Information Model of Interface to Network Security Functions Capability Interface

draft-xia-i2nsf-capability-interface-im-05

Liang Xia	Huawei
DaCheng Zhang	Alibaba
Edward Lopez	Fortinet
Nicolas BOUTHORS	Qosmos
Luyuan Fang	Microsoft

April 2016 Buenos Ayres

Monitoring Part of I2NSF Architecture

Service Layer

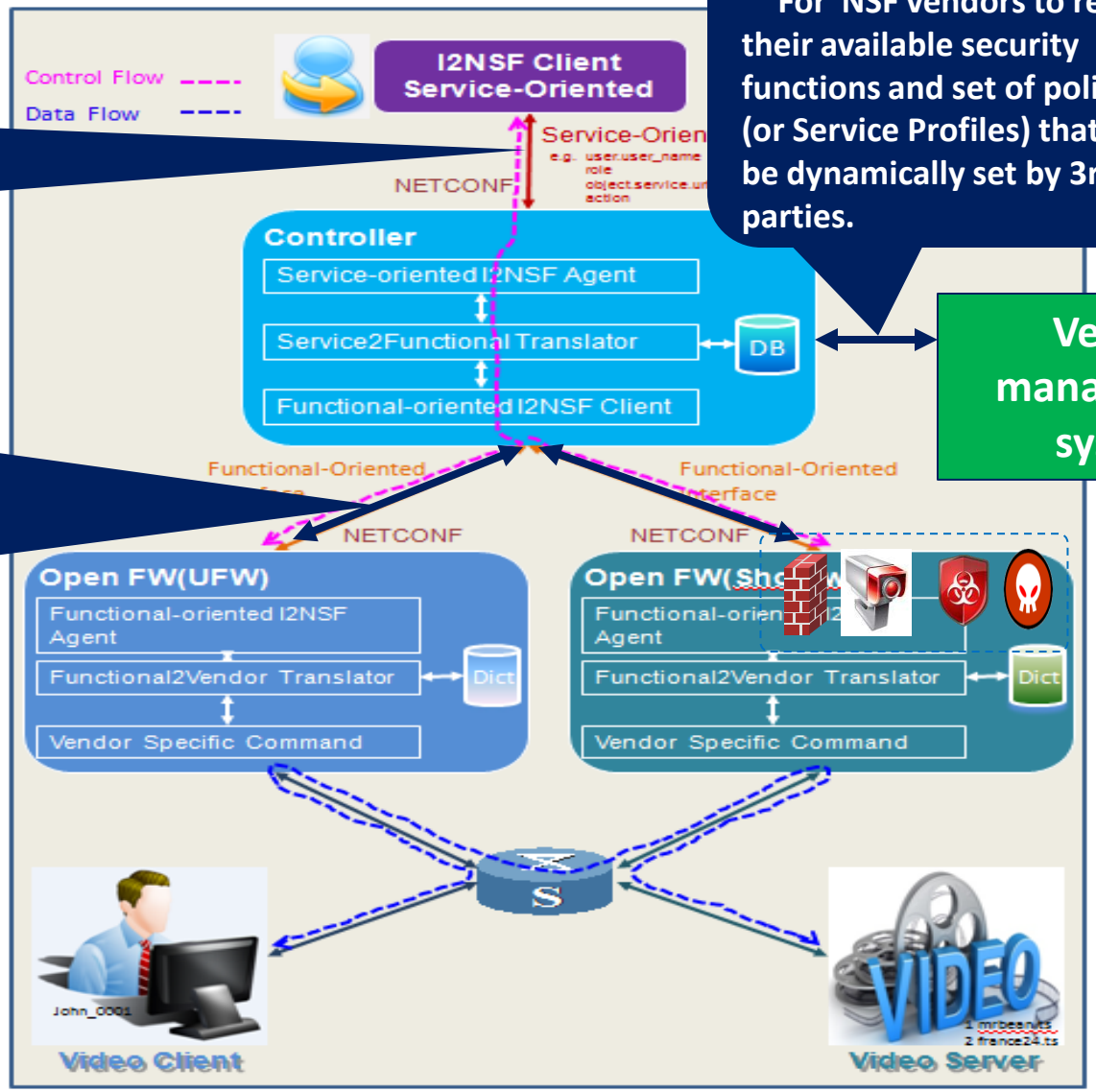
For clients or App Gateway to express and monitor security policies for their specific flows

