Revision Letter

Editor: Jaehoon Paul Jeong
Date: July 13, 2020

OLD: draft-ietf-i2nsf-capability-data-model-05
NEW: draft-ietf-i2nsf-capability-data-model-06

Dear Roman Danyliw,

I sincerely appreciate your detailed comments to improve our I2NSF Capability YANG Data Model.
I have addressed all your comments one by one. I use a bold font for your comments, and uses a regular font for my responses.

Hi!

I conducted an AD review of draft-ietf-i2nsf-capability-data-model-05. Thanks for the work in getting this document written. My most significant items are around aligning of the text in Section 4 with RFC8329 and the dependency on draft-dong-i2nsf-asf-config-01.

My detailed feedback is below.

(1) IDNits returned the following valid comment about references (many of the issue is noted were in the YANG module)
== Missing Reference: 'RFC3688' is mentioned on line 1764, but not defined
=> There is no reference to RFC3688 (The IETF XML Registry). Could you doublecheck your comment to let me follow it?

(2) Section 1. Typo. s/[draft-ietf-i2nsf-capability]../ [draft-ietf-i2nsf-capability]./
=> I removed an extra period after [draft-ietf-i2nsf-capability].

(3) Section 3. Is there a reason to rely on two expired drafts for terminology -- [draft-ietf-i2nsf-terminology] and [draft-ietf-supa-generic-policy-info-model]? In particular, couldn't RFC3444 provide the needed definitions of data and information models?
=> I removed these two expired drafts for terminology and added RFC3444 as a reference for data and information models.

(4) Section 4. I would have expected somewhere in this overview section an explicit enumeration of which I2NSF interfaces use this YANG module.
=> Registration Interface Data Model uses this YANG module, so I mention this explicitly as follows.

Section 4. Overview (Page 4): The 1st Paragraph

OLD	NEW
This section provides as overview of how the YANG data model can be used in the I2NSF framework described in [RFC8329]. Figure 1 shows the capabilities of NSFs in I2NSF Framework. As shown in this figure, an NSF Developer's Mgmt System can register NSFs and	This section provides as overview of how the YANG data model can be used in the I2NSF framework described in [RFC8329]. Figure 1 shows the capabilities (e.g., firewall and web filter) of NSFs in I2NSF Framework. As shown in this figure, an NSF Developer's Management System

the capabilities that the network security device can support. To register NSFs in this way, the Developer's Mgmt System utilizes this standardized capabilities YANG data model through its registration interface. With the capabilities of those network security devices maintained centrally, those security devices can be easily managed, which can resolve many of the problems described in [RFC8192]. The use cases are described below.	can register NSFs and the capabilities that the network security device can support. To register NSFs in this way, the Developer's Management System utilizes this standardized capability YANG data model through the I2NSF Registration Interface [draft-ietf-i2nsf-registration-interface-dm]. That is, this Registration Interface uses the YANG module described in this document to describe the capability of a network security function that is registered with the Security Controller. With the capabilities of those network security devices maintained centrally, those security devices can be easily managed, which can resolve many of the problems described in [RFC8192]. The use cases are described below.

(5) Section 4. Per "Figure 1 shows the capabilities of NSFs in I2NSF Framework."
-- Thanks for reusing the diagram from RFC8329 and annotating it with more detail. It helps connect the documents
-- It wasn't clear to me where the "capabilities" are on the diagram
=> I denote "Firewall" as a capability of a network security function in Figure 1 as follows.

