YANG Data Model for Access Control List Configuration
draft-huang-netmod-acl-01

Lisa Huang, yihuan@cisco.com
Alexander Clemm, alex@cisco.com
Andy Bierman, andy@yumaworks.com

11/6/2012
Purpose

- ACL: Access Control Lists
  Used to filter traffic (“Firewall Rules”); major part of device configurations
  No configuration complete without ACLs

- Today predominantly represented through vendor CLI
  Very common function but proprietary today – different flavors, different capabilities
  Geared towards human administrators - error proneness, learning curve

- Why a YANG data model?
  Netconf and YANG are intended for network device configuration
  Make ACL more accessible to automated applications, examples:
    - Dynamic Intrusion Protection Systems
    - Dynamic setup/configuration of services, e.g. temporary firewall rule adjustments for video conferences

- ACL Configuration will benefit from standardization
  Needed both by administrators and by applications
  Should be a standards-track Internet Draft
YANG Data Model for ACL configuration


- Modular and extensible ACL Management Framework

- 5 YANG modules
  
  ACL – items common to ACLs regardless of type – "abstract superclass"
  
  IP ACL, MAC ACL, ARP ACL as initial ACL types
    
    Items specific to ACL types; follow common design pattern
    
    Can be extended to other ACL types in the future
  
  Common datatypes
    
    Required by but not specific to ACL
    
    Could be reused by other modules

- Emphasis on configuration
  
  Very limited statistics
ACL concept

- **ACL**: Access Control List
  
  An ordered set of rules used to filter traffic on a networking device

- **Access Control Entry (ACE)**: a representation of a rule
  
  Left hand side: the matching criteria, or "filter"
  
  Right hand side: the action to take – permit/deny a packet
  
  Note: can generalize ACL with further actions: packet capture, audit logging, ...

- First rule that matches is applied
  
  Most specific rules first to avoid rule shadowing

- **ACLs** are applied against interfaces
  
  Interface refers to ACL (ACL specified independent of interface)
  
  Different interfaces can use different ACLs, or use the same
ACL Types covered in the data model

- **IP ACLs**
  - Filter traffic based on IP information in the Layer 3 header of packets.

- **MAC ACLs**
  - Filter traffic using the information in the Layer 2 header of each packet.

- **ARP ACLs**
  - Filter IP- and non-IP-based traffic by checking the Ethernet type code field in the Layer 2 header.

Each ACL includes only ACEs of its type (no mix and match)

Framework can be extended with additional ACL types
  - Augment ACL YANG module
  - Follow design pattern of other ACL types, leverage common ACL data types
ACL module overview

Conceptual diagram only – for specific parameters and mapping to YANG data nodes refer to the draft
Proposal Summary

• Make YANG Data Model for ACL a standards-track working group item
  ACLs are an important part of device configurations
  Proprietary today
  Enabler for many applications, generally related to security
  Will clearly benefit from standardization

• Rev 01 of draft has already been posted
  Extensible + modular framework
  Includes support for 3 different types of ACL, more can be added
  Covers comprehensive set of parameters;
  feature statements allow for customization and device adaptation

Possible items for discussion
  Inclusion of an RPC to test ACLs
  Support for statistics
Q & A
Thank You
YANG Module Structure

module: acl
  +--rw acls
    |  +--rw acl [name]
    |     |  +--rw name
    |     |  +--rw acl-type
    |     |  +--rw capture-session-id-global?
    |     |  +--rw (enable-match-counter-choices)?
    |     |  +--ro match?
    |
    +--rw port-groups
      |  +--rw port-group [name]
      |     |  +--rw name
      |     |  +--rw groups
      +--rw timerange-group
        |  +--rw timerange-group [name]
        |     |  +--rw name
        |     |  +--rw time-ranges
        +--rw ip-address-groups

Generic ACL aspects, common to each ACL type
Determines which types of ACEs can be inserted
Not configuration related, could be separated
Insertion point for specific ACL types (augmentation hook)
Auxiliary convenience objects to simplify reuse of port groupings and schedule information (could move outside acls container)
YANG module structure (contd.)

module: acl
  +--rw acls
  +--rw acl [name]
   | +--rw acl-ip:afi
   | +--rw acl-ip:ipv6-aces
   | | +--rw acl-ip:ipv6-ace [name]
   | | +--rw acl-ip:name
   | | +--rw (remark-or-ipv6-case)?
   | | | +--:(remark)
   | | | | +--rw acl-ip:acl-remark
   | | | +--:(ipv6-ace)
   | | | +--rw acl-ip:filters
   | | | | +-- filter parameters
   | | | | +--rw acl-ip:actions
   | | | | | +-- action parameters
   | | | | +-- ro acl-ip:match

