Assurance Activities Report

for

VMware ESXi 7.0 Update 3d

Version 1.0

28 July 2022

Prepared by:

Leidos Inc.
https://www.leidos.com/CC-FIPS140
Common Criteria Testing Laboratory
6841 Benjamin Franklin Drive
Columbia, MD 21046

Prepared for:

National Information Assurance Partnership
Common Criteria Evaluation and Validation Scheme

The Developer of the TOE:

VMware, Inc.
3401 Hillview Avenue
Palo Alto, CA 94304

The TOE Evaluation was Sponsored by:

VMware, Inc.
3401 Hillview Avenue
Palo Alto, CA 94304

Evaluation Personnel:

Anthony Apted
Pascal Patin

Common Criteria Version:

- Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model, Version 3.1, Revision 5, April 2017.
- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017.
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017.

Common Evaluation Methodology Version:

- Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, Version 3.1, Revision 5, April 2017.

Protection Profile:

- *PP-Configuration for Virtualization and Server Virtualization Systems*, Version 1.0, 4 Jun 2021, consisting of:
    - *Protection Profile for Virtualization*, Version 1.1, 14 Jun 2021 [PP_V]
    - *PP-Module for Server Virtualization*, Version 1.1, 14 Jun 2021 [MOD_SV]
    - *Functional Package for Transport Layer Security (TLS)*, Version 1.1, 1 March 2019 [PKG_TLS]

Revision History

Version	Date	Description
0.1	3 December 2021	Initial draft
0.2	18 March 2022	Updated for revised ST
1.0	28 July 2022	Final version for check-out

Contents

1 Introduction

This document presents results from performing assurance activities associated with the VMware ESXi™ 7.0 Update 3d product evaluation. This report contains sections documenting the performance of assurance activities associated with each of the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) as specified in the following documents:

- *Protection Profile for Virtualization*, Version 1.1, 14 Jun 2021 [PP_V]
- *Supporting Document Mandatory Technical Document PP-Module for Server Virtualization Systems*, Version 1.1, 14 Jun 2021 [MOD_SV_SD]
- *Functional Package for Transport Layer Security (TLS)*, Version 1.1, 1 March 2019 [PKG_TLS].

Note that, in accordance with NIAP Policy Letter #5, all cryptography in the TOE for which NIST provides validation testing of FIPS-approved and NIST-recommended cryptographic algorithms and their individual components must be NIST validated. The CCTL will verify that the claimed NIST validation complies with the NIAP-approved PP requirements the TOE claims to satisfy. The CCTL verification of the NIST validation will constitute performance of the associated assurance activity. As such, Test activities associated with functional requirements within the scope of Policy Letter #5 are performed by verification of the relevant CAVP certification and not through performance of any testing as specified in the PP or its supporting document.

1.1 Applicable Technical Decisions

As of the date of this AAR, NIAP has not issued any technical decisions related to [MOD_SV].

The NIAP Technical Decision referenced below applies to [PP_V]. Rationale is included for those Technical Decisions that do not apply to this evaluation.

TD0615: Audit generation for hypercalls implemented in HW

This TD is applicable to the TOE.

The NIAP Technical Decisions referenced below apply to [PKG_TLS]. Rationale is included for those Technical Decisions that do not apply to this evaluation.

TD0442: Updated TLS Ciphersuites for TLS Package

This TD is applicable to the TOE.

TD0469: Modification of test activity for FCS_TLSS_EXT.1.1 test 4.1

This TD is applicable to the TOE.

TD0499: Testing with pinned certificates

This TD modifies pre-conditions for test activities for FCS_TLSC_EXT.1.2, but does not affect the TOE as the TOE does not support certificate pinning.

TD0513: CA Certificate loading

This TD is applicable to the TOE.

TD0588: Session Resumption Support in TLS package

This TD is applicable to the TOE.

1.2 Evidence

[ST]	VMware ESXi 7.0 Update 3d Security Target, Version 1.0, 22 July 2022
[BACK]	TSS Annex – Backdoor Commands (Hypercalls), v1.0, 14 December 2021
[VDI]	TSS Annex – Virtual Device Interfaces, v1.0, 11 February 2022
[CCECG]	Guidance Supplement VMware ESXi 7.0 Update 3d, Version 1.0, 25 July 2022.

2 Security Functional Requirement Evaluation Activities

This section describes the evaluation activities associated with the SFRs defined in the ST and the results of those activities as performed by the evaluation team. The evaluation activities are derived from [PP_V], [MOD_SV_SD], and [PKG_TLS], and modified by applicable NIAP Technical Decisions. Evaluation activities for SFRs not claimed by the TOE have been omitted.

2.1 Security Audit (FAU)

2.1.1 Audit Data Generation (FAU_GEN.1)

2.1.1.1 TSS Activities

The evaluator shall check the TSS and ensure that it lists all of the auditable events and provides a format for audit records. Each audit record format type shall be covered, along with a brief description of each field. The evaluator shall check to make sure that every audit event type mandated by the PP-Configuration is described in the TSS.

Section 6.2 of [ST] ("Security Audit") states the TOE generates audit records for all audit events listed in Tables 6 and 7 of [ST]. Table 6 ("Auditable Events") lists the auditable events associated with each mandatory functional requirement specified in [PP_V], while Table 7 ("Additional Auditable Events Based on Selections") lists the auditable events associated with the selection-based requirements included from [PP_V], along with the auditable events associated with requirements drawn from [PKG_TLS]. The evaluator confirmed the lists of auditable events are complete and consistent with requirements in [PP_V], [MOD_SV], and [PKG_TLS].

Section 6.2 of [ST] also states each audit record includes date, time, applicable subject and object identities, the outcome of the auditable event, and any additional information as required by [PP_V], [MOD_SV], and [PKG_TLS]. It also references the supplemental administrative guidance for examples of each audit record.

2.1.1.2 Guidance Activities

The evaluator shall also make a determination of the administrative actions that are relevant in the context of this PP-Configuration. The evaluator shall examine the administrative guide and make a determination of which administrative commands, including subcommands, scripts, and configuration files, are related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the PP and PP-Modules. The evaluator shall document the methodology or approach taken while determining which actions in the administrative guide are security-relevant with respect to this PP-Configuration.

The evaluator examined the supplied guidance documentation, identifying all mechanisms available to the administrator for configuring and managing the capabilities of the TOE. Those mechanisms related to the SFRs specified in the ST were identified and mapped to the applicable SFRs. In addition, the evaluator sought to confirm that all SFRs that would be expected to have a management capability related to them had appropriate management capabilities identified in the guidance documentation.

The relevant administrative actions related to TSF data related to configuration changes comprise:

- Updating the Virtualization System
- Configuring Administrator password policy as defined in FIA_PMG_EXT.1
- Creating, configuring, and deleting VMs

- Setting default initial VM configurations
- Configuring virtual networks including VM
- Configuring and managing the audit system and audit data
- Configuring VM access to physical devices
- Configuring inter-VM data sharing
- Configuring removable media policy
- Configuring the cryptographic functionality
- Changing default authorization factors
- Configuring remote connection inactivity timeout
- Configuring lockout policy for unsuccessful authentication attempts through limiting number of attempts during a time period
- Configuring name/address of audit/logging server to which to send audit/logging records
- Configuring name/address of network time server
- Configuring banner
- Connecting/disconnecting removable devices to/from a VM
- Starting a VM
- Stopping/halting a VM
- Checkpointing a VM
- Suspending a VM
- Resuming a VM.

2.1.1.3 Test Activities

The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records for the events listed and administrative actions. For administrative actions, the evaluator shall test that each action determined by the evaluator above to be security relevant in the context of this PP is auditable. When verifying the test results, the evaluator shall ensure the audit records generated during testing match the format specified in the administrative guide, and that the fields in each audit record have the proper entries. Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly.

The evaluator performed actions, either independently or as part of a test activity, to generate all audit records on the TOE and confirmed that they were generated in the format specified in guidance.

2.1.2 Audit Review (FAU_SAR.1)

2.1.2.1 TSS Activities

None.

2.1.2.2 Guidance Activities

The evaluator shall review the operational guidance for the procedure on how to review the audit records.

Section 4.1.1 of [CCECG] ("Viewing Audit Records") describes the procedure for how to view the audit records. An administrator can view the locally stored audit records via the VIM API. The guidance includes a link to a script provided as sample code and available from developer.vmware.com.

2.1.2.3 Test Activities

The evaluator shall verify that the audit records provide all of the information specified in FAU_GEN.1 and that this information is suitable for human interpretation. The evaluation activity for this requirement is performed in conjunction with the evaluation activity for FAU_GEN.1.

The evaluator reviewed the audit records the TOE generated during testing and verified the audit records provided all the information specified in FAU_GEN.1. The evaluator also verified the audit information is suitable for human interpretation.

2.1.3 Protected Audit Trail Storage (FAU_STG.1)

2.1.3.1 TSS Activities

The evaluator shall ensure that the TSS describes how the audit records are protected from unauthorized modification or deletion. The evaluator shall ensure that the TSS describes the conditions that must be met for authorized deletion of audit records.

Section 6.2 of [ST] ("Security Audit") states audit records are stored on the TOE's file system as flat files. The TOE protects audit records from unauthorized access through file system permissions as well as through logical access controls on the TOE's management interfaces. Audit records can be reviewed using the TOE via the VIM API, but only the Administrator has the ability to do this. There is no interface to modify or manually delete stored audit records.

2.1.3.2 Guidance Activities

None.

2.1.3.3 Test Activities

The evaluator shall perform the following tests:

Test 1: The evaluator shall access the audit trail as an unauthorized Administrator and attempt to modify and delete the audit records. The evaluator shall verify that these attempts fail.

The evaluator attempted to access the audit trail as an unauthorized user. The evaluator determined the TOE does not provide any interface that enables an unprivileged or unauthorized user to access the audit trail or modify or delete audit records.

Test 2: The evaluator shall access the audit trail as an authorized Administrator and attempt to delete the audit records. The evaluator shall verify that these attempts succeed. The evaluator shall verify that only the records authorized for deletion are deleted.

The evaluator attempted to access the audit trail as an Administrator. The evaluator determined the TOE does not provide any interface that enables an Administrator to modify or delete audit records. That is, the TOE does not authorize any modification or deletion of audit records.

2.1.4 Off-Loading of Audit Data (FAU_STG_EXT.1)

2.1.4.1 TSS Activities

FAU_STG_EXT.1.1

The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided.

Section 6.2 of [ST] ("Security Audit") states the TOE has the ability to transmit audit data to a remote syslog server using TLS 1.2. Section 6.3 of [ST] ("Cryptographic Support") describes the TOE's implementation of TLS as both a TLS client and TLS server.

FAU_STG_EXT.1.2

The evaluator shall examine the TSS to ensure it describes what happens when the local audit data store is full.

Section 6.2 of [ST] states the TOE can be configured to specify the maximum size of local audit record storage. Local audit records are stored as flat files that are pre-allocated when the TOE is initially provisioned. When a file has reached its maximum capacity, the log is rolled over to the next file. This repeats in a FIFO order until all files have been filled, at which point the log is rolled back over to the first file, which is subsequently cleared to make room for the new audit data.

2.1.4.2 Guidance Activities

FAU_STG_EXT.1.1

The evaluator shall examine the operational guidance to ensure it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with the audit server.

Section 4.1.3 of [CCECG] ("Configuring Remote Audit Server") describes how to establish the trusted channel to the audit server and how to configure the TOE to communicate with the audit server over the trusted channel. Section 2.5 of [CCECG] ("Configuring the TOE Environmental Components") states the audit server must implement RFC 3164 ("The BSD Syslog Protocol") and RFC 5425 ("TLS Transport Mapping for Syslog"). Section 4.1.3 of [CCECG] also notes the firewall must be configured during system setup to permit outbound syslog access.

FAU_STG_EXT.1.2

The evaluator shall also examine the operational guidance to determine that it describes the relationship between the local audit data and the audit data that are sent to the audit log server.

For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store used as a buffer and "cleared" periodically by sending the data to the audit server.

Section 4.1 of [CCECG] ("Audit Configuration (FAU)") states the TOE supports both local storage of generated audit records and transmission of audit records to an external audit server. Both modes are disabled by default and each mode must be manually enabled. With both modes enabled, the TOE sends generated audit messages simultaneously to the local store and remote audit servers. Section 4.1.3 of [CCECG] additionally states the TOE transmits audit records to a configured syslog server in real-time—

configuration of a remote syslog server for audit logging does not prevent logs from being generated and stored locally.

2.1.4.3 Test Activities

FAU_STG_EXT.1.1

Testing of the trusted channel mechanism is to be performed as specified in the evaluation activities for FTP_ITC_EXT.1.

The evaluator shall perform the following test for this requirement:

Test 1: The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator's choice designed to generate audit data to be transferred to the audit server. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing.

The evaluator established a TLS session between the TOE and the external audit server during the conduct of tests associated with FCS_TLSC_EXT.1. These tests demonstrated the ability of the TOE to establish a secure connection with the external audit server and transfer audit records as encrypted Application Data.

FAU_STG_EXT.1.2

The evaluator shall perform operations that generate audit data and verify that this data is stored locally. The evaluator shall perform operations that generate audit data until the local storage space is exceeded and verifies that the TOE complies with the behavior defined in FAU_STG_EXT.1.2.

The evaluator followed the operational guidance to configure the TOE to store messages locally, and constrained the size of the local audit storage. The evaluator then generated audit data sufficient to exhaust the local storage space available for audit data. The evaluator verified the TOE overwrites the oldest stored audit records with newly generated audit records when the local audit trail exceeds available storage space.

2.2 Cryptographic Support (FCS)

2.2.1 Cryptographic Key Generation (FCS_CKM.1)

2.2.1.1 TSS Activities

The evaluator shall ensure that the TSS identifies the key sizes supported by the TOE. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme.

Section 6.3 of [ST] ("Cryptographic Support") states the TOE generates RSA and ECDSA keys in support of TLS. When ECDHE cipher suites are used, the TOE uses NIST curves P-256, P-384, and P-521 for key establishment. When RSA cipher suites are used, the TOE uses 2048 bit RSA keys.

2.2.1.2 Guidance Activities

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key generation schemes and key sizes for all uses defined in this PP.

Section 4.2.1 of [CCECG] ("Cryptographic Key Generation") identifies and describes the asymmetric keys the TOE uses during normal operation. The TOE generates the TLS key (2048 bit RSA) at first system boot, without administrator involvement.

The TOE generates ephemeral keys for TLS sessions (2048 bit RSA, ECDSA NIST curves P-256, P-384, P-521) using algorithms and key sizes that depend on the negotiated TLS cipher suite—the administrator does not exercise any control over the behavior of this function.

2.2.1.3 Test Activities

Modified in accordance with TD0572.

Key Generation for FIPS PUB 186-4 RSA Schemes

Performed in accordance with NIAP Policy Letter #5.

Key Generation for Elliptic Curve Cryptography (ECC)

Performed in accordance with NIAP Policy Letter #5.

Section 6.3 of [ST] ("Cryptographic Support"), Table 10 ("Cryptographic Functions") identifies the CAVP certifications verifying asymmetric key generation, as follows.

Algorithm	Tested Capabilities	Certificates
RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3	Key Generation Mode: B.3.3 Properties: Modulo: 2048 Primality Tests: C.2 Public Exponent Mode: Fixed Fixed Public Exponent: 010001 Public Key Format: Standard	CAVP #A1292 RSA KeyGen (FIPS186-4)
ECC schemes using "NIST curves" P-256, P-384, P-521, that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4	Curve: P-256, P-384, P-521 Secret Generation Mode: Extra Bits, Testing Candidates	CAVP #A1292 ECDSA KeyGen (FIPS186-4) ECDSA KeyVer (FIPS186-4)

2.2.2 Cryptographic Key Distribution (FCS_CKM.2)

2.2.2.1 TSS Activities

The evaluator shall ensure that the supported key establishment schemes correspond to the key generation schemes identified in FCS_CKM.1.1. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme.

Section 6.3 of [ST] ("Cryptographic Support") identifies the following key establishment schemes supported by the TOE:

- RSA-based key establishment schemes, used with RSA cipher suites for TLS
- Elliptic curve-based key establishment schemes, used with ECDHE cipher suites for TLS.

These key establishment schemes correspond to the key generation schemes specified in FCS_CKM.1.

2.2.2.2 Guidance Activities

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key establishment schemes.

Section 4.2.2 of [CCECG] ("Cryptographic Key Establishment") states the choice of TLS cipher suites governs the configuration of TLS key establishment schemes. Section 4.2.8 of [CCECG] ("TLS Protocol") instructs the administrator how to configure the set of TLS cipher suites the TOE supports.

2.2.2.3 Test Activities

The evaluator shall verify the implementation of the key establishment schemes of the supported by the TOE using the applicable tests below. **Key Establishment Schemes** ***RSAES-PKCS1-v1_5 Key Establishment Schemes*** The evaluator shall verify the correctness of the TSF's implementation of RSAES-PKCS1-v1_5 by using a known good implementation for each protocol selected in FTP_ITC_EXT.1 that uses RSAES-PKCS1-v1_5.

The evaluator verified the correctness of the TOE's implementation of RSAES-PKCS1-v1_5 while testing the TOE's implementation of TLS client and TLS server protocols (refer to test results for FCS_TLSC_EXT.1 and FCS_TLSS_EXT.1 respectively).

SP800-56A ECC Key Establishment Schemes Performed in accordance with NIAP Policy Letter #5.

Section 6.3 of [ST] ("Cryptographic Support"), Table 8 ("Cryptographic Functions") identifies the CAVP certifications verifying SP 800-56A key establishment schemes, as follows.

