Functional Package for Secure Shell (SSH)

Version: 1.0-ccuf
2021-03-11
National Information Assurance Partnership

Revision History

Version	Date	Comment
1.0 draft	2021-01-05	DRAFT: First publication as a Functional Package

Contents

1 Introduction

1.1 Overview

Secure Shell (SSH) is a protocol for secure remote login and other secure network services over an untrusted network. SSH software can act as a client, server, or both.

This *Functional Package for Secure Shell* provides a collection of Secure Shell (SSH) protocol related SFRs and Evaluation Activities (EAs) covering audit, authentication, cryptographic algorithms, and protocol negotiation. The intent of this package is to provide PP, cPP, and PP-Module authors with a readily consumable collection of SFRs and EAs to be integrated into their documents.

1.2 Terms

The following sections list Common Criteria and technology terms used in this document.

1.2.1 Common Criteria Terms

Assurance	Grounds for confidence that a TOE meets the SFRs [CC].
Base Protection Profile (Base-PP)	Protection Profile used as a basis to build a PP-Configuration.
Common Criteria (CC)	Common Criteria for Information Technology Security Evaluation (International Standard ISO/IEC 15408).
Common Criteria Testing Laboratory	Within the context of the Common Criteria Evaluation and Validation Scheme (CCEVS), an IT security evaluation facility, accredited by the National Voluntary Laboratory Accreditation Program (NVLAP) and approved by the NIAP Validation Body to conduct Common Criteria-based evaluations.
Common Evaluation Methodology (CEM)	Common Evaluation Methodology for Information Technology Security Evaluation.
Distributed TOE	A TOE composed of multiple components operating as a logical whole.
Operational Environment (OE)	Hardware and software that are outside the TOE boundary that support the TOE functionality and security policy.
Protection Profile (PP)	An implementation-independent set of security requirements for a category of products.
Protection Profile Configuration (PP-Configuration)	A comprehensive set of security requirements for a product type that consists of at least one Base-PP and at least one PP-Module.
Protection Profile Module (PP-Module)	An implementation-independent statement of security needs for a TOE type complementary to one or more Base Protection Profiles.
Security Assurance Requirement (SAR)	A requirement to assure the security of the TOE.
Security Functional Requirement (SFR)	A requirement for security enforcement by the TOE.
Security Target (ST)	A set of implementation-dependent security requirements for a specific product.
TOE Security Functionality	The security functionality of the product under evaluation.

(TSF)	
TOE Summary Specification (TSS)	A description of how a TOE satisfies the SFRs in an ST.
Target of Evaluation (TOE)	The product under evaluation.

1.2.2 Technical Terms

Secure Shell (SSH)	Cryptographic network protocol for initiating text-based shell sessions on remote systems.

1.3 Compliant Targets of Evaluation

The Target of Evaluation (TOE) in this Functional Package (FP) is a product which acts as an SSH client, SSH server, or both. This FP describes the extended security functionality of SSH in terms of [CC].

The contents of this Functional Package must be appropriately incoporated into a PP, cPP, or PP-Module. When this package is so incorporated, the ST must include selection-based requirements in accordance with the selections or assignments indicated in the incorporating document.

The PP, cPP, or PP-Module that instantiates this Package must typically include the following components in order to satisfy dependencies of this Package. It is the responsibility of the PP, cPP, or PP-Module author who incorporates this FP to ensure that dependence on these components is satisfied, either by the TOE or by assumptions about its Operational Environment.

An ST must identify the applicable version of the PP, cPP, or PP-Module and this Functional Package in its conformance claims.

