

# ENDL TEXAS

Date: 15 November 2007  
To: T10 Technical Committee  
From: Ralph O. Weber  
Subject: OSD-2 Security Enhancements

## Introduction

The SNIA OSD TWG has requested the following security enhancements in OSD-2:

- Addition of user object range capabilities
- Limiting attributes accesses to a single attributes page
- Definition of a boot epoch capability field that is compared with a Root Policy/Security attribute

Since the data structures affected by these changes overlap, no effort has been made to carefully identify them in the proposed changes.

## Revision History

- r0 Initial revision
- r1 Incorporate changes requested by the SNIA OSD TWG
- r2 Incorporated corrections requested by Daniel Messinger
- r3 Incorporated corrections requested by the September CAP working group
- r4 Add value wrapping rules for Boot Epoch
- r5 Incorporated corrections requested by the November CAP working group

Differences between r5 and r5 are indicated by change bars.

Unless otherwise indicated additions are shown in **blue**, deletions in **red-strikethrough**, and comments in **green**.

## Proposed Changes in OSD-2 r01

### 4.9.2.2 Capability format

#### 4.9.2.2.1 Introduction

A capability (see table 9) is included in a CDB to enable the device server to verify that the sender is allowed to perform the command functions (see 3.1.10) described by the CDB.

**Table 9 — Capability format**

| Bit<br>Byte                  | 7  | 6 | 5 | 4 | 3   | 2 | 1 | 0 |
|------------------------------|--|---|---|---|---|---|---|---|
| 0                            | Reserved   |   |   |   | CAPABILITY FORMAT ( <del>2h</del> <del>4h</del> ) |   |   |   |
|                              | ⋮ <span style="color: green;">{{no changes between byte 0 and byte 49}}</span> |   |   |   |   |   |   |   |
| 49                           | PERMISSIONS BIT MASK   |   |   |   |   |   |   |   |
| 53                           |  |   |   |   |   |   |   |   |
| 54                           | Reserved   |   |   |   |   |   |   |   |
| 55                           | OBJECT DESCRIPTOR TYPE   |   |   |   | Reserved  |   |   |   |
| 56                           | (MSB)  |   |   |   |   |   |   |   |
| 59                           | ALLOWED ATTRIBUTES ACCESS  |   |   |   |   |   |   |   |
|                              | (LSB)  |   |   |   |   |   |   |   |
| <del>60</del> <del>56</del>  | OBJECT DESCRIPTOR  |   |   |   |   |   |   |   |
| <del>103</del> <del>79</del> |  |   |   |   |   |   |   |   |

{{Note: The ALLOWED ATTRIBUTES ACCESS field (4 bytes) has been added to the capability format and the object descriptor field has increased in size from 24 bytes to 44 bytes, thus increasing the total capability size by 24 bytes. This size increase also applies to the CDB.}}

The CAPABILITY FORMAT field (see table 10) specifies the format of the capability. If capabilities are coordinated with the security manager, the capability format also is the credential format. The policy/storage manager shall set the CAPABILITY FORMAT field to ~~2h~~ ~~4h~~ (i.e., the format defined by this standard).

**Table 10 — Capability format values**

| Value                            | Description   |
|----------------------------------|---|
| 0h                               | No capability   |
| 1h                               | <del>The format defined by this standard</del> Obsolete |
| 2h                               | The format defined by this standard                     |
| <del>2h</del> <del>3h</del> - Fh | Reserved  |

If the CAPABILITY FORMAT field contains ~~2h~~ ~~4h~~, the device server shall verify that the command functions requested by a CDB are permitted by the capability as described in this subclause. The device server may verify that a command function is permitted after other command functions are completed. The device server shall verify that a command function is permitted before any part of the command function is performed. (E.g., the device server may delay verifying that the set attributes command functions specified by a set attributes list are allowed until the requested read command function is completed, but all the capability permissions concerning the setting attributes are to be verified before any attribute values are changed.)

...

The OBJECT DESCRIPTOR TYPE field (see table 14) specifies the format of information that appears in the OBJECT DESCRIPTOR field.