Algorithm	Tested Capabilities	Certificates
Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	Domain Parameter Generation Methods: P-256, P-384, P-521 Scheme: Ephemeral Unified: KAS Role: Initiator, Responder	CAVP #A1292 KAS-ECC-SSC Sp800-56Ar3

2.2.3 Cryptographic Key Destruction (FCS_CKM_EXT.4)

2.2.3.1 TSS Activities

The evaluator shall check to ensure the TSS lists each type of key and its origin and location in memory or storage. The evaluator shall verify that the TSS describes when each type of key is cleared.

Section 9 of [ST] ("Key Lifecycle"), Table 12 ("Key Usage") lists the keys used by the TOE. The table provides the following information for each listed key: the key type and size; its origin and use within the TOE; its location in memory or storage; and when and how it is cleared.

2.2.3.2 Guidance Activities

None.

2.2.3.3 Test Activities

For each key clearing situation the evaluator shall perform one of the following activities:

- The evaluator shall use appropriate combinations of specialized operational or development environments, development tools (debuggers, emulators, simulators, etc.), or instrumented builds (developmental, debug, or release) to demonstrate that keys are cleared correctly, including all intermediate copies of the key that may have been created internally by the TOE during normal cryptographic processing.
- In cases where testing reveals that third-party software modules or programming language run-time environments do not properly overwrite keys, this fact must be documented. Likewise, it must be documented if there is no practical way to determine whether such modules or environments destroy keys properly.
- In cases where it is impossible or impracticable to perform the above tests, the evaluator shall describe how keys are destroyed in such cases, to include:
  - Which keys are affected
  - The reasons why testing is impossible or impracticable
  - Evidence that keys are destroyed appropriately (e.g., citations to component documentation, component developer/vendor attestation, component vendor test results)
  - Aggravating and mitigating factors that may affect the timeliness or execution of key destruction (e.g., caching, garbage collection, operating system memory management)

Use of debug or instrumented builds of the TOE and TOE components is permitted in order to demonstrate that the TOE takes appropriate action to destroy keys. These builds should be based on the same source code as are release builds (of course, with instrumentation and debug-specific code added).

The TOE destroys keys stored in volatile memory when the process using the key exits. The evaluator caused a key value to be loaded into memory of a debug version of the TOE. The evaluator verified the TOE zeroized the allocated memory that contained the key data immediately after the process using the key terminated.

The TOE stores keys in non-volatile storage in a specific file and destroys the file prior to writing a new state. The evaluator located the key file in non-volatile storage and computed a cryptographic hash of the file contents (since the key data is stored compressed and encrypted in the key file). The evaluator then generated a new set of keys and recomputed the cryptographic hash. The evaluator confirmed the hash was different, confirming the file content had been changed. Additionally, the evaluator attempted to recover the original keys from their previous location in the partition and noted that the sectors where the original keys used to reside have been overwritten with data.

2.2.4 Cryptographic Operation (Hashing) (FCS_COP.1/Hash)

2.2.4.1 TSS Activities

The evaluator shall check that the association of the hash function with other TSF cryptographic functions (for example, the digital signature verification function) is documented in the TSS.

Section 6.3 of [ST] ("Cryptographic Support") states the TOE's hash function is used in support of digital signature and keyed-hash message authentication (HMAC) functions.

2.2.4.2 Guidance Activities

The evaluator checks the AGD documents to determine that any configuration that is required to be done to configure the functionality for the required hash sizes is present.

Section 4.2.4 of [CCECG] ("Cryptographic Operation") states the TOE defaults to using FIPS-validated cryptography for all cryptographic operations. The TOE does not require any additional configuration to ensure it uses only the specified hash sizes.

2.2.4.3 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.3 of [ST] ("Cryptographic Support"), Table 10 ("Cryptographic Functions") identifies the CAVP certifications verifying cryptographic hashing, as follows.

Algorithm	Tested Capabilities	Certificates
SHS as defined in FIPS Pub 180-4	SHA-1 SHA-256 SHA-384 SHA-512	CAVP #A1292 SHA-1, SHA2-256, SHA2-384, SHA2-512

2.2.5 Cryptographic Operation (Keyed Hash Algorithms) (FCS_COP.1/KeyedHash)

2.2.5.1 TSS Activities

The evaluator shall examine the TSS to ensure that it specifies the following values used by the HMAC function: key length, hash function used, block size, and output MAC length used.

Section 6.3 of [ST] ("Cryptographic Support") specifies the key length, hash function, block size, and output MAC length of the TOE's HMAC functions as follows:

HMAC Function	Key Length	Hash Function	Block Size	Output MAC Length
HMAC-SHA-1	160 bits	SHA-1	512 bits	160 bits
HMAC-SHA-256	256 bits	SHA-256	512 bits	256 bits
HMAC-SHA-384	384 bits	SHA-384	1024 bits	384 bits
HMAC-SHA-512	512 bits	SHA-512	1024 bits	512 bits

2.2.5.2 Guidance Activities

None.

2.2.5.3 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.3 of [ST] (“Cryptographic Support”), Table 10 (“Cryptographic Functions”) identifies the CAVP certifications verifying cryptographic keyed hashing, as follows.

Algorithm	Tested Capabilities	Certificates
HMAC that meets : FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code”, and FIPS Pub 180-4, “Secure Hash Standard”	HMAC-SHA1 MAC: 160 Key Length: 32, 56, 192, 256, 512 HMAC-SHA2-256 MAC: 256 Key Length: 32, 56, 192, 256, 512 HMAC-SHA2-384 MAC: 384 Key Length: 32, 56, 192, 256, 512 HMAC-SHA2-512 MAC: 512 Key Length: 32, 56, 192, 256, 512	CAVP #A1292 HMAC-SHA-1 HMAC-SHA2-256 HMAC-SHA2-384 HMAC-SHA2-512

2.2.6 Cryptographic Operation (Signature Algorithms) (FCS_COP.1/Sig)

2.2.6.1 TSS Activities

None.

2.2.6.2 Guidance Activities

None.

2.2.6.3 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.3 of [ST] (“Cryptographic Support”), Table 10 (“Cryptographic Functions”) identifies the CAVP certifications verifying digital signature services, as follows.

Algorithm	Tested Capabilities	Certificates
RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 4	**RSA Signature Generation** Signature Type: PKCS 1.5 Modulo: 2048 Hash Algorithm: SHA2-256 Hash Algorithm: SHA2-384 Hash Algorithm: SHA2-512 Modulo: 3072 Hash Algorithm: SHA2-256 Hash Algorithm: SHA2-384 Hash Algorithm: SHA2-512	CAVP #A1292 RSA SigGen (186-4)
	Signature Type: PKCSPSS Modulo: 2048 Hash: SHA2-256; Salt Length: 0 Hash: SHA2-384; Salt Length: 0 Hash: SHA2-512; Salt Length: 0 Modulo: 3072 Hash: SHA2-256; Salt Length: 0 Hash: SHA2-384; Salt Length: 0 Hash: SHA2-512; Salt Length: 0	
	RSA Signature Verification Signature Type: PKCS 1.5 Modulo: 2048 Hash Algorithm: SHA1 Hash Algorithm: SHA2-256 Hash Algorithm: SHA2-384 Hash Algorithm: SHA2-512 Modulo: 3072 Hash Algorithm: SHA1 Hash Algorithm: SHA2-256 Hash Algorithm: SHA2-384 Hash Algorithm: SHA2-512 Modulo: 4096 Hash Algorithm: SHA1 Hash Algorithm: SHA2-256 Hash Algorithm: SHA2-384 Hash Algorithm: SHA2-512	CAVP #A1292 RSA SigVer (186-4)

Algorithm	Tested Capabilities	Certificates
	Signature Type: PKCPSS Modulo: 2048 Hash: SHA1; Salt Length: 0 Hash: SHA2-256; Salt Length: 0 Hash: SHA2-384; Salt Length: 0 Hash: SHA2-512; Salt Length: 0 Modulo: 3072 Hash: SHA1; Salt Length: 0 Hash: SHA2-256; Salt Length: 0 Hash: SHA2-384; Salt Length: 0 Hash: SHA2-512; Salt Length: 0 Modulo: 4096 Hash: SHA1; Salt Length: 0 Hash: SHA2-256; Salt Length: 0 Hash: SHA2-384; Salt Length: 0 Hash: SHA2-512; Salt Length: 0	
ECDSA schemes using "NIST curves" P-256, P-384 and P-521 that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5	**ECDSA Signature Generation** Curve: P-256, P-384, P-521 Hash Algorithm: SHA2-256, SHA2-384, SHA2-512 **ECDSA Signature Verification** Curve: P-256, P-384, P-521 Hash Algorithm: SHA-1, SHA2-256, SHA2-384, SHA2-512	CAVP #A1292 ECDSA SigGen (186-4) ECDSA SigVer (186-4)

2.2.7 Cryptographic Operation (AES Data Encryption/Decryption) (FCS_COP.1/UDE)

2.2.7.1 TSS Activities

None.

2.2.7.2 Guidance Activities

None.

2.2.7.3 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.3 of [ST] ("Cryptographic Support"), Table 10 ("Cryptographic Functions") identifies the CAVP certifications verifying AES encryption and decryption, as follows.

Algorithm	Tested Capabilities	Certificates
AES as specified in ISO 18033-3, CBC as specified in ISO 10116	Direction: Decrypt, Encrypt Key Length: 128, 256	CAVP #A1292 AES-CBC

Algorithm	Tested Capabilities	Certificates
AES as specified in ISO 18033-3, GCM as specified in ISO 19772	Direction: Decrypt, Encrypt IV Generation: Internal Key Length: 128, 256	CAVP #A1292 AES-GCM
AES as specified in ISO 18033-3, CTR as specified in ISO 10116	Direction: Decrypt, Encrypt Key Length: 128, 256	CAVP #A1292 AES-CTR

2.2.8 Entropy for Virtual Machines (FCS_ENT_EXT.1)

2.2.8.1 TSS Activities

The evaluator shall verify that the TSS describes how the TOE provides entropy to Guest VMs, and how to access the interface to acquire entropy or random numbers. The evaluator shall verify that the TSS describes the mechanisms for ensuring that one VM does not affect the entropy acquired by another.

Section 6.3 of [ST] ("Cryptographic Support") states the TOE provides guest VMs with pass-through access to the TOE's physical entropy source (the Intel RDSEED instruction). The TOE enforces isolation between VMs, so there is no sharing of entropy between VMs that independently access the entropy source.

2.2.8.2 Guidance Activities

None.

2.2.8.3 Test Activities

The evaluator shall perform the following tests:

- **Test 1:** The evaluator shall invoke entropy from each Guest VM. The evaluator shall verify that each VM acquires values from the interface.

The evaluation team performed this test in conjunction with Test 2.

- **Test 2:** The evaluator shall invoke entropy from multiple VMs as nearly simultaneously as practicable. The evaluator shall verify that the entropy used in one VM is not identical to that invoked from the other VMs.

The evaluation team wrote a program to query /dev/random and executed it simultaneously on three guest VMs. The program generated four bytes of random data on each call, for a total of 50 distinct calls per VM, which amounted to 200 bytes of random data per VM. The program wrote the collected random data to a file, one for each VM. The evaluator calculated a cryptographic hash of the contents of each of the files and confirmed the hashes were different, confirming the entropy used in each VM was not identical to the entropy used in the other VMs.

2.2.9 HTTPS Protocol (FCS_HTTPS_EXT.1)

2.2.9.1 TSS Activities

The evaluator shall check the TSS to ensure that it is clear on how HTTPS uses TLS to establish an administrative session, focusing on any client authentication required by the TLS protocol vs. security administrator authentication which may be done at a different level of the processing stack.

Section 6.3 of [ST] ("Cryptographic Support") states the TOE implements TLS as a server for remote administration using HTTPS. The TOE's implementation of HTTPS conforms to RFC 2818. The TOE's TLS server implementation does not require mutual authentication of the TLS client.

2.2.9.2 Guidance Activities

None.

2.2.9.3 Test Activities

Testing for this activity is done as part of the TLS testing; this may result in additional testing if the TLS tests are done at the TLS protocol level.

Refer to the Test Activities for FCS_TLSS_EXT.1.

2.2.10 Cryptographic Operation (Random Bit Generation) (FCS_RBG_EXT.1)

Documentation shall be produced—and the evaluator shall perform the activities—in accordance with Annex D, Entropy Documentation and Assessment.

2.2.10.1 TSS Activities

None.

2.2.10.2 Guidance Activities

None.

2.2.10.3 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.3 of [ST] ("Cryptographic Support"), Table 10 ("Cryptographic Functions") identifies the CAVP certifications verifying deterministic random bit generation, as follows.

Algorithm	Tested Capabilities	Certificates
CTR_DRBG (AES) as specified in NIST SP 800-90A.	Mode: AES-256	CAVP #A1292 Counter DRBG CAVP #C499 Counter DRBG

2.2.11 TLS Protocol (FCS_TLS_EXT.1)

2.2.11.1 TSS Activities

The evaluator shall ensure that the selections indicated in the ST are consistent with selections in the dependent components.

In Section 5.2.2.11 of [ST] ("TLS Protocol (FCS_TLS_EXT.1)"), "TLS as a client" and "TLS as a server" are selected in FCS_TLS_EXT.1.1. The ST includes FCS_TLSC_EXT.1 and FCS_TLSS_EXT.1 from the selection-based requirements in [PKG_TLS_V1.1], consistent with the selections made in FCS_TLS_EXT.1.1.

2.2.11.2 Guidance Activities

None.

2.2.11.3 Test Activities

None.

2.2.12 TLS Client Protocol (FCS_TLSC_EXT.1)

2.2.12.1 TSS Activities

FCS_TLSC_EXT.1.1

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the cipher suites supported are specified. The evaluator shall check the TSS to ensure that the cipher suites specified include those listed for this component.

Section 6.3 of [ST] ("Cryptographic Support") states the TOE's TLS client implementation supports the following TLS cipher suites in the TOE's evaluated configuration:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384.

This list comprises exactly the list of supported cipher suites specified in the SFR.

FCS_TLSC_EXT.1.2

The evaluator shall ensure that the TSS describes the client's method of establishing all reference identifiers from the application-configured reference identifier, including which types of reference identifiers are supported (e.g. Common Name, DNS Name, URI Name, Service Name, or other application-specific Subject Alternative Names) and whether IP addresses and wildcards are supported. The evaluator shall ensure that this description identifies whether and the manner in which certificate pinning is supported or used by the product.

Section 6.3 of [ST] ("Cryptographic Support") states the TOE validates the reference identifier of a presented server certificate as part of certificate validation in the establishment of TLS connectivity. This is done through validation of the Common Name (CN) and Subject Alternative Name (SAN) certificate fields. The TOE expects the SAN field to contain a case-insensitive DNS name or IP address. Wildcards are supported for SANs that are DNS names. Certificate pinning is not supported.

FCS_TLSC_EXT.1.3

If the selection for authorizing override of invalid certificates is made, then the evaluator shall ensure that the TSS includes a description of how and when user or administrator authorization is obtained. The evaluator shall also ensure that the TSS describes any mechanism for storing such authorizations, such that future presentation of such otherwise-invalid certificates permits establishment of a trusted channel without user or administrator action.

In FCS_TLSC_EXT.1.3, the ST selects "with no exceptions".

2.2.12.2 Guidance Activities

FCS_TLSC_EXT.1.1

The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the product so that TLS conforms to the description in the TSS.

Section 4.2.8 of [CCECG] ("TLS Protocol") contains instructions on configuring the TOE so that TLS conforms to the description in the TSS.

Section 4.2.8.1 of [CCECG] ("TLS Protocol Versions") states the TOE allows only TLS v1.2 by default and directs the administrator not to modify the configuration to enable earlier versions of TLS.

Section 4.2.8.2 of [CCECG] ("TLS Cipher Suites") describes the TOE's default TLS cipher suite settings and explains that the default specification, combined with the fact that ECDSA server certificates cannot be loaded into the TOE, results in the TOE supporting precisely the sets of cipher suites specified for the TOE's TLS client and TLS server implementations described in the TSS.

FCS_TLSC_EXT.1.2

The evaluator shall verify that the AGD guidance includes instructions for setting the reference identifier to be used for the purposes of certificate validation in TLS.

Section 4.4.3 of [CCECG] ("X.509 Certificate Validation and Authentication") states the only secure outbound connection from the TOE is to an external syslog server using TLS (i.e., this is the only connection in the evaluated configuration where the TOE operates as the TLS client.

Section 2.5.2 of [CCECG] ("Audit Configuration") provides instructions for configuring remote auditing, including configuring the reference identifier used for certificate validation in TLS.

FCS_TLSC_EXT.1.3

None.

2.2.12.3 Test Activities

FCS_TLSC_EXT.1.1

The evaluator shall also perform the following tests:

Test 1: The evaluator shall establish a TLS connection using each of the cipher suites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an EAP session. It is sufficient to observe the successful negotiation of a cipher suite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the cipher suite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

The evaluator configured a TLS server to present each of the supported ciphers one at a time. The evaluator captured network traffic and observed that the TOE accepted each of the connections.

Test 2: The goal of the following test is to verify that the TOE accepts only certificates with appropriate values in the extendedKeyUsage extension, and implicitly that the TOE correctly parses the extendedKeyUsage extension as part of X.509v3 server certificate validation.

The evaluator shall attempt to establish the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage extension and verify that a connection is established. The evaluator shall repeat this test using a different, but otherwise valid and trusted, certificate that lacks the Server Authentication purpose in the extendedKeyUsage extension and ensure that a connection is not established. Ideally, the two certificates should be similar in structure, the types of identifiers used, and the chain of trust.

