Command Security via SAs
Goals

⭐ Per-Command Security

⭐ Fine-grained Reservations and/or Access Controls
  ✩ Tied to TBD Entities
  ✩ Inside Application Client
  ✩ Greater Command-Access Flexibility

⭐ Consideration for OS Performance
Securing Commands

(Comparison of Approaches)

07-069
Initiator

CDB
Capability & ICV

Target

122 bytes

07-149
Initiator

CDB
ID

ICV

Target

8 bytes

64 bytes
ID Options

(ICV is ICV …)

✔ SA Identifier
  ✽ AC_SAI + DS_SAI

✔ OS-Specific
  ✔ Setup as Synonym for SA ID
    (during SA Creation, i.e., validated)

✔ Tied to Program Running on OS
  + (Windows) Process ID
  + (Windows) Image Count
SA Extensions
(Extensions to SA Creation)

- Authentication Required (usage based)
- Synonym Setup

🌟 Commands Controls
→ Allowed When SA ID Present
→ Allowed for Others
✓ Allowed Lists Checked Against Permissions for Authenticated SA Creator
Command Controls
(Preliminary Format Ideas)

- Allowed Bit Mask (1 bit for each OP code)
- Exceptions Descriptors
  - Service Actions
  - Mode Page Codes
  - Log Page Codes
  - Diagnostic Page Codes
  - Reservations Modes
  - LBA Ranges
  - ...
Command Controls
(Preliminary Format Ideas)

- Prohibit All MAINTENANCE OUT except SET IDENTIFYING INFORMATION
- Allow All MODE SELECT(10) except Control Mode Page
- Prohibit All Reservations except All Registrants

→ Is the Allow/Prohibit Model Flexible Enough?
Capabilities Too?

☆ Could SA Extension be Capability?
(instead of bulky bit/exceptions format)

★ Somehow push to 1 Authentication
   (the Security Manager one)

★ I_T Nexus ID?
★ ICV?
Two Usage Models
(Good Reasons for Each)

Initiator → Target

- Usage validation in Target
- Decentralized Security
- More Smarts in Target
- Small Configurations

Initiator → Security Manager → Target

- Usage validation in SM
- Centralized Security
- Less Smarts in Target
- Large Configurations
Help!