Requirements from the

Functional Package for Secure Shell (SSH)

Version: 1.0

2019-08-21

National Information Assurance Partnership

Revision History

Version	Date	Comment

Introduction

Purpose. This document presents the functional and assurance requirements found in the *Functional Package for Secure Shell (SSH)*. Common Criteria evaluation, facilitated in the U.S. by the National Information Assurance Partnership (NIAP), is required for IA and IA-enabled products in National Security Systems according to CNSS Policy #11.

Using this document. This representation of the Protection Profile includes:

- *Security Functional Requirements* for use in evaluation. These are featured without the formal Assurance Activities specified in the Protection Profile, to assist the reader who is interested only in the requirements.

  It also includes, in tables shown later, particular types of security functional requirements that are not strictly required in all cases. These are:

  - *Selection-based Security Functional Requirements* which become required when certain selections are made inside the regular Security Functionality Requirements (as indicated by the **[selection:]** construct).
  - *Objective Security Functional Requirements* which are highly desired but not yet widely-available in commercial technology.
  - *Optional Security Functional Requirements* which are available for evaluation and which some customers may insist upon.

- *Security Assurance Requirements* which relate to developer support for the product under evaluation, development processes, and other non-functionality security relevant requirements.

Security Functional Requirements

Cryptographic Operation (AES-CTR Encryption/Decryption for SSH)

FCS_COP.1.1/SSH The **SSH software shall [selection: *perform*, *invoke-platform-provided*]** [*encryption/decryption services for data*] in accordance with a specified cryptographic algorithm [*AES-CTR (as defined in NIST SP 800-38A) mode*] and cryptographic key sizes [*128-bit, 256-bit*].

Application Note: This Package may be used for a TOE that conforms to a PP that permits the TOE's use of platform cryptography (such as the Protection Profile for Application Software). In this case, the TOE may rely on its platform to provide the cryptographic functionality used to support the TOE's SSH function. If the SSH software does provide its own cryptography, the ST should indicate which cryptographic SFRs from its claimed PP are used to implement SSH functionality.

SSH Protocol

FCS_SSH_EXT.1.1 The SSH software shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254 and [**selection**: *5647*, *5656*, *6187*, *6668*, *8332*, *no other RFCs*] as a [**selection**: *client*, *server*].

Application Note: The ST author selects which of the additional RFCs to which conformance is being claimed. An SSH product can implement additional RFCs, but only those listed in the selection can be claimed as conformant under CC. The RFC selections for this requirement need to be consistent with selections in later elements of this Functional Package (e.g., cryptographic algorithms permitted). RFC 4253 indicates that certain cryptographic algorithms are "REQUIRED." This means that from the Internet Engineering Task Force's (IETF's) perspective the implementation must include support, not that the algorithms must be enabled for use. For the purposes of this SFR's evaluation activity and this Functional Package overall, it is not necessary to ensure that algorithms listed as "REQUIRED" by the RFC but not listed in later elements of this Functional Package are actually implemented.

RFC 5647 applies when AEAD_AES_128_GCM or AEAD_AES_256_GCM is selected as an encryption algorithm in FCS_SSHC_EXT.1.3 or FCS_SSHS_EXT.1.3 and as a MAC algorithm in FCS_SSHC_EXT.1.5 or FCS_SSHS_EXT.1.5.

RFC 5656 applies when ecdsa-sha2-nistp256 or ecdsa-sha2-nistp384 is selected as a public key algorithm in FCS_SSHC_EXT.1.4 or FCS_SSHS_EXT.1.4, or when ecdh-sha2-nistp256, ecdh-sha2-nistp384, or ecdh-sha2-nistp521 is selected as a key exchange algorithm in FCS_SSHC_EXT.1.6 or FCS_SSHS_EXT.1.6.

RFC 6187 applies when x509v3-ecdsa-sha2-nistp256 or x509v3-ecdsa-sha2-nistp384 is selected as a public key algorithm in FCS_SSHC_EXT.1.4 or FCS_SSHS_EXT.1.4.

RFC 6668 applies when hmac-sha2-256 or hmac-sha2-512 is selected as a MAC algorithm in FCS_SSHC_EXT.1.5 or FCS_SSHS_EXT.1.5.

