ENDL
TEXAS

Date: 2 July 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

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 Byte	7	6	5	4	3	2	1	0
0	Reserved				CAPABILITY FORMAT (2h~~1h~~)			
	⋮ {{no changes between byte 0 and byte 49}}							
49	PERMISSIONS BIT MASK							
53								
54	Reserved							
55	(MSB)	ALLOWED ATTRIBUTES PAGE						
58								(LSB)
59~~55~~	OBJECT DESCRIPTOR TYPE				Reserved			
60~~56~~	OBJECT DESCRIPTOR							
95~~79~~								

{{Note: The ALLOWED ATTRIBUTES PAGE field (4 bytes) has been added and the object descriptor field has increased in size from 24 bytes to 36 bytes, thus increasing the total capability size by 16 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 ~~1h~~ (i.e., the format defined by this standard).

Table 10 — Capability format values

Value	Description
0h	No capability
1h	The format defined by previous versions of this standard
2h	The format defined by this standard
~~2h~~ 3h - Fh	Reserved

If the CAPABILITY FORMAT field contains 2h ~~1h~~, the device server shall verify that the command functions requested by a CDB are permitted by the capability as described in this subclause. If the CAPABILITY FORMAT field contains 1h, the device server should verify that the command functions requested by a CDB are permitted by the capability as described in a previous version of this standard. 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 ALLOWED ATTRIBUTES PAGE field constrains which attributes page(s) may be accessed in addition to the constraints placed on attributes access by the PERMISSIONS BIT MASK field. If the ALLOWED ATTRIBUTES PAGE field is set to FFFF FFFFh, then any attributes page may be accessed. Otherwise, any attempt to retrieve or set attributes in an attribute page whose page number does not match the value in the ALLOWED ATTRIBUTES PAGE field shall cause the command to 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 invalid attribute length is in a CDB field, the additional sense code shall be set to INVALID FIELD IN CDB; or
b) If the invalid attribute length is in the Data-Out Buffer, the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST.

…

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	U/C	A single collection or user object	4.9.2.2.2
2h	PAR	A single partition, including partition zero	4.9.2.2.3
3h	RANGE	A range of bytes within a single user object	4.9.2.2.4
~~3h~~ 4h - Fh		Reserved	

4.9.2.2.2 U/C capability object descriptor

If the object descriptor type is U/C (i.e., 1h), the OBJECT DESCRIPTOR field shall have the format shown in table 15, specifying a single collection or user object to which the capability allows access.{{add a space here}} 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 Byte	7	6	5	4	3	2	1	0
~~56~~60	~~(MSB)~~			~~POLICY ACCESS TAG~~				
~~59~~63								~~(LSB)~~
60	(MSB)			BOOT EPOCH				
61								(LSB)
62	(MSB)			POLICY ACCESS TAG				
63								(LSB)
64~~60~~	(MSB)			ALLOWED PARTITION_ID				
71~~67~~								(LSB)
72~~68~~	(MSB)			ALLOWED OBJECT_ID				
79~~75~~								(LSB)
80~~76~~				Reserved				
99~~79~~								

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, the contents of the BOOT EPOCH field shall be ignored. If the non-zero values in the BOOT EPOCH field contains zero 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.

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.}}

…

{{No other changes in 4.9.2.2.2.}}

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 Byte	7	6	5	4	3	2	1	0
~~56~~60	~~(MSB)~~			~~POLICY ACCESS TAG~~				
~~59~~63								~~(LSB)~~
60	(MSB)			BOOT EPOCH				
61								(LSB)
62	(MSB)			POLICY ACCESS TAG				
63								(LSB)
64~~60~~	(MSB)			ALLOWED PARTITION_ID				
71~~67~~								(LSB)
72~~68~~				Reserved				
99~~79~~								

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

…

{{No other changes in 4.9.2.2.3.}}

4.9.2.2.4 RANGE capability object descriptor

If the object descriptor type is RANGE (i.e., 3h), the OBJECT DESCRIPTOR field shall have the format shown in table x1, specifying a range of bytes in a single user object to which the capability allows access.

Table x1 — User object byte range descriptor format

Bit Byte	7	6	5	4	3	2	1	0
60	(MSB)	BOOT EPOCH						
61								(LSB)
62	(MSB)	POLICY ACCESS TAG						
63								(LSB)
64	(MSB)	ALLOWED PARTITION_ID						
71								(LSB)
72	(MSB)	ALLOWED OBJECT_ID						
79								(LSB)
80	(MSB)	ALLOWED RANGE LENGTH						
87								(LSB)
88	(MSB)	ALLOWED RANGE STARTING BYTE OFFSET						
95								(LSB)
96		Reserved						
99								

The BOOT EPOCH field, POLICY ACCESS TAG field, and ALLOWED PARTITION_ID field are described in 4.9.2.2.2.

The ALLOWED OBJECT_ID field specifies the 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 and the ALLOWED OBJECT_ID field contains zero; or
b) The OBJECT TYPE field contains 80h (i.e., USER) and the ALLOWED 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).

If 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, then 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 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, then 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 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 the RANGE capability object descriptor is equivalent to a U/C capability object descriptor (see 4.9.2.2.2) with the same values in the ALLOWED PARTITION_ID field, the ALLOWED OBJECT_ID field, the BOOT EPOCH field, and the POLICY ACCESS TAG 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 a non-zero value, 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.

4.9.2.3 Capabilities and commands allowed

…

Table 18 — Commands allowed by specific capability field values

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	U/C or RANGE
A CREATE command	USER	CREATE	U/C
A CREATE AND WRITE command	USER	CREATE and WRITE	U/C or RANGE
…	…	…	…
A READ command	USER	READ	U/C or RANGE
…	…	…	…
A WRITE command	USER	WRITE	U/C or RANGE
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.			

…

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 **Byte**	7	6	5	4	3	2	1	0
	⋮ {{no changes between byte 10 and byte 52}}							
52	Get and set attributes parameters [a]							
79								
80	Capability (see 4.9.2.2)							
175 ~~159~~								
176 ~~160~~	Security parameters (see 5.2.6)							
215 ~~199~~								

[a] See 5.2.2.

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

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. The processing of any condition (e.g., logical unit reset) established in response to an event (see SAM-4) shall cause one to be added to the previous value of the boot epoch attribute. 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.}}