Test 1 demonstrated the TOE accepts a connection when the TLS server presents a valid certificate that contains the Server Authentication purpose in the extendedKeyUsage extension. The evaluator then presented a certificate containing only client authentication in the extendedKeyUsage extension and observed that a connection was not established.

Test 3: The evaluator shall send a server certificate in the TLS connection that does not match the server-selected cipher suite (for example, send a ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA cipher suite or send a RSA certificate while using one of the ECDSA cipher suites.) The evaluator shall verify that the product disconnects after receiving the server's Certificate handshake message.

The evaluator configured the TLS server to present an RSA server certificate with an ECDSA ciphersuite and attempted a connection from the TOE. The evaluator confirmed that the TOE did not accept the connection.

Test 4: The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL cipher suite and verify that the client denies the connection.

The evaluator configured the TLS server to present the TLS_NULL_WITH_NULL_NULL cipher suite in the Server Hello message and attempted a connection from the TOE. The evaluator confirmed that the TOE did not accept the connection.

Test 5: The evaluator shall perform the following modifications to the traffic:

Test 5.1: Change the TLS version selected by the server in the Server Hello to an undefined TLS version (for example 1.5 represented by the two bytes 03 06) and verify that the client rejects the connection.

The evaluator modified the TLS server to present a Server Hello message with a non-supported TLS version (0x0377) and attempted a connection from the TOE. The evaluator confirmed that the TOE did not accept the connection.

Test 5.2: Change the TLS version selected by the server in the Server Hello to the most recent unsupported TLS version (for example 1.1 represented by the two bytes 03 02) and verify that the client rejects the connection.

The evaluator configured a TLS test server to forcibly select TLS 1.1 in its Server Hello message. The evaluator verified that TLS 1.1 was the selected version and that the TOE terminated the connection with a Protocol Version error after receiving the Server Hello message.

Test 5.3: [conditional] If DHE or ECDHE cipher suites are supported, modify at least one byte in the server's nonce in the Server Hello handshake message, and verify that the client does not complete the handshake and no application data flows.

The evaluator configured the TLS test server to modify the last two hexadecimal values of the server's nonce and observed the TOE rejected the connection and verified no application data flowed.

Test 5.4: Modify the server's selected cipher suite in the Server Hello handshake message to be a cipher suite not presented in the Client Hello handshake message. The evaluator shall verify that the client does not complete the handshake and no application data flows.

The evaluator configured the TLS test server to select the TLS_CHACHA20_POLY1305_SHA256 ciphersuite, which the TOE does not support. The evaluator observed the TOE terminated the connection with an Illegal Parameter error after receiving the Server Hello.

Test 5.5: [conditional] If DHE or ECDHE cipher suites are supported, modify the signature block in the server's Key Exchange handshake message, and verify that the client does not complete the handshake and no application data flows. This test does not apply to cipher suites using RSA key exchange. If a TOE only supports RSA key exchange in conjunction with TLS, then this test shall be omitted.

The evaluator configured the TLS test server to modify the signature block on the Server Key Exchange record. The evaluator observed the TOE terminated the connection with a Decrypt error after receiving the Server Key Exchange.

Test 5.6: Modify a byte in the Server Finished handshake message, and verify that the client does not complete the handshake and no application data flows.

The evaluator configured the TLS test server to modify a byte in the Server Finished handshake message. The evaluator observed the TOE did not complete the handshake and that no application data flowed.

Test 5.7: Send a message consisting of random bytes from the server after the server has issued the Change Cipher Spec message and verify that the client does not complete the handshake and no application data flows. The message must still have a valid 5-byte record header in order to ensure the message will be parsed as TLS.

The evaluator configured the TLS test server to send a message with garbled data after sending the Change Cipher Spec message. The evaluator observed the TOE did not complete the handshake and that no application data flowed.

Modified in accordance with TD0499.

FCS_TLSC_EXT.1.2

The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7.

Test 1: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection fails.

Note that some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass Test 1.

The evaluator attempted to connect to a TLS server, which identified itself with a certificate that had an incorrect CN and did not have the SAN extension. After receiving the certificate, the TOE terminated the connection with a Bad Certificate alert.

Test 2: The evaluator shall present a server certificate that contains a CN that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported SAN type.

The evaluator attempted to connect to a TLS server, which identified itself with a certificate that had a valid CN but an empty SAN extension. After receiving the certificate, the TOE terminated the connection with a Bad Certificate alert.

Test 3: [conditional] If the TOE does not mandate the presence of the SAN extension, the evaluator shall present a server certificate that contains a CN that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds. If the TOE does mandate the presence of the SAN extension, this Test shall be omitted.

The evaluator attempted to connect to a TLS server, which identified itself with a certificate that had a valid CN and no SAN extension. The evaluator observed the connection attempt succeeded.

Test 4: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches. The evaluator shall verify that the connection succeeds.

The evaluator attempted to connect to a TLS server, which identified itself with a certificate that had an invalid CN but a valid identifier in the SAN extension. The evaluator observed the connection attempt succeeded.

Test 5: The evaluator shall perform the following wildcard tests with each supported type of reference identifier. The support for wildcards is intended to be optional. If wildcards are supported, the first, second, and third tests below shall be executed. If wildcards are not supported, then the fourth test below shall be executed.

Test 5.1: [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard that is not in the left-most label of the presented identifier (e.g. foo.*.example.com) and verify that the connection fails.

The evaluator attempted to connect to a TLS server, which identified itself with a certificate with a CN of ‘tlss.*.ate’. After receiving the certificate, the TOE terminated the connection with a Bad Certificate alert.

Test 5.2: [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard in the left-most label but not preceding the public suffix (e.g. *.example.com). The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.example.com) and verify that the connection succeeds. The evaluator shall configure the reference identifier without a left-most label as in the certificate (e.g. example.com) and verify that the connection fails. The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.example.come) and verify that the connection fails.

The evaluator configured the reference identifier on the TOE with a single leftmost label in the reference identifier ('tlss.leidos.ate'). The evaluator attempted to connect to the TLS server, which identified itself with a certificate with a CN of '*.leidos.ate'. The evaluator observed this connection attempt succeeded.

Next, the evaluator configured the reference identifier on the TOE without a leftmost label in the reference identifier ('tlss.ate'). The evaluator attempted to connect to the TLS server, which identified itself with a certificate with a CN of '*.leidos.ate'. The evaluator observed this connection attempt failed.

Next, the evaluator configured the reference identifier on the TOE with two leftmost labels in the reference identifier ('test.tlss.leidos.ate'). The evaluator attempted to connect to the TLS server, which identified itself with a certificate with a CN of '*.leidos.ate'. The evaluator observed this connection attempt failed.

Test 5.3: [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard in the left-most label immediately preceding the public suffix (e.g. *.com). The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.com) and verify that the connection fails. The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.com) and verify that the connection fails.

The evaluator configured the reference identifier on the TOE with a single leftmost label in the reference identifier ('tlss.ate'). The evaluator attempted to connect to the TLS server, which identified itself with a certificate with a CN of '*.ate'. The evaluator observed this connection attempt failed.

Next, the evaluator configured the reference identifier on the TOE with two leftmost labels in the reference identifier ('tlss.leidos.ate'). The evaluator attempted to connect to the TLS server, which identified itself with a certificate with a CN of '*.ate'. The evaluator observed this connection attempt failed.

Test 5.4: conditional]: If wildcards are not supported, the evaluator shall present a server certificate containing a wildcard in the left-most label (e.g.*.example.com). The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.example.com) and verify that the connection fails.

The TOE supports wildcards, so this test is not applicable.

Test 6: conditional] If URI or Service name reference identifiers are supported, the evaluator shall configure the DNS name and the service identifier. The evaluator shall present a server certificate containing the correct DNS name and service identifier in the URIName or SRVName fields of the SAN and verify that the connection succeeds. The evaluator shall repeat this test with the wrong service identifier (but correct DNS name) and verify that the connection fails.

The TOE does not support URI or Service name reference identifiers, so this test is not applicable.

Test 7: [conditional] If pinned certificates are supported the evaluator shall present a certificate that does not match the pinned certificate and verify that the connection fails.

The TOE does not support pinned certificates, so this test is not applicable.

FCS_TLSC_EXT.1.3

Modified in accordance with TD0513.

The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted:

Test 1a: The evaluator shall demonstrate that a server using a certificate with a valid certification path successfully connects.

The evaluator opened a TLS connection from the TOE to a TLS test server. The root CA that issued the server's leaf certificate was in the TOE's trust store. The evaluator confirmed the connection attempt succeeded.

Note, the tests performed in conjunction with FIA_X509_EXT.1 Test 1 also demonstrate the TOE's ability to connect to a TLS server that presents a certificate with a valid certification path.

Test 1b: The evaluator shall modify the certificate chain used by the server in test 1a to be invalid and demonstrate that a server using a certificate without a valid certification path to a trust store element of the TOE results in an authentication failure.

The tests performed in conjunction with FIA_X509_EXT.1 Test 1 demonstrate the TOE will not connect to a TLS server that presents a certificate without a valid certification path.

Test 1c [conditional]: If the TOE trust store can be managed, the evaluator shall modify the trust store element used in Test 1a to be untrusted and demonstrate that a connection attempt from the same server used in Test 1a results in an authentication failure.

The evaluator removed the root CA's certificate from the TOE's trust store and re-attempted the connection. The evaluator confirmed the connection attempt failed with an 'unknown CA' error.

Test 2: The evaluator shall demonstrate that a server using a certificate which has been revoked results in an authentication failure.

This test was performed in conjunction with FIA_X509_EXT.1.1 Test 3. That test demonstrates that a TLS connection attempt was terminated after the TOE received a revoked certificate from a TLS server.

Test 3: The evaluator shall demonstrate that a server using a certificate which has passed its expiration date results in an authentication failure.

This test was performed in conjunction with FIA_X509_EXT.1.1 Test 2. That test demonstrates the TOE rejecting an expired certificate from a TLS server.

Test 4: The evaluator shall demonstrate that a server using a certificate which does not have a valid identifier results in an authentication failure.

This test was performed in conjunction with FCS_TLSC_EXT.1.2 Test 1. That test demonstrates the TOE rejecting a certificate with an invalid reference identifier from a TLS server.

2.2.13 TLS Client Support for Supported Groups Extension (FCS_TLSC_EXT.5)

2.2.13.1 TSS Activities

The evaluator shall verify that TSS describes the Supported Groups Extension.

Section 6.3 of [ST] ("Cryptographic Support") states the TOE uses secp256r1, secp384r1, and secp521r1 for key establishment when negotiating ECDHE cipher suites.

2.2.13.2 Guidance Activities

None.

2.2.13.3 Test Activities

The evaluator shall also perform the following test:

Test 1: The evaluator shall configure a server to perform key exchange using each of the TOE's supported curves and/or groups. The evaluator shall verify that the TOE successfully connects to the server.

The evaluator configured the TLS test server to perform key exchange using each of the TOE's supported curves in turn. The evaluator verified the TOE successfully connected to the server using each supported curve.

2.2.14 TLS Server Protocol (FCS_TLSS_EXT.1)

2.2.14.1 TSS Activities

FCS_TLSS_EXT.1.1

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the cipher suites supported are specified. The evaluator shall check the TSS to ensure that the cipher suites specified include those listed for this component.

Section 6.3 of [ST] ("Cryptographic Support") states the TOE's TLS server implementation supports the following TLS cipher suites:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384.

This list comprises exactly the list of supported cipher suites specified in the SFR.

FCS_TLSS_EXT.1.2

The evaluator shall verify that the TSS contains a description of the denial of old SSL and TLS versions consistent relative to selections in FCS_TLSS_EXT.1.2.

Section 6.3 of [ST] states the TOE uses TLS 1.2 for server communications and rejects client requests for all other TLS and SSL versions. This is consistent with the selections in FCS_TLSS_EXT.1.2.

FCS_TLSS_EXT.1.3

The evaluator shall verify that the TSS describes the key agreement parameters of the server's Key Exchange message.

Section 6.3 of [ST] states the TOE uses secp256r1, secp384r1, and secp521r1 for key establishment when negotiating ECDHE cipher suites, and uses 2048 bit RSA keys when RSA cipher suites are used.

2.2.14.2 Guidance Activities

FCS_TLSS_EXT.1.1

The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the product so that TLS conforms to the description in the TSS.

Section 4.2.8 of [CCECG] ("TLS Protocol") contains instructions on configuring the TOE so that TLS conforms to the description in the TSS.

Section 4.2.8.1 of [CCECG] ("TLS Protocol Versions") states the TOE allows only TLS v1.2 by default and directs the administrator not to modify the configuration to enable earlier versions of TLS.

Section 4.2.8.2 of [CCECG] ("TLS Cipher Suites") describes the TOE's default TLS cipher suite settings and explains that the default specification, combined with the fact that ECDSA server certificates cannot be loaded into the TOE, results in the TOE supporting precisely the sets of cipher suites specified for the TOE's TLS client and TLS server implementations described in the TSS.

FCS_TLSS_EXT.1.2

The evaluator shall verify that the AGD guidance includes any configuration necessary to meet this requirement.

Section 4.2.8.1 of [CCECG] ("TLS Protocol Versions") states the TOE allows only TLS v1.2 by default and directs the administrator not to modify the configuration to enable earlier versions of TLS.

FCS_TLSS_EXT.1.3

The evaluator shall verify that any configuration guidance necessary to meet the requirement must be contained in the AGD guidance.

The TOE uses the key agreement parameters specified in [ST] without any administrator configuration being needed.

2.2.14.3 Test Activities

FCS_TLSS_EXT.1.1

The evaluator shall also perform the following tests:

Test 1: The evaluator shall establish a TLS connection using each of the cipher suites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an EAP session. It is sufficient to observe the successful negotiation of a cipher suite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the cipher suite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

The evaluator set up a TLS client to connect to the TOE using each claimed cipher suite iteratively. The evaluator observed through packet captures that the TOE was capable of completing the connection with each specific cipher suite.

Test 2: The evaluator shall send a Client Hello to the server with a list of cipher suites that does not contain any of the cipher suites in the server's ST and verify that the server denies the connection. Additionally, the evaluator shall send a Client Hello to the server containing only the TLS_NULL_WITH_NULL_NULL cipher suite and verify that the server denies the connection.

The evaluator set up a TLS client to connect to the TOE using a cipher suite not in the TOE's claimed set of supported cipher suites. The evaluator confirmed the TOE refused the connection. The evaluator also configured the TLS client to send a Client Hello containing only the TLS_NULL_WITH_NULL_NULL cipher suite and confirmed the TOE also refused this connection.

Test 3: If RSA key exchange is used in one of the selected ciphersuites, the evaluator shall use a client to send a properly constructed Key Exchange message with a modified EncryptedPreMasterSecret field during the TLS handshake. The evaluator shall verify that the handshake is not completed successfully and no application data flows.

The evaluator configured a TLS client to send a properly constructed Key Exchange message with a modified EncryptedPreMasterSecret field during the TLS handshake. The evaluator observed the TOE stopped the TLS handshake and terminated the connection after receiving the Client Key Exchange message.

Modified in accordance with TD0469.

Test 4: The evaluator shall perform the following modifications to the traffic:

~~**Test 4.1:** Change the TLS version proposed by the client in the Client Hello to a non-supported TLS version (for example 1.3 represented by the two bytes 03 04) and verify that the server rejects the connection.~~

Test 4.2: Modify a byte in the data of the client's Finished handshake message, and verify that the server rejects the connection and does not send any application data.

The evaluator configured a TLS client to send a Finished handshake message with a modified data byte. The evaluator observed the TOE ended the connection with a TCP FIN packet and did not send any application data.

Modified in accordance with TD0588.

Test 4.3: Demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption): ~~Generate a Fatal Alert by sending a Finished message from the client before the client sends a ChangeCipherSpec message, and then send a Client Hello with the session identifier from the previous incomplete session, and verify that the server does not resume the session~~.

Test 4.3i [conditional]: If the TOE does not support session resumption based on session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2) or session tickets according to RFC5077, the evaluator shall perform the following test:

a) The evaluator shall send a Client Hello with a zero-length session identifier and with a SessionTicket extension containing a zero-length ticket.

The evaluator sent a Client Hello message to the TOE with a zero-length session identifier and a zero-length Session Ticket.

b) The evaluator shall verify the server does not send a NewSessionTicket handshake message (at any point in the handshake).

The evaluator verified the TOE did not send a NewSessionTicket handshake message at any point during the handshake.

c) The evaluator shall verify the Server Hello message contains a zero-length session identifier or passes the following steps:

The evaluator also verified the TOE's Server Hello message contained a zero-length session identifier.

Note: The following steps are only performed if the ServerHello message contains a non-zero length SessionID.

d) The evaluator shall complete the TLS handshake and capture the SessionID from the ServerHello.

e) The evaluator shall send a ClientHello containing the SessionID captured in step d). This can be done by keeping the TLS session in step d) open or start a new TLS session using the SessionID captured in step d).

f) The evaluator shall verify the TOE (1) implicitly rejects the SessionID by sending a ServerHello containing a different SessionID and by performing a full handshake (as shown in Figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data.

Since the TOE's Server Hello message contained a zero-length session identifier, the evaluator did not need to execute steps d), e) and f) of the test.

Test 4.3ii [conditional]: If the TOE supports session resumption using session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2), the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS):

a) The evaluator shall conduct a successful handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then initiate a new TLS connection and send the previously captured session ID to show that the TOE resumed the previous session by responding with ServerHello containing the same SessionID immediately followed by ChangeCipherSpec and Finished messages (as shown in Figure 2 of RFC 4346 or RFC 5246).

b) The evaluator shall initiate a handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then, within the same handshake, generate or force an unencrypted fatal Alert message immediately before the client would otherwise send its ChangeCipherSpec message thereby disrupting the handshake. The evaluator shall then initiate a new Client Hello using the previously captured session ID, and verify that the server (1) implicitly rejects the session ID by sending a ServerHello containing a different SessionID and performing a full handshake (as shown in figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data.

