Requirements for Client-facing Interface to Security controller

draft-ietf-i2nsf-client-facing-interface-req-02

Rakesh Kumar Juniper networks
Anil Lohiya Juniper networks
Dave Qi Bloomberg
Nabll Bitar Nokia
Senand Palislamovic Nokia
Liang Xia Huawei

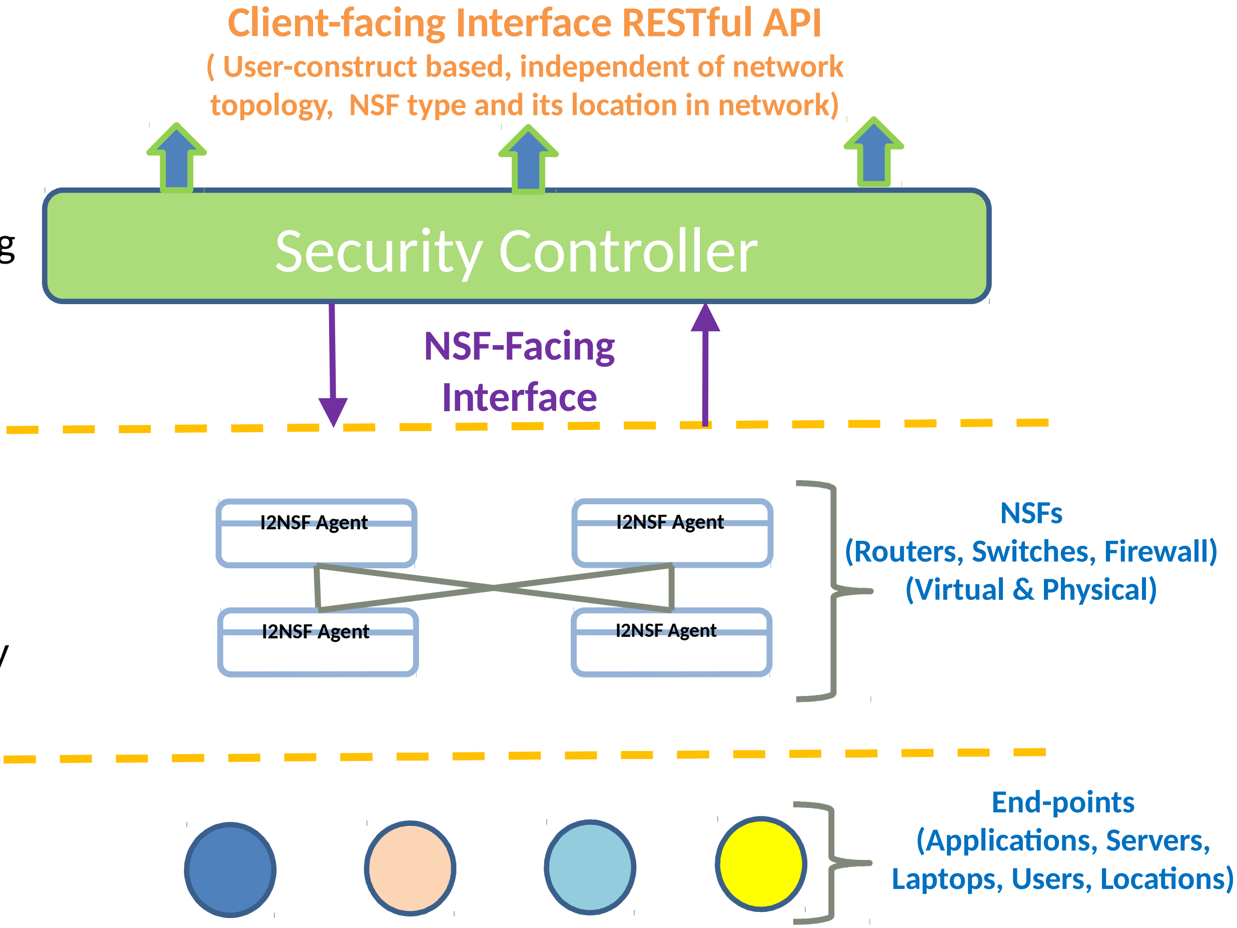
IETF-99, Prague
July 18, 2017

Agenda

- Draft overview
- Main updates outline
- Next steps and plans

Draft scope – Identify requirements to buil d I2NSF client-facing Interface

- End-user/application express security policies using client-facing interface
- All end-user interaction through an abstraction layer in security controller
- End-user security policies enforced on traffic originated and destined to end-points
- Security policy deployed in NSF by security controller


