Interface to Network Security
Functions
Problem Statement
July 2015

Linda Dunbar (linda.dunbar@huawei.com)
Myo Zarny (Myo.Zarny@gs.com)
Christian Jacquenet (Christian.jacquenet@orange.com)
Mohamed Boucadair (mohamed.boucadair@orange.com)
Shaibal Chakrabarty (shaibalc@us-ignite.org)
Multi-vendor & Multi-Types of NSFs
To be managed
Automation of the NSFs’ control & monitor

Security Controller (e.g. SDN)

Standard format for Rules, Signatures, …
to dynamically set/monitor/query the behavior of NSFs

It doesn’t require NFV, it doesn’t require provider domain. I2NSF is to facilitate automation.
Different vendor $\rightarrow$ Different Provisioning Formats

Vendor A

```
firewall name <name> default-action <action>
```

- **name**: The name of the firewall rule set.
- **action**: The default action to take if no matches are found within a rule set. Supported values are as follows:
  - `accept`: Accepts the packet.
  - `drop`: Drops the packet silently.
  - `reject`: Drops the packet with an ICMP “Destination Unreachable” message.

Vendor B

```
Action
 Use the Action field to define what occurs to traffic that matches the URL Filtering and Application Control rule. These are the Action options:

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AclX</td>
<td>Allows the traffic.</td>
</tr>
<tr>
<td>Block</td>
<td>Blocks the traffic. Shows a UserCheck Block message.</td>
</tr>
<tr>
<td>Limit</td>
<td>Defines the maximum bandwidth that is allowed for this rule. Select or create a Limit object that defines the bandwidth limits.</td>
</tr>
</tbody>
</table>
```

Vendor C

```
firewall name <name> rule <rule-num> limit
```

- Specifies traffic rate limiting parameters for a firewall rule.

**Syntax**

```
set firewall name rule rule-num limit [burst size | rate rate]
```

**Configuration Statement**

```
firewall {
    name name {
        rule rule-num {
            limit {
                burst size
                rate rate
            }
        }
    }
}
```
FW configuration: ports & links based

Virtual Networks Needs Group Policies & Abstraction. Need standard format for automation

<table>
<thead>
<tr>
<th>Active</th>
<th>Type</th>
<th>Rule</th>
<th>Protocol</th>
<th>Source</th>
<th>Port(s)</th>
<th>Destination</th>
<th>Port(s)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Access</td>
<td>Permit</td>
<td>UDP</td>
<td>IP or Host Name</td>
<td>ALL</td>
<td>Any</td>
<td>53</td>
<td>Example - Permit DNS request to this IP</td>
</tr>
<tr>
<td>No</td>
<td>Access</td>
<td>Permit</td>
<td>TCP</td>
<td>IP or Host Name</td>
<td>ALL</td>
<td>Any</td>
<td>110</td>
<td>Example - Permit POP access to this IP</td>
</tr>
<tr>
<td>No</td>
<td>Access</td>
<td>Permit</td>
<td>TCP</td>
<td>IP or Host Name</td>
<td>ALL</td>
<td>Any</td>
<td>25</td>
<td>Example - Permit SMTP access to this IP</td>
</tr>
<tr>
<td>No</td>
<td>Access</td>
<td>Deny</td>
<td>ALL</td>
<td>IP or Host Name</td>
<td>ALL</td>
<td>Any</td>
<td>ALL</td>
<td>Example - Deny all access to this IP</td>
</tr>
<tr>
<td>No</td>
<td>Access</td>
<td>Deny</td>
<td>ALL</td>
<td>IP or Host Name</td>
<td>ALL</td>
<td>Any</td>
<td>ALL</td>
<td>Example - Deny access to this Sub-net</td>
</tr>
<tr>
<td>No</td>
<td>Access</td>
<td>Deny</td>
<td>TCP</td>
<td>Any</td>
<td>ALL</td>
<td>Any</td>
<td>21</td>
<td>Example - Deny access to FTP sites</td>
</tr>
</tbody>
</table>
OpenStack FWaaS Rules Configuration

```json
{
    "firewall_rule": {
        "action": "allow",
        "description": "",
        "destination_ip_address": null,
        "destination_port": "80",
        "enabled": true,
        "firewall_policy_id": null,
        "id": "8722e0e0-9cc9-4490-9660-8c9a5732fbb0",
        "ip_version": 4,
        "name": "ALLOW_HTTP",
        "position": null,
        "protocol": "tcp",
        "shared": false,
        "source_ip_address": null,
        "source_port": null,
        "tenant_id": "45977fa2dbd7482098dd68d0d8970117"
    }
}
```
Summary of I2NSF Problems

• 3.1. Challenges Facing Security Service Providers
  - 3.1.1. Diverse types of Security Functions
  - 3.1.2. Diverse Interfaces to Control NSFs
  - 3.1.3. Diverse Interface to monitor the behavior of NSFs
  - 3.1.4. More Distributed NSFs and vNSFs
  - 3.1.5. More Demand to Control NSFs Dynamically
  - 3.1.6. Demand for multi-tenancy to control and monitor NSFs.
  - 3.1.7. Lack of Characterization of NSFs and Capability Exchange
  - 3.1.8. Lack of mechanism for NSFs to utilize external profiles

• 3.2. Challenges Facing Customers
  - 3.2.1. NSFs from heterogeneous administrative domains
  - 3.2.2. Today’s Control Requests are Vendors Specific
  - 3.2.3. Difficulty to Monitor the Execution of Desired Policies

• 3.3. Difficulty to Validate Policies across Multiple Domains
• 3.4. Lack of Standard Interface to Inject Feedback to NSF
• 3.5. Lack of Standard Interface for Capability Negotiation
Goal of I2NSF

- Specify and standardize corresponding information and data models for the dynamic provisioning, querying, monitoring of flow based network security functions

- Define Policy Enforcement Schemes for automated delivery of security services, Design feedback mechanisms for security service fulfillment and assurance purposes

• Other aspects of NSFs, such as device or network provisioning and configuration, are out of scope
Welcome to I2NSF Running Code

The running code is focused on the design of an I2NSF demo including the design of I2NSF client, I2NSF controller and NSF/vNSF. NETCONF protocol and YANG model are used for the I2NSF demo realization. The demo aims to enhance understanding of the I2NSF architecture and justify its feasibility.

I2NSF/Demo Description

Branch: master

<table>
<thead>
<tr>
<th>Component</th>
<th>Authored</th>
<th>Latest Commit</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>I2NSF Client</td>
<td>21 days</td>
<td>89acf</td>
<td></td>
</tr>
<tr>
<td>I2NSF Controller</td>
<td>21 days</td>
<td>89acf</td>
<td></td>
</tr>
<tr>
<td>UFW</td>
<td>21 days</td>
<td>89acf</td>
<td></td>
</tr>
<tr>
<td>Shorewall</td>
<td>21 days</td>
<td>89acf</td>
<td></td>
</tr>
</tbody>
</table>