The TOE does not support session resumption based on session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2), so this test is not applicable.

Test 4.3iii [conditional]: f the TOE supports session tickets according to RFC5077, the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS):

a) The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then attempt to correctly reuse the previous session by sending the session ticket in the ClientHello. The evaluator shall confirm that the TOE responds with a ServerHello with an empty SessionTicket extension, NewSessionTicket, ChangeCipherSpec and Finished messages (as seen in figure 2 of RFC 5077).

b) The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator will then modify the session ticket and send it as part of a new Client Hello message. The evaluator shall confirm that the TOE either (1) implicitly rejects the session ticket by performing a full handshake (as shown in figure 3 or 4 of RFC 5077), or (2) terminates the connection in some way that prevents the flow of application data.

The TOE does not support session resumption based on session tickets according to RFC5077, so this test is not applicable.

Test 4.4: Send a message consisting of random bytes from the client after the client has issued the ChangeCipherSpec message and verify that the server denies the connection.

The evaluator configured a TLS client to send a message other than the TLS Finished after the ChangeCipherSpec message. The evaluator verified the TOE denied the connection.

FCS_TLSS_EXT.1.2

Test 1: The evaluator shall send a Client Hello requesting a connection with version SSL 2.0 and verify that the server denies the connection. The evaluator shall repeat this test with SSL 3.0 and TLS 1.0, and TLS 1.1 if it is selected.

The evaluator attempted to connect to the TOE while requesting each of the unsupported SSL/TLS versions and verified the TOE refused each connection attempt.

FCS_TLSS_EXT.1.3

The evaluator shall conduct the following tests. The testing can be carried out manually with a packet analyzer or with an automated framework that similarly captures such empirical evidence. Note that this testing can be accomplished in conjunction with other testing activities. For each of the following tests, determining that the size matches the expected size is sufficient:

Test 1: [conditional] If RSA-based key establishment is selected, the evaluator shall attempt a connection using RSA-based key establishment with a supported size. The evaluator shall verify that the size used matches that which is configured. The evaluator shall repeat this test for each supported size of RSA-based key establishment.

The TOE supports RSA-based key establishment with 2048 bit keys. The evaluator confirmed the TOE was able to establish a connection with a TLS client using RSA key exchange and a 2048 bit key.

Test 2: [conditional] If finite-field (i.e. non-EC) Diffie-Hellman ciphers are selected, the evaluator shall attempt a connection using a Diffie-Hellman key exchange with a supported parameter size or supported group. The evaluator shall verify that the key agreement parameters in the Key Exchange message are the ones configured. The evaluator shall repeat this test for each supported parameter size or group.

The TOE does not support finite-field Diffie-Hellman key exchange, so this test is not applicable.

Test 3: conditional] If ECDHE ciphers are selected, the evaluator shall attempt a connection using an ECDHE ciphersuite with a supported curve. The evaluator shall verify that the key agreement parameters in the Key Exchange message are the ones configured. The evaluator shall repeat this test for each supported elliptic curve.

The TOE supports ECDHE ciphers with secp256r1, secp384r1, and secp521r1 NIST curves. The evaluator confirmed the TOE was able to establish a connection with a TLS client using each of the supported elliptic curves.

2.3 User Data Protection (FDP)

2.3.1 Hardware-Based Isolation Mechanisms (FDP_HBI_EXT.1)

2.3.1.1 TSS Activities

The evaluator shall ensure that the TSS provides evidence that hardware-based isolation mechanisms are used to constrain VMs when VMs have direct access to physical devices, including an explanation of the conditions under which the TSF invokes these protections.

Section 6.4 of [ST] ("User Data Protection") states the TOE uses Intel VT-X with Extended Page Tables (EPT) and VT-d to intercept access to all physical hardware resources and emulate those attempts in terms of virtual hardware. This interception is fundamental to virtualization and is not configurable.

2.3.1.2 Guidance Activities

The evaluator shall verify that the operational guidance contains instructions on how to ensure that the platform-provided, hardware-based mechanisms are enabled.

Section 4.3.4 of [CCECG] ("Hardware-based VM Isolation") states the only virtualization mode supported in ESXi 7.0 uses hardware-based virtualization in conjunction with hardware-based two-level page tables (Extended Page Tables). Usage of hardware-based VM isolation mechanisms is always enabled.

2.3.1.3 Test Activities

None.

2.3.2 Physical Platform Resource Controls (FDP_PPR_EXT.1)

2.3.2.1 TSS Activities

The evaluator shall examine the TSS to determine that it describes the mechanism by which the VMM controls a Guest VM's access to physical platform resources. This description shall cover all of the physical platforms allowed in the evaluated configuration by the ST. It should explain how the VMM distinguishes among Guest VMs, and how each physical platform resource that is controllable (that is, listed in the assignment statement in the first element) is identified to an Administrator.

Section 6.4 of [ST] ("User Data Protection") states the TOE uses Intel virtualization (VT-x) with Extended Page Tables (EPT) and Intel VT-d (I/O memory management unit virtualization) to intercept access to all hardware resources and emulate those attempts in terms of virtual hardware. It lists the following physical platform resources as configurable for physical access by a guest VM, noting that access requires both a global (host) configuration and a per-VM configuration that the TOE applies at configuration time. The TOE resolves access to allow or deny:

- USB devices—a guest VM may exchange USB packets with a host-connected USB device. No such devices are initially configured. An administrator uses VendorID and ProductID to identify USB devices and determine to which USB devices a guest VM is granted access.
- Network adapter—a guest VM may exchange Ethernet packets with the physical network when configured with a virtual switch that joins to the physical network. An administrator identifies network adapters via a physical NIC label that is referenced in the virtual switch associated with the adapter. This allows the administrator to determine the network adapters to which a guest VM is granted access.

The evaluator shall ensure that the TSS describes how the Guest VM is associated with each physical resource, and how other Guest VMs cannot access a physical resource without being granted explicit access. For TOEs that implement a robust interface (other than just "allow access" or "deny access"), the evaluator shall ensure that the TSS describes the possible operations or modes of access between a Guest VM's and physical platform resources.

Section 6.4 of [ST] states access of guest VMs to physical resources is applied at configuration time. Access is configured both globally (at the host level) and per guest VM. Access resolves to allow/deny with no finer-grained controls. The default per-VM access to both USB devices and network adapter is “deny”.

If physical resources are listed in the second element, the evaluator shall examine the TSS and operational guidance to determine that there appears to be no way to configure those resources for access by a Guest VM. The evaluator shall document in the evaluation report their analysis of why the controls offered to configure access to physical resources can't be used to specify access to the resources identified in the second element (for example, if the interface offers a drop-down list of resources to assign, and the denied resources are not included on that list, that would be sufficient justification in the evaluation report).

The ST lists the following physical platform resources in FDP_PPR_EXT.1.2: PCI Passthrough devices; and Storage Raw Device Mapping (SCSI passthrough). Section 6.4 of [ST] states in the evaluated configuration, the TOE and its operational environment are configured during initial setup in such a manner that access by Guest VMs to PCI passthrough devices and raw device mappings to logical unit numbers will not be permitted.

Section 2.4.6 of [CCECG] (“Additional Device Configuration”) provides guidance to the administrator on ensuring Guest VMs will be unable to access PCI passthrough devices and raw device mappings to logical unit numbers. For PCI passthrough devices, the administrator ensures all PCI devices are associated only with the host and are unavailable for association with a virtual machine. The guidance describes how the administrator can verify this using the Host Client by editing the settings of a powered off VM, selecting **Add other device**, and noting that the **PCI device** and **Dynamic PCI device** menu entries are not selectable.

For RDM passthrough of storage LUNs, the administrator ensures all local disks are formatted with VMFS volumes and mounted as datastores and that no unmounted storage volumes are available to the host. The guidance describes how the administrator can verify this using the Host Client by editing the settings of a powered off VM, selecting **Add hard disk**, and noting that the **New raw disk** menu entry is not selectable.

2.3.2.2 Guidance Activities

The evaluator shall examine the operational guidance to determine that it describes how an administrator is able to configure access to physical platform resources for Guest VMs for each platform allowed in the evaluated configuration according to the ST. The evaluator shall also determine that the operational guidance identifies those resources listed in the second and third elements of the component and notes that access to these resources is explicitly denied/allowed, respectively.

Section 4.3.2 of [CCECG] ("Physical Platform Resources") states most devices are implemented as virtual devices, including the mouse and keyboard. However, the TOE allows direct access to physical devices in limited scenarios. To remain compliant with [ST], the only physical devices a virtual machine is allowed to access are USB devices and network adapters. Raw disks and other devices (such as PCI passthrough devices, vGPU devices, and SCSI passthrough devices) are not to be used.

Section 4.3.2.1 of [CCECG] ("USB") describes how the administrator is able to configure access to a USB device.

Section 4.3.2.2 of [CCECG] ("Physical Network") states the TOE does not allow a Guest VM to access a physical network directly, except through the administrator configuring a virtual switch with a network interface attached to specific physical network interfaces. More commonly, the administrator can configure a virtual switch to connect to a specific VLAN on the physical network and then connect Guest VMs to that virtual switch. The guidance provides a hyperlink to the "Managing Virtual Switches in the VMware Host Client" web page, which provides guidance to configure virtual machine networking using virtual switches. This web page in turn provides links to web pages providing detailed instructions for adding a virtual switch, removing a virtual switch, and editing virtual switch settings using the Host Client.

2.3.2.3 Test Activities

Using the operational guidance, the evaluator shall perform the following tests for each physical platform identified in the ST:

Test 1: For each physical platform resource identified in the first element, the evaluator shall configure a Guest VM to have access to that resource and show that the Guest VM is able to successfully access that resource.

The evaluator configured a Guest VM to have access to a network adapter and to a USB. The evaluator confirmed the Guest VM could successfully access the network adapter and the USB when configured to have such access.

Test 2: For each physical platform resource identified in the first element, the evaluator shall configure the system such that a Guest VM does not have access to that resource and show that the Guest VM is unable to successfully access that resource.

The evaluator configured a Guest VM not to have access to a network adapter and to a USB. The evaluator confirmed the Guest VM could not access the network adapter or the USB when configured not to have such access.

Test 3: [conditional]: For TOEs that have a robust control interface, the evaluator shall exercise each element of the interface as described in the TSS and the operational guidance to ensure that the behavior described in the operational guidance is exhibited.

The evaluator covered this test while performing Tests 1, 2, 4, and 5 for this SFR.

Test 4: [conditional]: If the TOE explicitly denies access to certain physical resources, the evaluator shall attempt to access each listed (in FDP_PPR_EXT.1.2) physical resource from a Guest VM and observe that access is denied.

The evaluator examined the physical resources available to a Windows VM running on the TOE. There was no access to PCI passthrough devices, raw storage devices or to SCI passthrough through the Windows VM.

Test 5: [conditional]: If the TOE explicitly allows access to certain physical resources, the evaluator shall attempt to access each listed (in FDP_PPR_EXT.1.3) physical resource from a Guest VM and observe that the access is allowed. If the operational guidance specifies that access is allowed simultaneously by more than one Guest VM, the evaluator shall attempt to access each resource listed from more than one Guest VM and show that access is allowed.

The TOE does not explicitly allow access to any physical resources. Therefore, this test is not applicable.

2.3.3 Residual Information in Memory (FDP_RIP_EXT.1)

2.3.3.1 TSS Activities

The evaluator shall ensure that the TSS documents the process used for clearing physical memory prior to allocation to a Guest VM, providing details on when and how this is performed. Additionally, the evaluator shall ensure that the TSS documents the conditions under which physical memory is not cleared prior to allocation to a Guest VM, and describes when and how the memory is cleared.

Section 6.4 of [ST] ("User Data Protection") states the TOE zeroes out memory pages allocated to kernel threads at the time of allocation. The settings on the global `Mem.MemEagerZero` option and the per-VM `sched.mem.eagerZero` option determine how memory pages for guest VMs and user space applications are cleared:

- When `Mem.MemEagerZero` is set to 0 (the default), the TOE zeroes pages when they are allocated to guest VMs and user space applications. While this prevents exposing information from guest VMs to other clients, previous content can stay present in memory for an indeterminate period of time if the memory is not re-used
- When `Mem.MemEagerZero` is set to 1, the TOE zeroes pages when a user space application exits. For guest VMs, such pages are zeroed when the VM powers off, when its pages are migrated, or when virtual machine memory is reclaimed. This behaviour can be configured for guest VMs only by setting the per-VM `sched.mem.eagerZero` option to `TRUE`. Setting `Mem.MemEagerZero` to 1 overrides any per-VM setting of `sched.mem.eagerZero`.

2.3.3.2 Guidance Activities

None.

2.3.3.3 Test Activities

None.

2.3.4 Residual Information on Disk (FDP_RIP_EXT.2)

2.3.4.1 TSS Activities

The evaluator shall ensure that the TSS documents how the TSF ensures that disk storage is zeroed upon allocation to Guest VMs. Also, the TSS must document any conditions under which disk storage is not cleared prior to allocation to a Guest VM. Any file system format and metadata information needed by the evaluator to perform the below test shall be made available to the evaluator, but need not be published in the TSS.

Section 6.4 of [ST] ("User Data Protection") states physical disk storage may or may not be zeroed prior to provisioning to a guest VM in some scenarios for performance reasons. The provisioning policies for VMs include Thick Provision Lazy Zeroed, Thick Provision Eager Zeroed, and Thin Provisioned. For quick provisions, Thin Provision provides the most optimization by creating the disk with just header information. No additional storage blocks are allocated or zeroed out until first accessed for use. However, physical disk storage is cleared prior to access by the host or a guest VM in all scenarios. This is implemented by metadata in the VMFS file system (for Thick Provisioned) or VMDK format (for Thin Provisioned).

2.3.4.2 Guidance Activities

None.

2.3.4.3 Test Activities

The evaluator shall perform the following test:

Test 1: On the host, the evaluator creates a file that is more than half the size of a connected physical storage device (or multiple files whose individual sizes add up to more than half the size of the storage media). This file (or files) shall be filled entirely with a nonzero value. Then, the file (or files) shall be released (freed for use but not cleared). Next, the evaluator (as a VS Administrator) creates a virtual disk at least that large on the same physical storage device and connects it to a powered-off VM. Then, from outside the Guest VM, scan through and check that all the non-metadata (as documented in the TSS) in the file corresponding to that virtual disk is set to zero.

The evaluator performed this test using a 14.2 GB physical disk formatted to VMFS-6. The evaluator created a file with non-zero data that consumed all the free space available on the physical disk (12.8 GB). The evaluator then deleted the large file and confirmed its inode and all allocated blocks were released. The evaluator then used the physical disk to create a thick provisioned VM attached to a 12.8 GB virtual disk. The evaluator verified the virtual disk reused the blocks previously allocated to the large file in the first part of the test. Then, from outside the Guest VM, the evaluator checked the disk contents and confirmed the virtual disk was filled only with zeroes.

2.3.5 VM Separation (FDP_VMS_EXT.1)

2.3.5.1 TSS Activities

The evaluator shall examine the TSS to verify that it documents all inter-VM communications mechanisms (as defined above), and explains how the TSF prevents the transfer of data between VMs outside of the mechanisms listed in FDP_VMS_EXT.1.1.

Section 6.4 of [ST] ("User Data Protection") states the TOE supports communication between VMs through virtual networking, which the guest accesses via a virtual network interface controller (vNIC). A VM has no network connections unless explicitly configured. Administrators configure network connections to connect or disconnect virtual machines or the external network. A Guest VM cannot access the data of another Guest VM, or transfer data to another Guest VM other than through the virtual network mechanism when expressly enabled by an authorized Administrator.

2.3.5.2 Guidance Activities

The evaluator shall examine the operational guidance to ensure that it documents how to configure all inter-VM communications mechanisms, including how they are invoked and how they are disabled.

Section 4.3.1 of [CCECG] ("Virtual Networking") documents how the administrator configures virtual machine networking, which is the only mechanism the TOE implements for inter-VM communications. The guidance states virtual machines can communicate with each other if connected to the same Distributed Virtual Switch, or if connected to Distributed Virtual Switches that are connected to each other. The guidance provides a hyperlink to the "Network Virtual Machine Configuration" web page, which provides guidance to configure virtual machine networking using the Host Client. Section 4.3.1 of [CCECG] also describes how to disable access to a virtual switch.

2.3.5.3 Test Activities

The evaluator shall perform the following tests for each documented inter-VM communications channel:

Test 1:

a. Create two VMs without specifying any communications mechanism or overriding the default configuration.

b. Test that the two VMs cannot communicate through the mechanisms selected in FDP_VMS_EXT.1.1.

c. Create two new VMs, overriding the default configuration to allow communications through a channel selected in FDP_VMS_EXT.1.1.

d. Test that communications can be passed between the VMs through the channel.

e. Create two new VMs, the first with the inter-VM communications channel currently being tested enabled, and the second with the inter-VM communications channel currently being tested disabled.

f. Test that communications cannot be passed between the VMs through the channel.

g. As an Administrator, enable inter-VM communications between the VMs on the second VM.

h. Test that communications can be passed through the inter-VM channel.

i. As an Administrator again, disable inter-VM communications between the two VMs.

j. Test that communications can no longer be passed through the channel.

FDP_VMS_EXT.1.2 is met if communication is unsuccessful in step (b). FDP_VMS_EXT.1.3 is met if communication is successful in step (d) and unsuccessful in step (f).