Client-facing Interface RESTful API
(User-construct based, independent of network topology, NSF type and its location in network)
Security Controller
NSF-Facing Interface
I2NSF Agent
I2NSF Agent
I2NSF Agent
I2NSF Agent
NSFs
(Routers, Switches, Firewall)
(Virtual & Physical)
End-points
(Applications, Servers, Laptops, Users, Locations)

Main Updates Outline

- Introduce requirements preference
  - MUST
  - MAY
  - RECOMMENDED
- Several new requirements based on ONUG feedback
  - 3 new categories of security policy: Segmentation policies, Threat policies, Governance and Compliance policies
  - More fine-grained policy building blocks: Source Policy Endpoint Group, Destination Policy Endpoint Group, Direction, Threat Group, Match Condition, Exceptions, Actions…
  - Consistent policy enforcement: according to network/policy building blocks change, audit and log the change
- A lot of improved description

Draft overview – Designing Principles

- User-construct based modeling: abstract + decoupling
  - Easier for end-user to express policy which reflects business needs
  - Not dependent on low level network information
- More concretely:
  - Decoupling from low level network information: network topology, NSF ty pe/model/location…
  - Using Declarative/Descriptive model instead of Imperative/Prescriptive m odel
  - Being not dependent on NSFs’ operation in network, such as:
    - How to be connected in network
    - Control plane interactions: HA, scalability, etc
    - Data plane implementations: encap, sfc, etc
- Deployment Models: direct interaction, NMS proxy interaction

Draft overview – Set of requirements… (1/2)

- Functional requirements for interface, to support:
  - Multi-tenancy (isolation)： Policy-Administrator of Policy-Tenant manages Policy-User
  - Authentication and authorization
    - RBAC
  - Protection against:
    - attacks (DoS/DDoS)
    - Misconfiguration, Input data validation
  - Dynamic control of policy enforcement
    - Admin-Enforced
    - Time-Enforced
    - Event-Enforced
  - Definition of dynamic policy end group
    - User-Group, Device-Group, Application-Group, Location-Group
  - Security policy building blocks
    - 3 categories of security policy: Segmentation policies, Threat policies, Governance and Compliance policies
    - Building blocks: Source Policy Endpoint Group, Destination Policy Endpoint Group, Direction, Threat Group, Match Condition, Exceptions, Actions…

Draft overview – Set of requirements… (2/2)

- Comprehensive set of actions: Permit, Deny, Drop connection, Log, Authenticate connection, Quarantine/Redirect, Netflow, Count, Encrypt, Decrypt, Throttle, Mark, Instantiate-NSF
- Consistent policy enforcement: according to network/policy building blocks change, audit and log the change
- detect and correct policy conflicts, and backward compatibility
- Integration with external systems
  - Threat feeds, Honeypots
  - Security Information & Event Management (SIEM)
  - Network and Behavior analytic engines
- Telemetry data collection
  - Get data from NSF system logs, syslog, flow records, security violations
  - Export data to external systems for monitoring and analytics

➢ Operational requirements for interface

- APIs
  - API versioning for problem debugging, and backward compatibility
  - API extensibility
  - Data Model Transport: Yang + netconf/restconf
- Miscellaneous
  - Notification to end-user based on NSF events and policy violations
  - Test policies for conflicts before deploying
  - Affinity to allow end-user so that a policy is enforced on a specific NSF
    - Need to work on it some more

Next steps and plans for draft draft-ietf-i2nsf-client-facing-interface-req-03

- Add examples for requirement
  - Illustrate each requirement with use-case example for clarity
- Align to I2NSF Terminology draft
- Incorporate ideas from WG mailing discussions
  - Few comments received so far
    - Linda Dunbar
      - Clarification about actual requirements vs high level requirements
      - More clear clarification of the difference between “User-construct based poli cies” and the general “intent-based policy”
  - Solicit inputs on requirements
    - Get more use-cases from WG members in different segments
      - Service providers, Enterprise, cloud operators

Thanks!

Rakesh Kumar