VMware ESXi 7.0 Update 3d

Security Target

Version 1.0

22 July 2022

Prepared for:

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

Prepared by:

Accredited Testing and Evaluation Labs
6841 Benjamin Franklin Drive
Columbia, Maryland 21046

Revision History			
Version	**Date**	**Author**	**Modifications**
0.1	3/11/2021	Leidos	Initial draft/adaptation of previous ST to new ESXi version
0.2	4/30/2021	Leidos	Updates to address vendor feedback
0.3	5/12/2021	Leidos	Updates to address vendor feedback
0.4	5/14/2021	Leidos	Closure of vendor feedback
0.5	7/5/2021	Leidos	Updates to reflect new Virtualization PP materials
0.6	8/4/2021	Leidos	TOE version change
0.7	8/12/2021	Leidos	Updated to address reviewer feedback
0.8	8/25/2021	Leidos	Updated to address reviewer feedback
0.9	3/24/2022	Leidos	Updated to address reviewer feedback
1.0	7/22/2022	Leidos	Updated to final version

Table of Contents

List of Figures and Tables

1 Security Target Introduction

This section identifies the Security Target (ST), Target of Evaluation (TOE), document conventions, and terminology.

The Security Target (ST) includes the following additional sections:

- TOE Description (Section 2)
- Security Problem Definition (Section 3)
- Security Objectives (Section 4)
-

- IT Security Requirements (Section 0)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8)
- Key Lifecycle (Section 0).

1.1 Security Target, Target of Evaluation, and Common Criteria Identification

ST Title: VMware ESXi 7.0 Update 3d Security Target

ST Version: Version 1.0

ST Date: 22 July 2022

Target of Evaluation (TOE) Identification: VMware ESXi 7.0 Update 3d

TOE Developer: VMware, Inc.

Evaluation Sponsor: VMware, Inc.

CC Identification: Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, April 2017

1.2 Conformance Claims

This ST and the TOE it describes are conformant to the following CC specifications:

- *PP-Configuration for Virtualization and Server Virtualization Systems,* Version 1.0, 04 June 2021, which contains the following:
  - *Protection Profile for Virtualization*, Version 1.1, 14 June 2021 (Virtualization PP) with the following Optional, Selection-Based, and Objective requirements:
    - FCS_HTTPS_EXT.1
    - FIA_PMG_EXT.1
    - FIA_X509_EXT.1
    - FIA_X509_EXT.2
    - FPT_TUD_EXT.2
    - FTP_TRP.1
- *PP-Module for Server Virtualization Systems*, Version 1.1, 14 June 2021 (SV Module)
- *Functional Package for Transport Layer Security (TLS),* Version 1.1, 01 March 2019 (TLS Package), with the following Selection-Based and Objective requirements:
  - FCS_TLSC_EXT.1
  - FCS_TLSC_EXT.5
  - FCS_TLSS_EXT.1
- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017.

  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 5, April 2017.
  - Part 3 Extended
- The following NIAP Technical Decisions affect this evaluation:
  - TD0442: Updated TLS Ciphersuites for TLS Package
  - TD0469: Modification of test activity for FCS_TLSS_EXT.1.1 test 4.1
  - TD0513: CA Certificate loading
  - TD0588: Session Resumption Support in TLS package
  - TD0615: Audit generation for hypercalls implemented in HW

  TD0499 does not affect this evaluation because the TOE does not support certificate pinning.

1.3 Conventions

The Common Criteria allows for assignment, selection and iteration operations to be performed on security functional requirements. All these operations are used within this ST. These operations are performed as described in Part 2 of the CC, and selected presentation choices are discussed below to aid the Security Target reader:

- An assignment operation is indicated by [*italicized text within brackets*].
- Selections are denoted by [<u>underlined text within brackets</u>].
- Refinement of security requirements is identified using **bold text**. Any text removed is indicated with a strikethrough (Example: ~~TSF~~). If text is substituted (i.e. some text is removed in favor of some other text), only the addition is shown for readability (e.g., if a refinement changes a table reference from Table 1 to Table 6, it is formatted as "Table **6**"). Trivial grammatical changes to an SFR such as capitalization or pluralization changes are not formatted as refinements.
- Iterations are identified by appending a slash and a descriptive string following the component title; for example, FCS_COP.1/Hash and FCS_COP.1/KeyedHash refer to two iterations of the FCS_COP.1 component, one having to do with hash algorithms and one having to do with keyed hash algorithms.
- Operations completed by the PP author and reproduced exactly from the PP preserve the original brackets to show the completion of the operation but do not format the text. This distinguishes unaltered text from text completed by the ST author.

1.3.1 Terminology

Table 1: Terms and Definitions

Term	Definition
ESXCLI	Remote command line interface to the product.
ESXi	The TOE; VMware's enterprise server virtualization platform.

Host Client	Web client interface to the product.
Syslog Server	Remote entity to which the product sends log messages.
VIB	The file format for product updates.
VIM API	API for the ESXi virtualization infrastructure.
VMkernel	A specialized kernel for arbitrating/scheduling CPU/network/disk fairly and efficiently between virtual machines and user processes.
VT-d	Virtualization Technology for Directed I/O; an Intel processor feature that allows for direct guest VM access to physical PCI devices on a system. It ensures the PCI device can only access the guest VM for which it is configured.
VT-x	Intel virtualization technology; an intel processor feature that allows for hardware to be abstracted to different guest VMs such that multiple guest VMs can use the same hardware resource without awareness of how other guest VMs are using it.

1.3.2 Acronyms

Table 2: Acronyms

Acronym	Definition
AHCI	Advanced Host Controller Interface
API	Application Programming Interface
ATA	Advanced Technology Attachment
CBC	Cipher Block Chaining
CC	Common Criteria
CEM	Common Evaluation Methodology
CSP	Critical Security Parameters
DRBG	Deterministic Random Bit Generator
EHCI	Enhanced Host Controller Interface
EPT	Extended Page Tables
FDC	Floppy Disk Controller
FIPS	Federal Information Processing Standard
IOMMU	Input/Output Memory Management Unit
MMIO	Memory-mapped I/O
MSR	Model-Specific Register
NVM	Non-Volatile Memory
NVMe	NVM Express
OSP	Organizational Security Policy
PC	Personal Computer
PCI	Peripheral Component Interconnect [interface]
PP	Protection Profile

Acronym	Definition
RBG	Random Bit Generator
SAR	Security Assurance Requirement
SAS	Serial Attached SCSI
SATA	Serial ATA
SCSI	Small Computer System Interface
SMI	System Management Interrupt
SMM	System Management Mode
SFR	Security Functional Requirement
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality
UHCI	Universal Host Controller Interface
USB	Universal Serial Bus
VIB	Virtual Infrastructure Bundle
VIM	Virtual Infrastructure Management
VM	Virtual Machine
VMDK	Virtual Machine Disk
VMM	Virtual Machine Manager
XHCI	eXtensible Host Controller Interface

2 TOE Description

2.1 Introduction

VMware ESXi (the TOE) is a virtualization system that is capable of virtualizing server platforms. The TOE conforms to the Base Virtualization PP (also known as the “Virtualization PP”), the Server Virtualization PP-Module (also known as the “SV Module”), and the Functional Package for Transport Layer Security (TLS) (also known as the “TLS Package”). As such, the security-relevant functionality of the product is limited to the claimed requirements in those standards. The security-relevant functionality is described in sections 2.3 and 2.4. The product overview in section 2.2 below is intended to provide the reader with an overall summary of the entire product so that its intended usage is clear. The subset of the product functionality that is within the evaluation scope is subsequently described in the sections that follow it.