Component	Explanation
FCS_CKM.1	To support key generation for SSH, the incoporating document must include FCS_CKM.1 and specify the corresponding algorithm(s).
FCS_CKM.2	To support key establishment for SSH, the incoporating document must include FCS_CKM.2 and specify the corresponding algorithm(s).
FCS_COP.1	To support the cryptography needed for SSH communications, the incoporating document must include FCS_COP.1 (iterating as needed) to specify AES with corresponding key sizes and modes, digital signature generation and verification function (at least one of RSA or ECDSA), a cryptographic hash function, and a keyed-hash message authentication function. In particular, the incoporating document must support AES-CTR as defined in NIST SP 800-38A with key sizes of both 128 and 256 bits.
FCS_RBG_EXT.1	To support random bit generation needed for SSH key generation, the incorporating document must include a requirement that specifies the TOE's ability to invoke or provide random bit generation services, commonly identified as FCS_RBG_EXT.1.
FIA_X509_EXT.1	To support establishment of SSH communications using a public key algorithm that includes X.509, the incorporating document must include FIA_X509_EXT.1. Note however that support for X.509 is selectable and not mandatory.
FIA_X509_EXT.2	To support establishment of SSH communications using a public key algorithm that includes X.509, the incoporating document must include FIA_X509_EXT.2. Note however that support for X.509 is selectable and not mandatory.
FPT_STM.1	To support establishment of SSH communications using a public key algorithm that includes X.509, the incorporating document must include FPT_STM.1 or some other requirement that ensures reliable system time. Note however that support for time-based rekey thresholds is selectable and not mandatory.

2 Conformance Claims

Conformance Statement

An ST must claim exact conformance to this Package, as defined in the CC and CEM addenda for Exact Conformance, Selection-Based SFRs, and Optional SFRs (dated May 2017).

CC Conformance Claims

This Package is conformant to Parts 2 (extended) and 3 (conformant) of Common Criteria Version 3.1, Revision 5.

PP Claim

This Package does not claim conformance to any Protection Profile.

Package Claim

This Package does not claim conformance to any packages.

3 Security Functional Requirements

This chapter describes the security requirements which have to be fulfilled by the product under evaluation. Those requirements comprise functional components from Part 2 of [CC]. The following conventions are used for the completion of operations:

- **Refinement** operation (denoted by **bold text** or ~~strikethrough text~~): is used to add details to a requirement (including replacing an assignment with a more restrictive selection) or to remove part of the requirement that is made irrelevant through the completion of another operation, and thus further restricts a requirement.
- **Selection** (denoted by *italicized text*): is used to select one or more options provided by the [CC] in stating a requirement.
- **Assignment** operation (denoted by *italicized text*): is used to assign a specific value to an unspecified parameter, such as the length of a password. Showing the value in square brackets indicates assignment.
- **Iteration** operation: is indicated by appending the SFR name with a slash and unique identifier suggesting the purpose of the operation, e.g. "/EXAMPLE1."

3.1 Auditable Events for Mandatory SFRs

The auditable events specified in this Package are included in a Security Target if the incorporating PP, cPP, or PP-Module supports audit event reporting through FAU_GEN.1 and all other criteria in the incorporating PP or PP-Module are met.

Table 1: Auditable Events for Mandatory Requirements

Requirement	Auditable Events	Additional Audit Record Contents
FCS_SSH_EXT.1	**[selection:** *Failure to establish SSH connection, None***]**	Reason for failure. Non-TOE endpoint of attempted connection (IP Address)
FCS_SSH_EXT.1	**[selection:** *Establishment of SSH connection, None***]**	Non-TOE endpoint of connection (IP Address)
FCS_SSH_EXT.1	**[selection:** *Termination of SSH connection session, None***]**	Non-TOE endpoint of connection (IP Address)
FCS_SSH_EXT.1	**[selection:** *Dropping of packet(s) outside defined size limits, None***]**	Packet size

3.2 Cryptographic_Support_(FCS)

FCS_SSH_EXT.1 SSH Protocol

FCS_SSH_EXT.1.1

The TOE shall implement *SSH* acting as a [**selection**: *client*, *server*] in accordance with that complies with RFCs 4251, 4252, 4253, 4254, [**selection**: *4256*, *4344*, *5647*, *5656*, *6187*, *6668*, *8268*, *8308*, *8332*, *8731*, *no other RFCs*] and [*no other standard*].

