ENDL
TEXAS

Date: 7 February 2007
To: T10 Technical Committee
From: Ralph O. Weber
Subject: SPC-4: Encapsulated SCSI Commands

Revision History

r1 Incorporated changes requested by January '07 CAP working group.

Changed between r0 and r1 are marked with change bars.

Premise

In IPsec and FC-SP, transport layer security encapsulates frames in a technology called Encapsulated Security Protocol (ESP). The encapsulation adds security related information to frames that may be otherwise unchanged.

Sooner or later (i.e., about now), SCSI command layer security is going to require the encapsulation of SCSI CDBs for similar reasons.

N.B. Security may not be the only reason for encapsulating CDBs. Tunneling and bridging functions have been mentioned as other possible consumers of Encapsulated SCSI Commands (ESC).

Nota Bene

The changes proposed in r1 are the most restrictive that can be imagined. This is based on the belief that relaxing restrictions later is easier than tightening them. The limiting nature of the r1 proposal can be relaxed based on just cause presented in the form of a specific usage requirement example. However, suggestions of the form "wouldn't it be nice if …" most probably will meet with substantial resistance.

Proposed SPC-4 Changes

Reference SPC-4 r08. Blue for new text. ~~Red strikeout~~ for removed text. Green for proposer's notes.

{{New Definition}}

3.1.x Encapsulated SCSI command (ESC): A CDB in which another CDB is encapsulated and information is added to support extra processing (e.g., security features) before and/or after the processing of the encapsulated CDB. See 4.3.4.

{{New Acronym}}

ESC Encapsulated SCSI Command (see 3.1.x)

18484 Preston Road, Suite 102 PMB 178, Dallas, TX 75252 214-912-1373 972-596-2775 Fx ENDL_TX@computer.org

4.3 The Command Descriptor Block (CDB)

4.3.1 CDB usage and structure

A command is communicated by sending a command descriptor block (CDB) to the device server. For several commands, the CDB is accompanied by a list of parameters in the Data-Out Buffer. See the specific commands for detailed information.

If a logical unit validates reserved CDB fields and receives a reserved field within the CDB that is not zero, then the logical unit shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

If a logical unit receives a reserved CDB code value in a field other than the OPERATION CODE field, then the logical unit shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

The fixed length CDB formats are described in 4.3.2. The variable length CDB formats are described in 4.3.3. The ESC CDB format is described in 4.3.4. The CDB fields that are common to most commands are described in 4.3.5. The fields shown in 4.3.2 and 4.3.3 and described in 4.3.5 are used consistently by most commands. However, the actual usage of any field (except the OPERATION CODE field and the CONTROL field) is described in the subclause defining that command. If a device server receives a CDB containing an operation code that is invalid or not supported, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID COMMAND OPERATION CODE.

For all commands, if there is an invalid parameter in the CDB, the device server shall terminate the command without altering the medium.

The ESC CDB format (see 4.3.5) describes an encapsulating CDB and an encapsulated CDB that are packaged as one. The requirements in this subclause apply to the encapsulating CDB and the encapsulated CDB.

4.3.2 The fixed length CDB formats

…

4.3.3 The variable length CDB formats

…

4.3.4 Encapsulated CDBs

4.3.4.1 Model

4.3.4.1.1 Overview

With one exception, any SCSI CDB may be encapsulated in an ESC CDB. The ESC CDB shall not be encapsulated in another ESC CDB. Instead, a new layer of encapsulating parameters shall be added to the existing ESC CDB as described in 4.3.4.1.2.

The following types of encapsulating parameters are defined:

a) Prefix encapsulation parameters precede the encapsulated CDB in the ESC CDB; and
b) Postfix encapsulation parameters follow the encapsulated CDB in the ESC CDB.

An ESC CDB shall contain:

a) Prefix encapsulation parameters, or
b) Prefix encapsulation parameters and postfix encapsulation parameters.

Encapsulation parameters may be:

a) Added by application clients and removed by device servers, or
b) Added and removed by entities outside the scope of this standard that transport or process CDBs during their transfer from an application client to a device server.

Encapsulation parameters shall be added and removed as described in 4.3.4.1.2.

4.3.4.1.2 Adding and removing encapsulation parameters

Encapsulation parameters shall be added and removed in layers. The innermost layer of encapsulation parameters is the first layer that is added to a CDB that is not an ESC CDB. Additional layers are added around but not inside the existing encapsulation parameters. The most recently added layer is the outermost and is the first layer to be removed. The removal of the outermost layer creates a new outermost layer. This process is repeated until only one layer remains (i.e., until the outermost layer is also the innermost). The last (i.e., innermost) layer is removed by removing the ESC CDB from the originally encapsulated CDB.

The addition or removal of a layer of encapsulation parameters shall not alter the encapsulated CDB or any layer of encapsulation parameters except the one being added or removed.

