>
```

```
            </firewall-condition>
            <custom-condition>
              <destination-target>
                <dest-target>Example.com</dest-target>
              </destination-target>
            </custom-condition>
          </condition>
          <action>
            <primary-action>drop</primary-action>
          </action>
        </rule>
      </ietf-i2nsf-cfi-policy:policy>
```

In Section 4.

OLD	NEW
Figure 2 is an example XML code for this web filter: … The security policy name is "block_website" with the tag "name". The filtering condition has the source group "Staff_Member's_PC" with the tag "src", the destination website "Example.com" with the tag "dest", the filtering start time is the time "9:00AM" with the tag " time-span-start", and the filtering end time is the time "6:00PM" with the tag "time-span-end". The action is to "block" the packets satisfying the above condition, that is, to drop those packets.	Figure 2 is an example XML code for this web filter that is sent from the I2NSF User to the Security Controller via the Consumer-Facing Interface [consumer-facing-inf-dm]: … The security policy name is "block_website" with the tag "policy-name", and the security policy rule name is "block_website_during_working_hours" with the tag "rule-name". The filtering event has the time span where the filtering begin time is the time "09:00" (i.e., 9:00AM) with the tag "begin-time", and the filtering end time is the time "18:00" (i.e., 6:00PM) with the tag "end-time". The filtering condition has the source target of "Staff_Member's_PC" with the tag "src-target", the destination target of a website "Example.com" with the tag "dest-target". The action is to "drop" the packets satisfying the above event and condition with the tag "primary-action".

(17) Section 4. Grammatical Nit.

s/it is assumed that an NSF of firewall/ /it is assumed than a firewall NSF/

s/NSF of web filter/
/web filter NSF/

=> “NSF of web filter” is replaced by “web filter NSF”, and “NSF of firewall” is replaced by “firewall NSF” as follows:

In Section 4.

OLD	NEW
In this scenario, it is assumed that an NSF of firewall has the IP address and port number	In this scenario, it is assumed that a firewall NSF has the IP address and port number inspection

inspection capabilities and an NSF of web filter has URL inspection capability. … Finally, the Security Controller sends the low-level security rules of the IP address and port number inspection to the NSF of firewall and the low-level rules for URL inspection to the NSF of web filter.	capabilities and a web filter NSF has URL inspection capability. … Finally, the Security Controller sends the low-level security rules of the IP address and port number inspection to the firewall NSF and the low-level rules for URL inspection to the web filter NSF.

(18) Section 5. What is the purpose of including this section if there is an entire draft (draft-hyun-i2nsf-nsf-triggered-steering) focused on the topic?

=> The purpose of Section 5 is to include the traffic steering text in draft-hyun-i2nsf-nsf-triggered-steering because this draft will not be adopted as I2NSF WG draft. I delete the reference to the draft where I am the corresponding author, but add the reference to SFC [RFC7665].

In Section 3.

OLD	NEW
Also, the I2NSF framework can enforce multiple chained NSFs for the low-level security policies by means of SFC techniques for the I2NSF architecture described in [nsf-triggered-steering].	Also, the I2NSF framework can enforce multiple chained NSFs for the low-level security policies by means of SFC techniques for the I2NSF architecture [RFC7665].

In Section 4.

OLD	NEW
3. The firewall triggers the web filter to further inspect the packet, and the packet is forwarded from the firewall to the web filter. SFC technology can be utilized to support such packet forwarding in the I2NSF framework [nsf-triggered-steering].	3. The firewall triggers the web filter to further inspect the packet, and the packet is forwarded from the firewall to the web filter. SFC technology can be utilized to support such packet forwarding in the I2NSF framework [RFC7665].

=> I delete the reference to NFV for I2NSF [i2nsf-nfv-architecture] because the detailed NFV operations for I2NSF is out of scope for I2NSF in the text as follows:

In Section 7.

OLD	NEW
More details about the I2NSF framework based on the NFV reference architecture are described in [i2nsf-nfv-architecture].	

(19) Section 5, "To trigger an advanced security action in the I2NSF architecture, the current NSF appends a metadata describing the security capability required for the advanced action to the suspicious packet and sends the packet to the classifier."

** Editorial nit: s/NSF appends a metadata/NSF appends metadata/

=> We fix the typo as follows:

In Section 5.

OLD	NEW
To trigger an advanced security action in the I2NSF architecture, the current NSF appends a metadata describing the security capability required for the advanced action to the suspicious packet and sends the packet to the classifier.	To trigger an advanced security action in the I2NSF architecture, the current NSF appends metadata describing the security capability required for the advanced action to the suspicious packet to the network service header (NSH) of the packet [RFC8300]. It then sends the packet to the classifier.

