Role-Based Access Control

When a user is authenticated, and group membership is established, each action the user starts to perform must be authorized. Typically, the authorization is done based on rules configured in the AAA data model.

The first step in the implementation of role-based access control role-based access control (RBAC) is to enable the NETCONF Access Control Model (NACM), which is described by the RFC 6536 IETF standard. The authorization procedure first checks the value of */nacm/enable-nacm*. This leaf has a default of true, but if it is set to false, all access is permitted.

The characteristics of RBAC are as follows:

- Authorization is based on rules.
- Each role contains a set of rules: a set of privileges.
- Each group contains a set of users.
- Assign roles to user groups. Roles define which set of privileges are applied to all the users in the group.

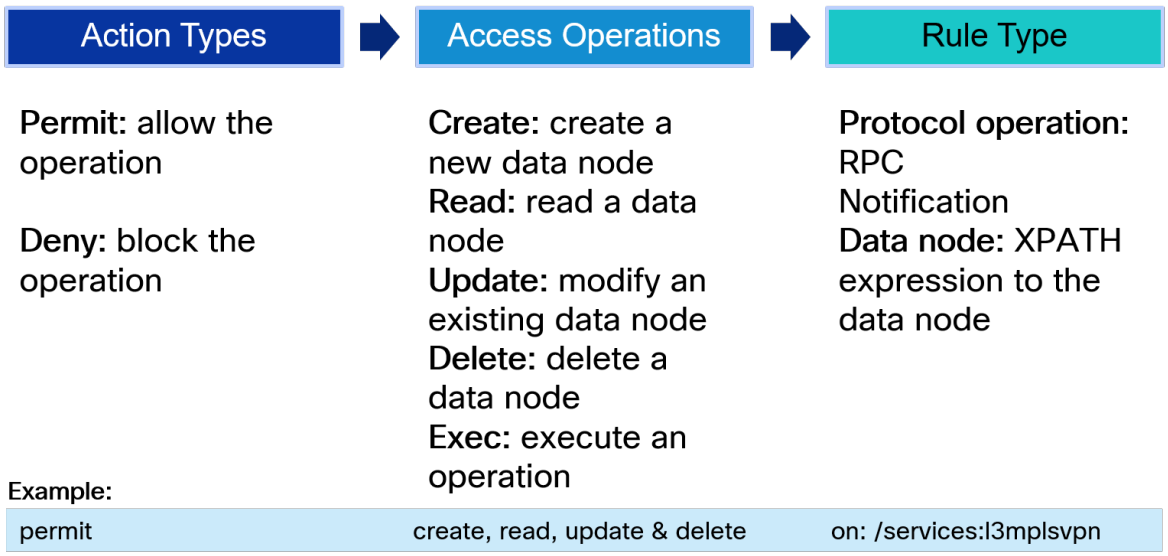
In the following figure, you can see how NACM can be enabled from NSO CLI.

```
admin@ncs(config)# nacm enable-nacm true
admin@ncs(config)# commit
```

Enable NETCONF Access Control Model (NACM)

RBAC Model

The following example illustrates the three objects that permit an operator to create, read, update, and delete any data under the */services:l3mplsvpn* configuration data tree.



The preceding figure illustrates the three sets of objects defining a rule:

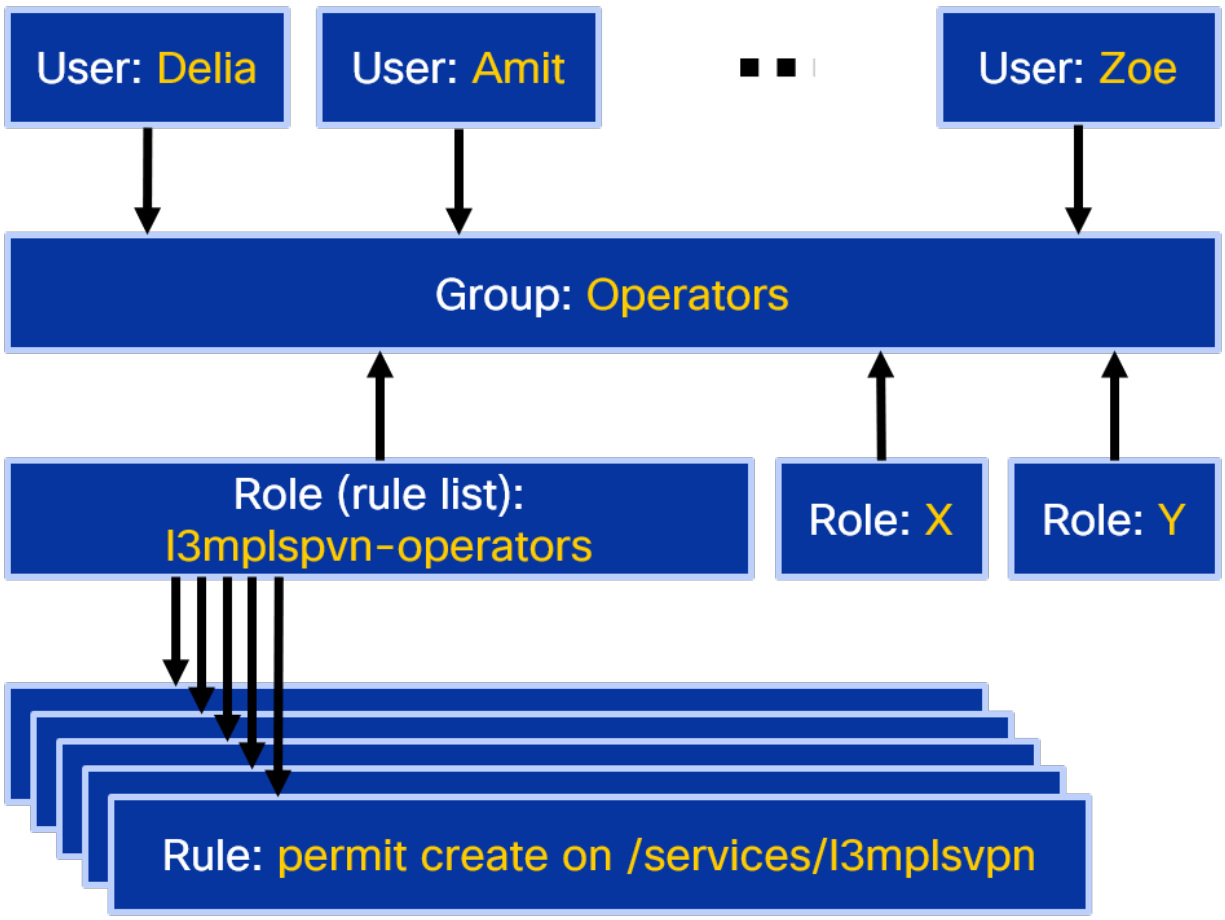
- **Action type options:** permit or deny an action

- **Access operation options:**
  - *Create*: create a new data node
  - *Read*: read a data node
  - *Update*: modify an existing data node
  - *Delete*: delete a data node
  - *Exec*: execute an operation
- **Rule type options:**
  - Protocol operation: control access to remote procedure call (RPC) operations
  - Notification: control access to notifications
  - *Data node*: control access to data nodes in the YANG data tree using Xpath

RBAC Principles

Implement RBAC as listed:

- Authenticate users.
- Map users to groups.
- Assign roles (rule lists) to groups.
- Assign privileges (rules) to roles (rule lists).

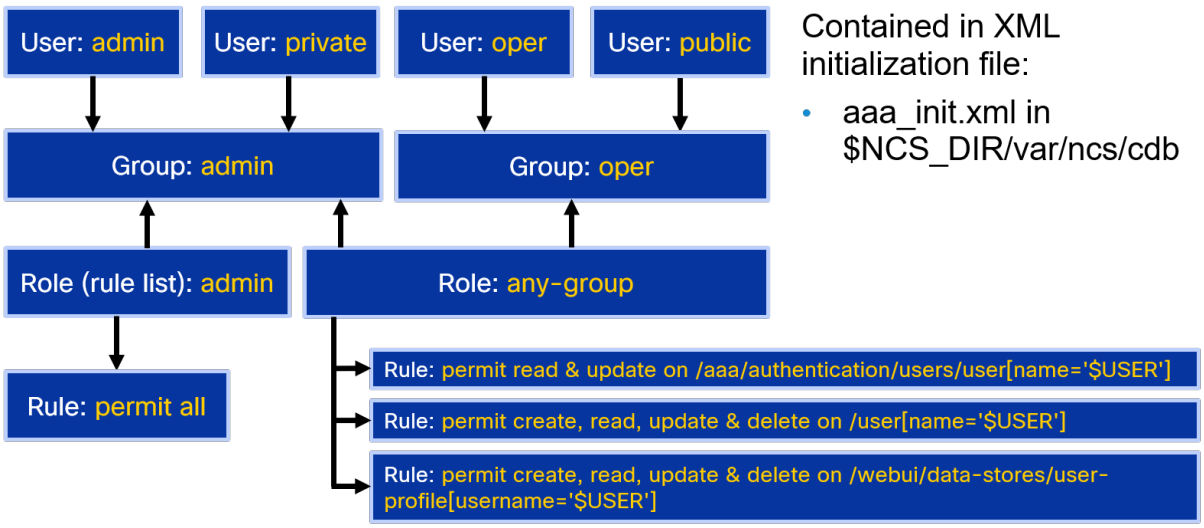


The previous figure illustrates the relationships between the various objects that are part of AAA and RBAC:

- Users are members of one or more groups. Example: users Delia, Amit, and Zoe belong to the group Operators.
- Group Operators are referenced by the rule list *l3mplspvn-operators*.
- Rule list *l3mplspvn-operators* references several individual rules.
- Rules provide access to individual actions.