2.2 Product Overview

VMware ESXi is a bare-metal or Type 1 hypervisor that is installed onto a computer system with no host platform OS, and serves as a virtual machine manager (VMM) and virtualization system. This allows for the instantiation of multiple virtual machines (VMs) onto a single physical platform. It also implements mechanisms to enforce logical separation of VMs from one another and from the hypervisor so that data transmission between these domains can only occur through authorized interfaces.

2.3 TOE Overview

The TOE is VMware ESXi 7.0 Update 3d, installed on a Dell PowerEdge R740 server platform with Intel Xeon 6230R “Cascade Lake” CPUs. The TOE is designed to act as a virtualization platform, providing the ability to implement and virtualize different workloads across multiple VMs. The TOE is a software-only TOE where the core component is installed directly on the bare metal hardware.

The logical boundary is summarized in section 2.4.2 below.

2.4 TOE Architecture

The VMware ESXi TOE consists solely of the VMware ESXi hypervisor.

2.4.1 Physical Boundary

The TOE is a Type 1 hypervisor, which means that it consists of software/firmware components that are installed directly onto a physical system without an intermediary operating system. Figure 1 shows the relationship between the TOE boundary and other components. Figure 2 shows the external interfaces and internal components of the TOE.

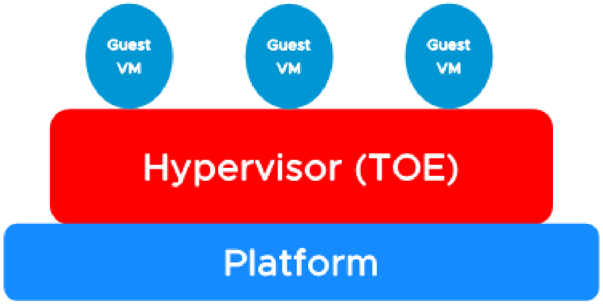

Figure 1: TOE Boundary

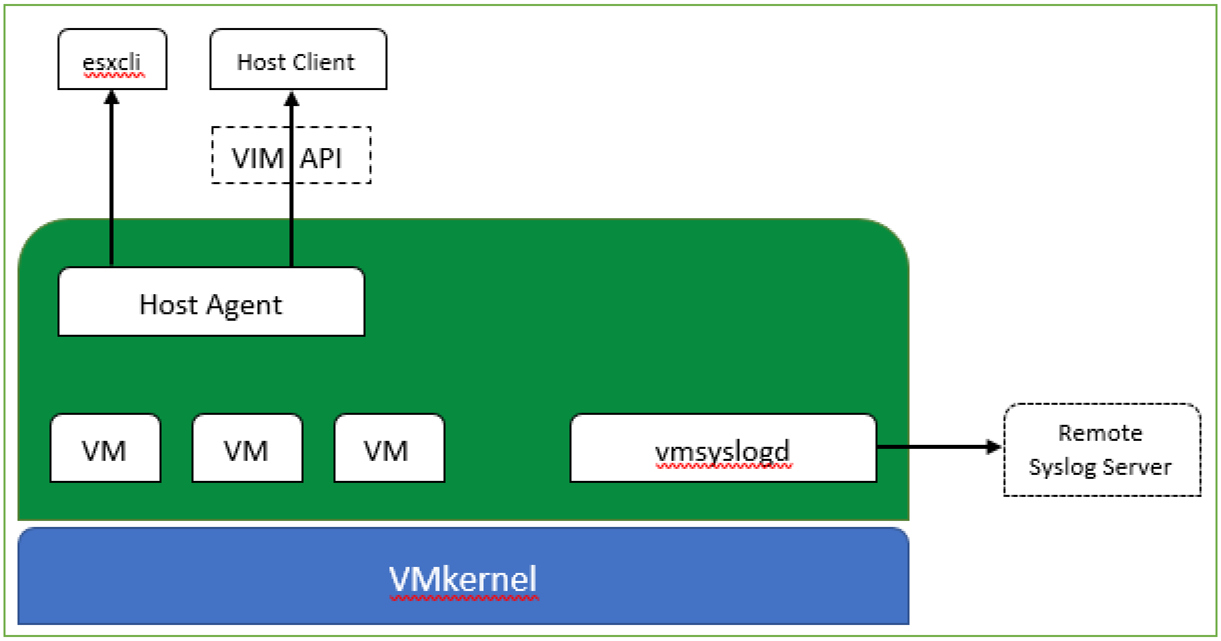

Figure 2: TOE External Interfaces and Internal Components

The physical hardware platform for this evaluation is a Dell PowerEdge R740 server with Intel Xeon Gold 6230R "Cascade Lake" CPUs. This CPU family was selected to provide the Intel VT and EPT feature support required by ESXi, as well as the RDSEED instruction used as an entropy source both internally and as a passthrough entropy source for virtual machines.

The TOE specifies only the above Intel Xeon 6230R "Cascade Lake" CPU. Usage of other Intel CPUs is subject to equivalence arguments which are outside the scope of this evaluation. Usage of AMD CPUs is outside the scope of this evaluation. Though VMware ESXi supports AMD CPUs with AMD-V, this configuration is not evaluated and thus usage of AMD CPUs is not NIAP-certified.

VMware ESXi additionally includes the following features that are not part of the evaluated TOE because they are outside the scope of the functionality described by the TOE's conformance claims:

- 3rd Party VIBs (distributed independently of VMware ESXi)
- Active Directory integration
- Common Information Model (CIM)
- Direct Console User Interface
- IPsec
- NSX software
- PCI passthrough (i.e. VMDirectPath I/O), including vGPU
- Physical optical drives (CD/DVD)
- Raw disks (RDM passthrough of storage LUNs)
- Remote shell (SSH)
- SCSI passthrough
- Simple Network Management Protocol (SNMP)
- USB passthrough
- vCenter Server software
- Virtual Shared Disks (Multiwriter disks)
- VM encryption
- VM Virtual Disk sharing
- vMotion
- VMware PowerCLI software
- vSAN software.

Additionally, the Guest VM software is not provided by VMware. Customers supply their own operating systems from 3rd party operating system vendors (e.g. Microsoft Windows Server 2016 or Red Hat Enterprise Linux 8). Guest VMs and their contents are outside the scope of the TOE.

2.4.2 Logical Boundary

The logical boundary of the TOE consists of:

- Timely Security Updates
- Security Audit
- Cryptographic Support
- User Data Protection
- Identification and Authentication
- Security Management
- Protection of the TSF
- TOE Access
- Trusted Path/Channels.