**Table 14 — Object descriptor types**

| Object Descriptor Type | Name                | Description                                   | Reference |
|------------------------|---------------------|---|-----------|
| 0h                     | NONE                | The OBJECT DESCRIPTOR field shall be ignored  |           |
| 1h                     | <del>U/G</del> USER | A single <del>collection or</del> user object | 4.9.2.2.2 |
| 2h                     | PAR                 | A single partition, including partition zero  | 4.9.2.2.3 |
| 3h                     | COL                 | A single collection                           | 4.9.2.2.4 |
| <del>3h</del> 4h - Fh  |                     | Reserved                                      |           |

The ALLOWED ATTRIBUTES ACCESS field (see table x1) places additional restrictions on the attributes that the command is able to access.

**Table x1 — ALLOWED ATTRIBUTES ACCESS field**

| Value           | Description  |
|-----------------|--|
| 0h              | No additional restrictions are placed on attributes accesses.  |
| 1h to FFFF FFEh | The contents of the Attributes Access attributes page attribute for the partition specified by the ALLOWED PARTITION_ID field in the capability object descriptor specified by the ALLOWED ATTRIBUTES ACCESS field restrict the attributes to which access is allowed as described in 7.1.2.x. |
| FFFF FFFFh      | Reserved   |

If the ALLOWED ATTRIBUTES ACCESS field specifies the attribute number of an attribute that is undefined (see 3.1.50) in the Attributes Access attributes page attribute for the partition specified by the ALLOWED PARTITION\_ID field in the capability object descriptor, then the command shall be terminated with a CHECK CONDITION, with the sense key set to ILLEGAL REQUEST, and the additional sense code shall be set to INVALID FIELD IN CDB

**4.9.2.2.2 U/G USER capability object descriptor**

If the object descriptor type is **U/G USER** (i.e., 1h), the OBJECT DESCRIPTOR field shall have the format shown in table 15, specifying a single ~~collection or~~ user object and a range of bytes with in that user object to which the capability allows access. ~~If the M\_OBJECT permission bit is set to one or the QUERY permission bit is set to one (see 4.9.2.2.1), the U/C capability object descriptor allows access to a single collection and the attributes associated with each user object in the collection.~~

**Table 15 — User object/~~collection~~ descriptor format**

| Bit<br>Byte                 | 7                                  | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----------------------------|------------------------------------|---|---|---|---|---|---|---|
| <del>60</del> <sup>56</sup> | (MSB)                              |   |   |   |   |   |   |   |
| <del>63</del> <sup>59</sup> | POLICY ACCESS TAG                  |   |   |   |   |   |   |   |
|                             | (LSB)                              |   |   |   |   |   |   |   |
| 64                          | (MSB)                              |   |   |   |   |   |   |   |
| 65                          | BOOT EPOCH                         |   |   |   |   |   |   |   |
|                             | (LSB)                              |   |   |   |   |   |   |   |
| 66                          | Reserved                           |   |   |   |   |   |   |   |
| 71                          | Reserved                           |   |   |   |   |   |   |   |
| <del>72</del> <sup>60</sup> | (MSB)                              |   |   |   |   |   |   |   |
| <del>79</del> <sup>67</sup> | ALLOWED PARTITION_ID               |   |   |   |   |   |   |   |
|                             | (LSB)                              |   |   |   |   |   |   |   |
| <del>80</del> <sup>68</sup> | (MSB)                              |   |   |   |   |   |   |   |
| <del>87</del> <sup>75</sup> | ALLOWED USER_OBJECT_ID             |   |   |   |   |   |   |   |
|                             | (LSB)                              |   |   |   |   |   |   |   |
| 88                          | (MSB)                              |   |   |   |   |   |   |   |
| 95                          | ALLOWED RANGE LENGTH               |   |   |   |   |   |   |   |
|                             | (LSB)                              |   |   |   |   |   |   |   |
| 96                          | (MSB)                              |   |   |   |   |   |   |   |
| 103                         | ALLOWED RANGE STARTING BYTE OFFSET |   |   |   |   |   |   |   |
|                             | (LSB)                              |   |   |   |   |   |   |   |
| <del>76</del>               | Reserved                           |   |   |   |   |   |   |   |
| <del>79</del>               | Reserved                           |   |   |   |   |   |   |   |

If the POLICY ACCESS TAG field contains a value other than zero, the policy access tag attribute ...