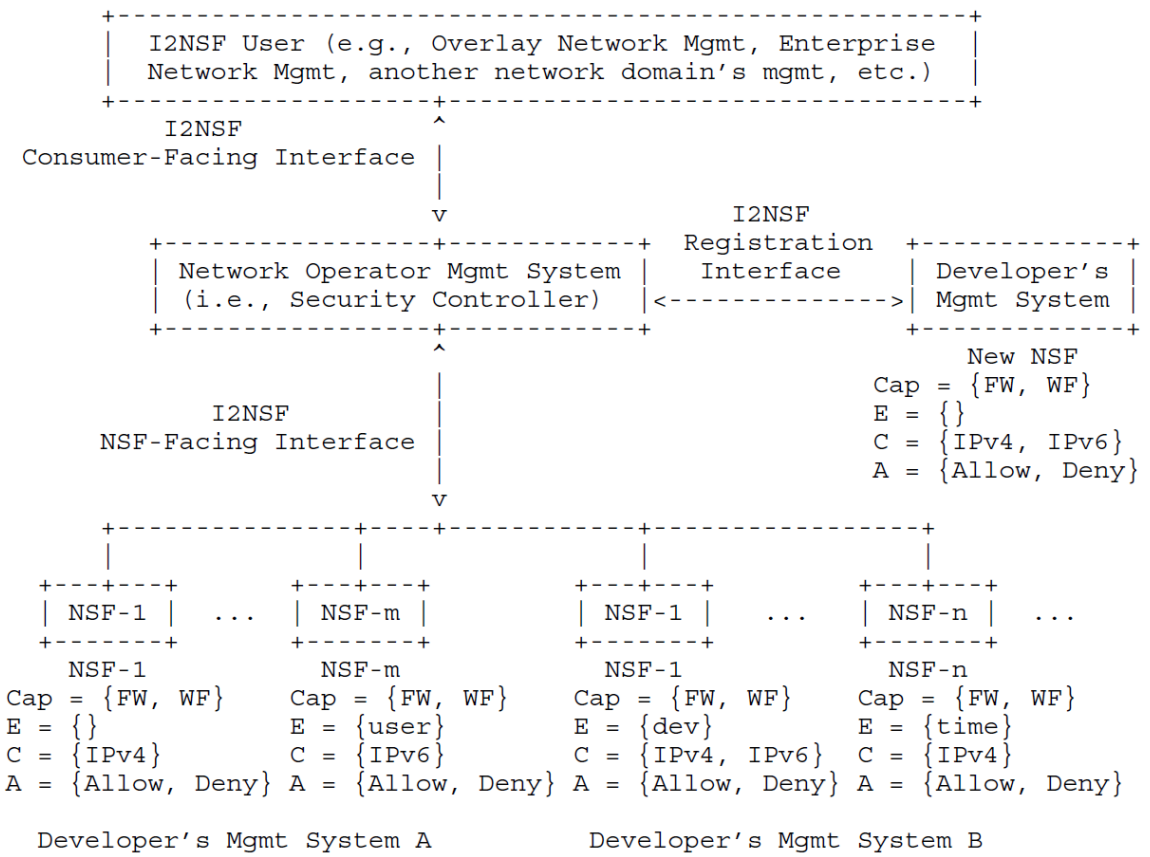

Figure 1: Capabilities of NSFs in I2NSF Framework

-- Is all of the detail under the NSFs (i.e., E, C and A) needed in the diagram? Text doesn't explain it or reference it. If kept, it should be explained and E, C and A should be defined (i.e., saying these correspond to Event, Condition and Action)
=> I defined E, C, and A as follows.

Section 4. Overview (Page 4): The 2nd Paragraph

NEW
In Figure 1, a new NSF at a Developer's Management Systems has capabilities of Firewall (FW) and Web Filter (WF), which are denoted as (Cap = {FW, WF}), to support Event-Condition-Action (ECA) policy rules where 'E', 'C', and 'A' mean "Event", "Condition", and "Action", respectively. The condition involves IPv4 or IPv6 datagrams, and the action includes "Allow" and "Deny" for those datagrams.

(6) Global. Editorial. Is there a reason to abbreviate "Mgmt" in "Developer's Mgmt System in the text? Recommend s/Developer's Mgmt System/Developer's Management System/g
=> I replaced Developer's Mgmt System with Developer's Management System in the text. But, in Figure 1, I use "Mgmt" for the space saving.

(7) Section 4. Per "To register NSFs in this way, the Developer's Mgmt System utilizes this standardized capabilities YANG data model through its registration interface.", this confused me a bit. Doesn't the Developer Management System use the model described in draft-ietf-i2nsf-registration-interface-dm for registration?
=> Yes, you are right. I clarify this sentence for the registration of NSFs as follows.

