A Proposal for Access Controls (aka SAN Boxes)

T10/99-278 revision 3

(Apropos T10/99-245r5)

Jim Hafner

IBM

March 8, 2000

Outline of talk

- Brief overview of changes in 99-245r5
- Brief comparison of two "access denied" models
- Outline of new LUN Mapping model and alternatives
- Outline of proxy model
- Other open design issues

Major Changes from 99-245r4

- Major rework of the basic model and proxy model
  - Jointly developed with Ralph Weber (ENDL) and David Chambliss (IBM)
  - Include "LUN Mapping" and "LUN Masking" (see 00-123r0)
- Some name changes (e.g., ACL Key is now called Management Identifier Key)
- Proposed changes to EXTENDED COPY in line with the modified proxy model
- MANAGE ACL no longer can reset to default state (must use the DISABLE ACCESS CONTROLS service action, formerly named RESET AC)
- PTPL (Persist Through Power-loss) is now mandatory

Major Changes from 99-245r4 *(continued)*

- Changes to proposed ASC/ASCQ values
- Removed N_PortID from TransportID for FCP
- TransportID for SPI has reference to glossary of SPI-3 for term "SCSI Address"

Stuff that stayed from rev4

- Configuration of (non-proxy) ACs requires "Management Identifier Key" shared between configuring application client and device
- Proxy ACs still available (revised model)
- Access granted with
  - AccessID identifier (as enrolled by initiator)
  - TransportID identifier (e.g., FC-WWN, now only persistent identifier)

A Tale of Two Models

- Old Model (99-245r4--):
  - all LUs are "visible" (always seen in INQUIRY/REPORT LUNS)
  - "inaccessable" to unauthorized initiators (CHECK CONDITION - ACCESS DENIED)
- New Model (99-245r5++):
  - inaccessable LUs are "invisible", i.e., not seen in INQUIRY/REPORT LUNS (LUN Masking)
  - LUN<->LU map is different for different initiators (LUN Mapping)

Old Access-denied Model

- Advantages:
  - easier dynamic reconfiguration (no host/PAM interlock)
  - global addressing based on consistent LUN<->LU mapping (good for copy services)
  - no changes needed to enable PAM's requirements for "inventory"
  - less intrusion in OS driver stack
    - no change to "LUN discovery"
  - minimal target resources

Old Access-denied Model *(continued)*

- Disadvantages:
  - waste of host resources
    - some large LUN values not accessable to some OSs
  - might not enable "boot off LUN0" requirements
  - not consistent with current VS implementations

New Access-denied Model: LUN Mapping

- Advantages:
  - already implemented in some form by many vendors using only TransportIDs
  - no waste of host resources
  - should work with all OSs without restriction

New Access-denied Model *(continued)*

- Disadvantages:
  - requires more target resources
  - requires tighter interlock between PAM and hosts (in case LUN Map changes)
  - needs additional facilities for PAM-inventory
  - (probably) requires more modifications to OS LUN discovery logic
  - LUNs are no longer global addresses!
  - more difficult for PAM to manage

New Model in Detail

- target creates a LUN Map according to rules
  - for consistency after resets and enrollments
  - specific LUN0 rule
  - LUN Map is "packed":
    - LUN0 first
    - TransportID-accessable LUs next
    - AccessID-accessable LUs next (if enrolled)
  - Proxy-accessable LUs come last (not necessarily packed

New Model in Detail *(continued)*

- LUN Map picture:

LUN Value	**Reason**
0	PAM authorized by TransportID, with specified LUN0 rule
0 m	PAM authorized by TransportID
m+1 n	PAM authorized by AccessID, after enrollment
>n	Via Proxy request

New Model in Detail *(continued)*

- "Access Controls Coordinator":
  - new entity in an SMU
  - handles all access control commands (at LUN0)
  - enforces access controls
  - manages LUN Map per initiator
  - responsibility encompasses all LUs in the device and all ports (like the task manager)
  - facilitates PAM inventory
  - manages iLUNs (internal LUNs)

New Model in Detail *(continued)*

- Host has three states:
  - not-enrolled
    - only TransportID LUs in LUN Map (plus Proxy LUs)
  - enrolled
    - all PAM-authorized LUs in LUN Map and accessable
  - de-enrolled
    - all PAM-authorized LUs in LUN Map
    - AccessID-authorized LUs inaccessable

New Model in Detail *(continued)*

- PAM/host/target interlock for LUN Map change
  - required only if a LUN "moves" to new LU; "adds" and "deletes" not a problem
  - in TransportID range for legacy systems and LUN0 boot
    - required PAM/host interlock (e.g., PAM tells host to reboot)
    - rare?
  - in AccessID range
    - change causes transition to "not-enrolled" state
    - host detects state change, re-enrolls, rediscovers LUN Map, bookkeeps new state

