Problem Statement

draft-dunbar-i2nsf-problem-statement

Sue Hares
Interface to Network Security Functions Problem Statement

• 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)

"If I have seen further, it is by standing upon the shoulders of giants".
Sir Isaac Newton
Multi-vendor & Multi-Types of NSFs
To be managed
Automation of the NSFs’ control & monitor

Security Controller (e.g. SDN)

Internet

Clients/Apps/etc

Attack Analyzers, Actions Intelligence

Incident reports from External Sources

Incidents Response Intelligence/Feedback from Network

Standard format for Rules, Signatures, ...

to dynamically set/monitor/query the behavior of NSFs

Feedback from NSFs/Network

NSF Pool

Multi-Vendors NSFs

It doesn’t require NFV, it doesn’t require provider domain. I2NSF is to facilitate automation.
Different vendor → 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.

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

- Specifies traffic rate limiting parameters for a firewall rule.

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

Configuration Statement:
```
firewall {
  name name {
    rule-rule-num {
      limit {
        burst size
        rate rate
      }
    }
  }
}
```

Vendor B

```
Action
```

- **Action** field to define what occurs to traffic that matches the URL Filtering and Application Control rule. These are the Action options:
  - **Adox**: Allows the traffic.
  - **Block**: Blocks the traffic. Shows a UserCheck Block message. If no UserCheck object is defined for this action, no message is displayed.
  - **Limit**: Defines the maximum bandwidth that is allowed for this rule. Select or create a Limit object that defines the bandwidth limits.

Vendor C

Difficult to achieve automated deployment.
FW configuration: ports & links based

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

Need standard method to express commonly used rules for virtual networks and groups.
OpenStack FWaaS Rules Configuration

```json
{
    "firewall_rule": {
        "action": "allow",
        "description": "",
        "destination_ip_address": null,
        "destination_port": "80",
        "enabled": true,
        "firewall_policy_id": null,
        "id": "8722e0e0-9cc9-4490-9660-9c9a5732fbb0",
        "ip_version": 4,
        "name": "ALLOW_HTTP",
        "position": null,
        "protocol": "tcp",
        "shared": false,
        "source_ip_address": null,
        "source_port": null,
        "tenant_id": "45977fa2dbd7482098dd68d0d6970117"
    }
}
```
Challenges (Section 3 of document)

Facing Service Providers (3.1)

- Diverse types of Security functions
- Diverse interfaces to control NSFs
- Diverse interface to monitor NSFs
- More Distributed NSFs and vNSFs
- More demand to control NSFs Dynamically
- Demand for multi-tenancy and control NSFs
- Lack of Characterization of NSF and Capability Exchange
- Lack of mechanism for SMFs to utilize external profiles

Facing Customers (3.2)

- NSF from heterogeneous administrative domains
- Control Requests are Vendors Specific
- Difficulty to Monitor the Execution of Desired Policies

Common Problems (3.4-3.6)

- Difficulty to Validate Policies across Multiple Domains
- Lack of Standard Interface to Inject Feedback to NSF
- Lack of Standard Interface for Capability Negotiation
Other Areas

• ETSI-NFV - EMS to VNF interface
  - Defines interface between EMS (element management system) and VNF
  - This matches I2NSF work

• OPNFV Moon project – An interface between EMS-VNF
  - Problems: NO dynamic control, only 1 definition, no room for existing vendor, no fine grain authentication, no allowance for central control

• CSA – 1 definition, 10 implementation agreements
  - All are concerned about the NMS-NSF interface
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>Commit SHA</th>
</tr>
</thead>
<tbody>
<tr>
<td>I2NSF Client</td>
<td>21 days ago</td>
<td>89acf</td>
<td>0452f</td>
</tr>
<tr>
<td>I2NSF Controller</td>
<td>21 days ago</td>
<td>89acf</td>
<td>0452f</td>
</tr>
<tr>
<td>UFW</td>
<td>21 days ago</td>
<td>89acf</td>
<td>0452f</td>
</tr>
<tr>
<td>Shorewall</td>
<td>21 days ago</td>
<td>89acf</td>
<td>0452f</td>
</tr>
</tbody>
</table>

Import to make steps toward Open Source for I2NSF