Application Note: The following mapping is provided as a guide to ST authors to ensure the appropriate RFC selections are made based on applicable selections in subsequent SFRs:

- RFC 4256: Select for keyboard-interactive authentication
- RFC 4344: Select for AES-128-CTR or AES-256-CTR
- RFC 5647: Select for AEAD_AES_128_GCM, AEAD_AES_256_GCM, or aes*-gcm@openssh.com
- RFC 5656: Select for elliptic curve cryptography
- RFC 6187: Select for X.509 certificate use
- RFC 6668: Select for HMAC-SHA-2 algorithms
- RFC 8268: Select for FFC DH groups with SHA-2
- RFC 8308: Select if RFC 8332 is selected
- RFC 8332: Select if SHA-2 is available with ssh-rsa

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 must be consistent with selections in later elements of this Functional Package (e.g., cryptographic algorithms permitted).

For the purposes of this package (and subsequent integration into cPPs) only the claimed algorithms listed in the package must be enabled for use.

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 4344 must be selected if aes128-ctr or aes256-ctr is selected in FCS_SSH_EXT.1.4.

RFC 4356 must be selected if "keyboard-interactive" is selected in FCS_SSH_EXT.1.2.

RFC 5647 must be selected when AEAD_AES_128_GCM, AEAD_AES_256_GCM, aes128-gcm@openssh.com, or aes256-gcm@openssh.com is selected as an encryption algorithm in FCS_SSH_EXT.1.4 and when AEAD_AES_128_GCM or AEAD_AES_256_GCM is selecyed as MAC algorithm in FCS_SSH_EXT.1.5.

RFC 5656 must be selected when ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521 is selected as a public key algorithm in FCS_SSH_EXT.1.2, or when ecdh-sha2-nistp256, ecdh-sha2-nistp384, or ecdh-sha2-nistp521 is selected as a key exchange algorithm in FCS_SSH_EXT.1.6, or when "RFC 5656" is selected in FCS_SSH_EXT.1.7.

RFC 6187 must be selected when x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp521, or x509v3-rsa2048-sha256 is selected as a public key algorithm in FCS_SSH_EXT.1.2.

RFC 6668 must be selected when hmac-sha2-256 or hmac-sha2-512 is selected as a MAC algorithm in FCS_SSH_EXT.1.5.

RFC 8268 must be selected when diffie-hellman-group14-sha256, diffie-hellman-group15-sha512, diffie-hellman-group16-sha512, diffie-hellman-group17-sha512, or diffie-hellman-group18-sha512 is selected as a key exchange algorithm in FCS_SSH_EXT.1.6.

RFC 8332 must be selected when rsa-sha2-256 or rsa-sha2-512 is selected as a public key algorithm in FCS_SSH_EXT.1.2.

RFC 8709 must be selected when ssh-ed25519 or ssh-ed448 is selected as a public key algorithm in FCS_SSH_EXT.1.2.

RFC 8731 must be selected when curve25519-sha256 or curve448-sha512 is selected as a key exchange algorithm in FCS_SSH_EXT.1.6.

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

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

FCS_SSH_EXT.1.2

The TSF shall ensure that the SSH protocol implementation supports the following authentication methods: [**selection**:

- *“password” (RFC 4252),*
- *“keyboard-interactive” (RFC 4256),*
- *“publickey” (RFC 4252): [**selection**:*
  - *ssh-rsa (RFC 4253),*
  - *rsa-sha2-256 (RFC 8332),*
  - *rsa-sha2-512 (RFC 8332),*
  - *ecdsa-sha2-nistp256 (RFC 5656),*
  - *ecdsa-sha2-nistp384 (RFC 5656),*
  - *ecdsa-sha2-nistp521 (RFC 5656),*
  - *ssh-ed25519 (RFC 8709),*
  - *ssh-ed448 (RFC 8709),*
  - *x509v3-ecdsa-sha2-nistp256 (RFC 6187),*
  - *x509v3-ecdsa-sha2-nistp384 (RFC 6187),*
  - *x509v3-ecdsa-sha2-nistp521 (RFC 6187),*
  - *x509v3-rsa2048-sha256 (RFC 6187)*

]

].

Application Note: Within SSH there are two types of authentication: user authentication and peer authentication. This SFR deals with the options supported for user authentication. Peer authentication is covered in FCS_SSHS_EXT.1.1 (for servers) and FCS_SSHC_EXT.1.1 (for clients).