2.4.2.1 Timely Security Updates

VMware maintains a product lifecycle for the TOE that includes a service-level agreement for the duration of product support and the process for reporting, diagnosing, and triaging security issues for the TOE.

2.4.2.2 Security Audit

The TOE's security audit function accepts audit records and stores them locally in pre-allocated files, as well as transmitting them to a remote syslog server via TLS. Each audit record contains relevant information about the audit event. Locally-stored audit records are reviewable by authorized subjects and protected from unauthorized deletion and modification.

2.4.2.3 Cryptographic Support

The TOE implements CAVP-validated cryptographic algorithms for its cryptographic services. These are used to support TLS and HTTPS communications. Trusted communications protocols are implemented using secure cryptographic parameters and in accordance with relevant standards. The TOE implements a NIST SP 800-90A conformant DRBG that is seeded with a combination of hardware and software entropy. The hardware entropy source used by the TOE is made available to Guest VMs through a passthrough interface.

2.4.2.4 User Data Protection

The TOE uses hardware-based mechanisms to constrain direct access of Guest VMs to PCI devices. Authorized subjects may configure a specific Guest VM to use USB and network interfaces, however access to PCI passthrough devices, vGPU devices, and SCSI passthrough devices are always prohibited. All volatile and non-volatile memory is cleared prior to allocation to a Guest VM so that domain separation between Guest VMs is enforced.

2.4.2.5 Identification and Authentication

To control access to the TSF, the TOE uses locally defined username/password credentials for authentication. All TSF-mediated actions require successful authentication prior to authorization. The TSF protects against brute-force password authentication attempts by locking an offending user account for a period of time when an excessive number of failed attempts have been accumulated. The TSF also enforces configuration of password complexity policies to further reduce the chance that a brute force authentication attack will succeed.

The TOE uses X.509 certificate validation services for TLS authentication and code signing. CRLs are used for revocation checking. The TSF rejects invalid certificates and those whose revocation status cannot be determined.

2.4.2.6 Security Management

The TOE includes management functions that allow for configuration of its own behavior as well as configuration and manipulation of Guest VMs, such as starting/stopping VMs, creating checkpoints for VMs, and configuring the VMs with virtual networking and physical device access. The TOE includes several management interfaces over which various management functions can be performed. The TOE implements role-based access control to grant members of different roles granular privileges to manage the TSF and its associated data. For the purpose of the evaluation, only the ‘Administrator’ role is defined.

The TOE also enforces physical and logical separation of management and operational networks and protects against data sharing between Guest VMs using virtual networking, unless specifically authorized by an Administrator.

2.4.2.7 Protection of the TSF

The TOE implements various mechanisms to protect itself from misuse. A Guest VM can only access devices assigned to it by an Administrator. Furthermore, the TOE validates parameters passed to virtual devices, and implements controls for transferring removable media between Guest VMs. The TOE includes a hypercall interface that allows Guest VMs to interact with the hypervisor. The TOE also uses hardware assists to eliminate the need for shadow page tables and reduce the use of binary translation.

The TOE enforces isolation between Guest VMs and between VMs and itself. It also implements various protection methods in the execution environment to protect against memory-based attacks. TOE updates are also integrity protected using code signing certificates.

2.4.2.8 TOE Access

The TOE supports the display of an advisory warning message regarding unauthorized use of the TOE before establishing an Administrator session.

2.4.2.9 Trusted Path/Channels

The TOE implements TLS and HTTPS for secure communications between itself and external entities, which include remote administrators and remote audit servers. The TOE also enforces unambiguous identification of Guest VMs to reduce the likelihood that a user will inadvertently input data to an unintended Guest VM.

2.5 TOE Documentation

VMware provides the following product documentation in support of the installation and secure use of the TOE:

- VMware vSphere Documentation portal
- VMware ESXi Installation and Setup (ESXi 7.0 Update 3 PDF)
- VMware ESXi Upgrade (ESXi 7.0 Update 3 PDF)
- VMware vSphere Security (ESXi 7.0 Update 3 PDF)
- vSphere Single Host Management – VMware Host Client (ESXi 7.0 Update 3 PDF)
- vSphere Virtual Machine Administration (ESXi 7.0 Update 3 PDF)
- Guidance Supplement for VMware ESXi 7.0 Update 3d, Version 1.0

This evaluation occurred with the versions of documentation listed above. The latest documents, which may be updated for releases following this evaluation, are maintained on the VMware documentation portal at https://docs.vmware.com/en/VMware-vSphere/index.html.

3 Security Problem Definition

This section defines the security problem that the TOE and its operational environment are intended to address. Specifically, the security problem comprises the following:

- Any known or assumed threats countered by the TOE or its operational environment.
- Any organizational security policies with which the TOE must comply.
- Any assumptions about the security aspects of the environment and/or the intended way the TOE should be used.

In general, the Virtualization PP and SV Module have presented a Security Problem Definition appropriate for a virtualization platform with the ability to implement and virtualize different workloads across multiple VMs, and as such is applicable to the TOE. The TLS Package is a collection of functional requirements and therefore does not define its own threats, assumptions, or organizational security policies.

4 Security Objectives

Security objectives are concise, abstract statements of the intended solution to the problem defined by the security problem definition. This high-level solution is divided into two parts: the security objectives for the TOE, and the security objectives for the TOE's operational environment. This section identifies the security objectives for the TOE and its supporting environment. Note that all security objectives were taken directly from the Virtualization PP; the SV Module and TLS Package do not separately define any objectives.

4.1 Security Objectives for the TOE

Table 3: Security Objectives for TOE

Objective	Description
O.VM_ISOLATION	VMs are the fundamental subject of the system. The VMM is responsible for applying the system security policy (SSP) to the VM and all resources. As basic functionality, the VMM must support a security policy that mandates no information transfer between VMs. The VMM must support the necessary mechanisms to isolate the resources of all VMs. The VMM partitions a platform's physical resources for use by the supported virtual environments. Depending on customer requirements, a VM may need a completely isolated environment with exclusive access to system resources or share some of its resources with other VMs. It must be possible to enforce a security policy that prohibits the transfer of data between VMs through shared devices. When the platform security policy allows the sharing of resources across VM boundaries, the VMM must ensure that all access to those resources is consistent with the policy. The VMM may delegate the responsibility for the mediation of resource sharing to select Service VMs; however in doing so, it remains responsible for mediating access to the Service VMs, and each Service VM must mediate all access to any shared resource that has been delegated to it in accordance with the SSP. Both virtual and physical devices are resources requiring access control. The VMM must enforce access control in accordance with system security policy. Physical devices are platform devices with access mediated via the VMM per the O.VMM_Integrity objective. Virtual devices may include virtual storage devices and virtual network devices. Some of the access control restrictions must be enforced internal to Service VMs, as may be the case for isolating virtual networks. VMMs may also expose purely virtual interfaces. These are VMM specific, and while they are not analogous to a physical device, they are also subject to access control. The VMM must support the mechanisms to isolate all resources associated with virtual networks and to limit a VM's access to only those virtual networks for which it has been configured. The VMM must also support the mechanisms to control the configurations of virtual networks according to the SSP.
O.VMM_INTEGRITY	Integrity is a core security objective for Virtualization Systems. To achieve system

