ENDL
TEXAS

Date: 7 May 2008
To: T10 Technical Committee
From: Ralph O. Weber
Subject: Allowing ESP-SCSI descriptors in variable length CDBs

Introduction

Document T10/08-145r2 requires the ability to place ESP-SCSI descriptors in a variable length CDB. This proposal shows the changes needed to allow that.

N.B. The smallest possible ESP-SCSI descriptor is 16 bytes. Therefore, ESP-SCSI descriptors cannot be placed in fixed length CDBs.

Revision History

r0 Initial revision
r1 Revised as requested by the May CAP working group.

Changes between r0 and r1 are marked with change bars.

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

Proposed Changes in SPC-4 r14

5.13.1.4 Types of attacks

The following types of attacks are considered:

a) Passive attacks (i.e., attacks that only require reading IUs), and
b) Active attacks (i.e., attacks that require the attacker to change communication and/or engage in communication).

More information on attack types is available in RFC 3552.

Simple passive attacks involve reading communicated data that the attacker was not intended to see (e.g., password, credit card number). More complex passive attacks involve post-processing the communicated data (e.g., checking a challenge-response pair against a dictionary to see if a common word was used as a password).

There are a wide variety of active attacks (e.g., spoofing, replay, insertion, deletion, known plaintext, and modification of communications). Man-in-the-middle attacks are a class of active attacks that involve the attacker inserting itself in the middle of communication, enabling it to intercept all communications without the knowledge of the communicating parties for various purposes (e.g., insertion, deletion, replay, modification and/or inspection via decryption of the communications).

…

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

5.13.7 ESP-SCSI for parameter data

5.13.7.1 Overview

Subclause 5.13.7 defines a method for transferring encrypted and/or integrity checked parameter data in data-in buffers, ~~and/or~~ data-out buffers, variable length CDBs (see 4.3.3), and extended CDBs (see 4.3.4). The method is based on the Encapsulating Security Payload (see RFC 4303) standard developed by the IETF. Because of the constrained usage of ESP-SCSI parameter data in data-in buffers and/or data-out buffers, the method defined in this standard differs from the one found in RFC 4303.

…

5.13.7.2 ESP-SCSI required inputs

…

5.13.7.3 ESP-SCSI data format before encryption and after decryption

Before data bytes are encrypted and after they are decrypted, they have the format shown in table 68.

Table 68 — ESP-SCSI data format before encryption and after decryption

Bit / Byte	7	6	5	4	3	2	1	0
0	UNENCRYPTED BYTES							
p-1								
p	PADDING BYTES							
j-1								
j	PAD LENGTH (j-p)							
j+1	MUST BE ZERO							

The UNENCRYPTED BYTES field contains the bytes that are to be protected via encryption or that have been decrypted.

Before encryption, the PADDING BYTES field contains zero to 255 bytes. The number of padding bytes is:

a) Defined by the encryption algorithm; or
b) The number needed to cause the length of all bytes prior to encryption (i.e., j+2) to be a whole multiple of the alignment (see table 407 in 7.6.3.6.2) for the encryption algorithm being used.

The contents of the padding bytes are:

a) Defined by the encryption algorithm; or
b) If the encryption algorithm does not define the padding bytes contents, a series of one byte binary values starting at one and incrementing by one in each successive byte (i.e., 01h in the first padding byte, 02h in the second padding byte, etc.).

If the encryption algorithm does not place requirements on the contents of the padding bytes (i.e., option b) is in effect), then after decryption the contents of the padding bytes shall be verified to match the series of one byte binary values described in this subclause. If this verification is not successful in a device server, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, the additional

sense code set to INVALID FIELD IN PARAMETER LIST, the SKSV bit set to one, and SENSE KEY SPECIFIC field set to indicate the last byte in the encrypted data as defined in 4.5.2.4.2. If this verification is not successful in an application client, the decrypted data should be ignored.

The PAD LENGTH field contains the number of bytes in the PADDING BYTES field.

The MUST BE ZERO field contains zero. After decryption, the contents of the MUST BE ZERO field shall be verified to be zero. If this verification is not successful in a device server, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, the additional sense code set to INVALID FIELD IN PARAMETER LIST, the SKSV bit set to one, and SENSE KEY SPECIFIC field set to indicate the last byte in the encrypted data as defined in 4.5.2.4.2. If this verification is not successful in an application client, the decrypted data should be ignored.