*{{Note: There is a blank line after table 16 that needs to be removed.}}*

If the BOOT EPOCH field contains zero or the boot epoch attribute in the Root Policy/Security attribute page (see 7.1.2.20) contains zero, the contents of the BOOT EPOCH field shall be ignored. If the non-zero values in the BOOT EPOCH field and the boot epoch attribute in the Root Policy/Security attribute page do not match, then the command shall be terminated with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set to INVALID FIELD IN CDB.

The ALLOWED PARTITION\_ID field specifies ...

The ALLOWED USER\_OBJECT\_ID field specifies the ~~Collection\_Object\_ID (see 4.6.6)~~ or User\_Object\_ID (see 4.6.5) of the OSD object to which the capability allows access. The command shall be terminated with a CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB, if:

- a) The command is not CREATE, or CREATE AND WRITE, ~~or CREATE COLLECTION~~ and the ALLOWED OBJECT\_ID field contains zero; or
- b) ~~The OBJECT TYPE field contains 40h (i.e., COLLECTION) and the ALLOWED OBJECT\_ID field contents do not match the contents of the CDB COLLECTION\_OBJECT\_ID field or REQUESTED COLLECTION\_OBJECT\_ID field; or~~
- c) The ~~OBJECT TYPE field contains 80h (i.e., USER)~~ and the ALLOWED USER\_OBJECT\_ID field contents do not match the contents of the CDB USER\_OBJECT\_ID field or REQUESTED USER\_OBJECT\_ID field.

The ALLOWED RANGE LENGTH field specifies number of bytes in the range of user object bytes to which the capability allows access.

The ALLOWED RANGE STARTING BYTE OFFSET field specifies the location of the first byte in the range of user object bytes to which the capability allows access relative to the first byte (i.e., byte zero).

The command shall be terminated with a CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB, if any of the following is true:

- a) The range of bytes specified by the CDB LENGTH field and STARTING BYTE ADDRESS field in a CREATE AND WRITE command (see 6.4), READ command (see 6.18), or WRITE command (see 6.28) is not inside the range of bytes specified by the ALLOWED RANGE LENGTH field and ALLOWED RANGE STARTING BYTE OFFSET field;
- b) The range of bytes specified by the CDB LENGTH field in an APPEND command (see 6.2) and the value in the user object logical length attribute in the User Object Information attributes page (see 7.1.2.11) is not inside the range of bytes specified by the ALLOWED RANGE LENGTH field and ALLOWED RANGE STARTING BYTE OFFSET field;
- c) The range of bytes specified by the CDB CLEAR LENGTH field and CLEAR STARTING BYTE ADDRESS field in a CLEAR command (see 6.3) is not inside the range of bytes specified by the ALLOWED RANGE LENGTH field and ALLOWED RANGE STARTING BYTE OFFSET field; or
- d) The range of bytes specified by the CDB PUNCH LENGTH field and PUNCH STARTING BYTE ADDRESS field in a PUNCH command (see 6.19) is not inside the range of bytes specified by the ALLOWED RANGE LENGTH field and ALLOWED RANGE STARTING BYTE OFFSET field.

If the ALLOWED RANGE LENGTH field is set to FFFF FFFF FFFF FFFFh and the ALLOWED RANGE STARTING BYTE OFFSET field is set to zero, then access is allowed to all bytes in the user object.

If the ALLOWED RANGE LENGTH field is set to FFFF FFFF FFFF FFFFh and the ALLOWED RANGE STARTING BYTE OFFSET field is set to a non-zero value, then access is allowed to all bytes from the allowed range starting byte to byte FFFF FFFF FFFF FFFFh. This shall not be considered an error.

**4.9.2.2.3 PAR capability object descriptor**

If the object descriptor type is PAR (i.e., 2h), the OBJECT DESCRIPTOR field shall have the format shown in table 17, specifying a single partition to which the capability allows access. For a LIST COLLECTION command with the M\_OBJECT bit set to one (see 4.9.2.2.1), the PAR capability object descriptor allows access to a single partition and the attributes associated with each collection in the partition. For the LIST command with the M\_OBJECT bit set to one, the PAR capability object descriptor allows access to:

- a) The root object and the attributes associated with each partition; or
- b) A partition and the attributes associated with each user object in the partition.