Objective	Description
	integrity, the integrity of each VMM component must be established and maintained. This objective concerns only the integrity of the VS—not the integrity of software running inside of Guest VMs or of the physical platform. The overall objective is to ensure the integrity of critical components of a VS. Initial integrity of a VS can be established through mechanisms such as a digitally signed installation or update package, or through integrity measurements made at launch. Integrity is maintained in a running system by careful protection of the VMM from untrusted users and software. For example, it must not be possible for software running within a Guest VM to exploit a vulnerability in a device or hypercall interface and gain control of the VMM. The vendor must release patches for vulnerabilities as soon as practicable after discovery.
O.PLATFORM_INTEGRITY	The integrity of the VMM depends on the integrity of the hardware and software on which the VMM relies. Although the VS does not have complete control over the integrity of the platform, the VS should as much as possible try to ensure that no users or software hosted by the VS can undermine the integrity of the platform.
O.DOMAIN_INTEGRITY	While the VS is not responsible for the contents or correct functioning of software that runs within Guest VMs, it is responsible for ensuring that the correct functioning of the software within a Guest VM is not interfered with by other VMs.
O.MANAGEMENT_ACCESS	VMM management functions include VM configuration, virtualized network configuration, allocation of physical resources, and reporting. Only authorized users (administrators) may exercise management functions. Because of the privileges exercised by the VMM management functions, it must not be possible for the VMM's management components to be compromised without administrator notification. This means that unauthorized users cannot be permitted access to the management functions, and the management components must not be interfered with by Guest VMs or unprivileged users on other networks—including operational networks connected to the TOE. VMMs include a set of management functions that collectively allow administrators to configure and manage the VMM, as well as configure Guest VMs. These management functions are specific to the VS and are distinct from any other management functions that might exist for the internal management of any given Guest VM. These VMM management functions are privileged, with the security of the entire system relying on their proper use. The VMM management functions can be classified into different categories and the policy for their use and the impact to security may vary accordingly. The management functions are distributed throughout the VMM (within the VMM and Service VMs). The VMM must support the necessary mechanisms to enable the control of all management functions according to the system security policy. When a management function is distributed among multiple Service VMs, the VMs must be protected using the security mechanisms of the Hypervisor and any Service VMs involved to ensure that the intent of the system security policy is not compromised. Additionally, since hypercalls permit Guest VMs to invoke the Hypervisor, and often

Objective	Description
	allow the passing of data to the Hypervisor, it is important that the hypercall interface is well-guarded and that all parameters be validated. The VMM maintains configuration data for every VM on the system. This configuration data, whether of Service or Guest VMs, must be protected. The mechanisms used to establish, modify and verify configuration data are part of the VS management functions and must be protected as such. The proper internal configuration of Service VMs that provide critical security functions can also greatly impact VS security. These configurations must also be protected. Internal configuration of Guest VMs should not impact overall VS security. The overall goal is to ensure that the VMM, including the environments internal to Service VMs, is properly configured and that all Guest VM configurations are maintained consistent with the system security policy throughout their lifecycle. Virtualization Systems are often managed remotely. For example, an administrator can remotely update virtualization software, start and shut down VMs, and manage virtualized network connections. If a console is required, it could be run on a separate machine or it could itself run in a VM. When performing remote management, an administrator must communicate with a privileged management agent over a network. Communications with the management infrastructure must be protected from Guest VMs and operational networks.
O.PATCHED_SOFTWARE	The VS must be updated and patched when needed in order to prevent the potential compromise of the VMM, as well as the networks and VMs that it hosts. Identifying and applying needed updates must be a normal part of the operating procedure to ensure that patches are applied in a timely and thorough manner. In order to facilitate this, the VS must support standards and protocols that help enhance the manageability of the VS as an IT product, enabling it to be integrated as part of a manageable network (e.g., reporting current patch level and patchability).
O.VM_ENTROPY	VMs must have access to good entropy sources to support security-related features that implement cryptographic algorithms. For example, in order to function as members of operational networks, VMs must be able to communicate securely with other network entities—whether virtual or physical. They must therefore have access to sources of good entropy to support that secure communication.
O.AUDIT	An audit log must be created that captures accesses to the objects the TOE protects. The log of these accesses, or audit events, must be protected from modification, unauthorized access, and destruction. The audit log must be sufficiently detailed to indicate the date and time of the event, the identify of the user, the type of event, and the success or failure of the event.
O.CORRECTLY_APPLIED_ CONFIGURATION	The TOE must not apply configurations that violate the current security policy. The TOE must correctly apply configurations and policies to newly created Guest VMs, as well as to existing Guest VMs when applicable configuration or policy changes are made. All changes to configuration and to policy must conform to the existing security policy. Similarly, changes made to the configuration of the TOE itself must not violate the existing security policy.

Objective	Description
O.RESOURCE_ALLOCATION	The TOE will provide mechanisms that enforce constraints on the allocation of system resources in accordance with existing security policy.

4.2 Security Objectives for the Operational Environment

Table 4: Security Objectives for the Operational Environment

Objective	Description
OE.CONFIG	TOE administrators will configure the VS correctly to create the intended security policy.
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.
OE.NON_MALICIOUS_USER	Users are trusted to not be willfully negligent or hostile and use the VS in compliance with the applied enterprise security policy and guidance.

4.3 Security Objective Rationale

The rationale for the Threats, Assumptions, and Objectives for this Protection Profile can be found in Section 4.3 of the Virtualization PP. In addition, the inclusion of the SV Module further aids in the mitigation of potential threats.

5 IT Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) met by the TOE.

5.1 Extended Requirements

All of the extended requirements in this ST have been drawn from the Virtualization PP, SV Module, and TLS Package. These documents define the following extended SAR and extended SFRs; since they have not been redefined in this ST, the Virtualization PP, SV Module, and TLS Package should be consulted for more information regarding these extensions to CC Parts 2 and 3.

- ALC_TSU_EXT.1
- FAU_STG_EXT.1
- FCS_CKM_EXT.4
- FCS_ENT_EXT.1
- FCS_HTTPS_EXT.1
- FCS_RBG_EXT.1
- FCS_TLS_EXT.1
- FCS_TLSC_EXT.1
- FCS_TLSC_EXT.5
- FCS_TLSS_EXT.1
- FDP_HBI_EXT.1
- FDP_PPR_EXT.1
- FDP_RIP_EXT.1
- FDP_RIP_EXT.2
- FDP_VMS_EXT.1
- FDP_VNC_EXT.1
- FIA_AFL_EXT.1
- FIA_UIA_EXT.1
- FIA_PMG_EXT.1
- FIA_X509_EXT.1
- FIA_X509_EXT.2
- FMT_MOF_EXT.1
- FMT_SMO_EXT.1
- FPT_DVD_EXT.1
- FPT_EEM_EXT.1
- FPT_HAS_EXT.1
- FPT_HCL_EXT.1
- FPT_RDM_EXT.1
- FPT_TUD_EXT.1
- FPT_TUD_EXT.2
- FPT_VDP_EXT.1
- FPT_VIV_EXT.1
- FTP_ITC_EXT.1

- FTP_UIF_EXT.1
- FTP_UIF_EXT.2

5.2 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by the TOE.