5.13.7.4 ESP-SCSI ~~data-out buffer parameter list~~ outbound data descriptors

5.13.7.4.1 Overview

When ESP-SCSI is used in a variable length CDB (see 4.3.3), an extended CDB (see 4.3.4), or parameter list data that appears in a data-out buffer, the parameter list data contains one or more descriptors selected based on the criteria shown in table 69.

Table 69 — ESP-SCSI data-out buffer parameter data descriptors

Descriptor name	External descriptor length [a]	Initialization vector present [b]	Reference
ESP-SCSI CDB	No	No	table 70 in 5.13.7.4.2
	No	Yes	table 71 in 5.13.7.4.2
ESP-SCSI data-out buffer	No	No	table 70 in 5.13.7.4.2
	No	Yes	table 71 in 5.13.7.4.2
ESP-SCSI data-out buffer without length	Yes	No	table 72 in 5.13.7.4.3
	Yes	Yes	table 73 in 5.13.7.4.3

[a] This is determined by the data format defined for the data-out buffer parameter data. If the format includes a length for the ESP-SCSI descriptor, then the answer to this question is yes.

[b] This is determined from the USAGE_DATA SA parameter (see 5.13.7.2).

5.13.7.4.2 ESP-SCSI CDBs or data-out buffer parameter lists including a descriptor length

If the USAGE_DATA SA parameter (see 5.13.7.2) indicates an encryption algorithm whose initialization vector size is zero, then the variable length CDB (see 4.3.3), an extended CDB (see 4.3.4), or data-out buffer parameter list descriptor shown in table 70 contains the ESP-SCSI data.

Table 70 — ESP-SCSI CDBs or data-out buffer parameter list descriptor without initialization vector

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				DESCRIPTOR LENGTH (n-1)				(LSB)
2								
3				Reserved				
4	(MSB)							
7				DS_SAI				(LSB)
8	(MSB)							
15				DS_SQN				(LSB)
16								
i-1				ENCRYPTED OR AUTHENTICATED DATA				
i	(MSB)							
n				INTEGRITY CHECK VALUE				(LSB)

The DESCRIPTOR LENGTH field, DS_SAI field, DS_SQN field, ENCRYPTED OR AUTHENTICATED DATA field, and INTEGRITY CHECK VALUE field are defined after table 71 in this subclause.

If the USAGE_DATA SA parameter indicates an indicates an encryption algorithm whose initialization vector size (i.e., s) is greater than zero, the variable length CDB (see 4.3.3), an extended CDB (see 4.3.4), or data-out buffer parameter data descriptor shown in table 71 contains the ESP-SCSI data.

Table 71 — ESP-SCSI CDBs or data-out buffer full parameter list descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)			DESCRIPTOR LENGTH (n-1)				
1								(LSB)
2				Reserved				
3								
4	(MSB)			DS_SAI				
7								(LSB)
8	(MSB)			DS_SQN				
15								(LSB)
16	(MSB)			INITIALIZATION VECTOR				
16+s-1								(LSB)
16+s				ENCRYPTED OR AUTHENTICATED DATA				
i-1								
i	(MSB)			INTEGRITY CHECK VALUE				
n								(LSB)

The DESCRIPTOR LENGTH field specifies the number of bytes that follow in the ESP-SCSI CDB or ESP-SCSI data-out buffer parameter list descriptor.

The DS_SAI field contains the value in the DS_SAI SA parameter (see 5.13.2.2) for the SA that is being used to prepare the ESP-SCSI CDB or ESP-SCSI data-out buffer parameter list descriptor. If the DS_SAI value is not known to the device server, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, the additional sense code set to INVALID FIELD IN PARAMETER LIST, the SKSV bit set to one, and SENSE KEY SPECIFIC field set as defined in 4.5.2.4.2.

The DS_SQN field should contain one plus the value in the application client's DS_SQN SA parameter (see 5.13.2.2) for the SA that is being used to prepare the ESP-SCSI CDB or ESP-SCSI data-out buffer parameter list descriptor. Before sending the ESP-SCSI CDB or ESP-SCSI data-out buffer parameter list, the application client should copy the contents of the DS_SQN field to its DS_SQN SA parameter.

The device server shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, the additional sense code set to INVALID FIELD IN PARAMETER LIST, the SKSV bit set to one, and SENSE KEY SPECIFIC field set as defined in 4.5.2.4.2 if any of the following conditions are detected:

a) The DS_SQN field is set to zero;
b) The value in the DS_SQN field is less than or equal to the value in the device server's DS_SQN SA parameter; or
c) The value in the DS_SQN field is greater than 32 plus the value in the device server's DS_SQN SA parameter.

