# Network Security Functions Facing Interface YANG Data Model

## (draft-kim-i2nsf-nsf-facing-interface-data-model-01)

IETF 98, Chicago, US

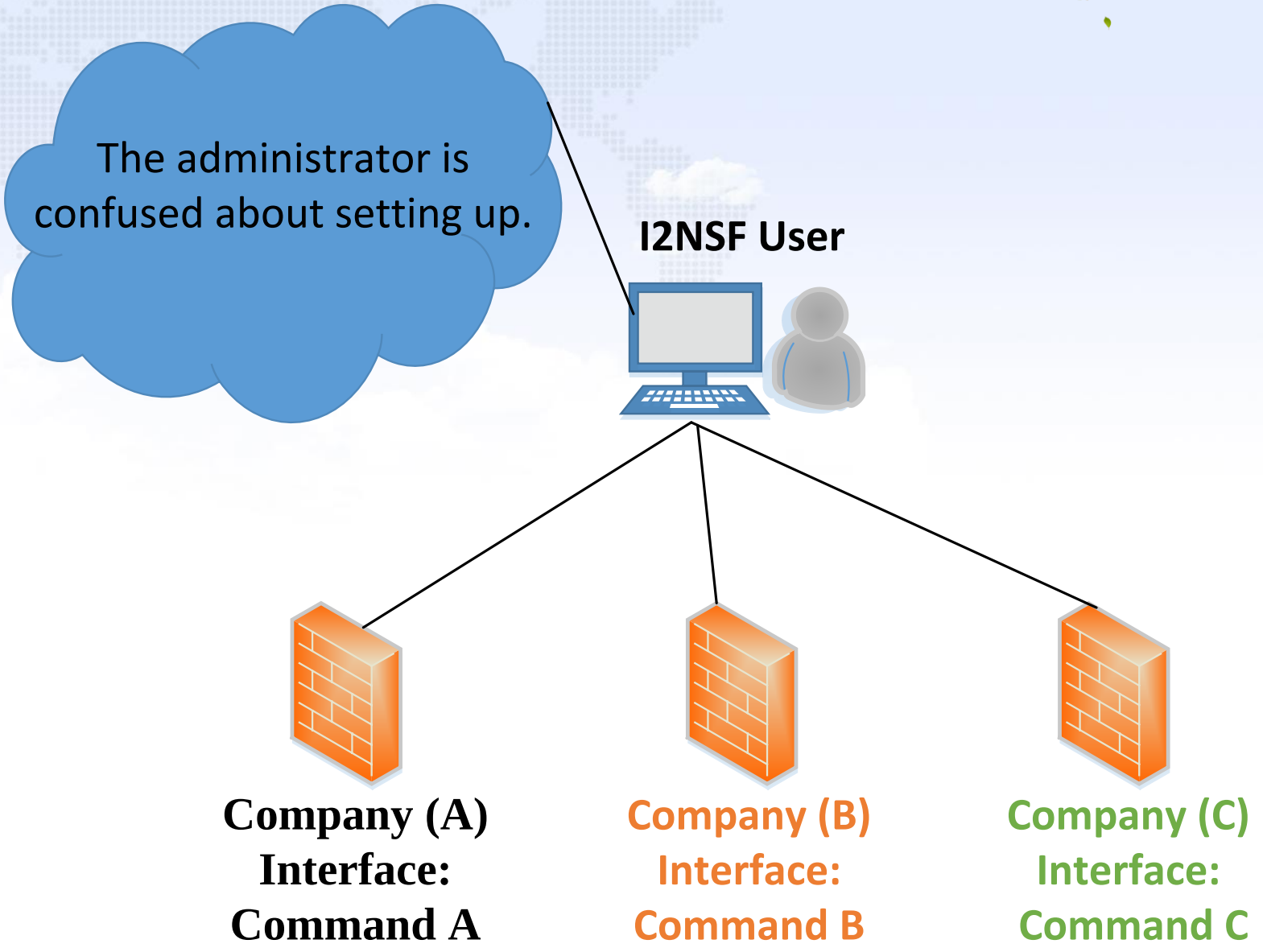
Mar. 27, 2017

Jinyoung Tim Kim, Jaehoon Paul Jeong*,