The evaluator created a pair of Linux virtual machines, one running Ubuntu Linux and another running Kali. The evaluator created the VMs using ESXi's virtual machine creation functionality with a disconnected network adapter.

The evaluator created the VMs with a disconnected network adapter rather than with no network adapter so that they could still be assigned an IP address. The evaluator then attempted to ping each VM from the other and saw that they could not communicate with each other.

The evaluator then re-created the VMs, this time configuring them with a connection to the same virtual test network. Again the VMs were configured with static IPs. Another attempt was made to have them ping each other, which this time was successful.

A third pair of new VMs was created. This time the Ubuntu VM was created without a network adapter connected, while the Kali VM's network adapter was connected. Another attempt was made for the VMs to ping each other. This time the attempt was not successful. The network adapter on the Ubuntu VM was then enabled and the two VMs could successfully ping each other. Both VMs' network adapters were then disconnected and another attempt was made to ping between them. The attempt did not succeed.

2.3.6 Virtual Networking Components (FDP_VNC_EXT.1)

2.3.6.1 TSS Activities

The evaluator shall examine the TSS (or a proprietary annex) to verify that it describes the mechanism by which virtual network traffic is ensured to be visible only to Guest VMs configured to be on that virtual network.

Section 6.4 of [ST] ("User Data Protection") states the TOE supports communication between VMs through virtual networking, which the guest accesses via a virtual network interface controller (vNIC). A VM has no network connections unless explicitly configured. Administrators configure network connections to connect or disconnect virtual machines or the external network. A Guest VM cannot access the data of another Guest VM, or transfer data to another Guest VM other than through the virtual network mechanism when expressly enabled by an authorized Administrator.

2.3.6.2 Guidance Activities

The evaluator must ensure that the Operational Guidance describes how to create virtualized networks and connect VMs to each other and to physical networks.

Section 4.3.1 of [CCECG] ("Virtual Networking") documents how the administrator configures virtual machine networking. The guidance states virtual machines can communicate with each other if connected to the same Distributed Virtual Switch, or if connected to Distributed Virtual Switches that are connected to each other. The guidance provides a hyperlink to the "Network Virtual Machine Configuration" web page, which provides guidance to configure virtual machine networking using the Host Client. Section 4.3.1 of [CCECG] also describes how to disable access to a virtual switch.

2.3.6.3 Test Activities

Test 1: The evaluator shall assume the role of the Administrator and attempt to configure a VM to connect to a network component. The evaluator shall verify that the attempt is successful. The evaluator shall then assume the role of an unprivileged user and attempt the same connection. If the attempt fails, or there is no way for an unprivileged user to configure VM network connections, the requirement is met.

The evaluator assumed the role of an administrator and attempted to configure a VM to have access to the VM network. The evaluator verified the attempt succeeded. The evaluator then assumed the role an unprivileged user. The evaluator confirmed the unprivileged user was unable to access the management interface used to configure VM network connections.

Test 2: The evaluator shall assume the role of the Administrator and attempt to configure a VM to connect to a physical network. The evaluator shall verify that the attempt is successful. The evaluator shall then assume the role of an unprivileged user and make the same attempt. If the attempt fails, or there is no way for an unprivileged user to configure VM network connections, the requirement is met.

The evaluator assumed the role of an administrator and attempted to configure a VM to have access to the physical network. The evaluator verified the attempt succeeded. The evaluator then assumed the role an unprivileged user. The evaluator confirmed the unprivileged user was unable to access the management interface used to configure VM network connections.

2.4 Identification and Authentication (FIA)

2.4.1 Authentication Failure Handling (FIA_AFL_EXT.1)

2.4.1.1 TSS Activities

None.

2.4.1.2 Guidance Activities

None.

2.4.1.3 Test Activities

The evaluator shall perform the following tests for each credential selected in FIA_AFL_EXT.1.1:

The evaluator will set an Administrator-configurable threshold n for failed attempts, or note the ST-specified assignment.

The evaluator configured the TOE to lock a user out after 5 failed login attempts for 90 seconds.

Test 1: The evaluator will attempt to authenticate remotely with the credential n-1 times. The evaluator will then attempt to authenticate using a good credential and verify that authentication is successful.

The evaluator made four attempts to login remotely to an account using an incorrect password. After attempting to login four times with a bad password, the evaluator logged in successfully with a good password.

Test 2: The evaluator will make n attempts to authenticate using a bad credential. The evaluator will then attempt to authenticate using a good credential and verify that the attempt is unsuccessful. Note that the authentication attempts and lockouts must also be logged as specified in FAU_GEN.1.

The evaluator made five attempts to login remotely to an account using an incorrect password. After attempting to login five times with a bad password, the account was locked and an attempt to login with a good password was rejected. The evaluator also verified the TOE logged the authentication attempts and account lock as specified in FAU_GEN.1.

After reaching the limit for unsuccessful authentication attempts the evaluator will proceed as follows:

Test 1: If the Administrator action selection in FIA_AFL_EXT.1.2 is selected, then the evaluator will confirm by testing that following the operational guidance and performing each action specified in the ST to re-enable the remote Administrator's access results in successful access (when using valid credentials for that Administrator).

The ST does not select Administrator action in FIA_AFL_EXT.1.2, so this test is not applicable.

Test 2: If the time period selection in FIA_AFL_EXT.1.2 is selected, the evaluator will wait for just less than the time period configured and show that an authentication attempt using valid credentials does not result in successful access. The evaluator will then wait until just after the time period configured and show that an authentication attempt using valid credentials results in successful access.

The evaluator made five attempts to login to the TOE with an incorrect password, which resulted in the test account being locked. The evaluator then attempted to login to the test account with a good password approximately one minute after the lockout period started. The login attempt was denied.

It should be noted that the lockout timer restarts every time an unsuccessful login attempt is made, even if that attempt is not because of a bad password. This means that when the evaluator logged in at around the one minute mark the timer was reset. After waiting more than 90 seconds since the last login attempt, the evaluator was able to login with a good password.

2.4.2 Multiple Authentication Mechanisms (FIA_UAU.5)

2.4.2.1 TSS Activities

None defined.

2.4.2.2 Guidance Activities

None.

2.4.2.3 Test Activities

If "username and password authentication" is selected, the evaluator will configure the VS with a known username and password and conduct the following tests:

Test 1: The evaluator will attempt to authenticate to the VS using the known username and password. The evaluator will ensure that the authentication attempt is successful.

The evaluator attempted to authenticate to the TOE using the known username and correct password associated with that username, and confirmed the authentication attempt succeeded.

Test 2: The evaluator will attempt to authenticate to the VS using the known username but an incorrect password. The evaluator will ensure that the authentication attempt is unsuccessful.

The evaluator attempted to authenticate to the TOE using the known username but an incorrect password, and confirmed the authentication attempt failed.

If "username and PIN that releases an asymmetric key" is selected, the evaluator will examine the TSS for guidance on supported protected storage and will then configure the TOE or OE to establish a PIN which enables release of the asymmetric key from the protected storage (such as a TPM, a hardware token, or isolated execution environment) with which the VS can interface.

The evaluator will then conduct the following tests:

Test 1: The evaluator will attempt to authenticate to the VS using the known user name and PIN. The evaluator will ensure that the authentication attempt is successful.

Test 2: The evaluator will attempt to authenticate to the VS using the known user name but an incorrect PIN. The evaluator will ensure that the authentication attempt is unsuccessful.

The ST does not select "username and PIN that releases an asymmetric key", so these tests are not applicable.

If "X.509 certificate authentication" is selected, the evaluator will generate an X.509v3 certificate for an Administrator user with the Client Authentication Enhanced Key Usage field set. The evaluator will provision the VS for authentication with the X.509v3 certificate. The evaluator will ensure that the certificates are validated by the VS as per FIA_X509_EXT.1.1 and then conduct the following tests:

Test 1: The evaluator will attempt to authenticate to the VS using the X.509v3 certificate. The evaluator will ensure that the authentication attempt is successful.

Test 2: The evaluator will generate a second certificate identical to the first except for the public key and any values derived from the public key. The evaluator will attempt to authenticate to the VS with this certificate. The evaluator will ensure that the authentication attempt is unsuccessful.

The ST does not select "X.509 certificate authentication", so these tests are not applicable.

If "SSH public-key credential authentication" is selected, the evaluator shall generate a public-private host key pair on the TOE using RSA or ECDSA, and a second public-private key pair on a remote client. The evaluator shall provision the VS with the client public key for authentication over SSH, and conduct the following tests:

Test 1: The evaluator will attempt to authenticate to the VS using a message signed by the client private key that corresponds to provisioned client public key. The evaluator will ensure that the authentication attempt is successful.

Test 2: The evaluator will generate a second client key pair and will attempt to authenticate to the VS with the private key over SSH without first provisioning the VS to support the new key pair. The evaluator will ensure that the authentication attempt is unsuccessful.

The ST does not select "SSH public-key credential authentication", so these tests are not applicable.

2.4.3 Administrator Identification and Authentication (FIA_UIA_EXT.1)

2.4.3.1 TSS Activities

The evaluator shall examine the TSS to determine that it describes the logon process for each logon method (local, remote (HTTPS, SSH, etc.)) supported for the product. This description shall contain information pertaining to the credentials allowed/used, any protocol transactions that take place, and what constitutes a "successful logon."

Section 6.9 of [ST] ("Trusted Path/Channels") states administrators access the TOE using the VIM API, the ESXi Host Client (Web UI), or ESXCLI. Each of these interfaces uses TLS/HTTPS.

Section 6.5 of [ST] ("Identification and Authentication") states the TOE implements username and password-based authentication for all remote interfaces to the TOE. If the credentials match valid entries (and have sufficient permissions), the user is successfully logged in to the TOE and given Administrator privileges.

2.4.3.2 Guidance Activities

The evaluator shall examine the operational guidance to determine that any necessary preparatory steps (e.g., establishing credential material such as pre-shared keys, tunnels, certificates) to logging in are described. For each supported login method, the evaluator shall ensure the operational guidance provides clear instructions for successfully logging on. If configuration is necessary to ensure the services provided before login are limited, the evaluator shall determine that the operational guidance provides sufficient instruction on limiting the allowed services.

Section 4.8.1 of [CCECG] ("Establishing Remote Administrative Sessions") provides clear instructions for successfully logging in via each of the TOE's supported login mechanisms (VIM API, Host Client, ESXCLI), including the necessary preparatory steps prior to logging in.

Through examination of [ST] and [CCECG], the evaluator determined the TOE does not provide any services prior to login.

2.4.3.3 Test Activities

None.

2.4.4 Password Management (FIA_PMG_EXT.1)

2.4.4.1 TSS Activities

None.

2.4.4.2 Guidance Activities

The evaluator shall examine the operational guidance to determine that it provides guidance to security administrators in the composition of strong passwords, and that it provides instructions on setting the minimum password length.

Section 4.4.2 of [CCECG] ("Password Management") provides guidance to security administrators in the composition of strong passwords, including use of the `Security.PasswordQualityControl` setting that specifies minimum password composition requirements, including minimum length. Section 4.4.2 provides a link to the online documentation page "Configure the Passwords and Account Lockout Policy in the VMware Host Client", which provides detailed instructions for setting the minimum password length.

2.4.4.3 Test Activities

The evaluator shall also perform the following test.

Test 1: The evaluator shall compose passwords that either meet the requirements, or fail to meet the requirements, in some way. For each password, the evaluator shall verify that the TOE supports the password. While the evaluator is not required (nor is it feasible) to test all possible combinations of passwords, the evaluator shall ensure that all characters, rule characteristics, and a minimum length listed in the requirement are supported, and justify the subset of those characters chosen for testing.

The evaluator composed one compliant password and five non-compliant passwords to test different use cases against the TOE. The non-compliant passwords covered the following password policy exceptions: too few characters; no uppercase characters; no lowercase characters; no digits; and no special characters.

2.4.5 X.509 Certificate Validation (FIA_X509_EXT.1)

2.4.5.1 TSS Activities

The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place. The evaluator ensures the TSS also provides a description of the certificate path validation algorithm.

Section 6.5 of [ST] ("Identification and Authentication") states the TOE validates X.509 certificates in compliance with RFC 5280 as part of TLS connection establishment and validation of trusted updates.

For TLS connection establishment, the TOE's TLS client implementation validates the certificate presented by the remote syslog server by checking for trusted root certificates in the TOE's configured TLS trust store.

For trusted updates, the TOE's update mechanism validates the certificate immediately prior to applying the update.

The evaluator shall examine the TSS to confirm that it describes the behavior of the TOE when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel. If the requirement that the administrator is able to specify the default action, then the evaluator shall ensure that the operational guidance contains instructions on how this configuration action is performed.

Section 6.5 of [ST] states, for TLS connection establishment, if validation fails, the TLS client will abandon the connection. If a connection cannot be established during the validity check of a certificate used in establishing a trusted channel, the TLS client will terminate the trusted channel. For trusted update, if validation fails, the update is not applied and an error is returned.

2.4.5.2 Guidance Activities

None.

2.4.5.3 Test Activities

The tests described must be performed in conjunction with the other Certificate Services evaluation activities, including the uses listed in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules:

Test 1: The evaluator shall demonstrate that validating a certificate without a valid certification path results in the function failing, for each of the following reasons, in turn:

- by establishing a certificate path in which one of the issuing certificates is not a CA certificate,
- by omitting the basicConstraints field in one of the issuing certificates,
- by setting the basicConstraints field in an issuing certificate to have CA=False,
- by omitting the CA signing bit of the key usage field in an issuing certificate, and
- by setting the path length field of a valid CA field to a value strictly less than the certificate path.

The evaluator shall then establish a valid certificate path consisting of valid CA certificates, and demonstrate that the function succeeds. The evaluator shall then remove trust in one of the CA certificates, and show that the function fails).

The evaluator created an intermediate CA with the basicConstraints extension set to false. (Note that this test also covers the requirement to test for where an issuing certificate is not a CA certificate, since basicConstraints is how this is determined). That intermediate CA was used to issue a leaf certificate for a TLS test server. The evaluator then attempted to establish a connection from the TOE to the TLS test server and confirmed the TOE did not accept the connection.

The evaluator created an intermediate CA whose certificate did not have the basicConstraints extension. The evaluator then attempted to establish a connection from the TOE to the TLS test server and confirmed the TOE did not accept the connection.

The evaluator created a certificate with the CA signing bit of the key usage field omitted. The evaluator then attempted to establish a connection from the TOE to the TLS test server and confirmed the TOE did not accept the connection.

The evaluator created a certificate chain with two intermediate CAs in it. The top level intermediate CA had basicConstraints set to true but had a Path Length constraint value of 0. The evaluator then attempted to establish a connection from the TOE to the TLS test server and confirmed the TOE did not accept the connection.

The evaluator established a valid certificate path consisting of valid CA certificates, and demonstrated that the TOE would accept the connection when the TLS server authenticated using the valid certification path.

The evaluator then removed the root CA certificate and an intermediate CA certificate from the previously successful chain of certificates and observed that the TOE refused to connect to the TLS server.

Test 2: The evaluator shall demonstrate that validating an expired certificate results in the function failing.

The evaluator created an expired certificate signed by a CA and verified the TOE rejected it.

Test 3: The evaluator shall test that the TOE can properly handle revoked certificates – conditional on whether CRL, OCSP, OCSP stapling, or OCSP multi-stapling is selected; if multiple methods are selected, then a test is performed for each method. The evaluator has to only test one up in the trust chain (future revisions may require to ensure the validation is done up the entire chain). The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator shall then attempt the test with a certificate that will be revoked (for each method chosen in the selection) and verify that the validation function fails.

The evaluator attempted to open a TLS connection from the TOE to a TLS server. The TLS server identified itself with a valid leaf certificate whose revocation status was verified by CRL. The validation function succeeded and the TOE accepted the connection. The evaluator then revoked the leaf certificate and repeated the connection attempt. The evaluator confirmed through packet captures that the TOE rejected the connection attempt due to the revoked certificate.

Test 4: If any OCSP option is selected, the evaluator shall present a delegated OCSP certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set and verify that validation of the CRL fails.

The ST does not select any OCSP option-the TOE supports only CRLs for determining certificate revocation status. The evaluator configured a test TLS server to present a certificate that utilizes CRLs for revocation checking and configured the CDP to be lacking the crlSign key usage. The evaluator attempted a connection from the TOE to the test TLS server and confirmed validation of the CRL failed and the connection attempt did not succeed.

Test 5: (Conditional on support for EC certificates as indicated in FCS_COP.1/SIG). The evaluator shall establish a valid, trusted certificate chain consisting of an EC leaf certificate, an EC Intermediate CA certificate not designated as a trust anchor, and an EC certificate designated as a trusted anchor, where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain.

The evaluator configured a test server to present a certificate that chained to an ECDSA CA certificate, where all certificates in the chain specified elliptic curve parameters as a named curve. The evaluator verified that the TOE accepted the connection.

Test 6: (Conditional on support for EC certificates as indicated in FCS_COP.1/SIG). The evaluator shall replace the intermediate certificate in the certificate chain for Test 5 with a modified certificate, where the modified intermediate CA has a public key information field where the EC parameters uses an explicit format version of the Elliptic Curve parameters in the public key information field of the intermediate CA certificate from Test 5, and the modified Intermediate CA certificate is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid

The evaluator changed the certificate from the previous test and modified it so that the intermediate CA utilized explicit curve parameters and verified that the TOE rejected the connection.

2.4.6 X.509 Certificate Authentication (FIA_X509_EXT.2)

2.4.6.1 TSS Activities

The evaluator shall check the TSS to ensure that it describes how the TOE chooses which certificates to use, and any necessary instructions in the administrative guidance for configuring the operating environment so that the TOE can use the certificates.