**Table 17 — Partition descriptor format**

| Bit<br>Byte                  | 7                    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |       |
|------------------------------|----------------------|---|---|---|---|---|---|---|-------|
| <del>60</del> <sup>56</sup>  | (MSB)                |   |   |   |   |   |   |   |       |
| <del>63</del> <sup>59</sup>  | POLICY ACCESS TAG    |   |   |   |   |   |   |   | (LSB) |
| <del>64</del> <sup>64</sup>  | (MSB)                |   |   |   |   |   |   |   |       |
| <del>65</del> <sup>65</sup>  | BOOT EPOCH           |   |   |   |   |   |   |   | (LSB) |
| <del>66</del> <sup>66</sup>  | Reserved             |   |   |   |   |   |   |   |       |
| <del>71</del> <sup>71</sup>  | Reserved             |   |   |   |   |   |   |   |       |
| <del>72</del> <sup>60</sup>  | (MSB)                |   |   |   |   |   |   |   |       |
| <del>79</del> <sup>67</sup>  | ALLOWED PARTITION_ID |   |   |   |   |   |   |   | (LSB) |
| <del>80</del> <sup>68</sup>  | Reserved             |   |   |   |   |   |   |   |       |
| <del>103</del> <sup>79</sup> | Reserved             |   |   |   |   |   |   |   |       |

The POLICY ACCESS TAG field and BOOT EPOCH field are is described in 4.9.2.2.2.

...

{{No other changes in 4.9.2.2.3.}}

#### 4.9.2.2.4 COL capability object descriptor

If the object descriptor type is COL (i.e., 3h), the OBJECT\_DESCRIPTOR field shall have the format shown in table x2, specifying a single collection or to which the capability allows access. If the M\_OBJECT permission bit is set to one or the QUERY permission bit is set to one (see 4.9.2.2.1), the COL capability object descriptor allows access to a single collection and the attributes associated with each user object in the collection.

**Table x2 — Collection descriptor format**

| Bit<br>Byte | 7     | 6                            | 5 | 4 | 3 | 2 | 1 | 0     |
|-------------|-------|------------------------------|---|---|---|---|---|-------|
| 60          | (MSB) | POLICY ACCESS TAG            |   |   |   |   |   | (LSB) |
| 63          |       |                              |   |   |   |   |   |       |
| 64          | (MSB) | BOOT EPOCH                   |   |   |   |   |   | (LSB) |
| 65          |       |                              |   |   |   |   |   |       |
| 66          |       | Reserved                     |   |   |   |   |   |       |
| 71          |       |                              |   |   |   |   |   |       |
| 72          | (MSB) | ALLOWED PARTITION_ID         |   |   |   |   |   | (LSB) |
| 79          |       |                              |   |   |   |   |   |       |
| 80          | (MSB) | ALLOWED COLLECTION_OBJECT_ID |   |   |   |   |   | (LSB) |
| 87          |       |                              |   |   |   |   |   |       |
| 88          |       | Reserved                     |   |   |   |   |   |       |
| 103         |       |                              |   |   |   |   |   |       |

The BOOT EPOCH field, POLICY ACCESS TAG field, and ALLOWED PARTITION\_ID field are described in 4.9.2.2.2.

The ALLOWED COLLECTION\_OBJECT\_ID field specifies the Collection\_Object\_ID (see 4.6.6) of the collection to which the capability allows access. The command shall be terminated with a CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB, if:

- a) The command is not CREATE COLLECTION and the ALLOWED COLLECTION\_OBJECT\_ID field contains zero; or
- b) The ALLOWED COLLECTION\_OBJECT\_ID field contents do not match the contents of the CDB COLLECTION\_OBJECT\_ID field or REQUESTED COLLECTION\_OBJECT\_ID field.

4.9.2.3 Capabilities and commands allowed

...

