EMC²
where information lives®

To: INCITS Technical Committee T10
From: David L. Black, EMC
Email: black_david@emc.com
Date: November 14, 2008
Subject: SPC-4: IKEv2-SCSI Authentication (08-423r3)

1) *Revision history*

Revision 0 (Oct 29, 2008): First revision (r0)

Revision 1 (Nov 4, 2008): Major rewrite to put most of the added text in the model clause, plus reorganize affected portions of the model clause. (r1)

Revision 2 (Nov 14, 2008): Incorporate changes from CAP meeting and include another subsection of the model clause in the new authentication subsection (r2).

Revision 3 (December 1, 2008): Incorporate editorial changes from interim call, and add raw RSA key mechanism to description (r3).

2) *Related documents*

spc4r17 – SCSI Primary Commands – 4

IETF RFC 4306 – Internet Key Exchange (IKEv2) Protocol

IETF RFC 5280 – Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile

IETF RFC 5282 – Using Authenticated Encryption Algorithms with the Encrypted Payload of the Internet Key Exchange version 2 (IKEv2) Protocol

3) *Overview*

Review of the IKEv2-SCSI specification has identified text that is incorrect when digital signatures are used for authentication. The erroneous text specifies that the recipient calculates the expected authentication value. This is applicable when a shared secret key is used for authentication (Shared Key Message Integrity Code), but is incorrect when a digital signature is used. This is because a digital signature is computed with the sender's private key, but the recipient checks it with the sender's public key. Asymmetric cryptography is used, so the recipient can check the signature, but cannot reproduce the signature on its own.

In considering how to correct this problem, a need has been identified for model clause material on IKEv2-SCSI authentication. This document proposes to add that model clause, incorporating some existing model clause material, and make the necessary changes elsewhere. This document also serves as a vehicle to make a couple of updates to Annex C, one of which has impacts elsewhere because it reflects IETF completion of standardization work motivated by IKEv2-SCSI.

Existing text is shown in **BLACK**, additions are shown in **GREEN**, text to be deleted is ~~**struck through in RED**~~ and comments (not to be included) are shown in **BLUE**.

Proposal:

2.5 IETF References

Remove RFC 3280 as it has been replaced by RFC 5280

~~RFC 3280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*~~

Add the following two references in the appropriate positions in the list of IETF RFCs:

RFC 5280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*

RFC 5282, *Using Authenticated Encryption Algorithms with the Encrypted Payload of the Internet Key Exchange version 2 (IKEv2) Protocol*

Editor: Please update all RFC 3280 references to reference RFC 5280.

5.14.4.3 ~~Handling of the Certificate Request payload and the Certificate payload~~ IKEv2-SCSI Authentication

Section 5.14.4.3 will contain the new model clause material on IKEv2-SCSI authentication. The existing contents of 5.14.4.3 are moved to a subsection of 5.14.4.3. The existing contents of 5.14.4.5 and 5.14.4.6 are also moved to subsections of 5.14.4.3.

5.14.4.3.1 Overview

IKEv2-SCSI authentication includes these security functions:

a) The application client and device server each establish an identity by demonstrating knowledge of a secret authentication key associated with that identity;
b) The application client demonstrates knowledge of the current device server capability information; and
c) The application client and device server check the integrity of the current IKEv2-SCSI CCS.

An IKEv2 SCSI authentication algorithm accomplishes these functions by generating and verifying authentication data based on a concatenation of bytes that includes device server capability information and an appropriate portion of the IKEv2-SCSI parameter data from the IKEv2-SCSI Key Exchange step (see 5.14.4.8).

An authentication key associated with an identity is used to generate authentication data. IKEv2-SCSI transfers the authentication data in the AUTHENTICATION DATA field of the IKEv2-SCSI

Authentication payload. The recipient uses a verification key associated with the identity to verify the authentication data. The identity is transferred in the IDENTIFICATION DATA field of the appropriate IKEv2-SCSI Identity payload or is obtained from a certificate transferred in the IKEv2-SCSI Certificate payload.