Section 4. Overview (Page 4): The 1st Paragraph

OLD	NEW
This section provides as overview of how the YANG data model can be used in the I2NSF framework described in [RFC8329]. Figure 1 shows the capabilities of NSFs in I2NSF Framework. As shown in this figure, an NSF Developer's Mgmt System can register NSFs and the capabilities that the network security device can support. To register NSFs in this way, the Developer's Mgmt System utilizes this standardized capabilities YANG data model through its registration interface. With the capabilities of those network security devices maintained centrally, those security devices can be easily managed, which can resolve many of the problems described in [RFC8192]. The use cases are described below.	This section provides as overview of how the YANG data model can be used in the I2NSF framework described in [RFC8329]. Figure 1 shows the capabilities (e.g., firewall and web filter) of NSFs in I2NSF Framework. As shown in this figure, an NSF Developer's Management System can register NSFs and the capabilities that the network security device can support. To register NSFs in this way, the Developer's Management System utilizes this standardized capability YANG data model through the I2NSF Registration Interface [draft-ietf-i2nsf-registration-interface-dm]. That is, this Registration Interface uses the YANG module described in this document to describe the capability of a network security function that is registered with the Security Controller. With the capabilities of those network

	security devices maintained centrally, those security devices can be easily managed, which can resolve many of the problems described in [RFC8192]. The use cases are described below.

(8) Section 4. Editorial. Per "... those security devices can be easily managed, ...", I might have used "more easily managed".
=> I reflected your comment.

(9) Section 4. Per "The use cases are described below.", where are those use cases described? Is this text a reference to the "Configuration Examples" in Appendix A?
=> A use case is explained as bulleted paragraphs. I clarified the use case more clearly as follows.

Section 4. Overview (Page 5): The 4th Paragraph

OLD	NEW
o If a network manager wants to apply security policy rules to block malicious users, it is a tremendous burden to apply all of the needed rules to NSFs one-by-one. This problem can be resolved by managing the capabilities of NSFs. If network manager wants to block malicious users with IPv6, the network manager sends the security policy rules to block the users to the Network Operator Mgmt System using I2NSF user (i.e., a web browser or a software). When the Network Operator Mgmt System receives the security policy rules, it automatically sends that security policy rules to appropriate NSFs (i.e., NSF-m in Developer's Mgmt System A and NSF-1 in Developer's Mgmt System B) which can support the capabilities (i.e., IPv6). Therefore, an I2NSF User need not consider NSFs where to which NSFs the rules apply. o If NSFs encounter the malicious packets, it is a tremendous burden for the network manager to apply the rule to block the malicious packets to NSFs one-by-one. This problem can be resolved by managing the capabilities of NSFs. If NSFs encounter the suspicious IPv4 packets, they can	A use case of an NSF with the capabilities of firewall and web filter is described as follows. o If a network manager wants to apply security policy rules to block malicious users with firewall and web filter, it is a tremendous burden for a network administrator to apply all of the needed rules to NSFs one by one. This problem can be resolved by managing the capabilities of NSFs in this document. o If a network administrator wants to block malicious users for IPv6 traffic, he sends a security policy rule to block the users to the Network Operator Management System using the I2NSF User (i.e., web application). o When the Network Operator Management System receives the security policy rule, it automatically sends that security policy rules to appropriate NSFs (i.e., NSF-m in Developer's Management System A and NSF-1 in Developer's Management System B) which can support the capabilities (i.e., IPv6). This lets an I2NSF User not consider NSFs where the rule is applied.

ask the Network Operator Mgmt System for information about the suspicious IPv4 packets in order to alter specific rules and/or configurations. When the Network Operator Mgmt System receives information, it inspects the information about the suspiciou IPv4 packets. If the suspicious packets are determined to be malicious packets, the Network Operator Mgmt System creates and sends the security policy rules blocking malicious packets to appropriate NSFs (i.e., NSF-1 in Developer's Mgmt System A and NSF-1 and NSF-n in Developer's Mgmt System B) which can support the capabilities (i.e., IPv4). Therefore, the new security policy rules blocking malicious packets can be applied to appropriate NSFs without humans intervention.	o If NSFs encounter the suspicious IPv6 packets of malicious users, they can filter the packets out according to the configured security policy rule. Therefore, the security policy rule against the malicious users' packets can be automatically applied to appropriate NSFs without human intervention.