FCS_SSH_EXT.1.3

The TSF shall ensure that, as described in RFC 4253, packets greater than [**assignment**: *number of bytes between 35,000 and 1 GB (inclusive)*] in an SSH transport connection are dropped.

Application Note: RFC 4253 (section 6.1) 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.

The upper bound on the packet size is driven by the size identified in FCS_SSH_EXT.1.8.

FCS_SSH_EXT.1.4

The TSF shall protect data in transit from unauthorised disclosure using one of the following mechanisms: [**selection**:

- *aes128-ctr (RFC 4344)*,
- *aes256-ctr (RFC 4344)*,
- *aes128-cbc (RFC 4253)*,
- *aes256-cbc (RFC 4253)*,
- *AEAD_AES_128_GCM (RFC 5647)*,
- *AEAD_AES_256_GCM (RFC 5647)*,
- *aes128-gcm@openssh.com (RFC 5647)*,
- *aes256-gcm@openssh.com (RFC 5647)*

].

Application Note: As described in RFC 5647, AEAD_AES_128_GCM and AEAD_AES_256_GCM need the corresponding MAC algorithm to be selected in FCS_SSH_EXT.1.5.

FCS_SSH_EXT.1.5

The TSF shall protect data in transit from modification, deletion, and insertion using one of the following mechanisms: [**selection**:

- *hmac-sha1 (RFC 4253)*,
- *hmac-sha2-256 (RFC 6668)*,
- *hmac-sha2-512 (RFC 6668)*,
- *AEAD_AES_128_GCM (RFC 5647)*,
- *AEAD_AES_256_GCM (RFC 5647)*,
- *implicit*

].

Application Note: As described in RFC 5647, AEAD_AES_128_GCM and AEAD_AES_256_GCM need the corresponding encryption algorithm to be selected. In AES-GCM mode, integrity is not provided using a MAC, it is implicit in AES-GCM mode itself. There is no need for a corresponding FCS_COP element. The FCS_COP element for AES would already cover this.

If the negotiated encryption algorithm is one of the aes*-gcm@openssh.com algorithms, then the MAC field is ignored during negotiation and implicitly selects AES-GCM for the MAC. “implicit” is not an SSH identifier and will not be seen on the wire; however, the negotiated MAC might be decoded as “implicit”.

FCS_SSH_EXT.1.6

The TSF shall establish a shared secret with its peer using one of the following mechanisms: [**selection**:

- *diffie-hellman-group14-sha1 (RFC 4253)*,
- *diffie-hellman-group14-sha256 (RFC 8268)*,
- *diffie-hellman-group15-sha512 (RFC 8268)*,
- *diffie-hellman-group16-sha512 (RFC 8268)*,
- *diffie-hellman-group17-sha512 (RFC 8268)*,
- *diffie-hellman-group18-sha512 (RFC 8268)*,
- *ecdh-sha2-nistp256 (RFC 5656)*,
- *ecdh-sha2-nistp384 (RFC 5656)*,
- *ecdh-sha2-nistp521 (RFC 5656)*,
- *curve25519-sha256 (RFC 8731)*,
- *curve448-sha512 (RFC 8731)*

] and no other mechanisms.

FCS_SSH_EXT.1.7

The TSF shall use *SSH KDF* as defined in [**selection**:

- *RFC 4253 (Section 7.2)*,
- *RFC 5656 (Section 4)*

] to derive the following cryptographic keys from a shared secret: *session keys*.

Application Note: RFC 4253 must be selected when the key establishment scheme (selected in FCS_SSH_EXT.1.6) uses finite field cryptography (FFC) and RFC 5656 when it uses elliptic curve cryptography (ECC).

RFC 4253 section 7.2 defines two KDFs for FFC based key establishment schemes. Therefore RFC 4253 should be selected if any of the RFC 4253 or RFC 8268 key establishment schemes are selected.

RFC 5656 section 4 defines KDFs used in ECC key establishment schemes and should be selected when RFC 5656 or RFC 8731 key establishment schemes are selected.

FCS_SSH_EXT.1.8

The TSF shall ensure that [**selection**:

- *a rekey of the session keys*,
- *connection termination*

] occurs when any of the following thresholds are met:

- one hour connection time
- no more than one gigabyte of transmitted data, or
- no more than one gigabyte of received data.