Encapsulation parameters shall be added to a CDB that is not an ESC CDB as follows:

1) The ESC CDB OPERATION CODE field (see 4.3.4.2.1) shall be set to 7Eh;
2) The DATA TRANSFER field (see 4.3.4.2.1) shall be set based on the contents of the encapsulated CDB;
3) The OUTERMOST ENCAPSULATION TYPE field (see 4.3.4.2.1) shall be set to indicate the type of encapsulation parameters (see 4.3.4.2.2) being added;
4) The prefix encapsulation parameters descriptor shall be added with the NEXT ENCAPSULATION TYPE field (see 4.3.4.2.3) set to zero;
5) The encapsulated CDB (see 4.3.4.2.1) shall be added; and
6) The postfix encapsulation parameters descriptor (if any) shall be added and the POSTFIX PARAMETERS OFFSET field in the prefix encapsulation parameters descriptor (see 4.3.4.2.3) shall be set to point to it.

Encapsulation parameters shall be added to an ESC CDB as follows:

1) The contents of the OUTERMOST ENCAPSULATION TYPE field (see 4.3.4.2.1) in the existing ESC CDB shall copied to the NEXT ENCAPSULATION TYPE field in the prefix encapsulation parameters descriptor (see 4.3.4.2.3) being added;
2) The OUTERMOST ENCAPSULATION TYPE field (see 4.3.4.2.1) shall be set to indicate the type of encapsulation parameters (see 4.3.4.2.2) being added;
3) The prefix encapsulation parameters descriptor shall be added;
4) The postfix encapsulation parameters descriptor (if any) shall be added and the POSTFIX PARAMETERS OFFSET field in the prefix encapsulation parameters descriptor (see 4.3.4.2.3) shall be set to point to it.

Unless the NEXT ENCAPSULATION TYPE field in the outermost prefix encapsulation parameters descriptor (see 4.3.4.2.3) contains zero (i.e., unless the outermost layer of encapsulation parameters is being removed), encapsulation parameters shall be removed from an ESC CDB as follows:

1) The contents of the NEXT ENCAPSULATION TYPE field in the outermost prefix encapsulation parameters descriptor shall be copied OUTERMOST ENCAPSULATION TYPE field (see 4.3.4.2.1) in the ESC CDB;
2) The outermost prefix and postfix encapsulation parameters descriptors shall be removed.

If the NEXT ENCAPSULATION TYPE field in the outermost prefix encapsulation parameters descriptor contains zero (i.e., if the outermost layer of encapsulation parameters is being removed), encapsulation parameters shall be removed from an ESC CDB by replacing ESC CDB with the encapsulated CDB contained in the ESC CDB.

4.3.4.2 The ESC CDB format

4.3.4.2.1 Overview

The first byte of an ESC CDB shall contain the operation code 7Eh. The CONTROL byte is the CONTROL byte in the encapsulated CDB (see table x1).

Table x1 — ESC CDB format

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Eh)							
1	DATA TRANSFER		Reserved					
2	Reserved							
3	OUTERMOST ENCAPSULATION TYPE							
Prefix encapsulation parameters descriptors								
4	Outermost prefix encapsulation parameters descriptor							
	. . .							
	Innermost prefix encapsulation parameters descriptor							
i+1	Encapsulated CDB							
k								
Postfix encapsulation parameters descriptors								
k+1	Innermost postfix encapsulation parameters descriptor							
	. . .							
	Outermost postfix encapsulation parameters descriptor							
n								

The DATA TRANSFER field (see table x2) indicates whether the encapsulated CDB uses a Data-In Buffer, a Data-Out Buffer, both, or neither.

Table x2 — DATA TRANSFER field

Code	Description
00b	No Data-In Buffer or Data-Out Buffer
01b	Data-In Buffer but no Data-Out Buffer
10b	Data-Out Buffer but no Data-In Buffer
11b	Both Data-In Buffer and Data-Out Buffer

The OUTERMOST ENCAPSULATION TYPE field (see 4.3.4.2.2) indicates the size and content of the outermost prefix encapsulation parameters descriptor that the ESC CDB adds to the encapsulated CDB. If the OUTERMOST ENCAPSULATION TYPE field contains zero, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

The prefix encapsulation parameters descriptors (see 4.3.4.2.3) contain the encapsulation parameters that the ESC CDB adds before the encapsulated CDB.

The encapsulated CDB bytes follow the prefix encapsulation parameters descriptors and precede the postfix encapsulation parameters descriptors (if any).

The postfix encapsulation parameters descriptors (see 4.3.4.2.4) contain the encapsulation parameters (if any) that the ESC CDB adds after the encapsulated CDB.

NOTE x1 - Several SCSI transport protocols (e.g., FCP and SAS) limit the total CDB size to 260 bytes.

4.3.4.2.2 Encapsulation type

The OUTERMOST ENCAPSULATION TYPE field (see 4.3.4.2.1) and NEXT ENCAPSULATION TYPE field (see 4.3.4.2.3) indicate the type of encapsulation parameters (see table x3) that the ESC CDB adds to the encapsulated CDB.