NSF Registration

For NSF vendors to register their available security functions and set of policies (or Service Profiles) that can be dynamically set by 3rd parties.

Capability Layer

For controller to define explicit rules for individual NSFs to treat packets, as well as methods to monitor the execution status of those functions



Vendor management system

Design Goals

- *A standard information model of capability interface for NSF:*
  - To realize the security policy provisioning which governs how the packets are treated by the NSF;
  - By building on the packet/flows-based paradigm;
- In order to:
  - Decouple network security controller from vendor-specific NSFs, and vice versa;
  - Abstract general network security capability to be managed flexibly and efficiently;
  - Avoid potential constraints on the NSFs.

3 Categories of Security Capabilities

1. Network security control:
   - Inspecting and processing the network packet/flow;
   - Packet contents, context information, actions;
   - Use a “Event-Condition-Action" paradigm;
2. Content security control:
   - Detect the malicious contents in application layer : file, url, data block, etc;
   - Security profiles or signature files with standardized input/output parameters;
   - Possibly need the standardized interface for updating its intelligence: signature, and algorithm.
3. Attack mitigation control:
   - Detect and mitigate various types of network attacks: DDoS attacks, Single-packet attacks, ipv6 related attack;
   - A standard interface for the security controller to choose and customize the given security capability.

Overall Structure for Information Model for security capability management

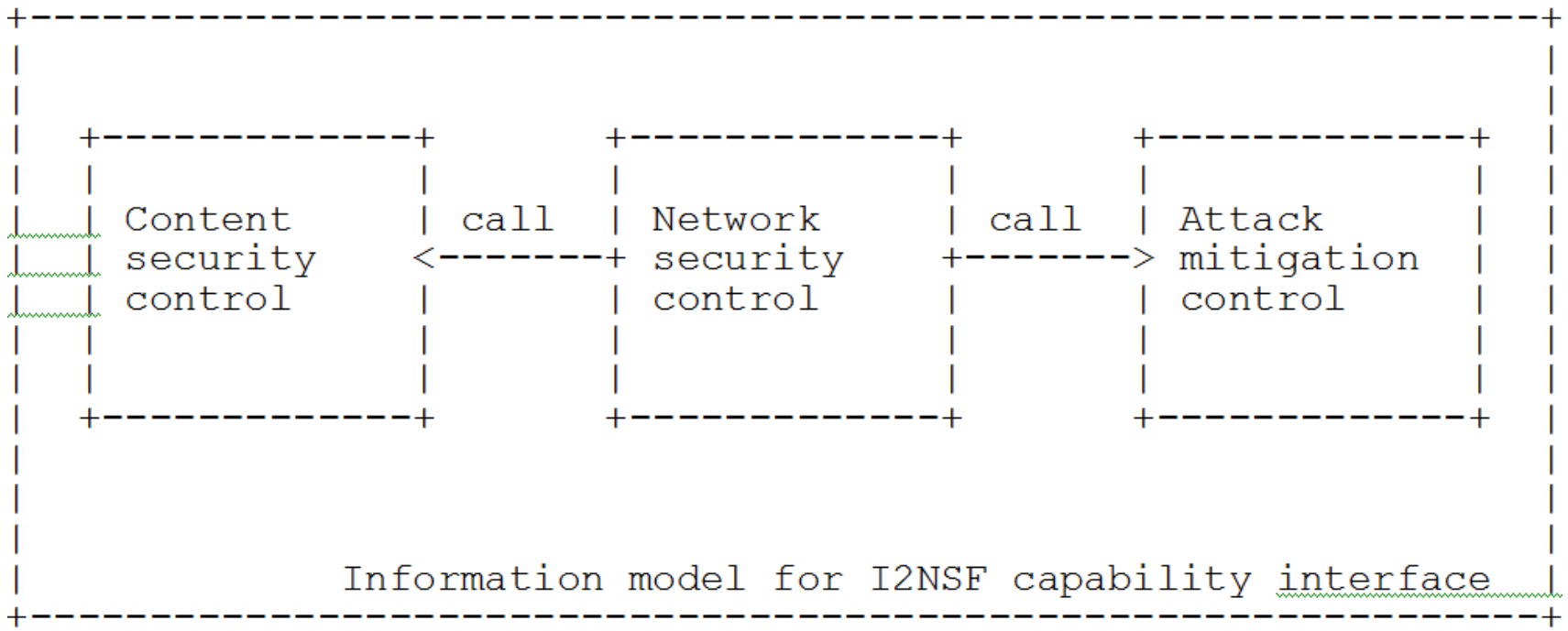
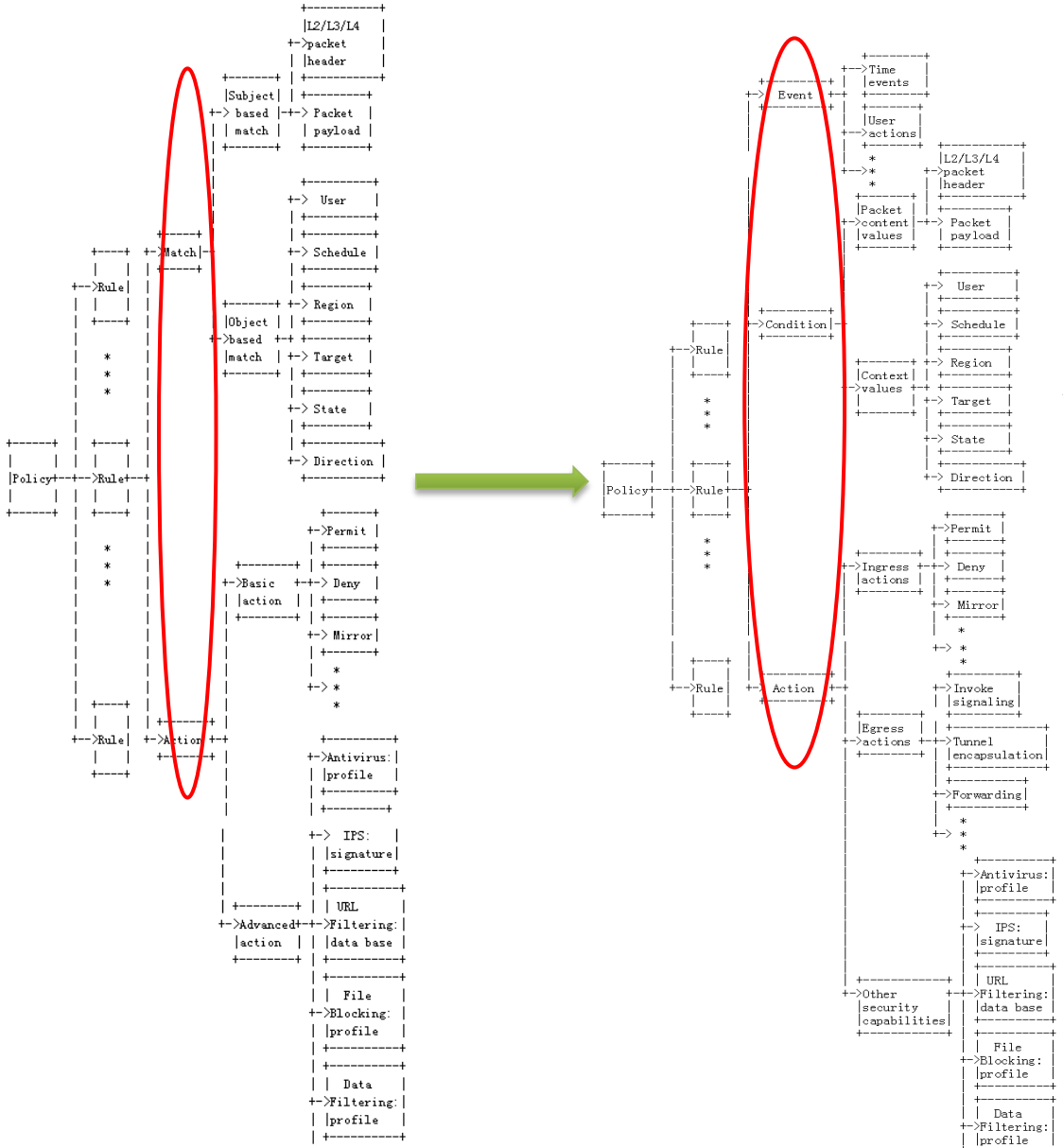