Section 6.5 of [ST] (“Identification and Authentication”) states the TOE maintains a repository for certificates and selects the appropriate certificate based upon the value in the extendedKeyUsage field. Section 4.4.3 of [CCECG] (“X.509 Certificate Validation and Authentication”) describes the TOE’s capabilities for managing certificate authorities trusted by the host.

The evaluator shall examine the TSS to confirm that it describes the behaviour of the TOE when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel. If the requirement states that the administrator specifies the default action, then the evaluator shall ensure that the operational guidance contains instructions on how this configuration action is performed.

Section 6.5 of [ST] states, for TLS connection establishment, if validation fails, the TLS client will abandon the connection. If a connection cannot be established during the validity check of a certificate used in establishing a trusted channel, the TLS client will terminate the trusted channel.

2.4.6.2 Guidance Activities

None.

2.4.6.3 Test Activities

The evaluator shall perform Test 1 for each function listed in FIA_X509_EXT.2.1 that requires the use of certificates:

Test 1: The evaluator shall demonstrate that using a certificate without a valid certification path results in the function failing. Using the administrative guidance, the evaluator shall then load a certificate or certificates needed to validate the certificate to be used in the function, and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates, and show that the function fails.

Evidence for this can be seen from FIA_X509_EXT.1 Test 1, as this test case is the same as the full chain test in FIA_X509_EXT.1 Test 1.

As described in the user documentation, all trusted CA certificates can be listed through the ESXCLI interface. The evaluator used the ESXCLI to list the trusted CA certificates in the TOE's trust store. This showed that a single trusted root CA was configured for the TOE's test configuration and validation of this CA in all situations was covered by the above FIA_X509 testing.

Test 2: The evaluator shall demonstrate that using a valid certificate requires that certificate validation checking be performed in at least some part by communicating with a non-TOE IT entity. The evaluator shall then manipulate the environment so that the TOE is unable to verify the validity of the certificate, and observe that the action selected in FIA_X509_EXT.2.2 is performed. If the selected action is administrator-configurable, then the evaluator shall follow the guidance documentation to determine that all supported administrator-configurable options behave in their documented manner.

The evaluator configured the operational environment so that the TOE could not connect to the CRL distribution point specified in the certificate. The evaluator confirmed through packet captures the TOE was unable to determine the revocation status of the presented certificate (because it could not connect to the CRL distribution point) and so did not accept the certificate.

2.5 Security Management (FMT)

2.5.1 Separation of Management and Operational Networks (FMT_SMO_EXT.1)

2.5.1.1 TSS Activities

The evaluator shall examine the TSS to verify that it describes how management and operational traffic is separated.

Section 6.6 of [ST] ("Security Management") states the TOE enforces separation of data sharing between management and operational networks. An administrator can configure separate networks for management and operation. The administrator does this either through physical means, where separate NICs are used for each network, or through logical means, where the management and operational networks are on separate VLANs. This ensures that communication on the management network does not occur on the same network as operational traffic.

2.5.1.2 Guidance Activities

The evaluator shall examine the operational guidance to verify that it details how to configure the VS to keep Management and Operational traffic separate.

Section 4.5.2 of [CCECG] ("Isolating VM Networks from the Management Network") details how the administrator configures the TOE to place the management network and guest networking on physically isolated networks.

2.5.1.3 Test Activities

The evaluator shall configure the TOE as documented in the guidance. If separation is logical, then the evaluator shall capture packets on the management network. If plaintext Guest network traffic is detected, the requirement is not met.

If separation uses trusted channels, then the evaluator shall capture packets on the network over which traffic is tunneled. If plaintext Guest network traffic is detected, the requirement is not met.

If data encryption is used, then the evaluator shall capture packets on the network over which the data is sent while a VM or other large data structure is being transmitted. If plaintext VM contents are detected, the requirement is not met.

The evaluator configured a packet capture on the TOE's management network. A VM on a guest network was then used to ping another VM on the same network. The second VM was then used to SCP a file to the first. A connection was also opened to the TOE's management interface. No traffic from the VMs on the guest network was seen on the management interface, all traffic to the TOE's management interface was protected by TLS, and no plaintext HTTP traffic was detected.

The ST does not select "data encryption" as a method for separating management and operational networks.

2.5.2 Management of Security Functions Behavior (FMT_MOF_EXT.1)

2.5.2.1 TSS Activities

The evaluator shall examine the TSS and Operational Guidance to ensure that it describes which security management functions require Administrator privilege and the actions associated with each management function. The evaluator shall verify that for each management function and role specified in the FMT_MOF_EXT.1.1 Server Virtualization Management Functions Table (Table 3), the defined role is able to perform all mandatory functions as well as all optional or selection-based functions claimed in the ST.

Section 6.6 of [ST] ("Security Management"), Table 11 ("TSF Management Functions by Interface"), lists each of the management functions specified in the FMT_MOF_EXT.1.1 Server Virtualization Functions Table, indicating which of the TOE's three management interfaces (VIM API, ESXi Host Client, ESXCLI) provides access to the function. All available functions require Administrator privilege. The evaluator confirmed the Administrator is able to perform each mandatory function as well as each of the optional and selection-based functions claimed in the ST.

2.5.2.2 Guidance Activities

The evaluator shall examine the Operational Guidance to ensure that it describes how the Administrator or User are able to perform each management function that the ST claims the TOE supports.

The evaluator shall verify for each claimed management function that the Operational Guidance is sufficiently detailed to allow the function to be performed.

The ST claims the TOE supports the management functions in the following list. The evaluator examined the guidance documentation and determined it describes how the administrator performs each of the claimed management functions and provides sufficient detail to enable the administrator to perform each

function. The list identifies, for each claimed function, the specific locations in the guidance documentation that describes how the administrator performs the function:

- Ability to update the Virtualization System—Section 4.6.1 of [CCECG] (“Trusted Updates”)
- Ability to configure Administrator password policy as defined in FIA_PMG_EXT.1—Section 4.4.2 of [CCECG] (“Password Management”)
- Ability to create, configure and delete VMs—Section 4.5.3 of [CCECG] (“Management APIs (Consolidated)”) provides a hyperlink to the “Virtual Machine Management with the VMware Host Client” web page, which describes how the administrator performs these functions using the Host Client. This section of [CCECG] also provides hyperlinks to web pages describing how to perform each function using the VIM API.
- Ability to set default initial VM configurations—Section 4.5.3 of [CCECG] states default initial VM configurations are always “empty”, but the administrator can override the “empty” configuration at the time of VM configuration, using either the Host Client or the VIM API. The [CCECG] describes how the administrator performs the override via each of these interfaces.
- Ability to configure virtual networks including VM—Section 4.3.1 of [CCECG] (“Virtual Networking”)
- Ability to configure and manage the audit system and audit data—Section 4.1 of [CCECG] (“Audit Configuration (FAU)”)
- Ability to configure VM access to physical devices—Section 4.3.2 of [CCECG] (“Physical Platform Resources”)
- Ability to configure inter-VM data sharing—covered in Section 4.3.1 of [CCECG] (“Virtual Networking”)
- Ability to configure removable media policy—Section 4.6.2 of [CCECG] (“Removable Devices and Media”)
- Ability to configure the cryptographic functionality—Section 4.2 of [CCECG] (“Cryptographic Configuration (FCS)”)
- Ability to change default authorization factors—Section 4.4 of [CCECG] (“Authentication Configuration (FIA)”)
- Ability to configure remote connection inactivity timeout—Section 4.7.1 of [CCECG] (“Session Timeouts”)
- Ability to configure lockout policy for unsuccessful authentication attempts through limiting number of attempts during a time period—Section 4.4.1 of [CCECG] (“Authentication Failure Handling”)
- Ability to configure name/address of audit/logging server to which to send audit/logging records—Section 4.1.3 of [CCECG] (“Configuring Remote Audit Server”)
- Ability to configure name/address of network time server—Section 4.5.3 of [CCECG] provides a hyperlink to the “Edit the Time Configuration of an ESXi Host in the VMware Host Client” web page, which describes how the administrator configures the host names or IP addresses of the time servers to be used by the TOE.
- Ability to configure banner—Section 4.7.2 of [CCECG] (“Administrative Access Banner”)

- Ability to connect/disconnect removable devices to/from a VM—Sections 4.3.2 and 4.6.2 of [CCECG] ("Physical Platform Resources" and "Removable Devices and Media" respectively).
- Ability to start a VM—Section 4.5.3 of [CCECG] provides a hyperlink to the "Power States of a Virtual Machine in the VMware Host Client" web page, which describes how the administrator starts ("powers on") a VM.
- Ability to stop/halt a VM—Section 4.5.3 of [CCECG] provides a hyperlink to the "Power States of a Virtual Machine in the VMware Host Client" web page, which describes how the administrator stops ("powers off") a VM.
- Ability to checkpoint a VM—Section 4.5.3 of [CCECG] provides a hyperlink to the "Using Snapshots to Manage Virtual Machines" web page, which describes the TOE's capabilities to take snapshots (the TOE's term for a checkpoint). The "Take a Snapshot in the VMware Host Client" web page describes the procedure to take a snapshot of a VM.
- Ability to suspend a VM—Section 4.5.3 of [CCECG] provides a hyperlink to the "Power States of a Virtual Machine in the VMware Host Client" web page, which describes how the administrator suspends a VM.
- Ability to resume a VM—Section 4.5.3 of [CCECG] notes resuming a VM is performed by starting a VM that is in a suspended state.

2.5.2.3 Test Activities

The evaluator shall test each management function for each role listed in the FMT_MOF_EXT.1.1 Server Virtualization Management Functions Table (Table 3) in the ST to demonstrate that the function can be performed by the roles that are authorized to do so and the result of the function is demonstrated. The evaluator shall also verify for each claimed management function that if the TOE claims not to provide a particular role with access to the function, then it is not possible to access the TOE as that role and perform that function.

The evaluator performed each management function as User and Administrator. In particular, the evaluator ensured that the User cannot perform the functions designated as not able to be performed by that role.

2.6 Protection of the TSF (FPT)

2.6.1 Non-Existence of Disconnected Virtual Devices (FPT_DVD_EXT.1)

2.6.1.1 TSS Activities

None.

2.6.1.2 Guidance Activities

None.

2.6.1.3 Test Activities

The evaluator shall connect a device to a VM, then from within the guest scan the VM's devices to ensure that the connected device is present--using a device driver or other available means to scan the VM's I/O ports or PCI Bus interfaces. (The device's interface should be documented in the TSS under FPT_VDP_EXT.1.) The evaluator shall remove the device from the VM and run the scan again. This requirement is met if the device's interfaces are no longer present.

The evaluator connected a device to a Guest VM. The evaluator verified the device was visible to the Guest VM. The evaluator then connected the device to a second VM and confirmed it was no longer visible to the first VM.

2.6.2 Execution Environment Mitigations (FPT_EEM_EXT.1)

2.6.2.1 TSS Activities

The evaluator shall examine the TSS to ensure that it states, for each platform listed in the ST, the execution environment-based vulnerability mitigation mechanisms used by the TOE on that platform. The evaluator shall ensure that the lists correspond to what is specified in FPT_EEM_EXT.1.1.

Section 2.4.1 of [ST] ("Physical Boundary") identifies the TOE is evaluated on a single platform, comprising a Dell PowerEdge R740 server with Intel Xeon Gold 6230R (Cascade Lake) CPUs.

Section 6.7 of [ST] ("Protection of the TSF") states the TOE leverages the capabilities of address-space randomization, memory execution protection, and stack buffer overflow protection to provide execution environment-based vulnerability mitigation mechanisms. This list corresponds to what is specified in FPT_EEM_EXT.1.1.

2.6.2.2 Guidance Activities

None.

2.6.2.3 Test Activities

None.

2.6.3 Hardware Assists (FPT_HAS_EXT.1)

2.6.3.1 TSS Activities

The evaluator shall examine the TSS to ensure that it states, for each platform listed in the ST, the hardware assists and memory-handling extensions used by the TOE on that platform. The evaluator shall ensure that these lists correspond to what is specified in the applicable FPT_HAS_EXT component.

Section 2.4.1 of [ST] ("Physical Boundary") identifies the TOE is evaluated on a single platform, comprising a Dell PowerEdge R740 server with Intel Xeon Gold 6230R (Cascade Lake) CPUs.

Section 6.7 of [ST] ("Protection of the TSF") states the TOE uses Intel VT-x virtualization technology to reduce the use of binary translation and uses Intel Extended Page Tables (EPT) to eliminate the need for shadow page tables. This list corresponds to what is specified in FPT_HAS_EXT.1.

2.6.3.2 Guidance Activities

None.

2.6.3.3 Test Activities

None.

2.6.4 Hypercall Controls (FPT_HCL_EXT.1)

2.6.4.1 TSS Activities

The evaluator shall examine the TSS (or proprietary TSS Annex) to ensure that all hypercall functions are documented at the level necessary for the evaluator to run the below test.

Documentation for each hypercall interface must include: how to invoke the interface, parameters and legal values, and any conditions under which the interface can be invoked (e.g., from guest user mode, guest privileged mode, during guest boot only).

Section 6.7 of [ST] ("Protection of the TSF") states guidance documentation provided by the vendor provides information on the hypercall interfaces, including all applicable functions, parameters, legal values, configuration settings, and how the functions are called.

The vendor has documented all hypercall interfaces in a proprietary TSS annex ([BACK]). The evaluator examined the information documented in [BACK] and confirmed it includes the following information for all identified hypercalls: how to invoke the interface; parameters and legal values; and any constraints on invocation of the interface.

2.6.4.2 Guidance Activities

None.

2.6.4.3 Test Activities

The evaluator shall perform the following test:

For each hypercall interface documented in the TSS or proprietary TSS Annex, the evaluator shall attempt to invoke the function from within the VM using an invalid parameter (if any). If the VMM or VS crashes or generates an exception, or if no error is returned to the guest, then the test fails. If an error is returned to the guest, then the test succeeds.

The evaluator used a tool provided by the vendor to invoke hypercalls on the TOE. The tool was written to supply a string of zeroes as an input argument, which is an invalid value for each of the hypercalls that the TOE supports. During testing it was verified that error values were returned to the guest VM when appropriate, and none of the invalid hypercalls resulted in either the TOE or the guest VM crashing.

2.6.5 Removable Devices and Media (FPT_RDM_EXT.1)

2.6.5.1 TSS Activities

The evaluator shall examine the TSS to ensure it describes the association between the media or devices supported by the TOE and the actions that can occur when switching information domains.

Section 6.7 of [ST] ("Protection of the TSF") states during regular operation of the TOE, an Administrator controls access to removable media, whether physical or virtual, by means of explicit configuration to permit access. Removable physical media applies to USB storage devices. Removable virtual media applies to virtual optical device images (e.g. ISO images). ISO images are presented read-only (no write access is permitted).

2.6.5.2 Guidance Activities

The evaluator shall examine the operational guidance to ensure it documents how an administrator or user configures the behavior of each media or device.

Section 4.3.2.1 of [CCECG] ("USB") documents how an administrator configures the behavior of USB devices.

Section 4.6.2.2 of [CCECG] ("CD-ROM devices") states an administrator can connect CD-ROM images in the form of files in ISO format within a datastore to a virtual machine. It provides a hyperlink to the "Add a CD or DVD Drive to a Virtual Machine in the VMware Host Client" web page, which documents how the administrator uses the ESXi Host Client to connect a CD-ROM image to a virtual machine. The [CCECG] notes the evaluated configuration does not support physical optical drives.

2.6.5.3 Test Activities

The evaluator shall perform the following test for each listed media or device:

Test 1: The evaluator shall configure two VMs that are members of different information domains, with the media or device connected to one of the VMs. The evaluator shall disconnect the media or device from the VM and connect it to the other VM. The evaluator shall verify that the action performed is consistent with the action assigned in the TSS.

The evaluator configured two VMs (VM1, VM2) that were members of different information domains. The evaluator granted VM1 access to a USB device and confirmed VM2 could not access it. The evaluator than removed the USB from VM1 and connected it to VM2 and confirmed VM1 could no longer access it. The actions performed by the evaluator were consistent with the action assigned in the TSS.

2.6.6 Trusted Updates to the Virtualization System (FPT_TUD_EXT.1)

2.6.6.1 TSS Activities

The evaluator shall verify that the TSS describes all TSF software update mechanisms for updating the system software. Updates to the TOE either have a hash associated with them, or are signed by an authorized source. The evaluator shall verify that the description includes either a digital signature or published hash verification of the software before installation and that installation fails if the verification fails. The evaluator shall verify that the TSS describes the method by which the digital signature or published hash is verified to include how the candidate updates are obtained, the processing associated with verifying the update, and the actions that take place for both successful and unsuccessful verification. If digital signatures are used, the evaluator shall also ensure the definition of an authorized source is contained in the TSS.

Section 6.7 of [ST] ("Protection of the TSF") states the TOE provides Administrators the ability to securely update the TOE software via a supported management interface. A digital certificate protects the integrity of TOE updates. Candidate updates are obtained from the vendor's official website, which is the only authorized source of updates. During installation of an update, the TOE automatically performs a digital signature validation check on the presented Code Signing certificate to ensure the update is correct and has not been modified or corrupted. This includes verifying the certificate chain, expiration date, and revocation status. If the digital signature validation check fails, the installation fails and an audit record is generated. Otherwise the installation proceeds, applying the update to the TOE as well as generating an audit record.

If the ST author indicates that a certificate-based mechanism is used for software update digital signature verification, the evaluator shall verify that the TSS contains a description of how the certificates are contained on the device. The evaluator also ensures that the TSS (or administrator guidance) describes how the certificates are installed/updated/selected, if necessary.