Table 18 — Commands allowed by specific capability field values (Sheet 1 of 2)

| Commands allowed and CDB fields whose contents are restricted by capability field contents, if any | Capability Field values that allow a command |                                     |                        |
|--|--|-------------------------------------|------------------------|
|  | Object Type Name                             | Permission Bits That Are Set To One | Object Descriptor Name |
| An APPEND command  | USER   | APPEND                              | <del>U/G</del><br>USER |
| A CLEAR command  | USER   | WRITE                               | <del>U/G</del><br>USER |
| A CREATE command   | USER   | CREATE                              | <del>U/G</del><br>USER |
| A CREATE AND WRITE command   | USER   | CREATE and WRITE                    | <del>U/G</del><br>USER |
| A CREATE COLLECTION command  | COLLECTION                                   | CREATE                              | <del>U/G</del><br>COL  |
| ...  | ...  | ...                                 | ...                    |
| A FLUSH command  | USER   | OBJ_MGMT                            | <del>U/G</del><br>USER |
| A FLUSH COLLECTION command   | COLLECTION                                   | OBJ_MGMT                            | <del>U/G</del><br>COL  |
| ...  | ...  | ...                                 | ...                    |
| A GET ATTRIBUTES command addressed to a user object  | USER   | see table 19                        | <del>U/G</del><br>USER |
| A GET ATTRIBUTES command addressed to a collection   | COLLECTION                                   | see table 19                        | <del>U/G</del><br>COL  |
| ...  | ...  | ...                                 | ...                    |
| A GET MEMBER ATTRIBUTES command addressed to a collection  | COLLECTION                                   | see table 19                        | <del>U/G</del><br>COL  |
| ...  | ...  | ...                                 | ...                    |
| A LIST COLLECTION command addressed to a collection with the LIST_ATTR bit to be set to zero       | COLLECTION                                   | READ                                | <del>U/G</del><br>COL  |
| A LIST COLLECTION command addressed to a collection  | COLLECTION                                   | READ and M_OBJECT                   | <del>U/G</del><br>COL  |

Combinations of OBJECT TYPE field, PERMISSION BITS field, and OBJECT DESCRIPTOR TYPE field values not shown in this table and table 19 are reserved.  
The capability fields not shown in this table may place additional limits on the objects that are allowed to be accessed.



**Table 18 — Commands allowed by specific capability field values (Sheet 2 of 2)**

| Commands allowed and CDB fields whose contents are restricted by capability field contents, if any  | Capability Field values that allow a command |                                     |                        |
|---|--|-------------------------------------|------------------------|
|   | Object Type Name                             | Permission Bits That Are Set To One | Object Descriptor Name |
| ...   | ...  | ...                                 | ...                    |
| A PERFORM TASK MANAGEMENT command with function code of ABORT TASK or QUERY TASK addressed to a user object   | USER   | DEV_MGMT                            | <del>U/G</del><br>USER |
| A PERFORM TASK MANAGEMENT command with function code of ABORT TASK or QUERY TASK addressed to a collection  | COLLECTION                                   | DEV_MGMT                            | <del>U/G</del><br>COL  |
| ...   | ...  | ...                                 | ...                    |
| A PUNCH command   | USER   | WRITE                               | <del>U/G</del><br>USER |
| A QUERY command addressed to a collection   | COLLECTION                                   | QUERY                               | <del>U/G</del><br>COL  |
| A READ command  | USER   | READ                                | <del>U/G</del><br>USER |
| A REMOVE command  | USER   | REMOVE                              | <del>U/G</del><br>USER |
| A REMOVE COLLECTION   | COLLECTION                                   | REMOVE                              | <del>U/G</del><br>COL  |
| A REMOVE MEMBER OBJECTS command addressed to a collection   | COLLECTION                                   | REMOVE and M_OBJECT                 | <del>U/G</del><br>COL  |
| ...   | ...  | ...                                 | ...                    |
| A SET ATTRIBUTES command addressed to a user object   | USER   | see table 19                        | <del>U/G</del><br>USER |
| A SET ATTRIBUTES command addressed to a collection  | COLLECTION                                   | see table 19                        | <del>U/G</del><br>COL  |
| ...   | ...  | ...                                 | ...                    |
| A SET MEMBER ATTRIBUTES command addressed to a collection   | COLLECTION                                   | see table 19                        | <del>U/G</del><br>COL  |
| ...   | ...  | ...                                 | ...                    |
| A WRITE command   | USER   | WRITE                               | <del>U/G</del><br>USER |
| Combinations of OBJECT TYPE field, PERMISSION BITS field, and OBJECT DESCRIPTOR TYPE field values not shown in this table and table 19 are reserved.<br>The capability fields not shown in this table may place additional limits on the objects that are allowed to be accessed. |  |                                     |                        |

...