Figure 1. The overall structure of information model for I2NSF Capability Interface

ECA Based Information Model



An example of an I2NSF ECA Policy Rule is, in pseudo-code:

IF <event-clause> is TRUE

IF <condition-clause> is TRUE

THEN execute <action-clause>

END-IF

END-IF

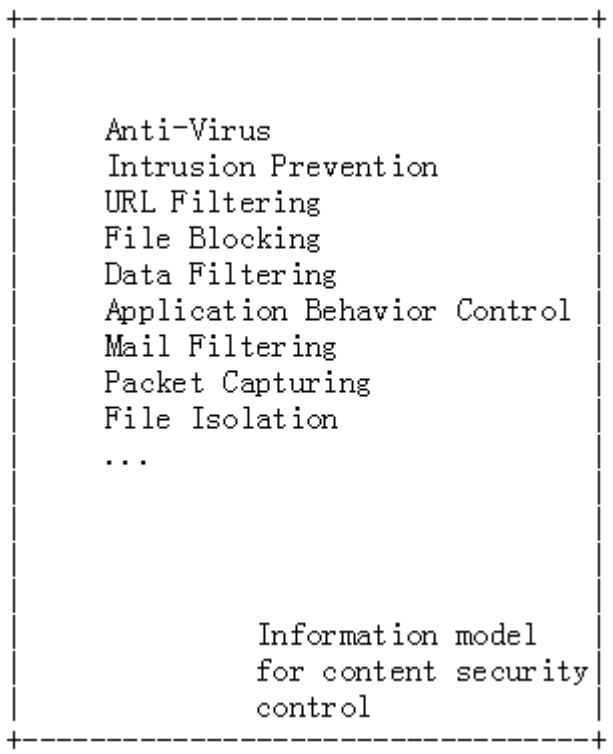
*In the above example, the Event, Condition, and Action portions of a Policy Rule are all **Boolean Clauses**.*