Table 5: TOE Security Functional Components

Requirement Class	Requirement Component
FAU: Security Audit	FAU_GEN.1 Audit Data Generation
	FAU_SAR.1 Audit Review
	FAU_STG.1 Protected Audit Trail Storage
	FAU_STG_EXT.1 Off-Loading of Audit Data
FCS: Cryptographic Support	FCS_CKM.1 Cryptographic Key generation
	FCS_CKM.2 Cryptographic Key Establishment
	FCS_CKM_EXT.4 Cryptographic Key Destruction
	FCS_COP.1/Hash Cryptographic Operation (Hashing)
	FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithms)
	FCS_COP.1/Sig Cryptographic Operation (Signature Algorithms)
	FCS_COP.1/UDE Cryptographic Operation (AES Data Encryption/Decryption)
	FCS_ENT_EXT.1 Entropy for Virtual Machines
	FCS_HTTPS_EXT.1 HTTPS Protocol
	FCS_RBG_EXT.1 Cryptographic Operation (Random Bit Generation)
	FCS_TLS_EXT.1 TLS Protocol
	FCS_TLSC_EXT.1 TLS Client Protocol
	FCS_TLSC_EXT.5 TLS Client Support for Supported Groups Extension
	FCS_TLSS_EXT.1 TLS Server Protocol
FDP: User Data Protection	FDP_HBI_EXT.1 Hardware-Based Isolation Mechanisms
	FDP_PPR_EXT.1 Physical Platform Resource Controls
	FDP_RIP_EXT.1 Residual Information in Memory
	FDP_RIP_EXT.2 Residual Information on Disk
	FDP_VMS_EXT.1 VM Separation
	FDP_VNC_EXT.1 Virtual Networking Components
FIA: Identification and Authentication	FIA_AFL_EXT.1 Authentication Failure Handling
	FIA_UAU.5 Multiple Authentication Mechanisms
	FIA_UIA_EXT.1 Administrator Identification and Authentication
	FIA_PMG_EXT.1 Password Management
	FIA_X509_EXT.1 X.509 Certificate Validation

Requirement Class	Requirement Component
	FIA_X509_EXT.2 X.509 Certificate Authentication
FMT: Security Management	FMT_MOF_EXT.1 Management of Security Functions Behavior
	FMT_SMO_EXT.1 Separation of Management and Operational Networks
FPT: Protection of the TSF	FPT_DVD_EXT.1 Non-Existence of Disconnected Virtual Devices
	FPT_EEM_EXT.1 Execution Environment Mitigations
	FPT_HAS_EXT.1 Hardware Assists
	FPT_HCL_EXT.1 Hypercall Controls
	FPT_RDM_EXT.1 Removable Devices and Media
	FPT_TUD_EXT.1 Trusted Updates to the Virtualization System
	FPT_TUD_EXT.2 Trusted Update Based on Certificates
	FPT_VDP_EXT.1 Virtual Device Parameters
	FPT_VIV_EXT.1 VMM Isolation from VMs
FTA: TOE Access	FTA_TAB.1 TOE Access Banner
FTP: Trusted Path/Channel	FTP_ITC_EXT.1 Trusted Channel Communications
	FTP_TRP.1 Trusted Path for Remote Administration
	FTP_UIF_EXT.1 User Interface: I/O Focus
	FTP_UIF_EXT.2 User Interface: Identification of VM

5.2.1 Security Audit (FAU)

5.2.1.1 Audit Data Generation (FAU_GEN.1)

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a. Start-up and shutdown of the audit functions
b. [All administrative actions relevant to claimed SFRs as defined in the Auditable Events Table from the Client and Server PP-Modules]
c. [Auditable events defined in Table **6**]
d. [Auditable events defined in Table 7 for Selection-Based SFRs,
e. Auditable events for the Functional Package for Transport Layer Security (TLS), version 1.1 listed in Table **7**].

Application Note: *Item b in this case refers to the auditable events defined in any claimed PP-Modules. The TOE only claims the Server PP-Module so there are no relevant administrative actions related to client virtualization. The auditable events table in the Server PP-Module consists of one entry for FMT_MOF_EXT.1; this has been added to Table 6.*

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a. Date and time of the event

b. Type of event
c. Subject and object identity (if applicable)
d. The outcome (success or failure) of the event
e. [Additional information defined in Table **6**]
f. [Additional information defined in Table 7 for Selection-Based SFRs,
g. Additional information for the Functional Package for Transport Layer Security (TLS), version 1.1 listed in Table **7**].

Table 6: Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	No events specified	
FAU_SAR.1	No events specified	
FAU_STG.1	No events specified	
FAU_STG_EXT.1	Failure of audit data capture due to lack of disk space or pre-defined limit.	No additional information.
	On failure of logging function, capture record of failure and record upon restart of logging function.	
FCS_CKM.1	No events specified	
FCS_CKM.2	No events specified	
FCS_CKM_EXT.4	No events specified	
FCS_COP.1/Hash	No events specified	
FCS_COP.1/KeyedHash	No events specified	
FCS_COP.1/Sig	No events specified	
FCS_COP.1/UDE	No events specified	
FCS_ENT_EXT.1	No events specified	
FCS_RBG_EXT.1	Failure of the randomization process.	No additional information.
FDP_HBI_EXT.1	No events specified	
FDP_PPR_EXT.1	Security policy violations.	Identifier for the security policy that was violated.
	Successful and failed VM connections to physical devices where connection is governed by configurable policy.	VM and physical device identifiers.
FDP_RIP_EXT.1	No events specified	
FDP_RIP_EXT.2	No events specified	
FDP_VMS_EXT.1	No events specified	
FDP_VNC_EXT.1	Successful and failed attempts to connect VMs to virtual and physical networking components.	VM and virtual or physical networking component identifiers.