IKEv2-SCSI Authentication is bidirectional (i.e., both the application client and the device server authenticate). IKEv2-SCSI Authentication is skipped when the application client and device server agree to do so during the Key Exchange step (see 5.14.4.3.4).

The following IKEv2-SCSI Authentication methods are defined:

a) Pre-Shared Key: The authentication key is also used as the verification key; and
b) Digital Signature: The verification key and authentication key form a public/private key pair. The authentication data is a digital signature based on asymmetric cryptography.

Certificates and the IKEv2-SCSI Certificate payload may be used to provide verification keys for digital signatures to application clients and device servers.

5.14.4.3.2 Pre-Shared Key Authentication

Pre-shared key authentication uses a single cryptographic algorithm to both generate and verify authentication data. A pre-shared key is associated with an identity that is transferred in the IDENTIFICATION DATA field of the appropriate Identification payload (see 7.6.5.3.4). The pre-shared key serves as both the authentication key and the verification key for the identity.

> Note: A pre-shared key that is not kept secret may compromise the security properties of IKEv2-SCSI.

If pre-shared key authentication is used, then the pre-shared key is the key for the cryptographic algorithm. Authentication data is generated by applying the cryptographic algorithm with this key to the input data.

Verification of the authentication data shall consist of:

1) Computing the expected contents of the AUTHENTICATION DATA field of the Authentication payload using the input data and a verification key associated with the identity received in the Identification payload (see 7.6.3.5.4); and
2) Comparing the expected contents of this field to the actual contents of the field.

Verification is successful if the expected contents match the actual contents, otherwise verification is not successful.

The contents of section 5.14.4.5 are moved to this location. Text that is unchanged except for relocation remains black.

The pre-shared key requirements in RFC 4306 shall apply to IKEv2-SCSI pre-shared keys, including the following requirements on interfaces for provisioning pre-shared keys:

~~A~~a) ASCII strings of at least 64 bytes shall be supported;
~~B~~b) A null terminator shall not be added to any input before it is used as a pre-shared key;
~~C~~c) A hexadecimal ASCII encoding of the pre-shared key shall be supported; and
~~D~~d) ASCII encodings other than hexadecimal may be supported. Support for any such encoding shall include specification of the algorithm for translating the encoding to a binary string as part of the interface;

The above list has been moved here from item f) below. In addition, item e) below is removed by this proposal.

~~If the Authentication payload (see 7.6.3.5.6) AUTHENTICATION DATA field contents are computed using pre-shared keys (e.g., if the applicable algorithm identifier is 00F9 0002h, Shared Key Message Integrity Code), then the~~ The following requirements for pre-shared keys apply in addition to those found in RFC 4306:

a) A pre-shared key shall be associated with one identity;

b) The same pre-shared key shall not be used to authenticate both an application client and a device server;

c) ~~Use of t~~The same pre-shared key should not be used for a group of application clients or a group of device servers ~~is strongly discouraged, because it enables any member of the group to impersonate any other member~~;

d) The means for provisioning pre-shared keys are outside the scope of this standard;

~~e) The pre-shared keys may be provisioned as follows:~~

~~A) At the time of manufacturing;~~

~~B) During device or system initialization; or~~

~~C) Any time thereafter;~~

~~f) The following requirements from RFC 4306 apply to the interfaces for provisioning pre-shared keys:~~

~~A) ASCII strings of at least 64 bytes shall be supported;~~

~~B) A null terminator shall not be added to any input before it is used as a pre-shared key;~~

~~C) A hexadecimal ASCII encoding of the pre-shared key shall be supported; and~~

~~D) ASCII encodings other than hexadecimal may be supported. Support for any such encoding shall include specification of the algorithm for translating the encoding to a binary string as part of the interface;~~

and

~~g~~e) Information about the size of the pre-shared key shall be stored at the same time that the pre-shared key is stored.

Editor: Please update cross-references to 5.14.4.5 to instead reference 5.14.4.3.2

5.14.4.3.3 Digital Signature Authentication

5.14.4.3.3.1 Overview

Digital signature authentication uses a matched pair of signature and verification cryptographic algorithms to generate and verify authentication data that is a digital signature. A public/private key pair is associated with an identity. The private key is used as the authentication key for the identity. The public key is used as the verification key for the identity.