## 5.2 Fields commonly used in OSD commands

### 5.2.1 Overview

OSD commands employ the basic CDB structure shown in 5.1. Within the basic CDB structure, the OSD service action specific fields are organized so that the same field is in the same location in all OSD CDBs (see table 44). OSD service action specific fields that are unique to a small number of CDBs are not shown in this subclause.

**Table 44 — OSD service action specific fields**

| Bit<br>Byte                                    | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--|---|---|---|---|---|---|---|---|
|  | ⋮ <span style="color: green;">{{no changes between byte 10 and byte 52}}</span> |   |   |   |   |   |   |   |
| 52<br>79                                       | Get and set attributes parameters <sup>a</sup>                                  |   |   |   |   |   |   |   |
| 80<br>183<br><del>159</del>                    | Capability (see 4.9.2.2)  |   |   |   |   |   |   |   |
| 184<br><del>160</del><br>223<br><del>199</del> | Security parameters (see 5.2.6)   |   |   |   |   |   |   |   |
| <sup>a</sup> See 5.2.2.                        |   |   |   |   |   |   |   |   |

{{The changes shown in table 44 must be made in all CDB definitions in clause 6.}}

...

### 7.1.2.1 Attributes pages overview

The attributes pages defined by this standard are shown in table 110.

**Table 110 — Attributes pages defined by this standard**

| Page Number   | Page Name                             | Page Format Defined | Support Requirements | Reference |
|---------------|---------------------------------------|---------------------|----------------------|-----------|
| ...           | ...                                   | ...                 | ...                  | ...       |
| P+3h          | Partition Timestamps                  | Yes                 | Mandatory            | 7.1.2.16  |
| P+4h          | <del>Reserved</del> Attributes Access | No                  | Mandatory            | 7.1.2.x   |
| P+5h          | Partition Policy/Security             | Yes                 | Mandatory            | 7.1.2.21  |
| P+6h to P+7Fh | Reserved                              |                     |                      |           |
| ...           | ...                                   | ...                 | ...                  | ...       |

...

**7.1.2.19 Collections attributes page**

{{Subclause 7.1.2.19 is shown only to indicate the position of 7.1.2.x. No changes are proposed in 7.1.2.19.}}

**7.1.2.x Attributes Access attributes page**

{{All of 7.1.2.x is new. The use of additions/deletions markups is suspended.}}

The Attributes Access attributes page (P+4h) shall contain the attributes listed in table x3.

**Table x3 — Attributes Access attributes page contents**

| Attribute Number | Length (bytes) | Attribute                 | Application Client Settable | OSD Logical Unit Provided |
|------------------|----------------|---------------------------|-----------------------------|---------------------------|
| 0h               | 40             | Page identification       | No                          | Yes                       |
| 1h to FFFF FFFEh | 0 or n         | Allowed attributes access | Yes                         | No                        |

The page identification attribute (number 0h) shall have the format described in 7.1.2.2 with the VENDOR IDENTIFICATION field containing the ASCII characters "INCITS" and the ATTRIBUTES PAGE IDENTIFICATION field containing the ASCII characters "T10 Attributes Access".

Each allowed attributes access attribute (1h to FFFF FFFEh) is:

- a) An undefined (i.e., zero length) attribute (see 3.1.50); or
- b) A defined attribute (see 3.1.15) that contains a list of attributes access descriptors (see table x4) each of which indicates an attribute or attributes to which access is allowed.

**Table x4 — Allowed attributes access attribute format**

| Bit Byte | 7  | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|--|---|---|---|---|---|---|---|
|          | Attributes access descriptors                      |   |   |   |   |   |   |   |
| 0        | Attributes access descriptor, first (see table x5) |   |   |   |   |   |   |   |
| 7        |  |   |   |   |   |   |   |   |
|          | ⋮  |   |   |   |   |   |   |   |
| n-7      | Attributes access descriptor, last (see table x5)  |   |   |   |   |   |   |   |
| n        |  |   |   |   |   |   |   |   |

Each attributes access descriptor (see table x5) indicates an attribute or set of attributes to which this descriptor allows access.