Requirement	Auditable Events	Additional Audit Record Contents
	Security policy violations.	Identifier for the security policy that was violated. VM and virtual or physical networking component identifiers.
	Administrator configuration of inter-VM communications channels between VMs.	VM and virtual or physical networking component identifiers.
FIA_AFL_EXT.1	Unsuccessful login attempts limit is met or exceeded.	Origin of attempt (e.g., IP address).
FIA_UAU.5	No events specified	
FIA_UIA_EXT.1	Administrator authentication attempts.	Provided user identity, origin of the attempt (e.g. console, remote IP address).
	All use of the identification and authentication mechanism.	
	[Start and end of administrator session.]	Start time and end time of administrator session.
FMT_MOF_EXT.1	Attempts to invoke any of the management functions listed in Table **8**	Success or failure of attempt Identity of actor
FMT_SMO_EXT.1	No events specified	
FPT_DVD_EXT.1	No events specified	
FPT_EEM_EXT.1	No events specified	
FDP_HAS_EXT.1	No events specified	
FPT_HCL_EXT.1[1]	[Invalid parameter to hypercall detected.]	Hypercall interface for which access was attempted.
	[Hypercall interface invoked when documented preconditions are not met.]	No additional information.
FPT_RDM_EXT.1	Connection/disconnection of removable media or device to/from a VM.	VM Identifier, Removable media/device identifier, event description or identifier (connect/disconnect, ejection/insertion, etc.)
	Ejection/insertion of removable media or device from/to an already connected VM.	
FPT_TUD_EXT.1	Initiation of Update.	No additional information.
	Failure of signature verification.	
FPT_VDP_EXT.1	No events specified	
FPT_VIV_EXT.1	No events specified	

[1] Entries modified per NIAP TD0615

Requirement	Auditable Events	Additional Audit Record Contents
FTA_TAB.1	No events specified	
FTP_ITC_EXT.1	Initiation of the trusted channel.	User ID and remote source (IP Address) if feasible.
	Termination of the trusted channel.	
	Failures of the trusted path functions.	
FTP_UIF_EXT.1	No events specified	
FTP_UIF_EXT.2	No events specified	

Table 7: Additional Auditable Events Based on Selections

Requirement	Auditable Events	Additional Audit Record Contents
FCS_HTTPS_EXT.1	Failure to establish a HTTPS session.	Reason for failure.
	Establishment/Termination of an HTTPS session.	Non-TOE endpoint of connection (IP address) for both successes and failures.
FCS_TLSC_EXT.1	Failure to establish a session.	Reason for failure.
	Failure to verify presented identifier.	Presented identifier and reference identifier.
	Establishment/Termination of a TLS session.	Non-TOE endpoint of connection.
FCS_TLSS_EXT.1	Failure to establish a TLS session.	Reason for failure.
FIA_X509_EXT.1	Failure to validate a certificate.	Reason for failure.
FIA_X509_EXT.2	None.	None.
FIA_PMG_EXT.1	None.	None.
FPT_TUD_EXT.2	None.	None.
FTP_TRP.1	Initiation of the trusted channel.	User ID and remote source (IP address) if feasible.
	Termination of the trusted channel.	
	Failure of the trusted channel functions.	

5.2.1.2 Audit Review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide [administrators] with the capability to read [all information] from the audit records.

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

5.2.1.3 Protected Audit Trail Storage (FAU_STG.1)

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.

FAU_STG.1.2 The TSF shall be able to [prevent] unauthorized modifications to the stored audit records in the audit trail.

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

FAU_STG_EXT.1.1 The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel as specified in FTP_ITC_EXT.1.

FAU_STG_EXT.1.2 The TSF shall [overwrite previous audit records according to the following rule: [*oldest audit records are overwritten by newest audit record*]] when the local storage space for audit data is full.

5.2.2 Cryptographic Support (FCS)

5.2.2.1 Cryptographic Key Generation (FCS_CKM.1)

FCS_CKM.1.1 The TSF shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm [

- 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],
- ECC schemes using ["NIST curves" P-256, P-384, and [P-521]] that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4]].

5.2.2.2 Cryptographic Key Establishment (FCS_CKM.2)

FCS_CKM.2.1 The TSF shall implement functionality to perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [

- RSA-based key establishment schemes that meets the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.2",
- Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"].

5.2.2.3 Cryptographic Key Destruction (FCS_CKM_EXT.4)

FCS_CKM_EXT.4.1 The TSF shall cause disused cryptographic keys in volatile memory to be destroyed or rendered unrecoverable.

FCS_CKM_EXT.4.2 The TSF shall cause disused cryptographic keys in non-volatile storage to be destroyed or rendered unrecoverable.

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

FCS_COP.1.1/Hash The TSF shall perform [cryptographic hashing] in accordance with a specified cryptographic algorithm [SHA-1, SHA-256, SHA-384, SHA-512] and message digest sizes [160, 256, 384, 512 bits] that meet the following: [FIPS PUB 180-4 “Secure Hash Standard”].

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

FCS_COP.1.1/KeyedHash The TSF shall perform [keyed-hash message authentication] in accordance with a specified cryptographic algorithm [HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512] and cryptographic key sizes [*160, 256, 384, 512 bits*] and message digest sizes [160, 256, 384, 512 bits] that meet the following: [FIPS PUB 198-1, "The Keyed-Hash Message Authentication Code”, FIPS PUB 180-4, “Secure Hash Standard”].

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

FCS_COP.1.1/Sig The TSF shall perform [cryptographic signature services (generation and verification)] in accordance with a specified cryptographic algorithm [

- RSA schemes using cryptographic key sizes [2048-bit or greater] that meet the following: [FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 4],
- 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]].

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

FCS_COP.1.1/UDE The TSF shall perform [encryption and decryption] in accordance with a specified cryptographic algorithm: [

- AES-GCM (as defined in NIST SP 800-38D),
- AES-CBC (as defined in FIPS PUB 197, and NIST SP 800-38A) mode]

and cryptographic key sizes [128-bit, 256-bit].

5.2.2.8 Entropy for Virtual Machines (FCS_ENT_EXT.1)

FCS_ENT_EXT.1.1 The TSF shall provide a mechanism to make available to VMs entropy that meets FCS_RBG_EXT.1 through [passthrough access to hardware entropy source].

FCS_ENT_EXT.1.2 The TSF shall provide independent entropy across multiple VMs.

5.2.2.9 HTTPS (FCS_HTTPS_EXT.1)

FCS_HTTPS_EXT.1.1 The TSF shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS.

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

FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation services in accordance with NIST Special Publication 800-90A using [CTR_DRBG (AES)].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by an entropy source that accumulates entropy from [a software-based noise source, a hardware-based noise source] with a minimum of [256 bits] of entropy at least equal to the greatest security strength according to NIST SP 800-57, of the keys and hashes that it will generate.

5.2.2.11 TLS Protocol (FCS_TLS_EXT.1)

FCS_TLS_EXT.1.1 The product shall implement [

- TLS as a client,
- TLS as a server].

5.2.2.12 TLS Client Protocol (FCS_TLSC_EXT.1)[2]

FCS_TLSC_EXT.1.1 The product shall implement TLS 1.2 (RFC 5246) and [no earlier TLS versions] as a client that supports the cipher suites: [

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246,
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,
- TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289]

and also supports functionality for [

- none].

[2] As modified by TD0442

FCS_TLSC_EXT.1.2 The product shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.1.3 The product shall not establish a trusted channel if the server certificate is invalid [with no exceptions].

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

FCS_TLSC_EXT.5.1 The product shall present the Supported Groups Extension in the Client Hello with the supported groups: [

- secp256r1,
- secp384r1,
- secp521r1].

5.2.2.14 TLS Server Protocol (FCS_TLSS_EXT.1)[3]

FCS_TLSS_EXT.1.1 The product shall implement TLS 1.2 (RFC 5246) and [no earlier TLS versions] as a server that supports the following cipher suites: [

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246,
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,
- TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

] and also supports functionality for [

- no session resumption or session tickets].