** What is the reference for this meta-data format?

=> The reference for this metadata format is RFC 8300 – Next Service Header. The metadata is added to a context header in the next service header for SFC: https://tools.ietf.org/html/rfc8300#page-7

In Section 5.

OLD	NEW
To trigger an advanced security action in the I2NSF architecture, the current NSF appends metadata describing the security capability required for the advanced action to the suspicious packet and sends the packet to the classifier.	To trigger an advanced security action in the I2NSF architecture, the current NSF appends metadata describing the security capability required for the advanced action to the suspicious packet to the network service header (NSH) of the packet [RFC8300]. It then sends the packet to the classifier.

(20) Section 6. What is the role of the DMS is this scenario? Why does the controller need to rely on [NFV MANO] if all information about the capabilities is already provided by the DMS? Since the SDN and other NSF operate using the same data model/interface, isn't the different between an SDN and NSF opaque to the controller? I would have assumed that an SDN is simply a specific type of NSF with particular capabilities.

=> The role of the DMS is to register the capabilities of an NSF into the Security Controller. The security controller relies on NFV MANO because NFV MANO is in charge of the life-cycle management of an NSF as a VNF running on top of the NFV infrastructure. I clarify these questions and comments in the text as follows:

In Section 6.

OLD	NEW
In this system, the enforcement of security policy rules is divided into the SDN forwarding elements (e.g., switch running as either a hardware middle box or a software virtual switch) and NSFs (e.g., firewall running in a form of a virtual network function [ETSI-NFV]).	In this system, the enforcement of security policy rules is divided into the SDN forwarding elements (e.g., switch running as either a hardware middle box or a software virtual switch) and NSFs (e.g., firewall running in a form of a virtual network function (VNF) [ETSI-NFV]). Note that NSFs are created or removed by the NFV Management and Orchestration (MANO) [ETSI-NFV-MANO], performing the life-cycle management of NSFs as VNFs. Refer to Section 7 for the detailed discussion of the NSF life-cycle management in the NFV MANO for I2NSF.

In Section 7.

OLD	NEW

The Developer's Management System (DMS) in the I2NSF framework is responsible for registering capability information of NSFs into the Security Controller. Those NSFs are created or removed by a virtual network functions manager (VNFM) in the NFV architecture that performs the life-cycle management of VNFs.	The Developer's Management System (DMS) in the I2NSF framework is responsible for registering capability information of NSFs into the Security Controller. However, those NSFs are created or removed by a virtual network functions manager (VNFM) in the NFV MANO that performs the life-cycle management of VNFs. Note that the life-cycle management of VNFs are out of scope for I2NSF.

=> Yes, SDN forwarding element (i.e., SDN switch) is a specific type of VNF rather than NSF because an NSF is for security services rather than for packet forwarding. I clarify these questions and comments in the text as follows:

In Section 6.

OLD	NEW
The distinction between software-based SDN forwarding elements and NSFs, which can both run as virtual network functions, may be necessary for some management purposes in this system. For this, we can take advantage of the NFV MANO where there is a subsystem that maintains the descriptions of the capabilities each VNF can offer [ETSI-NFV-MANO].	The distinction between software-based SDN forwarding elements and NSFs, which can both run as virtual network functions (VNFs), may be necessary for some management purposes in this system. Note that an SDN forwarding element (i.e., switch) is a specific type of VNF rather than an NSF because an NSF is for security services rather than for packet forwarding. For this distinction, we can take advantage of the NFV MANO where there is a subsystem that maintains the descriptions of the capabilities each VNF can offer [ETSI-NFV-MANO].

(21) Section 6. "By taking advantage of this capability of SDN, it is possible to optimize the process of security service enforcement in the I2NSF system." The proposed optimization isn't evident from this text.

=> For efficient firewall services, simple packet filtering can be performed by SDN forwarding elements, and complicated packet filtering based on packet payloads can be performed by a firewall NSF. This optimized firewall using SDN forwarding elements and a firewall NSF is more efficient than a firewall where SDN forwarding elements forward all the packets to a firewall NSF for packet filtering. This is because the packets to be filtered out can be dropped by SDN forwarding elements without consuming further network bandwidth due to the forwarding of the packets to the firewall NSF. I clarify this optimization in the text as follows:

In Section 6.