Application Note: This SFR defines three thresholds that need to be implemented. These thresholds were arrived at to ensure that the cryptographic key space for the symmetric session keys isn't exhausted (more detail can be found in RFC 4344 and RFC 4253). A rekey or connection termination needs to be performed whenever a threshold is reached for a given connection. The rekey applies to all session keys (encryption, integrity protection) for incoming and outgoing traffic.

It is acceptable for a TOE to implement lower thresholds than the maximum values defined in the SFR. If a threshold is configurable, the guidance documentation needs to specify how to configure that threshold.

It is possible that hardware limitation may prevent reaching data transfer threshold in less than one hour. In cases where data transfer threshold could not be reached due to hardware limitations it is acceptable to omit testing of this (SSH rekeying based on data transfer threshold). See Evaluation Activities for details.

Evaluation Activities

FCS_SSH_EXT.1:
TSS
The evaluator shall ensure that the selections indicated in the ST are consistent with selections in this and subsequent components. Otherwise, this SFR is evaluated by activities for other SFRs.
Guidance
There are no guidance evaluation activities for this component. This SFR is evaluated by activities for other SFRs
Tests
There are no test evaluation activities for this component. This SFR is evaluated by activities for other SFRs

FCS_SSHC_EXT.1 SSH Protocol - Client

This is a selection-based component. Its inclusion depends upon selection from .

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*, *aes128@openssh.com*, *aes256@openssh.com*, *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.

RFC 5647 applies only to the RFC-compliant implementation of GCM. A TOE that implements only the “@openssh.com” variant of GCM should not select 5647-compliant algorithms in FCS_SSHC_EXT.1.1. aes*-gcm@openssh.com is specified in Section 1.6 of the OpenSSH Protocol Specification (https://cvsweb.openbsd.org/cgi-bin/cvsweb/src/usr.bin/ssh/PROTOCOL?rev=1.31).

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*, *implicit*, *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.

The ST author selects “implicit” if and only if aes*-gcm@openssh.com is selected as an encryption algorithm. When aes*-gcm@openssh.com is negotiated as the encryption algorithm, the MAC algorithm field is ignored and GCM is implicitly used as the MAC. “implicit” is not an SSH algorithm identifier and will not be seen on the wire; however, the negotiated MAC might be decoded as “implicit”.

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.

Evaluation Activities

FCS_SSHC_EXT.1:

TSS

The evaluator shall check to ensure that the TSS contains a description of the public key algorithms that are acceptable for use for authentication and that this list conforms to FCS_SSHC_EXT.1.4. The evaluator shall also ensure that password-based authentication methods, if supported, are described.

Guidance

If the SSH client supports password-based authentication, the evaluator shall examine the guidance to determine that it includes instructions on how to configure whether the TSF uses password-based or public key-based authentication.

Tests

- ***Test 1:*** *The evaluator shall, for each public key algorithm supported, show that the TOE supports the use of that public key algorithm to authenticate a user connection to an SSH server. Any configuration activities required to support this test shall be performed according to instructions in the guidance documentation.*
- ***Test 2:*** *[conditional]: TOE supports password-based authentication] Using the guidance documentation, the evaluator shall configure the TOE to perform password-based authentication to an SSH server, and demonstrate that a user can be successfully authenticated by the TOE to an SSH server using a password as an authenticator.*

TSS

The evaluator shall check that the TSS describes how "large packets" in terms of RFC 4253 are detected and handled.

Guidance

There are no guidance evaluation activities for this element.

Tests

The evaluator shall perform the following test:

- ***Test 1:*** *The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this element, the packet is dropped.*

TSS

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that it specifies the supported encryption algorithms and any optional characteristics. The evaluator shall also check the TSS to ensure that the encryption algorithms specified are identical to those listed for this element.

Guidance

The evaluator shall check the guidance documentation to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements).