RFC 8332 applies when rsa-sha2-256 or rsa-sha2-512 is selected as a public key algorithm in FCS_SSHC_EXT.1.4 or FCS_SSHS_EXT.1.4.

If "client" is selected, then the ST must include the requirements from FCS_SSHC_EXT.1.

If "server" is selected, then the ST must include the requirements from FCS_SSHS_EXT.1.

Security Assurance Requirements

Selection-Based Security Functional Requirements

SSH Protocol - Client

FCS_SSHC_EXT.1.1 The SSH client shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based and [**selection**: *password-based*, *no other method*].

FCS_SSHC_EXT.1.2 The SSH client shall ensure that, as described in RFC 4253, packets greater than [**assignment**: *number of bytes*] bytes in an SSH transport connection are dropped.

Application Note: RFC 4253 provides for the acceptance of “large packets” with the caveat that the packets should be of “reasonable length” or dropped. The assignment should be filled

in by the ST author with the maximum packet size accepted, thus defining “reasonable length” for the TOE.

FCS_SSHC_EXT.1.3 The SSH client shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: aes128-ctr, aes256-ctr, [**selection**: *aes128-cbc*, *aes256-cbc*, *AEAD_AES_128_GCM*, *AEAD_AES_256_GCM*, *no other algorithms*].

Application Note: RFC 5647 specifies the use of the AEAD_AES_128_GCM and AEAD_AES_256_GCM algorithms in SSH. As described in RFC 5647, AEAD_AES_128_GCM and AEAD_AES_256_GCM can only be chosen as encryption algorithms when the same algorithm is being used as the MAC algorithm. If AES-GCM is selected, there should be corresponding FCS_COP entries in the ST.

FCS_SSHC_EXT.1.4 The SSH client shall ensure that the SSH transport implementation uses [**selection**: *ssh-rsa*, *rsa-sha2-256*, *rsa-sha2-512*, *ecdsa-sha2-nistp256*] and [**selection**: *ecdsa-sha2-nistp384*, *x509v3-ecdsa-sha2-nistp256*, *x509v3-ecdsa-sha2-nistp384*, *no other public key algorithms*] as its public key algorithm(s) and rejects all other public key algorithms.

Application Note: Implementations that select only ssh-rsa will not achieve the 112-bit security strength in the digital signature generation for SSH authentication as is recommended in NIST SP 800-131A. Future versions of this document may remove ssh-rsa as a selection. If "x509v3-ecdsa-sha2-nistp256" or "x509v3-ecdsa-sha2-nistp384" are selected, then the list of trusted certification authorities must be selected in FCS_SSHC_EXT.1.8. RFC 8332 specifies the use of rsa-sha2-256 or rsa-sha2-512 in SSH.

The SFRs for cryptographic key generation and certificate validation are inherited from the PP or PP-Module that includes this Package.

FCS_SSHC_EXT.1.5 The SSH client shall ensure that the SSH transport implementation uses [**selection**: *hmac-sha1*, *hmac-sha1-96*, *hmac-sha2-256*, *hmac-sha2-512*] and [**selection**: *AEAD_AES_128_GCM*, *AEAD_AES_256_GCM*, *no other MAC algorithms*] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

Application Note: RFC 5647 specifies the use of the AEAD_AES_128_GCM and AEAD_AES_256_GCM algorithms in SSH. As described in RFC 5647, AEAD_AES_128_GCM and AEAD_AES_256_GCM can only be chosen as MAC algorithms when the same algorithm is being used as the encryption algorithm. RFC 6668 specifies the use of the sha2 algorithms in SSH. The SFRs for cryptographic operations, encryption, and hashing are inherited from the PP or PP-Module that includes this Package.

FCS_SSHC_EXT.1.6 The SSH client shall ensure that [**selection**: *diffie-hellman-group14-sha1*, *ecdh-sha2-nistp256*] and [**selection**: *ecdh-sha2-nistp384*, *ecdh-sha2-nistp521*, *no other methods*] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHC_EXT.1.7 The SSH client shall ensure that the SSH connection be rekeyed after [**selection**: *no more than 2^{28} packets have been transmitted*, *no more than 1 gigabyte of data has been transmitted* , *no more than 1 hour*] using that key.