Indicates IP address type

ACLs can include “comment lines” for human/admin consumption
Included in YANG module to maintain consistency with CLI

“left hand side”

“right hand side”
Not configuration related, could be separated

Generic design pattern that is reflected in every ACL type
All ACL type specifics are in the filter parameters and in the actions
YANG module structure (contd.)

module: acl
   +-rw acls
   |  +-rw acl [name]
   |     |  +-rw acl-ip:afi
   |     |  +-rw acl-ip:ipv4-aces
   |     |     |  +-rw acl-ip:ipv4-ace [name]
   |     |     |     |  +-rw acl-ip:name
   |     |     |     |  +-rw (remark-or-ipv4-case)?
   |     |     |     |     |  +-:(remark)
   |     |     |     |     |     |  +-rw acl-ip:acl-remark
   |     |     |     |     |     |  +-:(ipv4-ace)
   |     |     |     |     |     |     |  +-rw acl-ip:filters
   |     |     |     |     |     |     |     |  +- filter parameters
   |     |     |     |     |     |     |     |     |  +-rw acl-ip:actions
   |     |     |     |     |     |     |     |     |  +- action parameters
   |     |     |     |     |     |     |     |  +- ro acl-ip:match

IPv4
(IPv4 and IPv6 specified in same submodule)

| Indicates IP address type
| ACLs can include “comment lines” for human/admin consumption
| Included in YANG module to maintain consistency with CLI
| “left hand side”
| “right hand side”
| Not configuration related, could be separated

Generic design pattern that is reflected in every ACL type
All ACL type specifics are in the filter parameters and in the actions
YANG module structure (contd.)

module: acl
  +--rw acls
  +--rw acl [name]
    | +--rw acl-mac:mac-aces
    | | +--rw acl-mac:mac-ace [name]
    | | +--rw acl-mac:name
    | | +--rw (remark-or-mac-case)?
    | | | +--:(remark)
    | | | | +--rw acl-mac:remark
    | | | | +--:(mac-ace)
    | | | +--rw acl-mac:filters
    | | | | +-- filter parameters
    | | | | +--rw acl-mac:actions
    | | | | | +-- action parameters
    | | | | +-- ro acl-mac:match

MAC
(separate module)

Generic design pattern that is reflected in every ACL type
All ACL type specifics are in the filter parameters and in the actions
YANG module structure (contd.)

module: acl
  +--rw acls
    +--rw acl [name]
      | +--rw acl-arp:arp-aces
      | | +--rw acl-arp:arp-ace [name]
      | | +--rw acl-arp:name
      | | +--rw (remark-or-arp-case)?
      | | | +--:(remark)
      | | | | +--rw acl-arp:remark
      | | | | +--:(arp-ace)
      | | | | +--rw acl-arp:filters
      | | | | | +-- filter parameters
      | | | | | +--rw acl-arp:actions
      | | | | | | +-- action parameters
      | | | | | | +-- ro acl-arp:match

ARP (separate module)

Generic design pattern that is reflected in every ACL type
All ACL type specifics are in the filter parameters and in the actions
YANG module structure (contd.)

```
module: acl
  +--rw acl-s
    +--rw acl [name]
      | +--rw acl-ip:ipv6-aces
      | | +--rw acl-ip:ipv6-ace [name]
      | | | +--rw acl-ip:name
      | | | +--rw (remark-or-ipv6-case)?
      | | | | +--:(ipv6-ace)
      | | | | | +--rw acl-ip:filters
      | | | | | | +-- rw (source-address-host-group)?
      | | | | | | +-- rw (dest-address-host-group)?
      | | | | | | +-- rw acl-ip:protocol?
      | | | | | | +-- rw acl-ip:enable-capture?
      | | | | | | +-- rw acl-ip:capture-session-id?
      | | | | | | +-- rw acl-ip:fragments?
      | | | | | | +-- rw acl-ip:time-range?
      | | | | | | +-- rw acl-ip:src-ports?
      | | | | | | +-- rw acl-ip:dest-ports?
      | | | | | | +-- ...
      | | | +--rw acl-ip: actions
      | | | | +-- rw acl-ip:action
      | | | | | +-- rw acl-ip:log?
```

IPv6-specific parameters

Common actions but could add IP-specific actions later, such as copy, route
## Types specific to ACL (ACL module)