Jung-Soo Park, Susan Hares, and Liang Xia

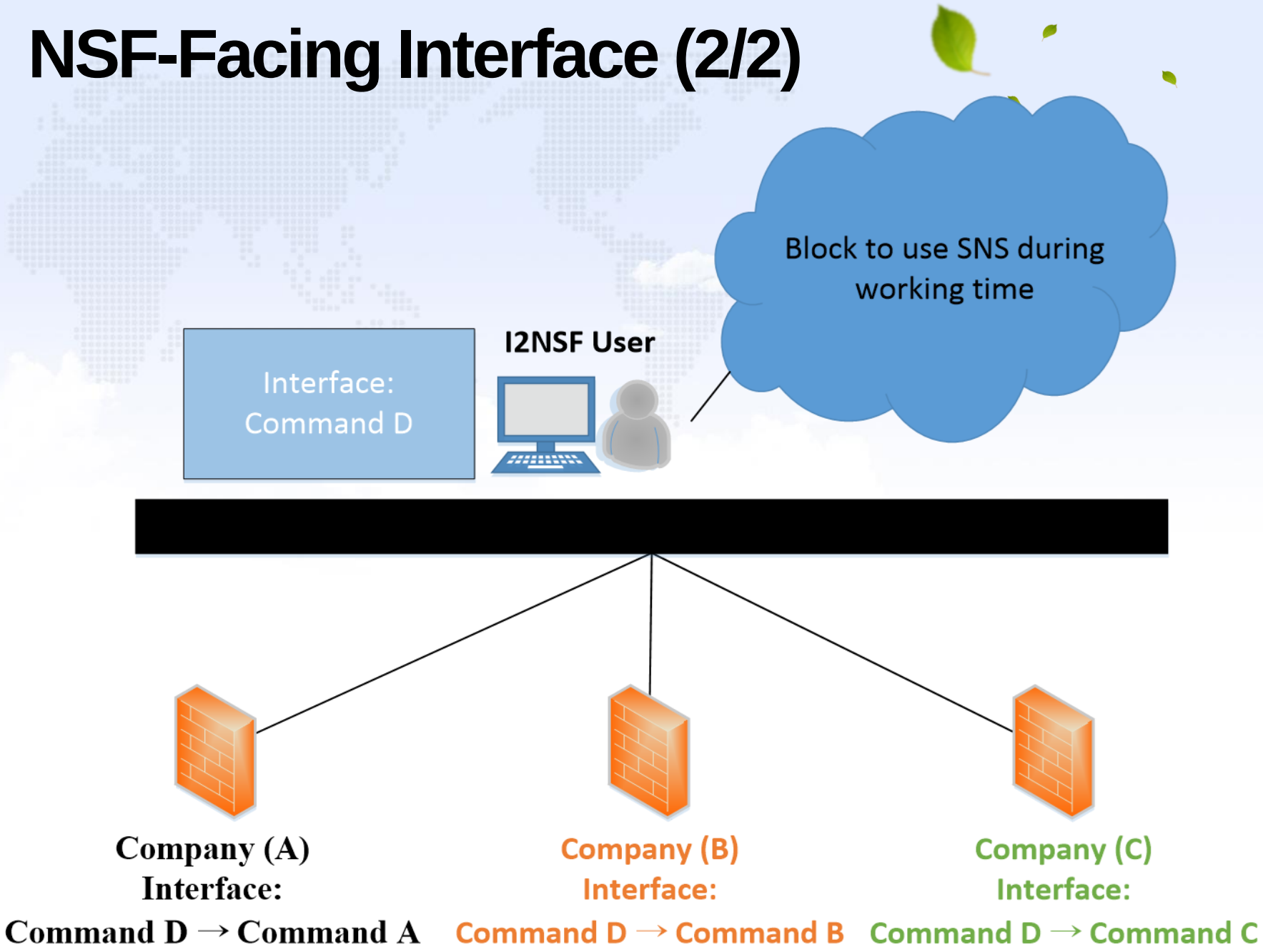
# Contents



# NSF-Facing Interface (1/2)



# NSF-Facing Interface (2/2)



# Introduction

- This draft is an updated version from draft-kim-i2nsf-nsf-facing-interface-data-model-00.