Note: A private authentication key that is not kept secret may compromise the security properties of IKEv2-SCSI.

If digital signature authentication is used, then the private key is the key for the signature algorithm. A digital signature is generated by applying the signature algorithm with this private key to the input data.

Verification of the digital signature shall consist of using the public verification key associated with the identity and the input data to verify the digital signature received as the contents of the AUTHENTICATION DATA field of the Authentication payload. Verification is successful if the digital signature is a valid digital signature over the input data, otherwise verification is not successful.

The means by which an application client or device server obtains a private authentication key are outside the scope of this standard. An identity and associated public verification key are obtained as follows:

a) If certificates are used for digital signature authentication, then the identity and the associated public verification key are obtained from a certificate transferred in the first IKEv2-SCSI Certificate payload (see 5.14.4.3.3.4); or
b) If certificates are not used for digital signature authentication, then the identity is transferred in the IDENTIFICATION DATA field of the appropriate IKEv2-SCSI Identification payload and the public verification key:
   A) May be transferred as a raw RSA key in an IKEv2-SCSI Certificate payload; or
   B) May be obtained by means that are outside the scope of this standard.

If certificates are not used for digital signature authentication, the association between the identity and the public key should be verified by means outside the scope of this standard.

5.14.4.3.3.2 Certificates and Digital Signature Authentication

A certificate (see RFC 5280) is a data structure that contains:
a) An identity;
b) A public key for that identity;
c) Additional relevant information that may constrain use of the public key;
d) The identity of a certification authority (see RFC 5280); and
e) A digital signature generated by that certification authority.

If the identity and associated public key used to verify a digital signature are obtained from a certificate, the certification path from the certificate to a trust anchor should be validated (see RFC 5280). If certification path validation is not successful, verification of the digital signature for that identity shall fail independent of whether the digital signature is valid. The means by which an application client or device server obtains a trust anchor are outside the scope of this standard.

5.14.4.3.3.3 Example of Certificate Use for Digital Signature Authentication

An example of certificate use involves an application client or device server that trusts a certification authority. Based on this trust, the public key of that certification authority is used to validate a certificate presented as part of authentication. Successful validation of that certificate establishes that the public key in that certificate is associated with the identity in the certificate. That public key is then used to verify the digital signature in the Authentication payload.

In this example, providing a certificate as part of the IKEv2-SCSI Authentication step (see 5.14.4.9) allows a single certification authority public key to serve as a trust anchor for verification of digital signatures for any identity that has been issued a certificate by that certification authority, avoiding the need to obtain a public key for each identity by other means.

Validating a certificate includes multiple checks beyond verifying the signature, and the validation may traverse a certification path composed of multiple certificates (see RFC 5280).

5.14.4.3.3.4 Handling of the Certificate Request payload and the Certificate payload

The contents of section 5.14.4.3 are moved here. Text that is unchanged except for relocation remains black.

As detailed in this subclause, a Certificate Request payload (see 7.6.3.5.5) in one set of parameter data requests the delivery of a Certificate payload (see 7.6.3.5.5) in the next set of parameter data transferred. The purpose of these IKEv2-SCSI protocol elements is as follows:

a) Each SA participant is allowed to require the delivery of a Certificate payload by the other SA participant for use in authentication; and
b) Each Certificate Request payload indicates the trust anchors (see RFC 4306) list used by the device server or application client when PKI-based Authentication is being used with certificates that are not self signed (see RFC ~~3280~~5280).

The presence of one or more Certificate Request payloads in the Key Exchange step SECURITY PROTOCOL IN command (see 5.14.4.8.3) parameter data indicates that the device server requires the application client to send a Certificate payload in the Authentication step SECURITY PROTOCOL OUT command (see 5.14.4.9.2).

The presence of one or more Certificate Request payloads in the Authentication step SECURITY PROTOCOL OUT command parameter list specifies that the application client requires the device server to send a Certificate payload in the Authentication step SECURITY PROTOCOL IN command (see 5.14.4.9.3).