OLD	NEW
By taking advantage of this capability of SDN, it is possible to optimize the process of security service enforcement in the I2NSF system.	By taking advantage of this capability of SDN, it is possible to optimize the process of security service enforcement in the I2NSF system. For example, for efficient firewall services, simple packet filtering can be performed by SDN forwarding elements (e.g., switches), and complicated packet filtering based on packet payloads can be performed by a firewall NSF. This optimized firewall using both SDN

	forwarding elements and a firewall NSF is more efficient than a firewall where SDN forwarding elements forward all the packets to a firewall NSF for packet filtering. This is because packets to be filtered out can be early dropped by SDN forwarding elements without consuming further network bandwidth due to the forwarding of the packets to the firewall NSF.

(22) Section 6. "Especially, SDN forwarding elements enforce simple packet filtering rules that can be translated into their packet forwarding rules, whereas NSFs enforce NSF-related security rules requiring the security capabilities of the NSFs."

** I found the use of the word "Especially" confusing

=> I delete the word “Especially” as follows:

In Section 6.

OLD	NEW
Especially, SDN forwarding elements enforce simple packet filtering rules that can be translated into their packet forwarding rules, whereas NSFs enforce NSF-related security rules requiring the security capabilities of the NSFs.	SDN forwarding elements enforce simple packet filtering rules that can be translated into their packet forwarding rules, whereas NSFs enforce NSF-related security rules requiring the security capabilities of the NSFs.

** I am not sure what distinction is being made between the SDN forwarding and NSF rules.

=> SDN forwarding rules are for SDN flow table entries and NSF rules are for firewall. Simple firewall rules can be enforced by SDN forwarding rules at SDN forwarding elements (i.e., SDN switches). I clarify this distinction in the text as follows:

In Section 6.

OLD	NEW
SDN forwarding elements enforce simple packet filtering rules that can be translated into their packet forwarding rules, whereas NSFs enforce NSF-related security rules requiring the security capabilities of the NSFs. For this purpose, the Security Controller instructs the SDN Controller via NSF-Facing Interface so that SDN forwarding elements can perform the required security services with flow tables under the supervision of the SDN Controller.	SDN forwarding elements enforce simple packet filtering rules that can be translated into their packet forwarding rules, whereas NSFs enforce complicated NSF-related security rules requiring the security capabilities of the NSFs. Note that SDN packet forwarding rules are for packet forwarding or filtering by flow table entries at SDN forwarding elements, and NSF rules are for security enforcement at NSFs (e.g., firewall). Thus, simple firewall rules can be enforced by SDN packet forwarding rules at SDN forwarding elements (e.g., switches). For the tasks for security enforcement (e.g., packet filtering), the Security Controller instructs the SDN Controller via NSF-Facing Interface so that SDN forwarding elements can perform the required security services with flow tables under the supervision of the SDN Controller.

(23) Section 6, "For this purpose, the Security Controller instructs the SDN Controller via NSF-Facing Interface so that SDN forwarding elements can perform the required security services with flow tables under the supervision of the SDN Controller."

** I wasn't sure what the "for this purpose" was referencing, what "purpose"?

=> I clarify “this purpose” with "the tasks for security enforcement (e.g., packet filtering)" as follows:

In Section 6.

OLD	NEW
For this purpose, the Security Controller instructs the SDN Controller via NSF-Facing Interface so that SDN forwarding elements can perform the required security services with flow tables under the supervision of the SDN Controller.	For the tasks for security enforcement (e.g., packet filtering), the Security Controller instructs the SDN Controller via NSF-Facing Interface so that SDN forwarding elements can perform the required security services with flow tables under the supervision of the SDN Controller.

(24) Section 6. Editorial Nit.

OLD:
"The following subsections introduce three use cases for cloud-based security services: (i) firewall system, (ii) deep packet inspection system, and (iii) attack mitigation system. [RFC8192]"

NEW:
The following subsections introduce three use cases from [RFC8192] for cloud-based security services: (i) firewall system, (ii) deep packet inspection system, and (iii) attack mitigation system."
=> This replacement is performed as follows:

In Section 6.

OLD	NEW
The following subsections introduce three use cases for cloud-based security services: (i) firewall system, (ii) deep packet inspection system, and (iii) attack mitigation system. [RFC8192]	The following subsections introduce three use cases from [RFC8192] for cloud-based security services: (i) firewall system, (ii) deep packet inspection system, and (iii) attack mitigation system.

(25) Section 6.1 - 6.3. It wasn't evident to me why these sections were in the document. The described procedures and benefits didn't read as being I2NSF specific and appear to primarily describe what's happening in the SDN (and not using the defined I2NSF interfaces).
