Extensions to the Access Control Lists (ACLs) YANG Model

draft-dbb-netmod-acl-00

O. Gonzalez de Dios, S. Barguil (**Telefonica**) M. Boucadair (**Orange**)

NETMOD WG Meeting

11th Nov 2021, Online IETF#112

Agenda

- Motivation & Approach
- Problem Statement
- Sample Gaps
- Next Steps

Motivation & Approach

- RFC 8519 defines a YANG data model for Access Control Lists (ACLs)
  - The model targets the configuration of the **forwarding behaviour** in a device
  - The model allows the definition of access-control-lists (ACLs), **entries** (ACEs), **matches,** and **actions**

- The authors have compiled the **operational experience** of Network Operators (Telefonica and Orange) in using ACLs in live deployments

- Based on that feedback from the operation teams, the authors have documented a set of improvements on the ACL model to cover the gaps

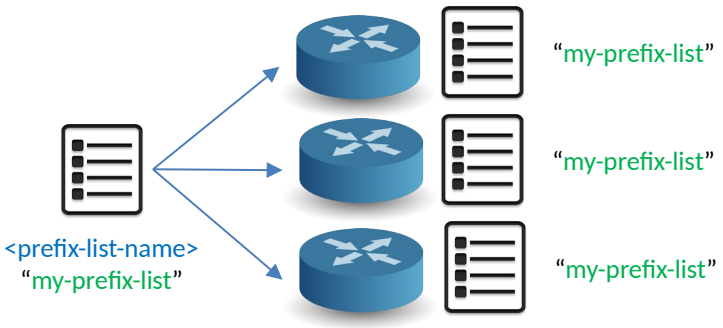
- RFC8519 is prepared for some augmentations/extensions, however:
  - The model structure suffers from a set of limitations
  - Not all the improvements are possible via augmentation with the current module structure

Problem Statement & Functional Gaps Found

- Suboptimal Configuration:
  - Lack of Manipulating Lists of Prefixes
- Manageability:
  - Impossibility to Use Aliases or Defined Sets
- Functionality:
  - Partial or Lack of IPv4/IPv6 Fragment Handling
  - Suboptimal TCP Flags Handling
  - Rate-Limit Actions
  - Payload-based Filtering
- Use the model at network level (current model is a device model)
  - Bind ACLs to Devices (Not Only Interfaces)
  - Reuse an ACLs Content & Defined sets Across Several Devices

Lack of Manipulating Lists of Prefixes (1)

- IP prefix related data nodes, e.g., "destination-ipv4-network" or "destination-ipv6-network", do not allow manipulating a list of IP prefixes, which may lead to manipulating large files
- RFC8519 supports only one IP address per A
- The same issue is encountered when ACLs have to be in place to mitigate DDoS attacks when a set of sources are involved in such an attack
- The situation is even worse when both a list of sources and destination prefixes are involved



Lack of Manipulating Lists of Prefixes (2)


```
{
    "ietf-access-control-list:acls": {
      "acl": [
        {
          "name": "first-prefix",
          "type": "ipv6-acl-type",
          "aces": {
            "ace": [
              {
                "name": "my-test-ace",
                "matches": {
                  "ipv6": {
                    "destination-ipv6-network":
                      "2001:db8:6401:1::/64",
                    "source-ipv6-network":
                      "2001:db8:1234::/96",
                    "protocol": 17,
                    "flow-label": 10000
                  },
                  "udp": {
                    "source-port": {
                      "operator": "lte",
                      "port": 80
                    },
                    "destination-port": {
                      "operator": "neq",
                      "port": 1010
                    }
                  }
                },
                "actions": {
                  "forwarding": "accept"
                }
              }
            ]
          }
        },
        {
          "name": "second-prefix",
          "type": "ipv6-acl-type",
          "aces": {
            "ace": [
              {
                "name": "my-test-ace",
                "matches": {
                  "ipv6": {
                    "destination-ipv6-network":
                      "2001:db8:6401:c::/64",
                    "source-ipv6-network":
                      "2001:db8:1234::/96",
                    "protocol": 17,
                    "flow-label": 10000
                  },
                  "udp": {
                    "source-port": {
                      "operator": "lte",
                      "port": 80
                    },
                    "destination-port": {
                      "operator": "neq",
                      "port": 1010
                    }
                  }
                },
                "actions": {
                  "forwarding": "accept"
                }
              }
            ]
          }
        }
      ]
    }
```