(10) Section 4. Per "Note that the NSF-Facing Interface ... and the NSF Monitoring Interface is used to ...", does this text need additional precision based on the definitions in RFC8329. Per RFC8329, the "NSF-Facing Interfaces" consists of the "NSF Operational and Administrative Interface" and a "Monitoring Interface". If draft-dong-i2nsf-asf-config is on the "Monitoring Interface", on which "sub-interface" of the "NSF-Facing Interface" does draft-ietf-i2nsf-nsf-facing-interface-dm belong?
=> draft-dong-i2nsf-asf-config uses the NSF-Facing Interface for advanced NSFs (e.g., anti-virus and anti-DDoS attack). I clarify the sentence as follows.

Section 4. Overview (Page 4): The 3rd Paragraph

OLD	NEW
Note that the NSF-Facing Interface is used to configure the security policy rules of the generic network security functions [draft-ietf-i2nsf-nsf-facing-interface-dm], and the NSF Monitoring Interface is used to configure the security policy rules of advanced network security functions [draft-dong-i2nsf-asf-config], respectively, according to the capabilities of NSFs registered with the I2NSF Framework.	Note that the NSF-Facing Interface is used to configure the security policy rules of the generic network security functions [draft-ietf-i2nsf-nsf-facing-interface-dm], and the configuration of advanced security functions over the NSF-Facing Interface is used to configure the security policy rules of advanced network security functions (e.g., anti-virus and anti-DDoS attack) [draft-dong-i2nsf-asf-config], respectively, according to the capabilities of NSFs registered with the I2NSF Framework.

(11) Figure 1. Editorial Nit. Is there are reason that the Registration interface has a bidirectional arrow between the network operator management system and the developer management system, but the there is no directionality on the consumer or NSF facing interface?
=> According to the Registration Interface data model draft [draft-ietf-i2nsf-registration-interface-dm], both the registration and query of an NSF are performed, so a bidirectional arrow is used. I put bidirectional arrows for both Consumer-Facing and NSF-Facing Interfaces in Figure 1.