If any Certificate payloads are included in the parameter data, the first Certificate payload shall contain the public key used to verify the Authentication payload. Additional Certificate payloads may be sent to assist in establishing a ~~chain of trust~~ certification path from the certificate in the first payload to a trust anchor (see RFC 4306 and RFC 5280).

The application client and device server may use different authentication methods that require or do not require the use of Certificate payloads. ~~, and t~~ The presence or absence of Certificate Request payloads and Certificate payloads may vary in any of the commands described in this subclause.

Editor: Please update cross-references to 5.14.4.3 to instead reference 5.14.4.3.3.4

5.14.4.3.4 Constraints on skipping the Authentication step

The contents of section 5.14.4.6 are moved here. Text that is unchanged except for relocation remains black.

In the Device Server Capabilities step (see 5.14.4.7), the parameter data returned by the SECURITY PROTOCOL IN command (see 7.6.2.3.2) contains the IKEv2-SCSI SA Creation Algorithms payload (see 7.6.3.5.11) that contains one or more SA_AUTH_OUT IKEv2-SCSI cryptographic algorithm descriptors (see 7.6.3.6.6) and one or more SA_AUTH_IN~~OUT~~ IKEv2-SCSI cryptographic algorithm descriptors.

The device server ~~permits~~ shall permit the Authentication step to be omitted (see 5.14.4.1) if ~~all of the following are true~~:

a) The ALGORITHM IDENTIFIER field is set to SA_AUTH_NONE (see 7.6.3.6.6) in one of the SA_AUTH_OUT IKEv2-SCSI cryptographic algorithm descriptors returned in the Device Server Capabilities step; and
b) The ALGORITHM IDENTIFIER field is set to SA_AUTH_NONE in one of the SA_AUTH_IN IKEv2-SCSI cryptographic algorithm descriptors returned in the Device Server Capabilities step.

The methods for configuring a device server to return SA_AUTH_NONE are outside the scope of this standard. Device servers shall not be manufactured to return SA_AUTH_NONE as an Authentication payload authentication algorithm type in the Device Server Capabilities step.

In the Key Exchange step SECURITY PROTOCOL OUT command (see 5.14.4.8.2), the application client requests that the Authentication step be omitted by setting the ALGORITHM IDENTIFIER field to SA_AUTH_NONE in:

a) ~~t~~The SA_AUTH_OUT cryptographic algorithm descriptor; and ~~in~~
b) ~~t~~The SA_AUTH_IN cryptographic algorithm descriptor

in the IKEv2-SCSI SA Cryptographic Algorithms payload (see 7.6.3.5.12).

To ensure adequate SA security, the application client should not select the SA_AUTH_NONE value as an Authentication payload authentication algorithm type unless:

a) An SA_AUTH_OUT IKEv2-SCSI cryptographic algorithm descriptor and an SA_AUTH_IN IKEv2-SCSI cryptographic algorithm descriptor from the Device Server Capabilities step indicates SA_AUTH_NONE availability; and
b) The application client is configured to omit the Authentication step.

NOTE 12 - When SA_AUTH_NONE is used, IKEv2-SCSI has no protection against any man-in-the-middle attacks. Enabling return of the SA_AUTH_NONE authentication algorithm type in the Device Capabilities step, and allowing an application client to select SA_AUTH_NONE in the Key Exchange step are administrative security policy decisions that absence of authentication is acceptable, ~~and should only be made with a full understanding of the security consequences of the lack of authentication~~. Such decisions should only be made in situations where active attacks on IKEv2-SCSI are not of concern (e.g., direct attachment of a SCSI initiator device and a SCSI target device, or an end-to-end secure service delivery subsystem such as ~~FCP over~~ Fibre Channel secured by an end-to-end FC-SP SA).

Editor: Please update cross-references to 5.14.4.6 to instead reference 5.14.4.3.4

~~5.14.4.5 IKEv2-SCSI usage of pre-shared keys~~

The contents of section 5.14.4.5 are moved to 5.14.4.3.2 above. Remove this section, renumbering other sections (and cross-references to them) appropriately.

~~5.14.4.6 Constraints on skipping the Authentication step~~

The contents of section 5.14.4.6 are moved to 5.14.4.3.4 above. Remove this section, renumbering other sections (and cross-references to them) appropriately.