Match Condition Details

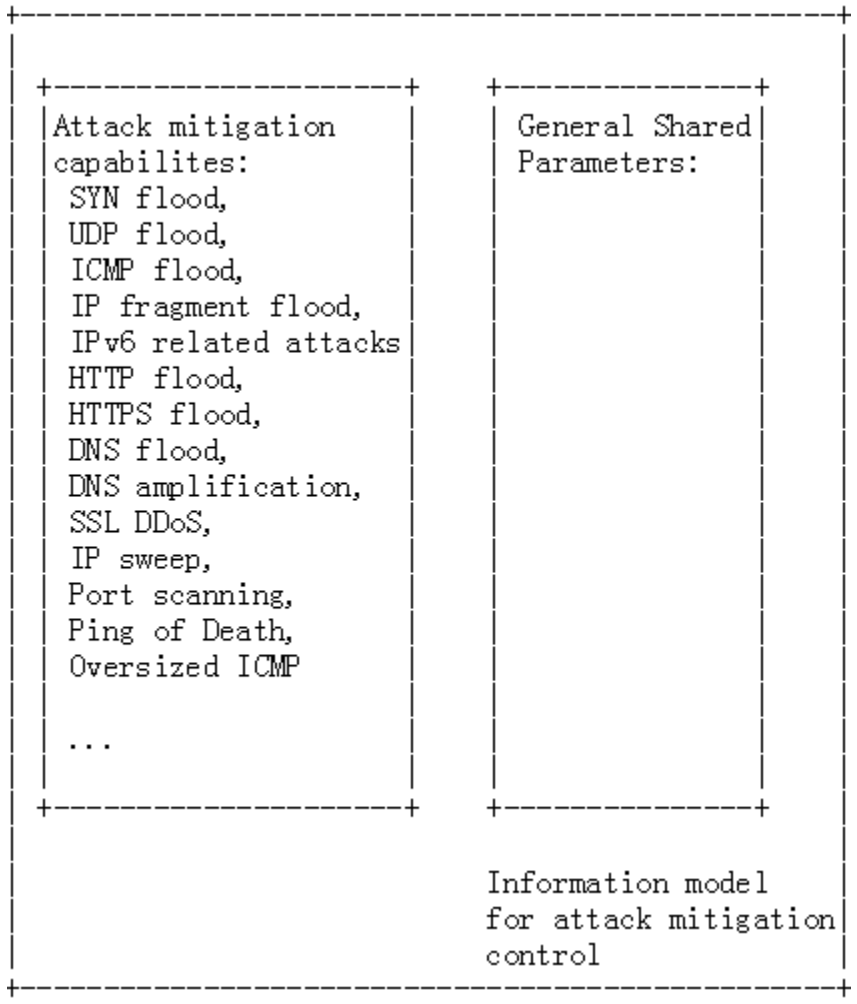
Match Condition	Attributes: Values
Time Event	TBD
User Actions Event	login, logout, violate ACL ...
Ethernet Frame Header	Source/Destination address s-VID/c-VID/EtherType
IPv4 Packet Header	src/dest address protocol src/dest port length flags ttl
IPv6 Packet Header	src/dest address protocol/nh src/dest port length traffic class hop limit flow label
TCP SCTP DCCP	Port syn ack fin rst psh urg window sockstress

User	
Schedule	time span days, minutes, seconds,
Region	country, province, city IP address, network section, network domain
Target	service: TCP, UDP, ICMP, HTTP... application: Gmail, QQ, MySQL... device: mobile phone, tablet, PC...
State	session state: new, established, related invalid, untracked access mode: WIFI, 802.1x, PPPOE, SSL...
Direction	Direction: from_client, from_server, bidirection, reversed

Information Model for Content Security Control



Information Model for Attack Mitigation Control



Information Model Grammar Details

<Policy> ::= <policy-name> <policy-id> (<Rule> ...)

<Rule> ::= <rule-name> <rule-id> <Match> <Action>

<Match> ::= [<subject-based-match>] [<object-based-match>]

<subject-based-match> ::= [<L234-packet-header> ...] [<packet-payload> ...]