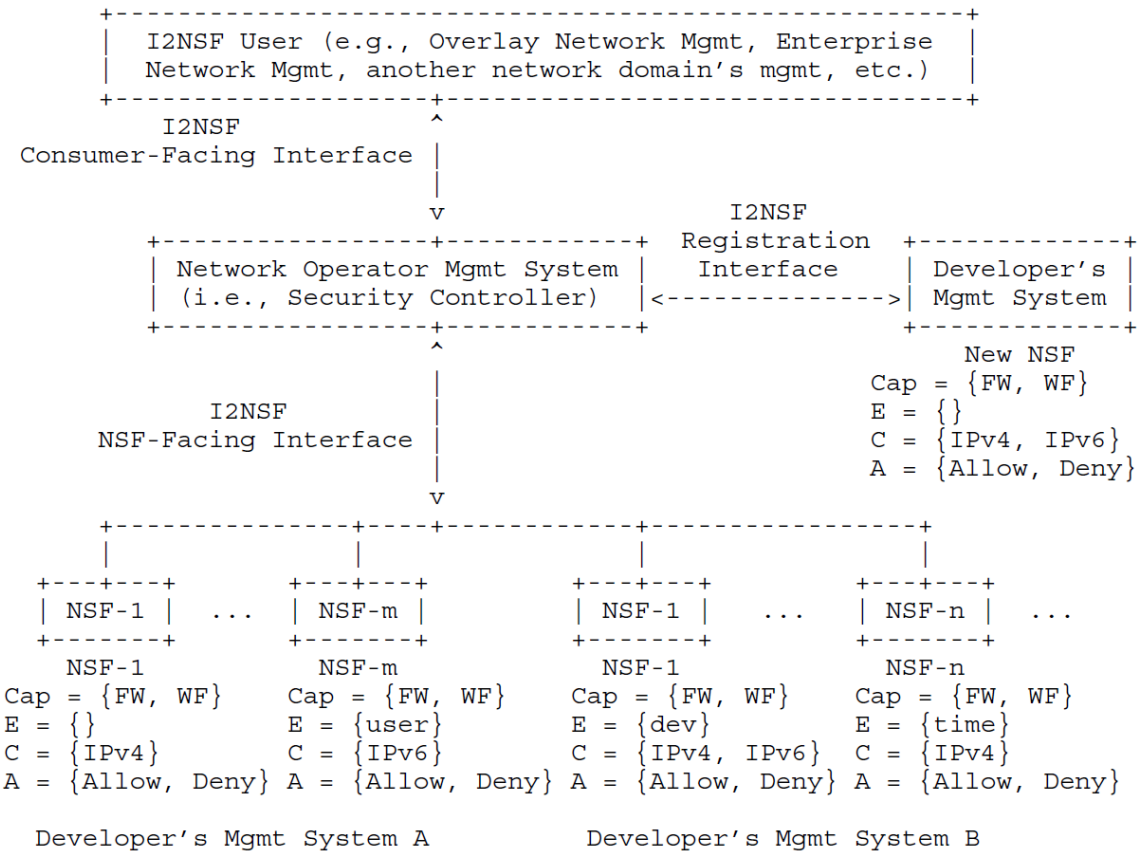

Figure 1: Capabilities of NSFs in I2NSF Framework

(12) Section 4. The bulleted list under Figure 1 is helpful in describing Figure 1. However, I'd recommend explicitly saying this is an example. Explain the use case up front and then narrate the flow clearly delineating what is in and out of scope of I2NSF. IMO, the text describes a number of internal processing functions which are in scope for standardization - please let me know if I'm reading it wrong.
=> I explicitly say a use case and clarify the explanation of the use case as follows.

Section 4. Overview (Page 5): The 4th Paragraph

OLD	NEW
o If a network manager wants to apply security policy rules to block malicious users, it is a tremendous burden to apply all of the needed	A use case of an NSF with the capabilities of firewall and web filter is described as follows.

rules to NSFs one-by-one. This problem can be resolved by managing the capabilities of NSFs. If network manager wants to block malicious users with IPv6, the network manager sends the security policy rules to block the users to the Network Operator Mgmt System using I2NSF user (i.e., a web browser or a software). When the Network Operator Mgmt System receives the security policy rules, it automatically sends that security policy rules to appropriate NSFs (i.e., NSF-m in Developer's Mgmt System A and NSF-1 in Developer's Mgmt System B) which can support the capabilities (i.e., IPv6). Therefore, an I2NSF User need not consider NSFs where to which NSFs the rules apply. o If NSFs encounter the malicious packets, it is a tremendous burden for the network manager to apply the rule to block the malicious packets to NSFs one-by-one. This problem can be resolved by managing the capabilities of NSFs. If NSFs encounter the suspicious IPv4 packets, they can ask the Network Operator Mgmt System for information about the suspicious IPv4 packets in order to alter specific rules and/or configurations. When the Network Operator Mgmt System receives information, it inspects the information about the suspiciou IPv4 packets. If the suspicious packets are determined to be malicious packets, the Network Operator Mgmt System creates and sends the security policy rules blocking malicious packets to appropriate NSFs (i.e., NSF-1 in Developer's Mgmt System A and NSF-1 and NSF-n in Developer's Mgmt System B) which can support the capabilities (i.e., IPv4). Therefore, the new security policy rules blocking malicious packets can be applied to appropriate NSFs without humans intervention.	o If a network manager wants to apply security policy rules to block malicious users with firewall and web filter, it is a tremendous burden for a network administrator to apply all of the needed rules to NSFs one by one. This problem can be resolved by managing the capabilities of NSFs in this document. o If a network administrator wants to block malicious users for IPv6 traffic, he sends a security policy rule to block the users to the Network Operator Management System using the I2NSF User (i.e., web application). o When the Network Operator Management System receives the security policy rule, it automatically sends that security policy rules to appropriate NSFs (i.e., NSF-m in Developer's Management System A and NSF-1 in Developer's Management System B) which can support the capabilities (i.e., IPv6). This lets an I2NSF User not consider NSFs where the rule is applied. o If NSFs encounter the suspicious IPv6 packets of malicious users, they can filter the packets out according to the configured security policy rule. Therefore, the security policy rule against the malicious users' packets can be automatically applied to appropriate NSFs without human intervention.