FCS_TLSS_EXT.1.2 The product shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, and [TLS 1.1].

FCS_TLSS_EXT.1.3 The product shall perform key establishment for TLS using [

- RSA with size [2048 bits],
- ECDHE parameters using elliptic curves [secp256r1, secp384r1, secp521r1]].

[3] As modified by TD0442 and TD0588

5.2.3 User Data Protection (FDP)

5.2.3.1 Hardware-Based Isolation Mechanisms (FDP_HBI_EXT.1)

FDP_HBI_EXT.1.1 The TSF shall use [[*Intel VT-x, EPT, VT-d*]] to constrain a Guest VM's direct access to the following physical devices: [[*PCI devices*]].

5.2.3.2 Physical Platform Resource Controls (FDP_PPR_EXT.1)

FDP_PPR_EXT.1.1 The TSF shall allow an authorized administrator to control Guest VM access to the following physical platform resources: [*USB, network adapter*].

FDP_PPR_EXT.1.2 The TSF shall explicitly deny all Guest VMs access to the following physical platform resources: [[*PCI Passthrough devices, Storage Raw Device Mapping (SCSI passthrough)*]].

FDP_PPR_EXT.1.3 The TSF shall explicitly allow all Guest VMs access to the following physical platform resources: [no physical platform resources].

5.2.3.3 Residual Information in Memory (FDP_RIP_EXT.1)

FDP_RIP_EXT.1.1 The TSF shall ensure that any previous information content of physical memory is cleared prior to allocation to a Guest VM.

5.2.3.4 Residual Information on Disk (FDP_RIP_EXT.2)

FDP_RIP_EXT.2.1 The TSF shall ensure that any previous information content of physical disk storage is cleared to zeros upon allocation to a Guest VM.

5.2.3.5 VM Separation (FDP_VMS_EXT.1)

FDP_VMS_EXT.1.1 The VS shall provide the following mechanisms for transferring data between Guest VMs: [virtual networking].

FDP_VMS_EXT.1.2 The TSF shall by default enforce a policy prohibiting sharing of data between Guest VMs.

FDP_VMS_EXT.1.3 The TSF shall allow Administrators to configure the mechanisms selected in FDP_VMS_EXT.1.1 to enable and disable the transfer of data between Guest VMs.

FDP_VMS_EXT.1.4 The VS shall ensure that no Guest VM is able to read or transfer data to or from another Guest VM except through the mechanisms listed in FDP_VMS_EXT.1.1.

5.2.3.6 Virtual Networking Components (FDP_VNC_EXT.1)

FDP_VNC_EXT.1.1 The TSF shall allow Administrators to configure virtual networking components to connect VMs to each other and to physical networks.

FDP_VNC_EXT.1.2 The TSF shall ensure that network traffic visible to a Guest VM on a virtual network—or virtual segment of a physical network—is visible only to Guest VMs configured to be on that virtual network or segment.

5.2.4 Identification and Authentication (FIA)

5.2.4.1 Authentication Failure Handling (FIA_AFL_EXT.1)

FIA_AFL_EXT.1.1 The TSF shall detect when [

- an administrator configurable positive integer within a [*1-10*]]

unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using [username and password].

FIA_AFL_EXT.1.2 When the defined number of unsuccessful authentication attempts has been met, the TSF shall: [prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password ~~or PIN~~ until an Administrator-defined time period has elapsed].

5.2.4.2 Password Management (FIA_PMG_EXT.1)

FIA_PMG_EXT.1.1 The TSF shall provide the following password management capabilities for administrative passwords:

a. Passwords shall be able to be composed of any combination of upper and lower case characters, digits, and the following special characters: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”]
b. Minimum password length shall be configurable
c. Passwords of at least 15 characters in length shall be supported

5.2.4.3 Multiple Authentication Mechanisms (FIA_UAU.5)

FIA_UAU.5.1 The TSF shall provide the following authentication mechanisms: [

- [local] authentication based on username and password

to support Administrator authentication.

FIA_UAU.5.2 The TSF shall authenticate any Administrator’s claimed identity according to the [*successful authentication through username and password*].

5.2.4.4 Administrator Identification and Authentication (FIA_UIA_EXT.1)

FIA_UIA_EXT.1.1 The TSF shall require Administrators to be successfully identified and authenticated using one of the methods in FIA_UAU.5 before allowing any TSF-mediated management function to be performed by that Administrator.

5.2.4.5 X.509 Certificate Validation (FIA_X509_EXT.1)

FIA_X509_EXT.1.1 The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.

- The certificate path must terminate with a trusted certificate.
- The TOE shall validate a certificate path by ensuring the presence of the basicConstraints extension, that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met.
- The TSF shall validate that any CA certificate includes caSigning purpose in the key usage field
- The TSF shall validate the revocation status of the certificate using [a CRL as specified in RFC 5759].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the EKU field.
    - OCSP certificates presented for OCSP responses shall have the OCSP Signing Purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the EKU field.

FIA_X509_EXT.1.2 The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

5.2.4.6 X.509 Certificate Authentication (FIA_X509_EXT.2)

FIA_X509_EXT.2.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [TLS, HTTPS, code signing for system software updates, [*code signing for integrity verification*]].

FIA_X509_EXT.2.2 When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [not accept the certificate].

5.2.5 Security Management (FMT)

5.2.5.1 Management of Security Functions Behavior (FMT_MOF_EXT.1)

FMT_MOF_EXT.1.1 The TSF shall be capable of supporting [remote] administration.

FMT_MOF_EXT.1.2 The TSF shall be capable of performing the following management functions, controlled by an Administrator or User as shown in Table **8**, based on the following key:

X = Mandatory (TOE must provide that function to that role)

N = Not Permitted (TOE must not provide that function to that role)

S = Selection-Based (TOE must provide that function to that role if the TOE claims a particular selection-based SFR)

Table 8: Server Virtualization Management Functions

No	Function	Administrator	User	Note (all SFR references from Base PP)
1	Ability to update the Virtualization System	X	N	See FPT_TUD_EXT.1
2	[Ability to configure Administrator password policy as defined in FIA_PMG_EXT.1]	S	N	Must be selected if ST includes FIA_PMG_EXT.1.
3	Ability to create, configure and delete VMs	X	**N**	
4	Ability to set default initial VM configurations	X	N	
5	Ability to configure virtual networks including VM	X	**N**	See FDP_VNC_EXT.1
6	Ability to configure and manage the audit system and audit data	X	N	
7	Ability to configure VM access to physical devices	X	**N**	See FDP_PPR_EXT.1
8	Ability to configure inter-VM data sharing	X	**N**	See FDP_VMS_EXT.1
9	~~Ability to enable/disable VM access to Hypercall functions~~	**N**	**N**	Management function 9 is no longer required
10	Ability to configure removable media policy	X	N	See FPT_RDM_EXT.1
11	Ability to configure the cryptographic functionality	X	N	See FCS_CKM.1, FCS_CKM.2, and FCS_COP.1/HASH. See also, the Functional Packages for Transport Layer Security (TLS) and for Secure Shell (SSH) if claimed for methods to configure their respective cryptographic functionality.
12	Ability to change default authorization factors	X	N	See FIA_PMG_EXT.1
13	Ability to enable/disable screen lock	**N**	**N**	
14	Ability to configure screen lock inactivity timeout	**N**	**N**	
15	Ability to configure remote connection inactivity timeout	X	N	
16	Ability to configure lockout policy for unsuccessful authentication attempts through [limiting number of attempts during a time period]	X	N	See FIA_AFL_EXT.1