- This draft defines a YANG data model corresponding to the information model for NSF-Facing Interface,
  - Referring to draft-xibassnz-i2nsf-capability-00.

# Update of Version

- Expansion of YANG data model with open source such as Suricata and Iptables.

- Expansion of a YANG data model with header of packets:
  - IPv4, IPv6, ICMP, TCP/UDP, etc.

- Replacement of “List” data type to “Leaf-list” data type.

# Expansion of YANG data model with open source such as Suricata and Iptables

```
module : ietf-i2nsf-nsf-facing-interface
+--rw cfg-network-security-control
|  +--rw policy
|     +--rw policy-name string
|     +--rw policy-id string
|     +--rw rules* [rule-id]
|        +--rw rule-name string
|        +--rw rule-id uint 8
|        +--rw rule-msg string
|        +--rw rule-rev uint 8
|        +--rw rule-gid uint 8
|        +--rw rule-class-type string
|        +--rw rule-reference string
|        +--rw rule-priority uint 8
|        +--rw event
|           +--rw user-security-event* [usr-sec-event-id]
|           |  +--rw usr-sec-event-id uint 8
|           |  +--rw usr-sec-event-content string
|           |  +--rw usr-sec-event-format uint 8
|           |  +--rw usr-sec-event-type uint 8
|           +--rw device-security-event* [dev-sec-event-id]
|           |  +--rw dev-sec-event-id uint 8
|           |  +--rw dev-sec-event-content string
|           |  +--rw dev-sec-event-format uint 8
|           |  +--rw dev-sec-event-type uint 8
|           |  +--rw dev-sec-event-type-severity uint 8
|           +--rw system-security-event* [sys-sec-event-id]
|           |  +--rw sys-sec-event-id uint 8
|           |  +--rw sys-sec-event-content string
|           |  +--rw sys-sec-event-format uint 8
|           |  +--rw sys-sec-event-type uint 8
|           +--rw time-security-event* [time-sec-event-id]
|           |  +--rw time-sec-event-id uint 8
|           |  +--rw time-sec-event-period-begin yang:date-and-time
|           |  +--rw time-sec-event-period-end yang:date-and-time
|           |  +--rw time-sec-evnet-time-zone string
```

[meta-settings]

# Expansion of YANG data model with packet header (1/2)

```
+--rw packet-security-ipv4-condition
|  +--rw pkt-sec-cond-ipv4-header-length* uint 8
|  +--rw pkt-sec-cond-ipv4-tos* uint 8
|  +--rw pkt-sec-cond-ipv4-total-length* uint 16
|  +--rw pkt-sec-cond-ipv4-id* uint 16
|  +--rw pkt-sec-cond-ipv4-fragment* uint 8
|  +--rw pkt-sec-cond-ipv4-fragment-offset* uint 16
|  +--rw pkt-sec-cond-ipv4-ttl* uint 8
|  +--rw pkt-sec-cond-ipv4-protocol* uint 8
|  +--rw pkt-sec-cond-ipv4-src* inet:ipv4-address
|  +--rw pkt-sec-cond-ipv4-dest* inet:ipv4-address
|  +--rw pkt-sec-cond-ipv4-ipopts string
|  +--rw pkt-sec-cond-ipv4-sameip boolean
|  +--rw pkt-sec-cond-ipv4-geoip* string
```