Proxy Model

- Initiator (with access) requests Access Controls Coordinator assign a Proxy Token to a specific LU
    - Proxy Token is passed on to third parties (e.g., in EXTENDED COPY target descriptor)
- Holder (third party) requests LUN value (new entry in LUN Map) for LU associated with Proxy Token
- Invalidating Proxy Token(s):
    - by initiator (with access) with Proxy Token
    - by initiator (with access) - clear all Proxy Tokens
    - by PAM with Proxy Token
    - by PAM - clear all Proxy Tokens
    - target reset (optional) or power cycle

Proxy Model *(continued)*

- Advantages:
  - no global LUN addressing of LUs required
  - Proxy Tokens can be forwarded
  - multiple Proxy Tokens for same LU enables independent access rights
  - each token (even if associated to same LU) can get distinct LUN; copy manager can better separate tasks
  - initiators can share a LU, pass independent Proxy Tokens and not conflict

Proposed Command Set Summary (IN)

- **IN** service actions (Opcode 86h)
  - **REPORT ACL** (mandatory)
    - for PAM to get current state (including outstanding Proxy Tokens)
  - **REPORT LU DESCRIPTIONS** (mandatory - TBD)
    - for PAM to get inventory data (iLUN list, READ CAPACITY, IDENTIFIER, etc)
  - **REPORT LUN MAP** (optional)
    - for host to get LUN->iLUN map
  - **REQUEST PROXY TOKEN** (optional)
    - for host to get Proxy Token for third party functions

Proposed Command Set Summary (OUT)

- **OUT** service actions (Opcode 87h)
  - **MANAGE ACL** (mandatory)
    - for PAM to manage ACL data
  - **DISABLE ACCESS CONTROLS** (mandatory)
    - for PAM to shut down all ACLs (factory default)
  - **ACCESS ID ENROLL** (mandatory)
  - **CANCEL ENROLLMENT** (mandatory)
    - for host to gain access and release access to LUs by AccessID
  - **REVOKE PROXY TOKEN** (optional)
  - **REVOKE ALL PROXY TOKENS** (optional)
    - for host to invalidate one or all Proxy Tokens
  - **ASSIGN PROXY LUN** (optional)
  - **RELEASE PROXY LUN** (optional)
    - for host to create and remove LUN entry for Proxy Token

ASC/ASCQ Summary

AS	ASCQ	Name	Function
20h	01h	ACCESS DENIED - ENROLLMENT CONFLICT	An enrolled or de-enrolled Initiator issues an ACCESS ID ENROLL service action with different AccessID
20h	02h	ACCESS DENIED - INITIATOR DE-ENROLLED	A de-enrolled initiator sends a restricted command to an AccessID-accessible logical unit
20h	03h	ACCESS DENIED - NO ACCESS RIGHTS	A not-enrolled initiator sends an ACCESS ID ENROLL service action and given AccessID has no access rights in the ACL data
20h	04h	ACCESS DENIED - INVALID MGMT ID KEY	The Management Identifier Key value does not match the value maintained by the access controls coordinator
20h	05h	ACCESS DENIED - INVALID LU IDENTIFIER	The LUN or ILUN does not correspond to an accessible logical unit
20h	06h *	ACDESS DENIED - INVALID PROXY TOKEN	The Proxy Token is not valid; it does not correspond to a logical unit
55h	05h	INSUFFICIENT ACCESS CONTROL RESOURCES	The device server has exhausted its resources for access controls

Open Questions

- Who owns LUN Map?
  - revision 6 will (almost surely) have PAM owning map
- Do we need/want INQUIRY bits?
- Do we need tighter PAM/host/target interlock?
- Access controls on sublogical units (e.g., elements in SMC or Object Groups in OSD)
- **How do we enable "override" of Management Identifier Key?**
  - concrete and specific suggestions are welcome

LUN Map Owner Options

- current: target ownership subject to rules (packing)
- alternative: PAM ownership
  - advantages
    - More like current implementations
    - less likely to create LUN "moves"
  - disadvantages
    - PAM configuration conflicts more likely
      - target will need rule to handle runtime conflicts
      - target may need "report conflict" capability
    - "no gaps" rule may not be possible

Other Design Points

- INQUIRY bit or bits?
  - "there is Access Controls Coordinator here"
  - "you see this LU because you're privileged"
- Tighter PAM/host/target LUN Map change interlock?
  - some alternatives:
    - if LUN "moves", put CHECK CONDITION state until cleared by specific host action
    - target refuses configuration command from PAM if causes a "move LUN" for a "connected initiator"
      - overrideable by PAM
      - (only useful if "target owns map")

Override Key Options

- unvalidated service action
- vendor-specific
- "state machine" - perhaps requiring physical access
- "private data" - available only to
  - initiator with access (e.g., serial number)
  - human with physical access (e.g., key on box)
- "fingerprints"

Contacts

- **Details**: `ftp://ftp.t10.org/t10/document.99/99-245r5.pdf`
- **e-mail**: `hafner@almaden.ibm.com`
- **phone**: 408-927-1892
- **fax**: 408-927-4182