Table x3 — OUTERMOST ENCAPSULATION TYPE field and NEXT ENCAPSULATION TYPE field

Code	Description	Size of encapsulation parameter descriptors (bytes)		Reference
		Prefix	Postfix	
00h	No next layer	n/a	n/a	4.3.4.2.1
Prefix and Postfix codes				
01h - 07h	Reserved			
Prefix only codes				
08h - FFh	Reserved		0	

4.3.4.2.3 Prefix encapsulations parameters descriptor formats

The format of a prefix encapsulation parameters descriptor is indicated by:

a) The contents of the OUTERMOST ENCAPSULATION TYPE field (see 4.3.4.2.1) for the outermost (i.e., first) prefix encapsulation parameters descriptor in the ESC CDB; or
b) The contents of the NEXT ENCAPSULATION TYPE field in the prefix encapsulation parameters descriptor that immediately precedes the current prefix encapsulation parameters descriptor for all other prefix encapsulation parameters descriptors in the ESC CDB.

If the applicable encapsulation type is 01h to 07h, inclusive, the prefix encapsulation parameters descriptor have the format shown in (see table x4).

Table x4 — Prefix encapsulation parameters descriptor format with postfix parameters present

Bit / Byte	7	6	5	4	3	2	1	0
0	NEXT ENCAPSULATION TYPE							
1	POSTFIX PARAMETERS OFFSET							
2	Prefix encapsulation parameters							
n								

If the applicable encapsulation type is 08h to FFh, inclusive, the prefix encapsulation parameters descriptor have the format shown in (see table x5).

Table x5 — Prefix encapsulation parameters descriptor format without postfix parameters

Bit / Byte	7	6	5	4	3	2	1	0
0	NEXT ENCAPSULATION TYPE							
1	Reserved							
2	Prefix encapsulation parameters							
n								

The NEXT ENCAPSULATION TYPE field (see 4.3.4.2.2) indicates the size and content of the next prefix encapsulation parameters descriptor that the ESC CDB adds to the encapsulated CDB. If this is the innermost (i.e., last) prefix encapsulation parameters descriptor in the ESC CDB, the next encapsulation type field shall be set to zero.

The POSTFIX PARAMETERS OFFSET field indicates the number of bytes that follow before the first byte in the corresponding postfix parameters descriptor (see 4.3.4.2.4) is encountered.

The format and content of the prefix encapsulation parameters depends on the applicable encapsulation type (see 4.3.4.2.2).

The number of bytes in a prefix encapsulation parameters descriptor shall be a multiple of four.

4.3.4.2.4 Postfix encapsulation parameter descriptor format

The postfix encapsulation parameters descriptor have the format shown in (see table x6).

Table x6 — Post encapsulation parameters descriptor format

Bit / Byte	7	6	5	4	3	2	1	0
0	Postfix encapsulation parameters							
n								

The format and content of the postfix encapsulation parameters depends on the applicable encapsulation type (see 4.3.4.2.2) for the prefix encapsulation parameters descriptor in which the POSTFIX PARAMETERS OFFSET field points to the postfix encapsulation parameters descriptor.

The number of bytes in a postfix encapsulation parameters descriptor shall be a multiple of four.

4.3.5 Common CDB fields

4.3.5.1 Operation code

The first byte of a SCSI CDB shall contain an operation code identifying the operation being requested by the CDB. Some operation codes provide for modification of their operation based on a service action (see 4.3.4.2). In such cases, the operation code and service action code combine to identify the operation being requested. The location of the SERVICE ACTION field in the CDB varies depending on the operation code value.

The OPERATION CODE (see table 10) of the CDB has a GROUP CODE field and a COMMAND CODE field. The three-bit GROUP CODE field provides for eight groups of command codes. The five-bit COMMAND CODE field provides for thirty-two command codes in each group. A total of 256 possible operation codes exist. Operation codes are defined in this standard and other command standards (see 3.1.17). The group code value (see table 11) shall determine the length of the CDB.

Table 10 — OPERATION CODE **byte**

Bit	7	6	5	4	3	2	1	0
	GROUP CODE			COMMAND CODE				

The value in the GROUP CODE field specifies one of the groups shown in table 11.

Table 11 — Group Code values

Group Code	Meaning	Typical CDB format
000b	6 byte commands	see table 3 in 4.3.2
001b	10 byte commands	see table 4 in 4.3.2
010b	10 byte commands	see table 4 in 4.3.2
011b	reserved [a]	
100b	16 byte commands	see table 6 and table 7 in 4.3.2
101b	12 byte commands	see table 5 in 4.3.2
110b	vendor specific	
111b	vendor specific	

[a] The format of the commands using the group code 011b and operation code 7Fh is described in 4.3.3. The format of the commands using the group code 011b and operation code 7Eh is described in 4.3.4. With the exception of operation ~~code~~ codes 7Fh and 7Eh, all group code 011b operation codes are reserved.