(13) Section 4. Per "If network manager wants to block malicious users with IPv6, the network manager sends the security policy rules to block the users to the Network Operator Mgmt System using I2NSF user ....", can you please clarify "malicious users with IPv6"; is the intent that the network manager is concerned about malicious IPv6 traffic?
=> Yes, your suggestion is accurate. I clarified the sentence in the above answer.

(14) Section 4. Bullet 1 under Figure 1. Per "a web browser or a software", what's the difference between a browser and software?
=> I replaced them with “web application”.

(15) Section 4. Editorial. Per the second bullet under Figure 1, "If NSFs encounter the malicious packets, it is a tremendous burden for the network manager to apply the rule to block the malicious packets to NSFs one-by-one. This problem can be resolved by managing the capabilities of NSFs.", delete this text. It is a duplicate of what was stated in the first bullet.
=> I revised the bulleted paragraphs of the use case in the answer for (12).

(16) Section 4. Per the paragraph, "If NSFs encounter the suspicious IPv4 packets, they can ask the Network Operator Mgmt System for information about the suspicious IPv4 packets in order to alter specific rules and/or configurations. When the Network ...", how much of that signaling is in scope for I2NSF?
=> The text about the signaling is deleted because it is out of scope for I2NSF.

(17) Section 4. Typo. s/suspiciou/suspicious/
=> The typo is fixed.

(18) Section 5.1. Editorial. s/The model includes NSF capabilities/The model describes NSF capabilities/
=> The replacement is done.

(19) Section 5.1. Editorial. "specify" is used twice in the sentence.
OLD
Time capabilities are used to specify the capabilities to specify when to execute the I2NSF policy rule.

NEW
Time capabilities are used to specify the capabilities which describe when to execute the I2NSF policy rule.
=> The replacement is done.

(20) Section 5.1. Editorial. This sentence didn't parse for me. The second contains duplicate text.
OLD
Event capabilities are used to specify capabilities how to trigger the evaluation of the condition clause of the I2NSF Policy Rule. The defined event capabilities are defined as system event and system alarm.

NEW
Event capabilities are used to specify the capabilities that describe the event that would trigger the evaluation of the condition clause of the I2NSF Policy Rule. The defined event capabilities are system event and system alarm.
=> The replacement is done.

(21) Section 5.1. A number of capabilities note that they can be extended which is a helpful feature. For example, "The condition capability can be extended according to specific vendor condition features." However, where is the guidance on doing that? Likewise, it might not be necessary to repeat this statement five times if the extension mechanism is the same.
=> According to the Capabilities Information Model draft, such a conditional capability can be tailored extended according to a vendor’s specific condition features. I removed the repeated statements.

Section 5.1. Network Security Function (NSF) Capabilities (Page 6): The 1st Paragraph

OLD	NEW
This section shows YANG tree diagram for NSF capabilities. This YANG tree diagram shows NSF capabilities. The model includes NSF capabilities. The NSF capabilities include time capabilities, event capabilities, condition capabilities, action capabilities, resolution strategy capabilities, and default action capabilities.	This section explains a YANG tree diagram of NSF capabilities and its features. Figure 2 shows a YANG tree diagram of NSF capabilities. The NSF capabilities in the tree include time capabilities, event capabilities, condition capabilities, action capabilities, resolution strategy capabilities, and default action capabilities. Those capabilities can be tailored or extended according to a vendor's specific requirements. Refer to the NSF capabilities information model for detailed discussion [draft-ietf-i2nsf-capability].