Tests

The evaluator shall perform the following tests:

- ***Test 1:*** *The evaluator shall establish an SSH connection using each of the encryption algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of this test.*
- ***Test 2:*** *The evaluator shall configure an SSH server to only allow the 3des-cbc encryption algorithm and no other encryption algorithms. The evaluator shall attempt to establish an SSH connection from the TOE to the SSH server and observe that the connection is rejected.*

TSS

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that it specifies the supported public key algorithms and any optional characteristics. The evaluator shall also check the TSS to ensure that the encryption algorithms specified are identical to those listed for this element.

Guidance

The evaluator shall check the guidance documentation to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements).

Tests

The evaluator shall perform the following tests:

- ***Test 1:*** *The evaluator shall establish a SSH connection using each of the public key algorithms specified by the requirement to authenticate an SSH server to the TOE. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.*
- ***Test 2:*** *The evaluator shall configure an SSH server to only allow the ssh-dsa public key algorithm and no other public key algorithms. The evaluator shall attempt to establish an SSH connection from the TOE to the SSH server and observe that the connection is rejected.*

TSS

The evaluator shall check the TSS to ensure that it lists the supported data integrity algorithms and that this list corresponds to the list in this element.

Guidance

The evaluator shall check the guidance documentation to ensure that it includes instructions to the administrator on how to ensure that only the allowed data integrity algorithms are used in SSH connections with the TOE (specifically, that the "none" MAC algorithm is not allowed).

Tests

The evaluator shall perform the following tests:

- ***Test 1:*** *The evaluator shall establish a SSH connection using each of the integrity algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of this test.*
- ***Test 2:*** *The evaluator shall configure an SSH server to only allow the "none" MAC algorithm. The evaluator shall attempt to connect from the TOE to the SSH server and observe that the attempt fails.*
- ***Test 3:*** *The evaluator shall configure an SSH server to only allow the hmac-md5 MAC algorithm. The evaluator shall attempt to connect from the TOE to the SSH server and observe that the attempt fails. To ensure the proposed MAC algorithm is used, the evaluator shall ensure a non-aes*- gcm@openssh.com encryption algorithm is negotiated while performing this test.*

TSS

The evaluator shall check the TSS to ensure that it lists the supported key exchange algorithms and that this list corresponds to the list in this element.

Guidance

The evaluator shall check the guidance documentation to ensure that it includes instructions to the administrator on how to ensure that only the allowed key exchange algorithms are used in SSH connections with the TOE.

Tests

The evaluator shall perform the following test:

- ***Test 1:*** *The evaluator shall configure an SSH server to permit all allowed key exchange methods. The evaluator shall then attempt to connect from the TOE to the SSH server using each allowed key exchange method and observe that each attempt succeeds.*

TSS

There are no TSS evaluation activities for this element.

Guidance

There are no guidance evaluation activities for this element.

Tests

The evaluator shall perform the following test for each rekeying method claimed in the ST:

The evaluator shall perform the following test:

- ***Test 1:*** *The evaluator will configure the TOE to create a log entry when a rekey occurs. The evaluator will connect to the TOE with an SSH client and cause a rekey to occur according to the selection(s) in the ST, and subsequently the evaluator uses available methods and tools to verify that rekeying occurs. This could be done, e.g., by checking that a corresponding audit event has been generated by the TOE, if the TOE supports auditing of rekey events.*

TSS

There are no TSS evaluation activities for this element.

Guidance

There are no guidance evaluation activities for this element.

Tests

The evaluator shall perform the following tests:

- ***Test 1:*** *The evaluator shall delete all entries in the TOE's list of recognized SSH server host keys and, if selected, all entries in the TOE's list of trusted certification authorities. The evaluator shall then initiate a connection from the TOE to an SSH server. The evaluator shall ensure that the TOE either rejects the connection or displays the SSH server's public key (either the key bytes themselves or a hash of the key using any allowed hash algorithm) and prompts the user to accept or deny the key before continuing the connection.*
- ***Test 2:*** *The evaluator shall add an entry associating a host name with a public key into the TOE's local database. The evaluator shall then replace, on the corresponding SSH server, the server's host key with a different host key. The evaluator shall initiate a connection from the TOE to the SSH server using password-based authentication, shall ensure that the TOE rejects the connection, and shall ensure that the password was not transmitted to the SSH server (for example, by instrumenting the SSH server with a debugging capability to output received passwords).*

FCS_SSHS_EXT.1 SSH Protocol - Server

This is a selection-based component. Its inclusion depends upon selection from .