[IPv4 Header]

```
+--rw packet-security-ipv6-condition
|  +--rw pkt-sec-cond-ipv6-dscp* string
|  +--rw pkt-sec-cond-ipv6-ecn* string
|  +--rw pkt-sec-cond-ipv6-traffic-class* uint 8
|  +--rw pkt-sec-cond-ipv6-flow-label* uint 32
|  +--rw pkt-sec-cond-ipv6-payload-length* uint 16
|  +--rw pkt-sec-cond-ipv6-next-header* uint 8
|  +--rw pkt-sec-cond-ipv6-hop-limit* uint 8
|  +--rw pkt-sec-cond-ipv6-src* inet:ipv6-address
|  +--rw pkt-sec-cond-ipv6-dest* inet:ipv6-address
```

[IPv6 Header]

# Expansion of YANG data model with packet header (2/2)

```
+--rw packet-security-tcp-condition
|  +--rw pkt-sec-cond-tcp-seq-num* uint 32
|  +--rw pkt-sec-cond-tcp-ack-num* uint 32
|  +--rw pkt-sec-cond-tcp-window-size* uint 16
|  +--rw pkt-sec-cond-tcp-falgs* uint 8
```

[TCP Header]

```
+--rw packet-security-udp-condition
|  +--rw pkt-sec-cond-udp-length* string
```

[UDP Header]

```
+--rw packet-security-icmp-condition
   +--rw pkt-sec-cond-icmp-type* uint 8
   +--rw pkt-sec-cond-icmp-code* uint 8
   +--rw pkt-sec-cond-icmp-seq-num* uint 32
```

[ICMP Header]

