Netconf Protocol: Security Considerations

Wes Hardaker
<hardaker@tislabs.com>

2004.Aug.05

Netconf Authentication and Access Control

- There is inherent access control now:
  - The authentication process should result in an entity whose permissions and capabilities are known to the device. These permissions must be enforced during the NETCONF session. For example, if the native use interface restricts users from changing the network interface configuration, the user should not be able to change this configuration data using NETCONF.

- This implies that all netconf operations/data:
  - MUST be mappable to existing access control specifications
    - Not likely always possible
    - Existing models are CLI based and very different
  - MUST be checked against both:
    - device access control
    - future netconf access control systems
    - accept by one, deny by other = ?
    - Completely standardized access control may never happen

Netconf Authentication and Access Control

- Existing access control systems aren’t network based
  - Can’t say "must encrypt this data in transit"
  - Can’t say "must not touch this except at the device"

Netconf Authentication and Access Control

- Recommendation:
  - Drop the existing requirement
  - "Netconf MUST NOT be implemented without a suitable access control mechanism"

Netconf protocol chaining

- Some operations work on remote datasets
  - copy-config
  - URL based: ftp, http

- Recommendation:
  - Discussion of login credentials and how to pass them
    - Explicit passing
    - Implicit passing
  - copy-config MUST only operate over secure URL transports?

Netconf Locking: DOS

- Not new. Long discussed. Discussed in document.

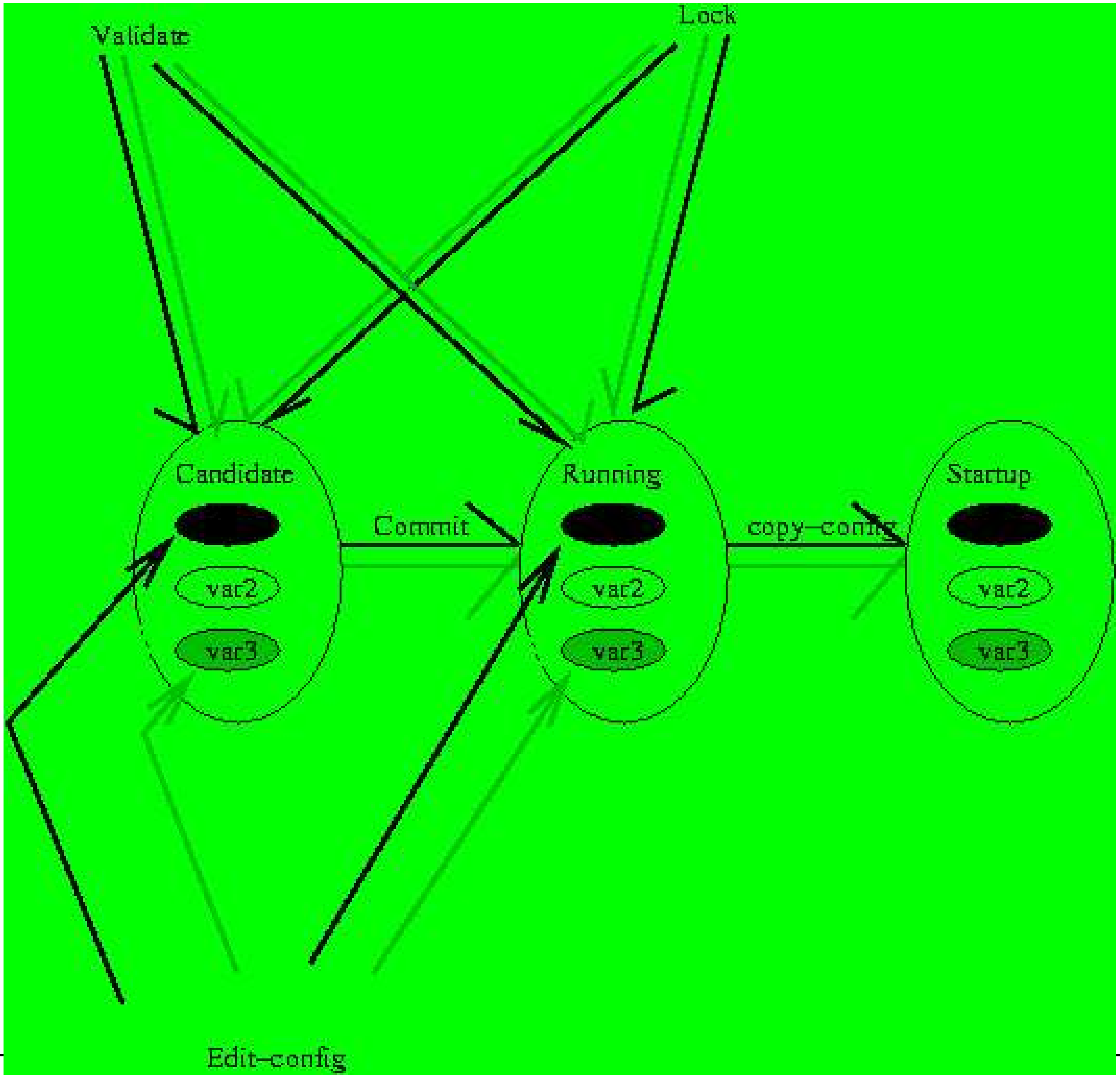
- Global locks mean global lock-outs
  - Grants absolute permission to lock objects otherwise unmanageable by a user
  - Kill-session can be used to remove locks
  - But there is a race condition

- Point:
  - Locks as is add insecurity if granted to peons

Netconf Operations: Micro vs Global

- Netconf Assumptions:
  - Configuration stores are always shared
  - IE, there is not one candidate per user

Netconf Operations: Micro vs Global



Netconf Operations: Micro vs Global

- Consider policy:
  - User P1 can only edit Var1, can’t edit Var2
  - User P2 can edit Var2
- Easy:
  - EDIT-CONFIG must disallow P1 from being able to edit Var2
- Global operations add complexity to the ACM (assume Var2 modified)
  - P1 can not COMMIT() if Var2 is modified
  - P1 can not COPY-CONFIG(running, startup) if Var2 is modified
  - VALIDATE(candidate) must not disclose errors about Var2 to P1
  - P1 can not CONFIRM changes if Var2 is modified
  - P1 can not DISCARDS-CHANGES if Var2 modified
- If P2 modifies Var2, P1 can’t do any global operations

Netconf Operations: Micro vs Global

- The state we’re in:
  - MUST NOT give peon a lock
  - MUST give peon a lock
- No secure state for multi-role enviornments
- Recommendation:
  - "Netconf 1.0 MUST NOT be used in restricted-role environments"
  - OR
  - Fix the problems

Netconf Operations: lock

- canditate config is locked by R
- running config is not

- Can someone else perform a commit?