Section 6.7 of [ST] states certificates used by the update process are shipped as part of the TOE software. Certificates are updated only as a result of a validated software update.

2.6.6.2 Guidance Activities

None.

2.6.6.3 Test Activities

The evaluator shall perform the following tests:

Test 1: The evaluator performs the version verification activity to determine the current version of the product. The evaluator obtains a legitimate update using procedures described in the operational guidance and verifies that it is successfully installed on the TOE. After the update, the evaluator performs the version verification activity again to verify the version correctly corresponds to that of the update.

The evaluator performed the version verification activity through both the Host Client and the ESXCLI to determine the current version of the product. The evaluator then obtained a legitimate update and verified it installed successfully on the TOE.

Test 2: The evaluator performs the version verification activity to determine the current version of the product. The evaluator obtains or produces illegitimate updates as defined below, and attempts to install them on the TOE. The evaluator verifies that the TOE rejects all of the illegitimate updates. The evaluator performs this test using all of the following forms of illegitimate updates:

1) A modified version (e.g. using a hex editor) of a legitimately signed or hashed update
2) An image that has not been signed/hashed
3) An image signed with an invalid hash or invalid signature (e.g. by using a different key as expected for creating the signature or by manual modification of a legitimate hash/signature)

The evaluator attempted to install an update that had been modified and verified the TOE rejected the update attempt.

The evaluator attempted to install an unsigned update and verified the TOE rejected the update attempt.

The evaluator attempted to install an update with an invalid signature and verified the TOE rejected the update attempt.

2.6.7 Trusted Update Based on Certificates (FPT_TUD_EXT.2)

2.6.7.1 Test Activities

The evaluation activity for this requirement is performed in conjunction with the evaluation activity for FIA_X509_EXT.1 and FIA_X509_EXT.2.

2.6.8 Virtual Device Parameters (FPT_VDP_EXT.1)

2.6.8.1 TSS Activities

The evaluator shall examine the TSS to ensure it lists all virtual devices accessible by the guest OS.

Section 6.7 of [ST] ("Protection of the TSF") lists the following virtual devices as within the scope of evaluation:

- Network controllers:
  - E1000e
  - VMXNET3
- Storage controllers:
  - LSILogic Parallel
  - LSILogic SAS
  - PVSCSI (VMware paravirtual SCSI)
  - AHCI (SATA)
  - NVMe
- USB controllers:
  - UHCI (USB 1.0)
  - EHCI (USB 2.0)
  - XHCI (USB 3.0)
- Traditional PC (Non-PCI) devices:
  - Serial port
  - Parallel port
  - Floppy Disk Controller (FDC).

The TSS, or a separate proprietary document, must also document all virtual device interfaces at the level of I/O ports or PCI Bus interfaces - including port numbers (absolute or relative to a base), port name, address range, and a description of legal input values.

Section 3.1 of [VDI] ("PCI Devices") states all virtual devices, with the exception of the Super I/O chip, are accessed using the PCI Bus interface. The presence of a virtual device's PCI ID in the PCI device list indicates presence of the virtual PCI device. Each virtual PCI device has a per-device configuration space accessed through the PCI Bus interface. Section 3.1 of [VID] includes a table summarizing the memory map layout of the per-device configuration layout, which is common across all PCI devices. The table describes the following for each field defined in the memory map layout:

- Address within the memory map of the field
- Size, in bytes, of the field
- The type of the field, which can be one of the following—Read/Write; Read Only; Write Only; or Undefined
- Description of the purpose or content of the field, including where applicable, invalid values, meanings and allowed values of specific bits in a field, and reserved bits.

Section 3.2 of [VID] ("Traditional PC") states the TOE uses a Super I/O virtual device to implement floppy drive controller, serial ports, and parallel ports. It includes a table that provides addresses to access the various device I/O ports. The table includes the following information:

- I/O port address

- Size of the I/O port field
- Type of the field, which can be one of the following—Read/Write; Read Only; Write Only; or Undefined
- Symbol associated with the I/O port (e.g., COM1, LPT2)
- Description of the purpose or content of the field, including where applicable, invalid values, meanings and allowed values of specific bits in a field, and reserved bits.

The TSS must also describe the expected behavior of the interface when presented with illegal input values. This behavior must be deterministic and indicative of parameter checking by the TSF.

Section 6.7 of [ST] references [VID] for information on virtual device interfaces, including the response to illegal values.

Section 2 of [VID] (“Parameter Validation”) states physical devices in general (and their virtualized counterparts) do not have the concept of illegal or out of range input values, so typically there are no "bad values" that can be passed to the device. The device's writable control registers usually consist of bit fields, where each bit controls a specific aspect of the device. Since bit values are either zero or one, no illegal or out of range values can be specified. Furthermore, even if a device's writable control registers have specified values other than just zero or one, behavior when those specified values are not provided is often undefined by the physical device’s specification sheet. This is because users of the devices are expected to follow the device’s specification. However, there are two characteristics of physical devices that require special consideration when such devices are virtualized:

- Writable control registers may have reserved or undefined bits indicated by the device's specification sheet. These bits are expected never to be modified.
- The device's specification sheet defines specific semantic behavior expected from the user of the device; that is, a specified order of register reads and writes.

Device specifications expect the users of the devices to follow the specification. They are expected not to modify reserved or undefined bits, and are expected to precisely follow the defined semantic behavior. Not following the specification results in undefined behavior. This can include causing the entire system to malfunction. For virtual devices, however, only the VM using the virtual device should be affected, as is the case with the TOE.

A VM executing on the TOE has access only to the virtual devices defined for it. No VM can access another VM's virtual devices. A VM that misuses a virtual device may freeze, panic, or in general malfunction in various ways. However, the TOE itself will not be degraded, and other VMs will not be affected. The TOE may also under rare circumstances cause the VM to crash with a core dump. This is clearly indicated in the VM-specific log (named vmware.log), located in the VM's directory on the system. Such controlled crashes are noted in the log by "panic" messages tagged with VERIFY or NOT_IMPLEMENTED. So although the user of a virtual device may violate the device's specification, doing so will not adversely affect the security or integrity of the TOE. Thus, there are two ways unique to virtualization where a virtualized device based on a physical device may be tested with illegal input values:

- Modifying reserved or undefined bits, since the specifications typically say not to modify them. Usually the defined response is that the modifications are ignored. In the rare case that such changes cause a VM to malfunction, neither the entire system nor other VMs will be adversely affected.

- Not following the device's semantic behavior, as defined by the specification. The results are, by definition, undefined. However, again at most the VM may malfunction, but neither the entire system nor other VMs will be adversely affected.

The evaluator must ensure that there are no obvious or publicly known virtual I/O ports missing from the TSS.

The documentation provided by the vendor covers obvious and publicly known virtual I/O ports.

There is no expectation that evaluators will examine source code to verify the “all” part of the evaluation activity.

2.6.8.2 Guidance Activities

None.

2.6.8.3 Test Activities

For each virtual device interface, the evaluator shall attempt to access the interface using at least one parameter value that is out of range or illegal. The test is passed if the interface behaves in the manner documented in the TSS. Interfaces that do not have input parameters need not be tested. This test can be performed in conjunction with the tests for FPT_DVD_EXT.1.

The evaluator used a customized VM provided by the vendor to access underlying virtual device interfaces. For each PCI type device, the validator attempted to alter the Vendor ID register value, which is supposed to be a read only value. The evaluator verified the attempt failed and the Vendor ID register value was unchanged, for each PCI type device.

2.6.9 VMM Isolation from VMs (FPT_VIV_EXT.1)

2.6.9.1 TSS Activities

The evaluator shall verify that the TSS (or a proprietary annex to the TSS) describes how the TSF ensures that guest software cannot degrade or disrupt the functioning of other VMs, the VMM or the platform. And how the TSF prevents guests from invoking higher-privilege platform code, such as the examples in the note.

Section 6.7 of [ST] (“Protection of the TSF”) states software running in a VM is not able to degrade or disrupt the functioning of other VMs, the VMM, or the platform. There are no design or implementation flaws that bypass or defeat VM isolation. Specifically, the TOE uses Intel’s VT-x and VT-d hardware virtualization support to ensure that VMs are isolated from each other and the TOE, and cannot interfere with a VM’s device access. The TOE does not provide a mechanism or ability for guest software to directly call platform APIs or to directly generate physical System Management Interrupts (SMIs). System Management Mode (SMM) and SMIs are both virtualized, and thus handled by the virtual firmware (in guest) and not by the physical hardware. Virtual SMIs are not correlated with physical SMIs. In the evaluated configuration, no platform firmware, I/O ports, or MMIO registers are directly mapped into the address space or I/O space of the guest VM. Although the TOE provides a host mechanism for updating microcode on the CPU, any attempt by guest software to update the microcode via the virtualized MSR (Model-Specific Register) will be logged and then dropped.

2.6.9.2 Guidance Activities

None.

2.6.9.3 Test Activities

None.

2.7 TOE Access (FTA)

2.7.1 TOE Access Banner (FTA_TAB.1)

2.7.1.1 TSS Activities

None.

2.7.1.2 Guidance Activities

None.

2.7.1.3 Test Activities

The evaluator shall configure the TOE to display the advisory warning message "TEST TEST Warning Message TEST TEST". The evaluator shall then log out and confirm that the advisory message is displayed before login can occur.

There are two banners the TOE can be configured to display, one prior to login and one after. Both of these were configured with the message "Common Criteria Test Message". The evaluator confirmed the message was displayed prior to logging in to the TOE and again after the evaluator had logged in to the TOE.

2.8 Trusted Path/Channel (FTP)

2.8.1 Trusted Channel Communications (FTP_ITC_EXT.1)

2.8.1.1 TSS Activities

The evaluator will review the TSS to determine that it lists all trusted channels the TOE uses for remote communications, including both the external entities and remote users used for the channel as well as the protocol that is used for each.

Section 6.9 of [ST] ("Trusted Path/Channels") identifies the following trusted channels the TOE uses for remote communications:

- Remote administrator access to the TOE via the VIM API, ESXi Host Client, or ESXCLI, using TLS/HTTPS
- Exporting audit records to an external syslog server over TLS.

2.8.1.2 Guidance Activities

None.

2.8.1.3 Test Activities

The evaluator will configure the TOE to communicate with each external IT entity and type of remote user identified in the TSS. The evaluator will monitor network traffic while the VS performs communication with each of these destinations. The evaluator will ensure that for each session a trusted channel was established in conformance with the protocols identified in the selection.

Refer to test results for FTP_TRP.1 and FCS_TLSS_EXT.1, which cover remote administrator access to the TOE, and test results for FAU_STG_EXT.1 and FCS_TLSC_EXT.1, which cover export of audit records to an external syslog server.

2.8.2 Trusted Path (FTP_TRP.1)

2.8.2.1 TSS Activities

The evaluator shall examine the TSS to determine that the methods of remote TOE administration are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST.

Section 6.9 of [ST] ("Trusted Path/Channels") states remote administrators are able to access the TOE via the following interfaces:

- VIM API
- ESXi Host Client
- ESXCLI.

Each of these interfaces uses TLS/HTTPS. This is consistent with the protocols listed as providing protection of communications between the TOE and remote administrators in FTP_ITC_EXT.1. The ST includes requirements for HTTPS (FCS_HTTPS_EXT.1) and TLS (FCS_TLSS_EXT.1).

2.8.2.2 Guidance Activities

The evaluator shall confirm that the operational guidance contains instructions for establishing the remote administrative sessions for each supported method.

Section 4.8.1 of [CCECG] ("Establishing Remote Administrative Sessions") provides instructions for establishing a remote administrative session with the TOE using each of the supported methods (i.e., VIM API, ESXi Host Client, and ESXCLI).

2.8.2.3 Test Activities

The evaluator shall also perform the following tests:

Test 1: The evaluators shall ensure that communications using each specified (in the operational guidance) remote administration method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.

The evaluator followed the operational guidance and established remote administrative connections to the TOE using each of the methods described in the operational guidance (i.e., VIM API, ESXCLI, and ESXi Host Client). The evaluator ensured successful connection to the TOE using each method for remote administration.

Test 2: For each method of remote administration supported, the evaluator shall follow the operational guidance to ensure that there is no available interface that can be used by a remote user to establish remote administrative sessions without invoking the trusted path.

The evaluator followed the operational guidance and confirmed no interface was available for a remote user to establish communications with the TOE without invoking the trusted path.

Test 3: The evaluator shall ensure, for each method of remote administration, the channel data is not sent in plaintext.

The evaluator confirmed the TOE uses TLS to protect remote administrative sessions using each of the methods described in the operational guidance. The evaluator confirmed via packet captures that channel data was not communicated in plain text.

Test 4: The evaluator shall ensure, for each method of remote administration, modification of the channel data is detected by the TOE.

As the channel data uses TLS, evidence for this can be seen in the FCS_TLSS_EXT test cases. In particular, FCS_TLSS_EXT.1.1 Test 4.5 shows that the TOE will terminate a TLS connection when receiving garbled Application Data from a client.

Additional evaluation activities are associated with the specific protocols.

Refer to test results for FCS_TLSS_EXT.1.

2.8.3 User Interface: I/O Focus (FTP_UIF_EXT.1)

2.8.3.1 TSS Activities

The evaluator shall ensure that the TSS lists the supported user input devices.

Section 6.9 of [ST] ("Trusted Path/Channels") indicates the TOE supports keyboard and mouse for user input to the TOE via the ESXi Host Client. The VIM API, being a programmatic interface, does not have user interface capabilities, while the ESXCLI is an administrative interface to the TOE and not to any VM.

2.8.3.2 Guidance Activities

The evaluator shall ensure that the operational guidance specifies how the current input focus is indicated to the user.

Section 4.8.3 of [CCECG] ("User Interface Indicators") states ESXi Host Client is the only interface that supports direct viewing of a VM console. Administrators using the Host Client can open the VM console window as either an embedded window within the Host Client UI, as a distinct browser tab, or as a distinct browser window. Within the Host Client UI, VM consoles are identified by the name of the VM in the window's title bar. The VM with input focus is the top-most window.

2.8.3.3 Test Activities

For each supported input device, the evaluator shall demonstrate that the input from each device listed in the TSS is directed to the VM that is indicated to have the input focus.

The evaluator attempted to access a VM on the TOE using the Host Client UI. When the evaluator clicked on the thumbnail of the VM, the UI opened up a session with the VM and focuses on that window.

2.8.4 User Interface: Identification of VM (FTP_UIF_EXT.2)

2.8.4.1 TSS Activities

The evaluator shall ensure that the TSS describes the mechanism for identifying VMs to the user, how identities are assigned to VMs, and how conflicts are prevented.

Section 6.9 of [ST] ("Trusted Path/Channels") provides the following information regarding the mechanism for identifying VMs to the user, how identities are assigned, and how conflicts are prevented:

- VIM API—when using this interface, VMs are identified by a unique system-assigned identifier (the Managed Object Identifier, or MOID); this short name is better suited to programmatic API access. The VIM API, being a programmatic interface, does not have user interface capabilities
- ESXi Host Client—when using this interface, an administrator may access the virtual machine's console. The virtual machine with input focus is indicated by the topmost window being displayed. Keyboard input is directed to the topmost window being displayed. Mouse input is directed to the topmost window if (and only if) the cursor is presently over the topmost window. The virtual machine is identified by a name supplied by the Administrator when creating the virtual machine and checked for uniqueness; this name is displayed in the title bar for the active virtual machine console window
- ESXCLI—this interface is an administrative interface to the TOE and not to any VM, so the notion of uniquely identifying VMs is not applicable.

2.8.4.2 Guidance Activities

None.

2.8.4.3 Test Activities

The evaluator shall perform the following test: The evaluator shall attempt to create and start at least three Guest VMs on a single display device where the evaluator attempts to assign two of the VMs the same identifier. If the user interface displays different identifiers for each VM, then the requirement is met. Likewise, the requirement is met if the system refuses to create or start a VM when there is already a VM with the same identifier.

The evaluator attempted to import into the TOE a VM with the same ID as an already existing VM. The evaluator verified the TOE refused to create or start a second VM with the same identifier as an already running VM.

3 Security Assurance Requirements

3.1 Security Targeted Evaluation (ASE)

As per ASE activities defined in [CEM] plus the TSS evaluation activities defined for any SFRs claimed by the TOE.

3.2 Development (ADV)

The information about the TOE is contained in the guidance documentation available to the end user as well as the TOE Summary Specification (TSS) portion of the ST. The TOE developer must concur with the description of the product that is contained in the TSS as it relates to the functional requirements. The evaluation activities contained in Section 5.2 [of the PP] should provide the ST authors with sufficient information to determine the appropriate content for the TSS section.

3.2.1 Basic Functional Specification (ADV_FSP.1)

There are no specific evaluation activities associated with these SARs. The functional specification documentation is provided to support the evaluation activities described in Section 5.2 [of the PP], and other activities described for AGD, ATE, and AVA SARs. The requirements on the content of the functional specification information is implicitly assessed by virtue of the other evaluation activities being performed; if the evaluator is unable to perform an activity because there is insufficient interface information, then an adequate functional specification has not been provided.

3.3 Guidance Documents (AGD)

The guidance documents will be provided with the developer's security target. Guidance must include a description of how the authorized user verifies that the Operational Environment can fulfill its role for the security functionality. The documentation should be in an informal style and readable by an authorized user.

Guidance must be provided for every operational environment that the product supports as claimed in the ST. This guidance includes:

- instructions to successfully install the TOE in that environment; and
- instructions to manage the security of the TOE as a product and as a component of the larger operational environment.

Guidance pertaining to particular security functionality is also provided; specific requirements on such guidance are contained in the evaluation activities specified with individual SFRs where applicable.