NACM Default Configuration

The following figure illustrates the default AAA and RBAC configuration after NSO installation. The configuration is loaded from an initialization XML file (*$NCS_DIR/var/ncs/cdb/aaa_init.xml*).



The configuration that would be used for the example figure is shown below:

```
admin@ncs# show running-config nacm
nacm groups group admin
 user-name [ admin private ]
!
nacm groups group oper
 user-name [ bob oper public ]
!
nacm rule-list admin
 group [ admin ]
 rule any-access
  action permit
 !
!
nacm rule-list any-group
group [ * ]
 rule tailf-aaa-authentication
  module-name       tailf-aaa
  path              /aaa/authentication/users/user[name='$USER']
  access-operations read,update
```

```
   action            permit
  !
  rule tailf-aaa-user
   module-name       tailf-aaa
     path              /user[name='$USER']
   access-operations create,read,update,delete
   action            permit
  !
  rule tailf-webui-user
   module-name       tailf-webui
  path              /webui/data-stores/user-profile[username='$USER']
   access-operations create,read,update,delete
   action            permit
  !
 !
```

Default NACM Behavior

When processing rules, the default NACM behavior is listed as follows:

- If no matching rule is found in any of the rule lists in any rule-list entry that matches the user's groups, the data model node for which access is requested is examined for presence of the NACM extensions:
  - If the **nacm:default-deny-all** extension is specified for the data model node, access is denied.
  - If the **nacm:default-deny-write** extension is specified for the data model node, and "create," "update," or "delete" access is requested, access is denied.
- If examination of the NACM extensions did not result in access being denied, the value (permit or deny) of the relevant default leaf is examined:
  - If "read" access is requested, the value of */nacm/read-default* determines whether access is permitted or denied.
  - If "create," "update," or "delete" access is requested, the value of */nacm/write-default* determines whether access is permitted or denied.
  - If "exec" access is requested, the value of */nacm/exec-default* determines whether access is permitted or denied.

The following figure illustrates how to add NACM rules to a custom subtree in a YANG model.

```
module l3mplsvpn{
  namespace "http://cisco.com/example/l3mplsvpn";
  prefix l3mplsvpn;
  import ietf-inet-types  { prefix inet;  }
  import tailf-ncs        { prefix ncs;   }
  import tailf-common     { prefix tailf; }

  import ietf-netconf-acm { prefix nacm;  }

  augment /ncs:services {
    list l3mplsvpn {

      nacm:default-deny-all;

      leaf vpn-name { ... }
      leaf vpn-id { ... }
      leaf customer { ... }
      list link {
        leaf link-name { ... }
        leaf link-id { ... }
        leaf device { ... }
        leaf interface { ... }
        leaf remote-ip { ... }
        leaf ce-ip { ... }
        leaf pe-ip { ... }
        leaf routing-protocol { ... }
        leaf site-name { ... }
      }
    }
  }
}
```

Example: Configuring Access for Operations

The following configuration is used to create a NACM rule for the following:

- Matches users from the group **ncsoper**.
- Allows operators to create and update device configurations.

```
admin@ncs(config)# nacm rule-list oper rule allow-create-update-on-device
path /devices access-operations create,update action permit
```

The following configuration is used to create a NACM rule for the following:

- Allows operators to create, update, and delete data by using the l3vpn service package.

```
admin@ncs(config)# nacm rule-list oper rule allow-l3vpn path /services/l3vpn
access-operations create,update,delete action permit
```

The following configuration is used to create a NACM rule for the following:

- Allows operators to execute **check-sync** and **sync-to** actions against devices in the CDB.