RFC8519

Gap


```
{
    "ietf-access-control-list:acls": {
      "acl": [
        {
          "name": "prefix-list-support",
          "type": "ipv6-acl-type",
          "aces": {
            "ace": [
              {
                "name": "my-test-ace",
                "matches": {
                  "ipv6": {
                    "destination-ipv6-network": [
                      "2001:db8:6401:1::/64",
                      "2001:db8:6401:c::/64"
                    ],
                    "source-ipv6-network":
                      ["2001:db8:1234::/96« ],
                    "protocol": 17,
                    "flow-label": 10000
                  },
                  "udp": {
                    "source-port": {
                      "operator": "lte",
                      "port": 80
                    },
                    "destination-port": {
                      "operator": "neq",
                      "port": 1010
                    }
                  }
                },
                "actions": {
                  "forwarding": "accept"
                }
              }
            ]
          }
        }
      ]
    }
```

Impossibility to Use Aliases & Defined Sets

- Manageability:
  - Network Operators maintain prefix lists to be used in ACLs. This facilitates the management of the ACLs
  - Note that routing policies (RFC 9067) already has the notion of prefix-sets to share among the different policies

- The prefix list can be generalized by introducing the concept of "aliases" or "defined sets"

- The defined sets are **reusable definitions** across several ACLs. Each category could be modelled in YANG as a list of parameters related to the class it represents. The following sets can be considered:
  - **Prefix sets**: Used to create lists of IPv4 or IPv6 prefixes
  - **Protocol sets**: Used to create a list of protocols
  - **Port number sets**: Used to create lists of port values (TCP, UDP or any other transport protocol that makes uses of port numbers)
  - **ICMP sets**: lists of ICMP-based filters. This applies only when the protocol is set to ICMP or ICMPv6 in the “protocol” match

Suboptimal Handling of Flags

- ACLs are used locally in devices but are triggered by other tools such as BGP Flow Spec
  - RFC8519 ***does not easily map to the filtering rules*** conveyed in messages triggered by these tools: simplify network operations
- Also, RFC8519 ***does not support fragment handling capability*** for IPv6 but offers a partial support for IPv4 by means of 'flags'

Gap

```
{
       "ietf-access-control-list:acls": {
         "acl": [{
           "name": "tcp-flags-example-with-rate-limit",
           "aces": {
             "ace": [{
               "name": "rate-limit-syn",
               "matches": {
                 "tcp": {
                   "flags-bitmask": {
                     "operator": "match",
                     "bitmask": 2
                   }
                 }
               },
               "actions": {
                 "forwarding": "accept"
               }
             }]
           }
         }]
       }
     }
```

Next Step: Open Questions to the WG

- We would like to work on the YANG module, but we need some guidance from the WG

- How does the Working Group suggest to approach the enhancements?
  - New version of the ACL model, minimizing non backwards compatible changes
  - Augmenting RFC 8519 in a new module. All existing structures are not touched
  - Else?

- ACL Network Model vs. Device Model
  - Separate module for the network module?
  - Is the network module to be worked in ***netmod wg***?

- Questions & Suggestions are welcome

Appendix

IPv6 Fragment Handling

Gap

```
{
       "ietf-access-control-list:acls": {
         "acl": [
           {
             "name": "dns-fragments",
             "type": "ipv6-acl-type",
             "aces": {
               "ace": [
                 {
                   "name": "drop-all-fragments",
                   "matches": {
                     "ipv6": {
                       "fragment": {
                         "operator": "match",
                         "type": "isf"
                       }
                     }
                   },
                   "actions": {
                     "forwarding": "drop"
                   }
                 },
                 {
                   "name": "allow-dns-packets",
                   "matches": {
                     "ipv6": {
                       "destination-ipv6-network": "2001:db8::/32"
                     },
                     "udp": {
                       "destination-port": {
                         "operator": "eq",
                         "port": 53
                       }
                     }
                   },
                   "actions": {
                     "forwarding": "accept"
                   }
                 }
               ]
             }
           }
         ]
       }
     }
```

Rate-Limit Actions

- RFC8519 specifies that forwarding actions can be 'accept' (i.e., accept matching traffic), 'drop' (i.e., drop matching traffic without sending any ICMP error message), or ‘reject' (i.e., drop matching traffic and send an ICMP error message to the source)

- However, there are situations where the matching traffic can be accepted, but with a **rate-limit policy**. Such capability is not currently supported by the ACL model

Gap

```
{
      "ietf-access-control-list:acls": {
        "acl": [{
          "name": "tcp-flags-example-with-rate-limit",
          "aces": {
            "ace": [{
              "name": "rate-limit-syn",
              "matches": {
                "tcp": {
                  "flags-bitmask": {
                    "operator": "match",
                    "bitmask": 2
                  }
                }
              },
              "actions": {
                "forwarding": "accept",
                "rate-limit": "20.00"
              }
            }]
          }
        }]
      }
    }
```