ENDL
TEXAS

Date: 10 January 2007
To: T10 Technical Committee
From: Ralph O. Weber
Subject: SPC-4: Encapsulated SCSI Commands

"The time has come", the Walrus said,
"To talk of many things:
Of shoes – and ships – and sealing wax –
Of cabbages – and kings –
And why the sea is boiling hot –
And whether pigs have wings."
— Lewis Carroll – *Alice In Wonderland*

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).

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 an information is added to support extra processing (e.g., security features) for the encapsulated CDB. See 4.3.4.

{{New Acronym}}

ESC Encapsulated SCSI Command (see 3.1.x)

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.

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

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 The ESC CDB format

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	ENCAPSULATED OPERATION CODE							
2	ENCAPSULATION TYPE							
3	ENCAPSULATION PARAMETERS LENGTH (n-3)							
4	Encapsulation parameters							
n								
n+1	Encapsulated CDB							
m								

The ENCAPSULATED OPERATION CODE field shall contain the operation code byte from the encapsulated CDB.

{{The ENCAPSULATED OPERATION CODE field is a lightning rod for discussion.}}

The ENCAPSULATION TYPE field (see table x2) indicates the type of encapsulation parameters that the ESC CDB adds to the encapsulated CDB.

Table x2 — ENCAPSULATION TYPE field

Code	Description	Reference
00h	Vendor specific	
01h - FFh	Reserved	

The ENCAPSULATION PARAMETERS LENGTH field indicates the number of bytes that follow in the encapsulation parameters.

{{Encapsulation type should indicate a fixed length for encapsulation parameters.}}

The format of the encapsulation parameters depends on the encapsulation type.

The encapsulated CDB bytes are the last bytes in the ESC CDB.

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

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 **<Super-script>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.