<table>
<thead>
<tr>
<th>YANG type</th>
<th>base type</th>
</tr>
</thead>
<tbody>
<tr>
<td>acl-comparator</td>
<td>enumeration</td>
</tr>
<tr>
<td>acl-action</td>
<td>enumeration</td>
</tr>
<tr>
<td>acl-remark</td>
<td>string</td>
</tr>
<tr>
<td>acl-type-ref</td>
<td>identityref</td>
</tr>
<tr>
<td>acl-ref</td>
<td>leafref</td>
</tr>
<tr>
<td>port-group-ref</td>
<td>leafref</td>
</tr>
<tr>
<td>ip-address-group-ref</td>
<td>leafref</td>
</tr>
<tr>
<td>time-range-ref</td>
<td>leafref</td>
</tr>
<tr>
<td>weekdays</td>
<td>bits</td>
</tr>
<tr>
<td>acl-name-string</td>
<td>string</td>
</tr>
</tbody>
</table>
### Common types – common module (required but not specific to ACL)

<table>
<thead>
<tr>
<th>YANG type</th>
<th>base type</th>
</tr>
</thead>
<tbody>
<tr>
<td>cos</td>
<td>uint8</td>
</tr>
<tr>
<td>tos</td>
<td>uint8</td>
</tr>
<tr>
<td>precedence</td>
<td>uint8</td>
</tr>
<tr>
<td>tcp-flag-type</td>
<td>enumeration</td>
</tr>
<tr>
<td>ether-type</td>
<td>string</td>
</tr>
<tr>
<td>ip-protocol</td>
<td>uint8</td>
</tr>
<tr>
<td>igmp-code</td>
<td>uint8</td>
</tr>
<tr>
<td>icmp-type</td>
<td>uint32</td>
</tr>
<tr>
<td>icmp-code</td>
<td>uint32</td>
</tr>
<tr>
<td>vlan-identifier</td>
<td>uint16</td>
</tr>
<tr>
<td>time-to-live</td>
<td>uint8</td>
</tr>
</tbody>
</table>
Example

• **ACL Example:**
  Denies TELNET traffic from 14.3.6.234 bound for host 6.5.4.1 from leaving.
  Denies all TFTP traffic bound for TFTP servers.
  Permits all other IP traffic.

• **ACL CLI:**
  ```
  access-list ip iacl
  deny tcp 14.3.6.234 0.0.0.0 host 6.5.4.1 eq 23
  deny udp any any eq tftp
  permit ip any any
  ```
<acl>
  <name>iacl</name>
  <acl-type>ip-acl</acl-type>
  <enable-match-counter>false</enable-match-counter>
  <acl-ip:afi>ipv4</acl-ip:afi>
  <acl-ip:ipv4-aces>
    <acl-ip:ipv4-ace>
      <acl-ip:name>deny10</acl-ip:name>
      <acl-ip:filters>
        <acl-ip:protocol>6</acl-ip:protocol>
        <acl-ip:source-address> 14.3.6.234 </acl-ip:source-address>
        <acl-ip:source-mask>0.0.0.0</acl-ip:source-mask>
        <acl-ip:dest-host-address> 6.5.4.1 </acl-ip:dest-host-address>
        <acl-ip:des-comparator>eq</acl-ip:des-comparator>
        <acl-ip:des-port>23</acl-ip:des-port>
      </acl-ip:filters>
      <acl-ip:actions>
        <acl-ip:action>deny</acl-ip:action>
      </acl-ip:actions>
    </acl-ip:ipv4-ace>
    ....
  </acl-ip:ipv4-aces>
</acl>
....
XML instantiation (contd.)

....
  <acl-ip:ipv4-ace>
    <acl-ip:name>permit-20</acl-ip:name>
    <acl-ip:filters>
      <acl-ip:protocol>17</acl-ip:protocol>
      <acl-ip:ip-source-any/>
      <acl-ip:ip-dest-any/>
      <acl-ip:des-comparator>eq</acl-ip:des-comparator>
      <acl-ip:des-port>69</acl-ip:des-port>
    </acl-ip:filters>
    <acl-ip:actions>
      <acl-ip:action>deny</acl-ip:action>
    </acl-ip:actions>
  </acl-ip:ipv4-ace>

  <acl-ip:ipv4-ace>
    <acl-ip:name>any-30</acl-ip:name>
    <acl-ip:filters>
      <acl-ip:ip-source-any/>
      <acl-ip:ip-dest-any>
    </acl-ip:filters>
    <acl-ip:actions>
      <acl-ip:action>permit</acl-ip:action>
    </acl-ip:actions>
  </acl-ip:ipv4-ace>
</acl-ip:ipv4-aces>
</acl>
</acls>
Differences to draft-huang-netmod-acl-00

• “Streamlined” and simplified the design to become more modular. Specifically, defined common groupings for ACEs (ACE-COMMON) and ACE filters (FILTER-COMMON), that span across different types of ACEs/ACLs and that are now used by the different containers.

• Removed the action to reset counters

• Removed the discriminators to distinguish between different ACL types

• Changed capitalizations of node definitions to lower-case to be more conformant with industry best practices

• Added an ACL config example

• Added a list of open issues

• Some general cleanup