NETCONF Access Control

draft-ietf-netconf-access-control-04
IETF 81, July 2011

Andy Bierman
andy.bierman@brocade.com

Martin Bjorklund
mbj@tail-f.com

Agenda

- Changes to NACM Draft
- Open Issues

Changes to NACM draft (1)

- Introduced rule-lists to group related rules together.
- Moved "module-rule", "rpc-rule", "notification-rule", and "data-rule" into one common "rule", with a choice to select between the four variants.
- Changed "superuser" to "recovery session", and adjusted text throughout document for this change.

Changes to NACM draft (2)

- Clarified behavior of global default NACM parameters, enable-nacm, read-default, write-default, exec-default.
- Clarified when access control is applied during system initialization.

Open Issues

- Write access control rule processing
  - edit or copy-config with no actual changes
  - copy-config running to startup
  - cancel-commit or commit timeout and revert
  - edit-config default-operation = merge/replace
  - commit with edits from multiple sessions

Edit Without Changes

- NACM has default setup
  - no rules, read-default permit, write-permit deny
- @mycfg.xml = get-config source=running
- edit-config target=running config=@mycfg.xml
- No config values are actually be changed but the request is a write on config data
- Does this request succeed of fail with access-denied?

copy-config to startup

- Session A has write access to /foo leaf
- Session A needs to do copy-config from running to startup to save changes to /foo
- Access control must permit this operation
  - Only the nodes that session A actually altered are subject to access control

cancel-commit or revert

- What user/group is used (if any) when checking access control to apply the changes that result from a cancel-commit or revert when the confirmed-commit timeout occurs?
  - none (no access control checked)
- Session that was allowed to alter config may not exist anymore; may not have correct permissions anyway (e.g., create access but no delete access)

edit-config default-operation

- Appears in PDU that a write operation is requested, even for ancestor nodes of the target leaf 'a':
  - <foo>
    <bar>
    <a>some value
    </bar>
    </foo>
- Merge or replace could be interpreted as an edit request. Only default-operation=none is clearly interpreted as a non-edit

commit for multiple edits

- Whichever session issues the commit operation must have correct permissions for all changes to running
  - Only nodes actually altered are checked
- What if NACM rules change during editing of candidate
  - Check edits to candidate may not have same results when applied to running

Proposed Solution

- Change NACM so only altered nodes are checked for access control
  - Currect text says effective operation is used to determine if access control check is needed
  - Effective operation is derived from the PDU only and does not check if the node value is actually changing
  - Change this text to require server to identify actual altered nodes instead
- Problem: lets attacker fish for config
  - no access-denied means config guessed OK