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*, *aes128-gcm@openssh.com*, *aes256-gcm@openssh.com*, *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.

RFC 5647 applies only to the RFC compliant implementation of GCM. A TOE that implements only the “@openssh.com” variant of GCM should not select 5647-compliant algorithms in FCS_SSHS_EXT.1.1. aes*-gcm@openssh.com is specified in Section 1.6 of the OpenSSH Protocol Specification (https://cvsweb.openbsd.org/cgi-bin/cvsweb/src/usr.bin/ssh/PROTOCOL?rev=1.31).

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*, *implicit*, *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.

The ST author selects “implicit” if and only if aes*-gcm@openssh.com is selected as an encryption algorithm. When aes*-gcm@openssh.com is negotiated as the encryption algorithm, the MAC algorithm field is ignored and GCM is implicitly used as the MAC. “implicit” is not an SSH algorithm identifier and will not be seen on the wire; however, the negotiated MAC might be decoded as “implicit”.

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.

Evaluation Activities

FCS_SSHS_EXT.1:
TSS
The evaluator shall check to ensure that the TSS contains a description of the public key algorithms that are acceptable for use for authentication and that this list conforms to FCS_SSHS_EXT.1.4. The evaluator shall also ensure that password-based authentication methods, if supported, are described.
Guidance
If the SSH server supports password-based authentication, the evaluator shall examine the guidance to determine that it includes instructions on how to configure whether the TSF uses password-based or public key-based authentication.
Tests
The evaluator shall perform the following tests:

- ***Test 1:*** *The evaluator shall, for each public key algorithm supported, show that the TOE supports the use of that public key algorithm to authenticate a user connection from an SSH client. Any configuration activities required to support this test shall be performed according to instructions in the guidance documentation.*
- ***Test 2:*** *The evaluator shall choose one public key algorithm supported by the TOE. The evaluator shall generate a new key pair for that algorithm without configuring the TOE to recognize the public key for authentication. The evaluator shall use an SSH client to attempt to connect to the TOE with the new key pair and demonstrate that authentication fails.*
- ***Test 3:*** *[conditional]: TOE supports password-based authentication] Using the guidance documentation, the evaluator shall configure the TOE to perform password-based authentication on a client and demonstrate that a user can be successfully authenticated by the TOE using a password as an authenticator.*
- ***Test 4:*** *[conditional]: TOE supports password-based authentication] The evaluator shall use an SSH client to enter an incorrect password to attempt to authenticate to the TOE and demonstrate that the authentication fails.*

TSS
The evaluator shall check that the TSS describes how "large packets" in terms of RFC 4253 are detected and handled.
Guidance
There are no guidance evaluation activities for this element.
Tests
The evaluator shall perform the following test:

- ***Test 1:*** *The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this element, the packet is dropped.*

TSS
The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that it specifies the supported encryption algorithms and any optional characteristics. The evaluator shall also check the TSS to ensure that the encryption algorithms specified are identical to those listed for this element.
Guidance
The evaluator shall check the guidance documentation to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements).
Tests
The evaluator shall perform the following tests:

- ***Test 1:*** *The evaluator shall initiate an SSH connection using each of the encryption algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of this test.*
- ***Test 2:*** *The evaluator shall configure an SSH client to only propose the 3des-cbc encryption algorithm and no other encryption algorithms. The evaluator shall attempt to establish an SSH connection from this client to the TOE server and observe that the connection is rejected.*

TSS
The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that it specifies the supported public key algorithms and any optional characteristics. The evaluator shall also check the TSS to ensure that the encryption algorithms specified are identical to those listed for this element.
Guidance
The evaluator shall check the guidance documentation to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements).
Tests

The evaluator shall perform the following tests:

- ***Test 1:*** *Using an appropriately configured client, the evaluator shall establish an SSH connection using each of the public key algorithms specified by the requirement to authenticate to the TOE. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of this test.*
- ***Test 2:*** *The evaluator shall configure an SSH client to propose only the ssh-dsa public key algorithm and no other public key algorithms. Using this client, the evaluator shall attempt to establish an SSH connection to the TOE and observe that the connection is rejected.*