(22) Section 5.1. A number of the described capability types state that they are described in detail in draft-ietf-i2nsf-capability. For example, "The condition capability is described in detail in [draft-ietf-i2nsf-capability]." I had difficulty locating which specific section to review. Also, for the default action capabilities, no described of "pass, drop .. mirror" was found in draft-ietf-i2nsf-capability. Please provide a specific section number for the event, condition, action, resolution strategy and default action in draft-ietf-i2nsf-capability.
=> I provide specific section numbers for those capabilities as follows.

Section 5.1. Network Security Function (NSF) Capabilities (Page 7): The 3rd Paragraph

NEW
Time capabilities are used to specify the capabilities which describe when to execute the I2NSF policy rule. The time capabilities are defined in terms of absolute time and periodic time. The absolute time means the exact time to start or end. The periodic time means repeated time like day, week, or month. See Section 3.4.6 (Capability Algebra) in [draft-ietf-i2nsf-capability] for more information about the time-based condition (e.g., time period) in the capability algebra. Event capabilities are used to specify the capabilities that describe the event that would trigger the evaluation of the condition clause of the I2NSF Policy Rule. The defined event capabilities are system event and system alarm. See Section 3.1 (Design Principles and ECA Policy Model Overview) in [draft-ietf-i2nsf-capability] for more information about the event in the ECA policy model. Condition capabilities are used to specify capabilities of a set of attributes, features, and/or values that are to be compared with a set of known attributes, features, and/or values in order to determine whether or not the set of actions in that (imperative) I2NSF policy rule can be executed. The condition capabilities are classified in terms of generic network security functions and advanced network

security functions. The condition capabilities of generic network security functions are defined as IPv4 capability, IPv6 capability, TCP capability, UDP capability, and ICMP capability. The condition capabilities of advanced network security functions are defined as anti-virus capability, anti-DDoS capability, IPS capability, HTTP capability, and VoIP/VoLTE capability. See Section 3.1 (Design Principles and ECA Policy Model Overview) in [draft-ietf-i2nsf-capability] for more information about the condition in the ECA policy model. Also, see Section 3.4.3 (I2NSF Condition Clause Operator Types) in [draft-ietf-i2nsf-capability] for more information about the operator types in an I2NSF condition clause.

Action capabilities are used to specify the capabilities that describe the control and monitoring aspects of flow-based NSFs when the event and condition clauses are satisfied. The action capabilities are defined as ingress-action capability, egress-action capability, and log- action capability. See Section 3.1 (Design Principles and ECA Policy Model Overview) in [draft-ietf-i2nsf-capability] for more information about the action in the ECA policy model. Also, see Section 7.2 (NSF-Facing Flow Security Policy Structure) in [RFC8329] for more information about the ingress and egress actions. In addition, see Section 9.1 (Flow-Based NSF Capability Characterization) for more information about logging at NSFs.

Resolution strategy capabilities are used to specify the capabilities that describe conflicts that occur between the actions of the same or different policy rules that are matched and contained in this particular NSF. The resolution strategy capabilities are defined as First Matching Rule (FMR), Last Matching Rule (LMR), Prioritized Matching Rule (PMR), Prioritized Matching Rule with Errors (PMRE), and Prioritized Matching Rule with No Errors (PMRN). See Section 3.4.2 (Conflict, Resolution Strategy and Default Action) in [draft-ietf-i2nsf-capability] for more information about the resolution strategy.

Default action capabilities are used to specify capabilities of how to execute I2NSF policy rules when no rule matches a packet. The default action capabilities are defined as pass, drop, alert, and mirror. See Section 3.4.2 (Conflict, Resolution Strategy and Default Action) in [draft-ietf-i2nsf-capability] for more information about the default action.

IPsec method capabilities are used to specify capabilities of how to support an Internet Key Exchange (IKE) for the security communication. The default action capabilities are defined as IKE or IKE-less. See [draft-ietf-i2nsf-sdn-ipsec-flow-protection] for more information about the SDN-based IPsec flow protection in I2NSF.