FCS_SSHC_EXT.1.8 The SSH client shall ensure that the SSH client authenticates the identity of the SSH server using a local database associating each host name with its corresponding public key or [**selection**: *a list of trusted certification authorities* , *no other methods*] as described in RFC 4251 section 4.1.

Application Note: The selection for "a list of trusted certification authorities" can only be chosen if "x509v3-ecdsa-sha2-nistp256" or "x509v3-ecdsa-sha2-nistp384" are selected in FCS_SSHC_EXT.1.4.

SSH Protocol - Server

FCS_SSHS_EXT.1.1 The SSH server shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based and [**selection**: *password-based*, *no other method*].

FCS_SSHS_EXT.1.2 The SSH server shall ensure that, as described in RFC 4253, packets greater than [**assignment**: *number of bytes*] bytes in an SSH transport connection are dropped.

Application Note: RFC 4253 provides for the acceptance of “large packets” with the caveat that the packets should be of “reasonable length” or dropped. The assignment should be filled in by the ST author with the maximum packet size accepted, thus defining “reasonable length” for the TOE.

FCS_SSHS_EXT.1.3 The SSH server shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: aes128-ctr, aes256-ctr, [**selection**: *aes128-cbc*, *aes256-cbc*, *AEAD_AES_128_GCM*, *AEAD_AES_256_GCM*, *no other algorithms*].

Application Note: RFC 5647 specifies the use of the AEAD_AES_128_GCM and AEAD_AES_256_GCM algorithms in SSH. As described in RFC 5647, AEAD_AES_128_GCM

and AEAD_AES_256_GCM can only be chosen as encryption algorithms when the same algorithm is being used as the MAC algorithm.

FCS_SSHS_EXT.1.4 The SSH server shall ensure that the SSH transport implementation uses [**selection**: *ssh-rsa*, *rsa-sha2-256*, *rsa-sha2-512*, *ecdsa-sha2-nistp256*] and [**selection**: *ecdsa-sha2-nistp384*, *x509v3-ecdsa-sha2-nistp256*, *x509v3-ecdsa-sha2-nistp384*, *no other public key algorithms*] as its public key algorithm(s) and rejects all other public key algorithms.

Application Note: Implementations that select only ssh-rsa will not achieve the 112-bit security strength in the digital signature generation for SSH authentication as is recommended in NIST SP 800-131A. Future versions of this document may remove ssh-rsa as a selection. RFC 8332 specifies the use of rsa-sha2-256 or rsa-sha2-512 in SSH. The SFRs for cryptographic key generation and certificate validation are inherited from the PP or PP-Module that includes this Package.

FCS_SSHS_EXT.1.5 The SSH server shall ensure that the SSH transport implementation uses [**selection**: *hmac-sha1*, *hmac-sha1-96*, *hmac-sha2-256*, *hmac-sha2-512*] and [**selection**: *AEAD_AES_128_GCM*, *AEAD_AES_256_GCM*, *no other MAC algorithms*] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

Application Note: RFC 5647 specifies the use of the AEAD_AES_128_GCM and AEAD_AES_256_GCM algorithms in SSH. As described in RFC 5647, AEAD_AES_128_GCM and AEAD_AES_256_GCM can only be chosen as MAC algorithms when the same algorithm is being used as the encryption algorithm. RFC 6668 specifies the use of the sha2 algorithms in SSH. The SFRs for cryptographic operations, encryption and hashing, are inherited from the PP or PP-Module that includes this Package.

FCS_SSHS_EXT.1.6 The SSH server shall ensure that [**selection**: *diffie-hellman-group14-sha1*, *ecdh-sha2-nistp256*] and [**selection**: *ecdh-sha2-nistp384*, *ecdh-sha2-nistp521*, *no other methods*] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHS_EXT.1.7 The SSH server shall ensure that the SSH connection be rekeyed after [**selection**: *no more than 2^{28} packets have been transmitted*, *no more than 1 gigabyte of data has been transmitted*, *no more than 1 hour*] using that key.

Objective Security Functional Requirements

Optional Security Functional Requirements