# Replacement of “List” with “Leaf-list” for more compact expression

```
+--rw condition
  +--rw packet-security-condition
  |  +--rw packet-security-mac-condition* [pkt-sec-cond-mac-id]
  |  |  +--rw pkt-sec-cond-mac-id uint 8
  |  |  +--rw pkt-sec-cond-mac-dest inet:port-number
  |  |  +--rw pkt-sec-cond-mac-src inet:port-number
  |  |  +--rw pkt-sec-cond-mac-8021q string
  |  |  +--rw pkt-sec-cond-mac-ether-type string
  |  |  +--rw pkt-sec-cond-mac-tci string
  |  +--rw packet-security-ipv4-condition* [pkt-sec-cond-ipv4-id]
  |  |  +--rw pkt-sec-cond-ipv4-id uint 8
  |  |  +--rw pkt-sec-cond-ipv4-src inet:ipv4-address
  |  |  +--rw pkt-sec-cond-ipv4-dest inet:ipv4-address
  |  |  +--rw pkt-sec-cond-ipv4-protocol string
  |  |  +--rw pkt-sec-cond-ipv4-dscp string
  |  |  +--rw pkt-sec-cond-ipv4-ecn string
  |  |  +--rw pkt-sec-cond-ipv4-length string
  |  |  +--rw pkt-sec-cond-ipv4-ttl
  |  +--rw packet-security-ipv6-condition* [pkt-sec-cond-ipv6-id]
  |  |  +--rw pkt-sec-cond-ipv6-id uint 8
  |  |  +--rw pkt-sec-cond-ipv6-src inet:ipv6-address
  |  |  +--rw pkt-sec-cond-ipv6-dest inet:ipv6-address
  |  |  +--rw pkt-sec-cond-ipv6-dscp string
  |  |  +--rw pkt-sec-cond-ipv6-ecn string
  |  |  +--rw pkt-sec-cond-ipv6-flow-label string
  |  |  +--rw pkt-sec-cond-ipv6-payload-length string
  |  |  +--rw pkt-sec-cond-ipv6-next-header string
  |  |  +--rw pkt-sec-cond-ipv6-hop-limit string
  |  +--rw packet-security-tcp-condition* [pkt-sec-cond-tcp-id]
  |  |  +--rw pkt-sec-cond-tcp-id uint 8
  |  |  +--rw pkt-sec-cond-tcp-src-port inet:port-number
  |  |  +--rw pkt-sec-cond-tcp-dest-port inet:port-number
  |  |  +--rw pkt-sec-cond-tcp-seq-num string
  |  |  +--rw pkt-sec-cond-tcp-falgs string
  |  +--rw packet-security-udp-condition* [pkt-sec-cond-udp-id]
  |     +--rw pkt-sec-cond-udp-id uint 8
  |     +--rw pkt-sec-cond-udp-src-port inet:port-number
  |     +--rw pkt-sec-cond-udp-dest-port inet:port-number
  |     +--rw pkt-sec-cond-udp-length string
```

```
+--rw condition
  +--rw packet-security-condition* [pkt-security-id]
  |  +--rw pkt-security-id uint 8
  |  +--rw packet-security-mac-condition
  |  |  +--rw pkt-sec-cond-mac-dest* inet:port-number
  |  |  +--rw pkt-sec-cond-mac-src* inet:port-number
  |  |  +--rw pkt-sec-cond-mac-8021q* string
  |  |  +--rw pkt-sec-cond-mac-ether-type* string
  |  |  +--rw pkt-sec-cond-mac-tci* string
  |  +--rw packet-security-ipv4-condition
  |  |  +--rw pkt-sec-cond-ipv4-header-length* uint 8
  |  |  +--rw pkt-sec-cond-ipv4-tos* uint 8
  |  |  +--rw pkt-sec-cond-ipv4-total-length* uint 16
  |  |  +--rw pkt-sec-cond-ipv4-id* uint 16
  |  |  +--rw pkt-sec-cond-ipv4-fragment* uint 8
  |  |  +--rw pkt-sec-cond-ipv4-fragment-offset* uint 16
  |  |  +--rw pkt-sec-cond-ipv4-ttl* uint 8
  |  |  +--rw pkt-sec-cond-ipv4-protocol* uint 8
  |  |  +--rw pkt-sec-cond-ipv4-src* inet:ipv4-address
  |  |  +--rw pkt-sec-cond-ipv4-dest* inet:ipv4-address
  |  |  +--rw pkt-sec-cond-ipv4-ipopts string
  |  |  +--rw pkt-sec-cond-ipv4-sameip boolean
  |  |  +--rw pkt-sec-cond-ipv4-geoip* string
  |  +--rw packet-security-ipv6-condition
  |  |  +--rw pkt-sec-cond-ipv6-dscp* string
  |  |  +--rw pkt-sec-cond-ipv6-ecn* string
  |  |  +--rw pkt-sec-cond-ipv6-traffic-class* uint 8
  |  |  +--rw pkt-sec-cond-ipv6-flow-label* uint 32
  |  |  +--rw pkt-sec-cond-ipv6-payload-length* uint 16
  |  |  +--rw pkt-sec-cond-ipv6-next-header* uint 8
  |  |  +--rw pkt-sec-cond-ipv6-hop-limit* uint 8
  |  |  +--rw pkt-sec-cond-ipv6-src* inet:ipv6-address
  |  |  +--rw pkt-sec-cond-ipv6-dest* inet:ipv6-address
  |  +--rw packet-security-tcp-condition
  |  |  +--rw pkt-sec-cond-tcp-seq-num* uint 32
  |  |  +--rw pkt-sec-cond-tcp-ack-num* uint 32
  |  |  +--rw pkt-sec-cond-tcp-window-size* uint 16
  |  |  +--rw pkt-sec-cond-tcp-falgs* uint 8
  |  +--rw packet-security-udp-condition
  |  |  +--rw pkt-sec-cond-udp-length* string
  |  +--rw packet-security-icmp-condition
  |     +--rw pkt-sec-cond-icmp-type* uint 8
  |     +--rw pkt-sec-cond-icmp-code* uint 8
  |     +--rw pkt-sec-cond-icmp-seq-num* uint 32
```

# Next Steps

- We will verify our YANG data model by implementing a prototype with other open source (e.g., Snort) other than Suricata.

- We will extend our YANG data model to support other security controls other than Network security control, such as Content security control and Attack mitigation  control.