(23) Section 5.1. Editorial. These sentences didn't parse for me.
OLD
Action capabilities are used to specify capabilities of how to control and monitor aspects of flow-based NSFs when the event and condition clauses are satisfied.

NEW

Action capabilities are used to specify the capabilities that describe the control and monitoring aspects of flow-based NSFs when the event and condition clauses are satisfied.
=> The replacement is done.

OLD
Resolution strategy capabilities are used to specify capabilities of how to resolve conflicts that occur between the actions of the same or different policy rules that are matched and contained in this particular NSF.

NEW
Resolution strategy capabilities are used to specify the capabilities that describe conflicts that occur between the actions of the same or different policy rules that are matched and contained in this particular NSF.
=> The replacement is done.

OLD
Default action capabilities are used to specify capabilities of how to execute I2NSF policy rules when no rule matches a packet.

NEW
Default action capabilities are used to specify the capabilities that describe how to execute I2NSF policy rules when no rule matches a packet.
=> The replacement is done.

(24) Section 6.1. Update the copyright date and revision date to be in 2020.
=> The correction is done.

(25) Section 6.1. Given that draft-ietf-i2nf-monitoring-data-model is referenced in the YANG model for event and system alarm, please make it a normative reference.
=> The correction is done.

(26) Section 6.1. identity ingress/egress-action-capability. I found draft-ietf-i2nsf-capability-04 to be an unexpected reference. There is no mention of ingress or egress in that document.
=> RFC8329 (I2NSF Framework) is referenced since it discusses the ingress or egress.

(27) Section 6.1. identity pass, drop, reject, alert, mirror. I found draft-ietf-i2nsf-capability-04 to be an unexpected reference. There is no mention of pass, drop, reject, alert or mirror in that document.
=> RFC8329 (I2NSF Framework) is referenced since it discusses the pass, drop, and mirror. Reject is deleted because it is the same as drop. I2NSF NSF Monitoring Data Model Draft [draft-ietf-i2nsf-nsf-monitoring-data-model-03] is referenced since it discusses an alarm that is the same as alert.

(28) Section 6.1. In the advanced-nsf-capability section, there are multiple normative references to draft-dong-i2nsf-asf-config-01, an expired, individual draft. Additionally, Section 4 notes how it supports the advanced capabilities. This draft is a substantial portion of the YANG module added in -03. What's the plan on resolving this dependency?
=> This draft will be developed by I2NSF WG later.

(29) Section 6.1. Typo. s/Funtion/Function/
=> The type is fixed.

(30) Section 6.1. The list of references in generic-nsf-capabilities don't line up with those in the child leaflist(s). For example, RFC 792 is mentioned in the top level reference list but not in any of the child leaflist (specifically not in leaf-list icmp-capability)
=> The top-level references are lined up with the child leaflist(s).

(31) Section 6.1. Typo. s/smae/same/
=> The type is fixed.

(32) Section 8. A few clarifying updates to the template:
OLD
These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations.
ietf-i2nsf-capability: The attacker may provide incorrect information of the security capability of any target NSF by illegally modifying this.

NEW
These data nodes may be considered sensitive or vulnerable in some network environments. Write operations to these data nodes could have a negative effect on network and security operations.
ietf-i2nsf-capability: An attacker could alter the security capabilities associated with an NSF whereby disabling or enabling the evasion of security mitigations.
=> The replacement is done.

OLD
ietf-i2nsf-capability: The attacker may gather the security capability information of any target NSF and misuse the information for subsequent attacks.

NEW
ietf-i2nsf-capability: An attacker could gather the security capability information of any NSF and use this information to evade detection or filtering.
=> The replacement is done.

Regards,
Roman

Thanks for your help and support.

Best Regards,
Paul
--
==========
Mr. Jaehoon (Paul) Jeong, Ph.D.
Associate Professor
Department of Computer Science and Engineering
Sungkyunkwan University
Office: +82-31-299-4957
Email: jaehoon.paul@gmail.com, pauljeong@skku.edu
Personal Homepage: http://iotlab.skku.edu/people-jaehoon-jeong.php