3.3.1 Operational User Guidance (AGD_OPE.1)

3.3.1.1 AGD_OPE.1 Evaluation Activity

The operational guidance shall contain instructions for configuring the password characteristics, number of allowed authentication attempt failures, the lockout period times for inactivity, and the notice and consent warning that is to be provided when authenticating.

Section 4.5.3 of [CCECG] ("Management APIs (Consolidated)") provides references to descriptions of each of these functions in the guidance documentation. The evaluator examined the references and the referenced guidance documentation and confirmed the documentation contains instructions to perform each of these functions—refer to the analysis in Section 2.5.2.2 above.

The operational guidance shall contain step-by-step instructions suitable for use by an end-user of the VS to configure a new, out-of-the-box system into the configuration evaluated under this Protection Profile.

Section 2.4 of [CCECG] ("Installation of the TOE") provides step-by-step instructions suitable for the administrator (the intended end-user of the VS) to configure a new, out-of-the-box system into the evaluated configuration.

The documentation shall describe the process for verifying updates to the TOE, either by checking the hash or by verifying a digital signature. The evaluator shall verify that this process includes the following steps:

- Instructions for querying the current version of the TOE software.
- For hashes, a description of where the hash for a given update can be obtained. For digital signatures, instructions for obtaining the certificate that will be used by the FCS_COP.1/SIG mechanism to ensure that a signed update has been received from the certificate owner. This may be supplied with the product initially, or may be obtained by some other means.
- Instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory).
- Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes generation of the hash/digital signature.

Section 2.4.3 of [CCECG] ("Verify Software") describes the process for querying the current version of the TOE software.

Section 4.6.1 of [CCECG] ("Trusted Updates") states the TOE uses code signing certificates to validate TOE updates. The initial installation process of the TOE includes automatically loading the root CAs the TOE uses for update verification.

Section 2.4.2 of [CCECG] ("Obtaining Software") provides instructions to obtain TOE updates.

Section 4.6.1 of [CCECG] provides instructions to initiate the update process. The update process includes automatic (i.e., without administrator involvement) of the verification of the digital signature used to protect the update. The administrator discerns whether or not the update process was successful by querying the version of the TOE as described in Section 2.4.3 of [CCECG].

3.3.2 Preparative Procedures (AGD_PRE.1)

3.3.2.1 AGD_PRE.1 Evaluation Activity

As indicated in the introduction above, there are significant expectations with respect to the documentation—especially when configuring the operational environment to support TOE functional requirements. The evaluator shall check to ensure that the guidance provided for the TOE adequately addresses all platforms (that is, combination of hardware and operating system) claimed for the TOE in the ST.

The operational guidance shall contain step-by-step instructions suitable for use by an end-user of the VS to configure a new, out-of-the-box system into the configuration evaluated under this Protection Profile.

The TOE comprises a single version of ESXi software installed on a single platform as specified in [ST]. The guidance documentation in [CCECG] and additional referenced guidance adequately addresses the platform claimed in [ST].

Section 2.4 of [CCECG] ("Installation of the TOE") provides step-by-step instructions suitable for the administrator (the intended end-user of the VS) to configure a new, out-of-the-box system into the evaluated configuration.

3.4 Life-Cycle Support (ALC)

At the assurance level specified for TOEs conformant to this PP, life-cycle support is limited to an examination of the TOE vendor's development and configuration management process in order to provide a baseline level of assurance that the TOE itself is developed in a secure manner and that the developer has a well-defined process in place to deliver updates to mitigate known security flaws. This is a result of the critical role that a developer's practices play in contributing to the overall trustworthiness of a product.

3.4.1 Labelling of the TOE (ALC_CMC.1)

3.4.1.1 ALC_CMC.1 Evaluation Activity

The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST.

Section 1.1 of [ST] ("Security Target, Target of Evaluation, and Common Criteria Identification") identifies the specific version of the TOE that meets the requirements of the ST as VMware ESXi 7.0 Update 3d.

The evaluator shall check the AGD guidance and TOE samples received for testing to ensure that the version number is consistent with that in the ST.

Both the cover page and section 1.1 of [CCECG] ("Purpose") identify the TOE as VMware ESXi 7.0 Update 3d, consistent with the TOE identification in [ST].

The evaluation team checked the TOE sample provided for testing and confirmed the version number was 7.0 Update 3d, consistent with [ST] and [CCECG].

If the vendor maintains a website advertising the TOE, the evaluator shall examine the information on the website to ensure that the information in the ST is sufficient to distinguish the product.

Section 2.4.2 of [CCECG] ("Obtain Software") states the evaluated version of the product is VMware ESXi 7.0 Update 3d and provides instructions to identify and download the evaluated version from the product support website.

3.4.2 TOE CM Coverage (ALC_CMS.1)

3.4.2.1 ALC_CMS.1 Evaluation Activity

The evaluator shall ensure that the developer has identified (in public-facing development guidance for their platform) one or more development environments appropriate for use in developing applications for the developer's platform. For each of these development environments, the developer shall provide information on how to configure the environment to ensure that buffer overflow protection mechanisms in the environment are invoked (e.g., compiler and linker flags). The evaluator shall ensure that this documentation also includes an indication of whether such protections are on by default, or have to be specifically enabled.

This activity is not relevant to the Type 1 hypervisor embodied in the TOE.

The evaluator shall ensure that the TSF is uniquely identified (with respect to other products from the TSF vendor), and that documentation provided by the developer in association with the requirements in the ST is associated with the TSF using this unique identification.

Section 1.1 of [ST] ("Security Target, Target of Evaluation, and Common Criteria Identification") identifies the specific version of the TOE that meets the requirements of the ST as VMware ESXi 7.0 Update 3d. This provides a unique identification that distinguishes the TOE from other versions of VMware ESXi and from other VMware products.

Both the cover page and section 1.1 of [CCECG] ("Purpose") identify the TOE as VMware ESXi 7.0 Update 3d, consistent with the TOE identification in [ST].

3.4.3 Timely Security Updates (ALC_TSU_EXT.1)

This component requires the TOE developer, in conjunction with any other necessary parties, to provide information as to how the VS is updated to address security issues in a timely manner. The documentation describes the process of providing updates to the public from the time a security flaw is reported/discovered, to the time an update is released. This description includes the parties involved (e.g., the developer, hardware vendors) and the steps that are performed (e.g., developer testing), including worst case time periods, before an update is made available to the public.

Section 6.1 of [ST] ("Timely Security Updates") states the vendor releases security updates as software updates (installable ISO images) or patches (installed using command line or ESXi Host Client) distributed from VMware's website. The operational guidance for the TOE describes the patching mechanism. The vendor expects customers to apply updates and patches to address bugs or security issues found after the product releases.

The vendor maintains a Security Response Policy that covers commitments for timely response to reported vulnerabilities, as well as a Lifecycle Policy that defines the length of product support. In the following policy summary, comments in *italics* represent details specific to ESXi

- Supported products will receive security fixes during the "General Support" lifecycle phase. For ESXi this is 5 years beginning from General Availability.
    - Critical severity fixes—work commences immediately to produce a fix or corrective action in the shortest commercially reasonable time.
    - Important severity fixes—fix with the next planned maintenance or update release, or in the form of a patch.
    - Moderate and Low severity fixes—fix with the next planned minor or major release. *For ESXi, minor or major releases occur approximately every 2 years, depending on commercial needs. ESXi 7.0 is a major release.*
- When security issues are privately reported, VMware attempts to provide a fix simultaneous with public notification of the issue.
- When security issues are known publicly, VMware will acknowledge the report and supply any known mitigations, with a further notification when a fix is available.
- An e-mail address (security@vmware.com) and PGP keys (KB 1055) are available for reporting security issues to VMware. VMware's Security Response Center acknowledges security issues immediately upon receipt, with 24/7 monitoring.

3.5 Tests (ATE)

Testing is specified for functional aspects of the system as well as aspects that take advantage of design or implementation weaknesses. The former is done through the ATE_IND family, while the latter is through the AVA_VAN family. At the assurance level specified in this PP, testing is based on advertised functionality and interfaces with dependency on the availability of design information. One of the primary outputs of the evaluation process is the test report as specified in the following requirements.

3.5.1 Independent Testing – Conformance (ATE_IND.1)

Testing is performed to confirm the functionality described in the TSS as well as the administrative (including configuration and operation) documentation provided. The focus of the testing is to confirm that the requirements specified in Section 5.1 [of the PP] are being met, although some additional testing is specified for SARs in Section 5.2 [of the PP]. The evaluation activities identify the additional testing activities associated with these components. The evaluator produces a test report documenting the plan for and results of testing, as well as coverage arguments focused on the platform/TOE combinations that are claiming conformance to this PP.

3.5.1.1 ATE_IND.1 Evaluation Activity

The evaluator shall prepare a test plan and report documenting the testing aspects of the system. While it is not necessary to have one test case per test listed in an evaluation activity, the evaluators must document in the test plan that each applicable testing requirement in the ST is covered.

The Test Plan identifies the platforms to be tested, and for those platforms not included in the test plan but included in the ST, the test plan provides a justification for not testing the platforms. This justification must address the differences between the tested platforms and the untested platforms, and make an argument that the differences do not affect the testing to be performed. It is not sufficient to merely assert that the differences have no affect; rationale must be provided. If all platforms claimed in the ST are tested, then no rationale is necessary.

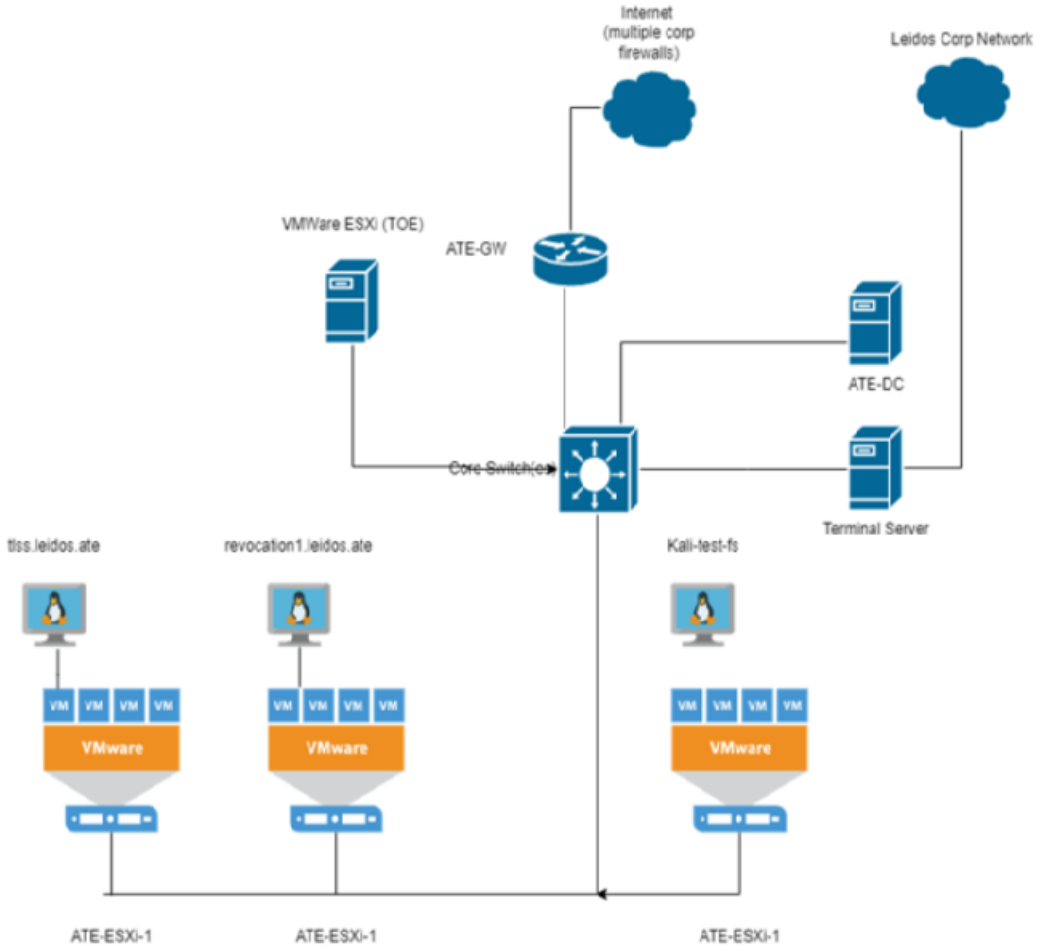
The test plan describes the composition of each platform to be tested, and any setup that is necessary beyond what is contained in the AGD documentation. It should be noted that the evaluators are expected to follow the AGD documentation for installation and setup of each platform either as part of a test or as a standard pre-test condition. This may include special test drivers or tools. For each driver or tool, an argument (not just an assertion) is provided that the driver or tool will not adversely affect the performance of the functionality by the TOE and its platform. This also includes the configuration of cryptographic engines to be used. The cryptographic algorithms implemented by these engines are those specified by this PP and used by the cryptographic protocols being evaluated (IPsec, TLS/HTTPS, SSH).

The test plan identifies high-level test objectives as well as the test procedures to be followed to achieve those objectives. These procedures include expected results. The test report (which could just be an annotated version of the test plan) details the activities that took place when the test procedures were executed, and includes the actual results of the tests. This shall be a cumulative account, so if there was a test run that resulted in a failure; a fix installed; and then a successful re-run of the test, the report would show a “fail” and “pass” result (and the supporting details), and not just the “pass” result.

Testing of the TOE was performed at the Leidos Accredited Testing and Evaluation Lab located in Columbia, Maryland from May 2021 through July 2022.

The evaluation team compiled a detailed test plan and report with a complete set of activities that followed [PP_V], [MOD_SV], and [PKG_TLS]. The evaluation team executed and documented each test case with evidence of passing results and a pass verdict.

The following figure depicts the test environment established for testing the TOE.



The test configuration included the following devices in the operational environment of the TOE:

- TOE platform (physical)—execution environment for the TOE, comprising:
  - Dell PowerEdge R740 server platform with an Intel Xeon 6230R (Cascade Lake) CPU
- ATE-GW (physical)—main router/gateway, running PfSense 2.4.4-RELEASE-p2 operating system
- ATE-DC (physical)—main Domain Controller for test environment and DNS server, running Windows Server 2016 version 1607
- ATE-ESXi-1 (physical)—virtualization server for test tool VMs, running VMware ESXi 6.5.0
- Terminal server (physical)—provides tester access to Test Environment from Corporate Network, running Windows Server 2016 version 1607
- TLSS.leidos.ate (virtual)—hosts TLS test tools, running Ubuntu 18.04.5 and following additional software:
  - Proprietary TLS test tools
  - OpenSSL 1.1.1
  - Wireshark 2.6.10
  - SSLyze 1.4.3
  - VMware backdoorpkg program (internal VMware test tool used for hypercall testing, no version provided)

- Kali-Test-FS (virtual)—Kali VM for testing and hosting ESXCLI, running Kali 2019.3 and following additional software:
  - VMware ESXCLI 7.0.0
  - OpenSSL 1.1.1
  - Wireshark 2.6.10
- Syslog1.leidos.ate (virtual)—hosts external audit server (syslog), running Ubuntu 18.04.2 LTS and following additional software:
  - rsyslogd 8.32.0.

It should be noted that TLSS.leidos.ate is also capable of functioning as an external audit server, using OpenSSL's s_server functionality to receive audit records over TLS. That device was used as an audit server for some test activities.

A test VM provided by VMware was used to test FPT_VDP_EXT.1. This VM was run on the TOE. Due to its extremely basic nature it had no network connectivity and hence no IP address.

3.6 Vulnerability Assessment (AVA)

For the first generation of this Protection Profile, the evaluation lab is expected to survey open sources to learn what vulnerabilities have been discovered in these types of products. In most cases, these vulnerabilities will require sophistication beyond that of a basic attacker. Until penetration tools are created and uniformly distributed to the evaluation labs, evaluators will not be expected to test for these vulnerabilities in the TOE. The labs will be expected to comment on the likelihood of these vulnerabilities given the documentation provided by the vendor. This information will be used in the development of penetration testing tools and for the development of future PPs.

3.6.1 Vulnerability Survey (AVA_VAN.1)

3.6.1.1 AVA_VAN.1 Evaluation Activity

As with ATE_IND the evaluator shall generate a report to document their findings with respect to this requirement. This report could physically be part of the overall test report mentioned in ATE_IND, or a separate document. The evaluator performs a search of public information to determine the vulnerabilities that have been found in virtualization in general, as well as those that pertain to the particular TOE. The evaluator documents the sources consulted and the vulnerabilities found in the report. For each vulnerability found, the evaluator either provides a rationale with respect to its non-applicability or the evaluator formulates a test (using the guidelines provided in ATE_IND) to confirm the vulnerability, if suitable. Suitability is determined by assessing the attack vector needed to take advantage of the vulnerability. For example, if the vulnerability can be detected by pressing a key combination on boot-up, a test would be suitable at the assurance level of this PP. If exploiting the vulnerability requires expert skills and an electron microscope, for instance, then a test would not be suitable and an appropriate justification would be formulated.

The evaluation team performed searches of the National Vulnerability Database (https://nvd.nist.gov/) on 28 July 2022, using the following search terms based on identified components, modules, and features of the TOE, as presented in [ST] and [CCECG]:

- Vmware esxi 7.0.0
- Vmware esxi 7.0
- Esxi 7.0
- vmdk
- Virtual Machine Disk
- Virtual Machine Manager
- vCenter Server
- Dell PowerEdge R740 server
- Intel Xeon Gold 6230.

The evaluation team did not identify any vulnerabilities that pertained to the TOE. The evaluation team determined that no residual vulnerabilities exist that are exploitable by attackers with Basic Attack Potential.