7.6.3.5.6 Authentication payload

In the middle of p.487 (after first numbered list):

When processing the Authentication step SECURITY PROTOCOL OUT command, the device server shall ~~compute the expected~~ verify the contents of the AUTHENTICATION DATA field by applying the algorithm specified by the ALGORITHM IDENTIFIER field in the SA_AUTH_OUT IKEv2-SCSI cryptographic algorithm descriptor in the IKEv2-SCSI SA Cryptographic Algorithms payload (see 7.6.3.5.12) in the Key Exchange step (see 5.14.4.8) as described in table 431 (see 7.6.3.6.6) and this subclause to the following concatenation of bytes:

No change to list of bytes to concatenate. Next change follows that list.

If the ~~expected~~ verification of the contents of the AUTHENTICATION DATA field ~~do not match actual contents of the AUTHENTICATION DATA field,~~ is not successful, then:

In the middle of p.488 (after first numbered list):

After GOOD status is received for the Authentication step SECURITY PROTOCOL IN command, the application client should ~~compute the expected~~ verify the contents of the AUTHENTICATION DATA field by applying the algorithm specified by the ALGORITHM IDENTIFIER field in the SA_AUTH_IN IKEv2-SCSI cryptographic algorithm descriptor in the IKEv2-SCSI SA Cryptographic Algorithms payload (see 7.6.3.5.12) in the Key Exchange step (see 5.14.4.8) as described in table 431 (see 7.6.3.6.6) and this subclause to the following concatenation of bytes:

No change to list of bytes to concatenate. Next change follows that list.

If the ~~expected~~ verification of the contents of the AUTHENTICATION DATA field ~~do not match actual contents of the AUTHENTICATION DATA field,~~ is not successful, then the application client should abandon the IKEv2-SCSI CCS and notify the device server that it is abandoning the IKEv2-SCSI CCS as described in 5.14.4.12.

7.6.3.6.2 Encryption algorithm (ENCR) IKEv2-SCSI cryptographic algorithm descriptor

Add "RFC 5282" as an additional Reference for the AES-CCM and AES-GCM rows in table 423.

7.6.3.6.6 IKEv2-SCSI authentication algorithm IKEv2-SCSI cryptographic algorithm descriptor

Add a new "Authentication Reference" column to table 431 between the Support and Reference columns (only these 3 columns are shown below), add a new table footnote d, and rename the "Reference" column to "Algorithm Reference" without change to the contents of that column. Table footnotes a-c are unchanged and are not shown below (the editor may wish to re-alphabetize the footnotes):

Description	Support	Authentication Reference[d]
SA_AUTH_NONE	Optional	n/a
RSA Digital Signature with SHA-1 [a]	Optional	5.14.4.3.3
Shared Key Message Integrity Code	Optional	5.14.4.3.2
ECDSA with SHA-256 on the P-256 curve [a]	Optional	5.14.4.3.3
ECDSA with SHA-512 on the P-521 curve [a]	Optional	5.14.4.3.3
Vendor Specific	Optional	
Restricted	Prohibited	
Reserved		

Footnotes a-c are unchanged.

[d] The reference in the Authentication Reference column specifies how the algorithm shall be used to generate and verify the contents of the AUTHENTICATION DATA field of the Authentication Payload.

C.1 IKEv2 protocol details and variations for IKEv2-SCSI

Modify item y) and add new items ab) and ac) at the end of the list:

y) ~~The description of how combined mode algorithms are used in the Encrypted payload in this standard predates the definition of equivalent functionality in IETF standards.~~ IETF standards omit the Integrity transform instead of using AUTH_COMBINED;

ab) The critical (CRIT) bit is set to one in all IKEv2-SCSI payloads specified in this standard and these payloads are required to be recognized by all IKEv2-SCSI implementations. RFC 4306 sets the Critical (C) bit to zero in all IKEv2 payloads specified in RFC 4306, and requires that all IKEv2 implementations recognize all payloads specified in RFC 4306.

ac) Use of the Identification payloads is required by IKEv2-SCSI, whereas IKEv2 allows the Identification payloads to be omitted.