If the DS_SQN SA parameter is equal to FFFF FFFF FFFF FFFFh, the device server shall delete the SA.

The INITIALIZATION VECTOR field, if any, contains a value that is used as an input into the encryption algorithm and/or integrity algorithm specified by the SA specified by the DS_SAI field. The INITIALIZATION VECTOR field is not encrypted. The encryption algorithm and/or integrity algorithm may define additional requirements for the INITIALIZATION VECTOR field.

The ENCRYPTED OR AUTHENTICATED DATA field contains:

a) If an encryption algorithm for the SA specified by the DS_SAI field is not ENCR_NULL, encrypted data bytes for the following:
   1) The bytes in the UNENCRYPTED BYTES field (see 5.13.7.3);
   2) The bytes in the PADDING BYTES field (see 5.13.7.3);
   3) The PAD LENGTH field byte (see 5.13.7.3); and
   4) The MUST BE ZERO field byte (see 5.13.7.3);

   or
b) Otherwise, the unencrypted data bytes.

If the integrity algorithm for the SA specified by the DS_SAI field is AUTH_COMBINED (see 5.13.7.2), then the AAD input to the encryption algorithm is composed of the following bytes, in order:

1) The bytes in the DS_SAI field; and
2) The bytes in the DS_SQN field.

The INTEGRITY CHECK VALUE field contains a value that is computed as follows:

a) If the integrity algorithm is not AUTH_COMBINED, the integrity check value is computed using the specified integrity algorithm with the following bytes as inputs, in order:
   1) The bytes in the DS_SAI field;
   2) The bytes in the DS_SQN field;
   3) The bytes in the INITIALIZATION VECTOR field, if any; and
   4) The bytes in the ENCRYPTED OR AUTHENTICATED DATA field after encryption, if any, has been performed;

   or
b) If the integrity algorithm is AUTH_COMBINED, the integrity check value is computed as an additional output of the specified encryption algorithm.

Upon receipt of ESP-SCSI CDB or ESP-SCSI data-out buffer parameter data, the device server shall compute an integrity check value for the ESP-SCSI CDB or ESP-SCSI parameter data as specified by the algorithms specified by the SA specified by the DS_SAI field using the inputs shown in this subclause. If the computed integrity check value does not match the value in the INTEGRITY CHECK VALUE field, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, the additional sense code set to INVALID FIELD IN PARAMETER LIST, the SKSV bit set to one, and SENSE KEY SPECIFIC field set as defined in 4.5.2.4.2.

If the command is not terminated due to a sequence number error or a mismatch between the computed integrity check value and the contents of the INTEGRITY CHECK VALUE field, then the device server shall copy the contents of the received DS_SQN field to its DS_SQN SA parameter.

5.13.7.4.3 ESP-SCSI data-out buffer parameter lists for externally specified descriptor length

…

5.13.7.5 ESP-SCSI data-in buffer parameter data descriptors

5.13.7.5.1 Overview

A device server shall transfer ESP-SCSI parameter data descriptors in a data-in buffer only in response to a request that specifies an SA using the AC_SAI SA parameter and DS_SAI SA parameter values (see 5.13.2.2). If the specified combination of AC_SAI and DS_SAI values in a command that requests the transfer of ESP-SCSI parameter data descriptors is not known to the device server, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, the additional sense code set to INVALID FIELD IN PARAMETER LIST or to INVALID FIELD IN CDB, the SKSV bit set to one, and SENSE KEY SPECIFIC field set as defined in 4.5.2.4.2.

When ESP-SCSI is used in parameter data which appears in a data-in buffer, the parameter data contains one or more descriptors selected based on the criteria shown in table 74.

Table 74 — ESP-SCSI data-in buffer parameter data descriptors

Descriptor name	External descriptor length [a]	Initialization vector present [b]	Reference
ESP-SCSI data-in buffer	No	No	table 75 in 5.13.7.5.2
	No	Yes	table 76 in 5.13.7.5.2
ESP-SCSI data-in buffer without length	Yes	No	table 77 in 5.13.7.5.3
	Yes	Yes	table 78 in 5.13.7.5.3

[a] This is determined by the data format defined for the data-in buffer parameter data. If the format includes a length for the ESP-SCSI descriptor, then the answer to this question is yes.

[b] This is determined from the USAGE_DATA SA parameter (see 5.13.7.3).

If ESP-SCSI parameter data descriptors are used in a data-in buffer, then the outbound data (see 5.13.7.4) should include at least one ESP-SCSI descriptor using the same SA to thwart known plaintext attacks (see 5.13.1.4).

…