=> The procedures and benefits of the three use cases are removed from the text as follows. The description of Firewall is enhanced with a time-based firewall in Section 6.1.

In Section 6.1 – 6.3.

NEW
6.1. Firewall: Centralized Firewall System A centralized network firewall can manage each network resource and apply common rules to individual network elements (e.g., switch). The centralized network firewall controls each forwarding element, and firewall rules can be added or deleted dynamically. A time-based firewall can be enforced with packet filtering rules and a time span (e.g., work hours).

With this time-based firewall, a time-based security policy can be enforced, as explained in Section 4. For example, employees at a company are allowed to access social networking service websites during lunch time or after work hours.

6.2. Deep Packet Inspection: Centralized VoIP/VoLTE Security System

A centralized VoIP/VoLTE security system can monitor each VoIP/VoLTE flow and manage VoIP/VoLTE security rules, according to the configuration of a VoIP/VoLTE security service called VoIP Intrusion Prevention System (IPS). This centralized VoIP/VoLTE security system controls each switch for the VoIP/VoLTE call flow management by manipulating the rules that can be added, deleted or modified dynamically.

The centralized VoIP/VoLTE security system can cooperate with a network firewall to realize VoIP/VoLTE security service. Specifically, a network firewall performs the basic security check of an unknown flow's packet observed by a switch. If the network firewall detects that the packet is an unknown VoIP call flow's packet that exhibits some suspicious patterns, then it triggers the VoIP/VoLTE security system for more specialized security analysis of the suspicious VoIP call packet.

6.3. Attack Mitigation: Centralized DDoS-attack Mitigation System

A centralized DDoS-attack mitigation can manage each network resource and configure rules to each switch for DDoS-attack mitigation (called DDoS-attack Mitigator) on a common server. The centralized DDoS-attack mitigation system defends servers against DDoS attacks outside the private network, that is, from public networks. Servers are categorized into stateless servers (e.g., DNS servers) and stateful servers (e.g., web servers). For DDoS-attack mitigation, the forwarding of traffic flows in switches can be dynamically configured such that malicious traffic flows are handled by the paths separated from normal traffic flows in order to minimize the impact of those malicious traffic on the the servers. This flow path separation can be done by a flow forwarding path management scheme based on [AVANT-GUARD]. This management should consider the load balance among the switches for the defense against DDoS attacks.

So far this section has described the three use cases for network-based security services using the I2NSF framework with SDN networks.

(26) Section 6.3, Typo. s/the the/the/
=> The typo is corrected as follows:

In Section 6.3.

OLD	NEW
For DDoS-attack mitigation, the forwarding of traffic flows in switches can be dynamically configured such that malicious traffic flows are handled by the paths separated from normal traffic flows in order to minimize the impact of those malicious traffic on the the servers.	For DDoS-attack mitigation, the forwarding of traffic flows in switches can be dynamically configured such that malicious traffic flows are handled by the paths separated from normal traffic flows in order to minimize the impact of those malicious traffic on the servers.

(27) Section 7. "Those NSFs are created or removed by a virtual network functions manager (VNFM) in the NFV architecture that performs the life-cycle management of VNFs. The Security Controller controls and monitors the configurations (e.g., function parameters and security policy rules) of VNFs."

Is the VNFM in scope for I2NSF?
=> No, the VNFM is out of scope for I2NSF. We specify that the life-cycle management is out of scope

for I2NSF.

In Section 7.

OLD	NEW
Those NSFs are created or removed by a virtual network functions manager (VNFM) in the NFV architecture that performs the life-cycle management of VNFs.	Those NSFs cab be created or removed by a virtual network functions manager (VNFM) in the NFV architecture that performs the life-cycle management of VNFs. Note that the life-cycle management of VNFs are out of scope for I2NSF.

If the Security Controller monitors/controls the VNFs, is it using [nsf-monitoring-dm] and [nsf-facing-inf-dm]?
=> Yes, it is. The controlling and monitoring of the VNFs are performed by the Security Controller via NSF-Facing Interface along with NSF monitoring capability.

In Section 7.

OLD	NEW
Those NSFs are created or removed by a virtual network functions manager (VNFM) in the NFV architecture that performs the life-cycle management of VNFs.	The Security Controller controls and monitors the configurations (e.g., function parameters and security policy rules) of VNFs via NSF-Facing Interface along with NSF monitoring capability [nsf-facing-inf-dm][nsf-monitoring-dm].

Roman