**Table x5 — Attributes access descriptor**

| Bit<br>Byte | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0                |             |
|-------------|-------------|---|---|---|---|---|---|------------------|-------------|
| 0           | (MSB) _____ |   |   |   |   |   |   | ATTRIBUTES PAGE  | _____ (LSB) |
| 3           |             |   |   |   |   |   |   |                  |             |
| 4           | (MSB) _____ |   |   |   |   |   |   | ATTRIBUTE NUMBER | _____ (LSB) |
| 7           |             |   |   |   |   |   |   |                  |             |

The ATTRIBUTES PAGE field identifies the page number of one attribute or set of attributes to which access is allowed.

The ATTRIBUTE NUMBER field identifies:

- a) The attribute number within the attributes page specified by the ATTRIBUTES PAGE field of the one attribute to which access is allowed; or
- b) The value FFFF FFFFh indicates that access is allowed to all the attributes in the attributes page specified by the ATTRIBUTES PAGE field.

If the ALLOWED ATTRIBUTES ACCESS field in a capability (see 4.9.2.2) specifies the number of a defined attribute in an Attributes Access attributes page and the command attempts to retrieve or set an attribute that is not identified in at least one attributes access descriptor, then the command shall be terminated with a CHECK CONDITION, with the sense key set to ILLEGAL REQUEST, and the additional sense code set as follows:

- a) If the disallowed attribute access is specified in CDB fields, the additional sense code shall be set to INVALID FIELD IN CDB; or
- b) If the disallowed attribute access is specified in the Data-Out Buffer, the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST.

An Attributes Access attributes page allowed access attribute serves only to disallow access to attributes to which access would otherwise be possible and this is accomplished by omitting them from the attributes identified in the attributes access descriptors. The Attributes Access attributes page does not grant access to attributes that a command would not otherwise be able to access.

{{Use of additions/deletions markups resumes here.}}

{{Annex C must be updated to add the Attributes Access attributes page defined by this proposal.}}

### 7.1.2.20 Root Policy/Security attributes page

The Root Policy/Security attributes page (R+5h) shall contain the attributes listed in table 136.

**Table 136 — Root Policy/Security attributes page contents**

| Attribute Number         | Length (bytes) | Attribute                                 | Application Client Settable | OSD Logical Unit Provided |
|--------------------------|----------------|---|-----------------------------|---------------------------|
| 0h                       | 40             | Page identification                       | No                          | Yes                       |
| 1h                       | 1              | Default security method                   | Yes                         | Yes                       |
| 2h                       | 6              | Oldest valid nonce limit                  | No                          | Yes                       |
| 3h                       | 6              | Newest valid nonce limit                  | No                          | Yes                       |
| 4h to 5h                 |                | Reserved                                  | No                          |                           |
| 6h                       | 1              | Partition default security method         | Yes                         | Yes                       |
| 7h                       | 2              | Supported security methods                | No                          | Yes                       |
| 8h                       |                | Reserved                                  | No                          |                           |
| 9h                       | 6              | Adjustable clock                          | Yes                         | Yes                       |
| Ah                       | 2              | Boot epoch                                | Yes                         | Yes                       |
| Ah Bh to 7FFCh           |                | Reserved                                  | No                          |                           |
| 7FFDh                    | 0 or 7         | Master key identifier                     | No                          | Yes                       |
| 7FFEh                    | 0 or 7         | Root key identifier                       | No                          | Yes                       |
| 7FFFh to 7FFF FFFFh      |                | Reserved                                  | No                          |                           |
| 8000 0000h to 8000 000Fh | 1              | Supported integrity check value algorithm | No                          | Yes                       |
| 8000 0010h to 8000 001Fh | 1              | Supported DH group                        | No                          | Yes                       |
| 8000 0020h to FFFF FFFEh |                | Reserved                                  | No                          |                           |

...

When the OSD device is manufactured, the boot epoch attribute (number Ah) should be set to a non-zero value. The processing of any SCSI device condition (e.g., logical unit reset) established in response to an event (see SAM-4) shall cause the boot epoch attribute to be updated as follows:

- a) If the boot epoch attribute is not set to FFFFh, then one shall be added to the boot epoch value; or
- b) If the boot epoch attribute is set to FFFFh, then the boot epoch value shall be set to one.

The boot epoch attribute is compared to the BOOT EPOCH field in capabilities processed as described in 4.9.2.2.

NOTE x1 - The application client may change the boot epoch attribute to any value (see table 136).

{{Annex C must be updated to add the boot epoch attribute defined by this proposal.}}