<L234-packet-header> ::= [<address-scope>] [<layer-2-header>] [<layer-3-header>] [<layer-4-header>]

<address-scope> ::= <route-type> (<ipv4-route> | <ipv6-route> | <mpls-route> | <mac-route> | <interface-route>)

<route-type> ::= <IPV4> | <IPV6> | <MPLS> | <IEEE_MAC> | <INTERFACE>

<ipv4-route> ::= <ip-route-type> (<destination-ipv4-address> | <source-ipv4-address> | (<destination-ipv4-address> <source-ipv4-address>))

<destination-ipv4-address> ::= <ipv4-prefix>

<source-ipv4-address> ::= <ipv4-prefix>

<ipv4-prefix> ::= <IPV4_ADDRESS> <IPV4_PREFIX_LENGTH>

<ipv6-route> ::= <ip-route-type> (<destination-ipv6-address> | <source-ipv6-address> | (<destination-ipv6-address> <source-ipv6-address>))

<destination-ipv6-address> ::= <ipv6-prefix>

<source-ipv6-address> ::= <ipv6-prefix>

<ipv6-prefix> ::= <IPV6_ADDRESS> <IPV6_PREFIX_LENGTH>

<ip-route-type> ::= <SRC> | <DEST> | <DEST_SRC>

<layer-3-header> ::= <ipv4-header> | <ipv6-header>

<ipv4-header> ::= <SOURCE_IPv4_ADDRESS> <DESTINATION_IPv4_ADDRESS> <PROTOCOL> [<TTL>] [<DSCP>]

<ipv6-header> ::= <SOURCE_IPV6_ADDRESS> <DESTINATION_IPV6_ADDRESS> <NEXT_HEADER> [<TRAFFIC_CLASS>] [<FLOW_LABEL>] [<HOP_LIMIT>]

<object-based-match> ::= [<user> ...] [<schedule>] [<region>] [<target>] [<state>]

<user> ::= (<login-name> <group-name> <parent-group> <password> <expired-date> <allow-multi-account-login> <address-binding>) | <tenant> | <VN-id>

<schedule> ::= <name> <type> <start-time> <end-time> <weekly-validity-time>

<type> ::= <once> | <periodic>

<target> ::= [<service>] [<application>] [<device>]

<service> ::= <name> <id> <protocol> [<protocol-num>] [<src-port>] [<dest-port>]

<protocol> ::= <TCP> | <UDP> | <ICMP> | <ICMPv6> | <IP>

<application> ::= <name> <id> <category> <subcategory> <data-transmission-model> <risk-level> <signature>

<category> ::= <business-system> | <Entertainment> | <internet> | <network> | <general>

<subcategory> ::= <Finance> | <Email> | <Game> | <media-sharing> | <social-network> | <web-posting> | <proxy> | ...

<data-transmission-model> ::= <client-server> | <browser-based> |<networking> | <peer-to-peer> | <unassigned>

<risk-level> ::= <Exploitable> | <Productivity-loss> | <Evasive> | <Data-loss> | <Malware-vehicle> |<Bandwidth-consuming> | <Tunneling>

<signature> ::= <server-address> <protocol> <dest-port-num> <flow-direction> <object> <keyword>

<flow-direction> ::= <request> | <response> | <bidirection>

<object> ::= <packet> | <flow>

<context based match> ::= [<user-group> ...] [<session-state>] [<schedule>] [<region-group>]

<user-group> ::= <user>...

<user> ::= (<login-name> <group-name> <parent-group> <password> <expired-date> <allow-multi-account-login> <address-binding>) | <tenant> | <VN-id>

<session-state> ::= <new> | <established> | <related> | <invalid> | <untracked>

<schedule> ::= <name> <type> <start-time> <end-time> <weekly-validity-time>

<type> ::= <once> | <periodic>

<action> ::= <basic-action> [<advanced-action>]

<basic-action> ::= <pass> | <deny> | <mirror> | <call-function> | <encapsulation>

<advanced-action> ::= [<profile-antivirus>] [<profile-IPS>] [<profile-url-filtering>] [<profile-file-blocking>] [<profile-data-filtering>] [<profile-application-control>]

Next Step

- Solicit Comments

- More detailed contents, including:
  - content security control IM;
  - attack mitigation control IM;
  - others.

- Call for adoption

Thanks!

Liang Xia (Frank)