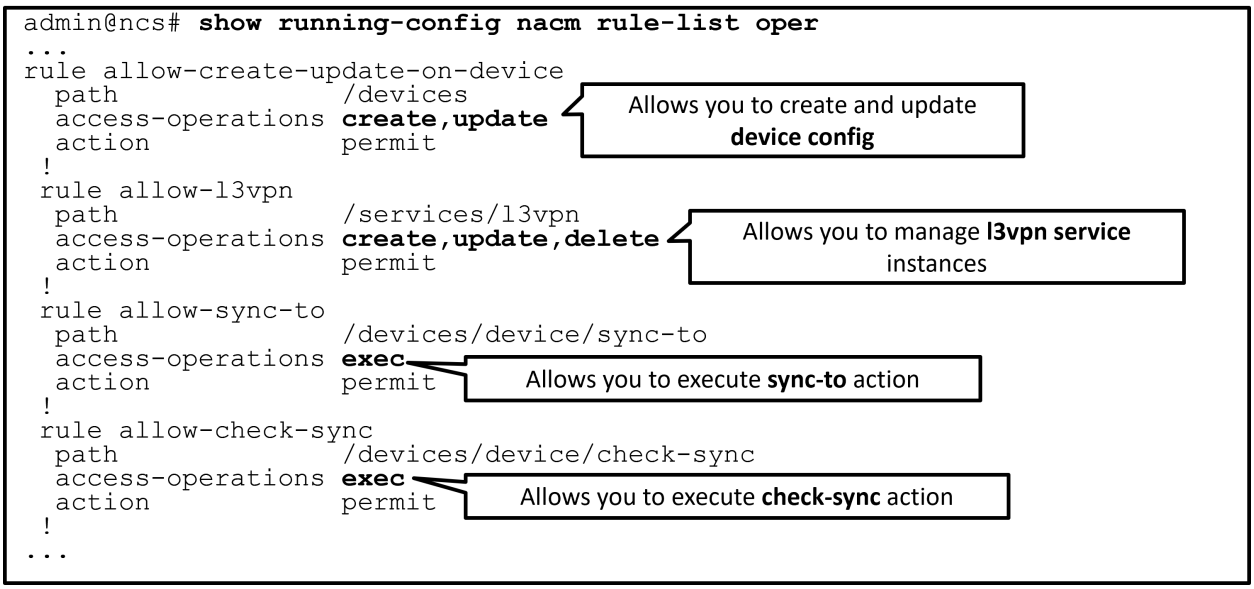
```
admin@ncs(config)# nacm rule-list oper rule allow-check-sync path
/devices/device/check-sync access-operations exec action permit
admin@ncs (config-rule-allow-check-sync)# top
admin@ncs (config)# nacm rule-list oper rule allow-sync-to path
/devices/device/sync-to access-operations exec action permit
admin@ncs (config-rule-allow-sync-to)# top
admin@ncs (config)# commit
```

The full configuration of the role shown in the previous examples is:

```
admin@ncs# show running-config nacm rule-list oper
 group [ ncsoper ]
 rule tailf-aaa-user
  module-name       tailf-aaa
  path              /user[name='$USER']
  access-operations create,read,update,delete
  action            permit
 !
 rule tailf-webui-user
  module-name       tailf-webui
  path              /webui/data-stores/user-profile[username='$USER']
  access-operations create,read,update,delete
  action            permit
 !
 rule tailf-aaa-alluser
  module-name tailf-aaa
  path        /user
  action      deny
 !
 rule tailf-aaa-aaa
  module-name tailf-aaa
  path        /aaa
  action      deny
 !
 rule nacm
  module-name ietf-netconf-acm
  path        /
  action      deny
 !
 rule read-only
  path              /
  access-operations read
  action            permit
 !
 rule allow-create-update-on-device
  path              /devices
  access-operations create,update
  action            permit
 !
 rule allow-l3vpn
  path              /services/l3vpn
  access-operations create,update,delete
```

```
   action            permit
  !
  rule allow-sync-to
   path              /devices/device/sync-to
  access-operations exec
   action            permit
  !
  rule allow-check-sync
   path              /devices/device/check-sync
   access-operations exec
   action            permit
  !
  cmdrule c-logout
   command logout
   action  deny
  !
  cmdrule j-logout
   command "request system logout"
   action  deny
  !
  cmdrule any-command
   action permit
  !
 !
```

A summary of the rules and their short descriptions can be seen in the following figure.



Example: Monitoring Access for Services

The following figure illustrates the creation of rule lists and rules to control access to data trees:

- The group command is used to map the rule list to one or multiple groups of users.
- Each rule list can have multiple rules. The example shows two rules.
- The first rule "permit-read":
  - Matches read access to any data within the /services data tree.

  - Permits read access.

- The second rule "deny-by-default":
  - Matches create, update, write, and exec access to any data within the /services data tree.
  - Denies create, update, write, and exec access.

The following figure shows the output of a configured NACM rule which matches users from the group **monitor** and gives them read access to the /services data tree.