TSS
The evaluator shall check the TSS to ensure that it lists the supported data integrity algorithms and that this list corresponds to the list in this element.
Guidance
The evaluator shall check the guidance documentation to ensure that it includes instructions to the administrator on how to ensure that only the allowed data integrity algorithms are used in SSH connections with the TOE (specifically, that the "none" and "hmac-md5" MAC algorithms are not allowed).
Tests
The evaluator shall perform the following tests:

- ***Test 1:*** *Using an appropriately configured client, the evaluator shall establish a SSH connection with the TOE using each of the integrity algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.*
- ***Test 2:*** *The evaluator shall configure an SSH client to only propose the "none" MAC algorithm. Using this client, the evaluator shall attempt to connect to the TOE and observe that the attempt fails.*
- ***Test 3:*** *The evaluator shall configure an SSH client to only propose the hmac-md5 MAC algorithm. Using this client, the evaluator shall attempt to connect to the TOE and observe that the attempt fails. To ensure the proposed MAC algorithm is used, the evaluator shall ensure a non-aes*- gcm@openssh.com encryption algorithm is negotiated while performing this test.*

TSS
The evaluator shall check the TSS to ensure that it lists the supported key exchange algorithms and that this list corresponds to the list in this element.
Guidance
The evaluator shall check the guidance documentation to ensure that it includes instructions to the administrator on how to ensure that only the allowed key exchange algorithms are used in SSH connections to the TOE.
Tests
The evaluator shall perform the following tests:

- ***Test 1:*** *For each of the allowed key exchange methods, the evaluator shall configure an SSH client to propose only that method and then attempt to connect to the TOE. The evaluator shall confirm that each attempt succeeds.*
- ***Test 2:*** *The evaluator shall configure an SSH client to only allow the diffie-hellman-group1-sha1 key exchange. The evaluator shall attempt to use this SSH client to connect to the TOE and confirm that this attempt fails.*

TSS
There are no TSS evaluation activities for this element.
Guidance
If the TOE has the ability to generate a log when an SSH rekey occurs, the evaluator shall examine the operational guidance to verify that it describes any configuration that is needed for this to be performed.
Tests
The evaluator shall perform the following test for each rekeying method claimed in the ST:

The evaluator shall perform the following test:

- ***Test 1:*** *The evaluator will configure the TOE to create a log entry when a rekey occurs. The evaluator will connect to the TOE with an SSH client and cause a rekey to occur according to the selection(s) in the ST, and subsequently the evaluator uses available methods and tools to verify that rekeying occurs. This could be done, e.g., by checking that a corresponding audit event has been generated by the TOE, if the TOE supports auditing of rekey events.*

Appendix A - Implementation-Dependent Requirements

Implementation-Dependent Requirements are dependent on the TOE implementing a particular function. If the TOE fulfills any of these requirements, the vendor must either add the related SFR or disable the functionality for the evaluated configuration.

Appendix B - References

ext-comp-def

Identifier	Title
[CC]	Common Criteria for Information Technology Security Evaluation - • Part 1: Introduction and General Model, CCMB-2017-04-001, Version 3.1, Revision 5, April 2017. • Part 2: Security Functional Components, CCMB-2017-04-002, Version 3.1, Revision 5, April 2017. • Part 3: Security Assurance Components, CCMB-2017-04-003, Version 3.1, Revision 5, April 2017.
[GPOSPP]	Protection Profile for General Purpose Operating Systems
[MDMPP]	Protection Profile for Mobile Device Management
[AppPP]	Protection Profile for Application Software
[VirtPP]	Protection Profile for Virtualization

Appendix C - Acronyms

Acronym	Meaning
Base-PP	Base Protection Profile
CC	Common Criteria
CEM	Common Evaluation Methodology
OE	Operational Environment
PP	Protection Profile
PP-Configuration	Protection Profile Configuration
PP-Module	Protection Profile Module
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SSH	Secure Shell
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TSF Interface
TSS	TOE Summary Specification