No	Function	Administrator	User	Note (all SFR references from Base PP)
17	[Not applicable]	**N**	**N**	Must be selected if "directory-based" is selected anywhere in FIA_UAU.5.1 in the Base Virtualization PP.
18	Ability to configure name/address of audit/logging server to which to send audit/logging records	X	N	See FAU_STG_EXT.1
19	Ability to configure name/address of network time server	X	**N**	
20	Ability to configure banner	X	**N**	See FTA_TAB.1
21	Ability to connect/disconnect removable devices to/from a VM	**X**	**N**	See FPT_RDM_EXT.1
22	Ability to start a VM	**X**	**N**	
23	Ability to stop/halt a VM	**X**	**N**	
24	Ability to checkpoint a VM	**X**	**N**	
25	Ability to suspend a VM	**X**	**N**	
26	Ability to resume a VM	**X**	**N**	
27	[Not applicable]	**N**	N	This function must be selected if "allow the administrator to choose whether to accept the certificate in these cases" in FIA_X509_EXT.2.2 in the Base-PP.

Application Note: *The PP-Module defines some role-function combinations as optional. For each optional role-function combination, this ST specifies whether or not the TOE supports them and indicates this through refinements. All instances of 'O' (for Optional) in the PP-Module are replaced with 'X' or 'N" based on whether or not the corresponding claim is made; the removal of the 'O' is not shown as a refinement for readability purposes.*

The TOE claims FIA_PMG_EXT.1 so function 2 is supported. The TOE is intended to run on a headless server so functions 13 and 14 are not supported. The TOE does not claim directory-based authentication so function 17 is not supported. The TOE does not allow for configuration of how a certificate is handled when its revocation status is undetermined so function 27 is not supported.

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

FMT_SMO_EXT.1.1 The TSF shall support the separation of management and operational network traffic through [separate physical networks, separate logical networks, trusted channels as defined in FTP_ITC_EXT.1].

5.2.6 Protection of the TSF (FPT)

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

FPT_DVD_EXT.1.1 The TSF shall prevent Guest VMs from accessing virtual device interfaces that are not present in the VM's current virtual hardware configuration.

5.2.6.2 Execution Environment Mitigations (FPT_EEM_EXT.1)

FPT_EEM_EXT.1.1 The TSF shall take advantage of execution environment-based vulnerability mitigation mechanisms supported by the Platform such as: [

- Address-space randomization,
- Memory execution protection (e.g., DEP),
- Stack buffer overflow protection

].

5.2.6.3 Hardware Assists (FPT_HAS_EXT.1)

FPT_HAS_EXT.1.1 The VMM shall use [*Intel VT-x*] to reduce or eliminate the need for binary translation.

FPT_HAS_EXT.1.2 The VMM shall use [*Intel Extended Page Tables (EPT)*] to reduce or eliminate the need for shadow page tables.

5.2.6.4 Hypercall Controls (FPT_HCL_EXT.1)

FPT_HCL_EXT.1.1 The TSF shall validate the parameters passed to Hypercall interfaces prior to execution of the VMM functionality exposed by each interface.

5.2.6.5 Removable Devices and Media (FPT_RDM_EXT.1)

FPT_RDM_EXT.1.1 The TSF shall implement controls for handling the transfer of virtual and physical removable media and virtual and physical removable media devices between information domains.

FPT_RDM_EXT.1.2 The TSF shall enforce the following rules when [*physical USB media, CD/DVD image media*] are switched between information domains, then [

- the Administrator has granted explicit access for the media or device to be connected to the receiving domain

].

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

FPT_TUD_EXT.1.1 The TSF shall provide Administrators the ability to query the currently executed version of the TOE firmware/software as well as the most recently installed version of the TOE firmware/software.

FPT_TUD_EXT.1.2 The TSF shall provide administrators the ability to manually initiate updates to TOE firmware/software and [no other update mechanism].

FPT_TUD_EXT.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a [digital signature mechanism using certificates] prior to installing those updates.

5.2.6.7 Trusted Update Based on Certificates (FPT_TUD_EXT.2)

FPT_TUD_EXT.2.1 The TSF shall not install an update if the code signing certificate is deemed invalid.

5.2.6.8 Virtual Device Parameters (FPT_VDP_EXT.1)

FPT_VDP_EXT.1.1 The TSF shall provide interfaces for virtual devices implemented by the VMM as part of the virtual hardware abstraction.

FPT_VDP_EXT.1.2 The TSF shall validate the parameters passed to the virtual device interface prior to execution of the VMM functionality exposed by those interfaces.

5.2.6.9 VMM Isolation from VMs (FPT_VIV_EXT.1)

FPT_VIV_EXT.1.1 The TSF must ensure that software running in a VM is not able to degrade or disrupt the functioning of other VMs, the VMM, or the Platform.

FPT_VIV_EXT.1.2 The TSF must ensure that a Guest VM is unable to invoke platform code that runs at a privilege level equal to or exceeding that of the VMM without involvement of the VMM.

5.2.7 TOE Access (FTA)

5.2.7.1 TOE Access Banner (FTA_TAB.1)

FTA_TAB.1.1 Before establishing an administrative user session, the TSF shall display a security Administrator-specified advisory notice and consent warning message regarding use of the TOE.

5.2.8 Trusted Path/Channels (FTP)

5.2.8.1 Trusted Channel Communications (FTP_ITC_EXT.1)

FTP_ITC_EXT.1.1 The TSF shall use [

- TLS as conforming to the Functional Package for Transport Layer Security,
- TLS/HTTPS as conforming to FCS_HTTPS_EXT.1] and [
- certificate-based authentication of the remote peer

] to provide a trusted communication channel between itself and:

- audit servers (as required by FAU_STG_EXT.1), and
- [remote administrators (as required by FTP_TRP.1.1 if selected in FMT_MOF_EXT.1.1 in the Client or Server PP-Module),

- Separation of management and operational networks (if selected in FMT_SMO_EXT.1)

] that is logically distinct from other communication paths and provides assured identification of its endpoints and protection of the communicated data from disclosure and detection of modification of the communicated data.

5.2.8.2 Trusted Path (FTP_TRP.1)

FTP_TRP.1.1 The TSF shall use a trusted channel as specified in FTP_ITC_EXT.1 to provide a trusted communication path between itself and [remote] administrators that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [modification, disclosure].

FTP_TRP.1.2 The TSF shall permit [remote administrators] to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for [[*all remote administration actions*]].

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

FTP_UIF_EXT.1.1 The TSF shall indicate to users which VM, if any, has the current input focus.

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

FTP_UIF_EXT.2.1 The TSF shall support the unique identification of a VM's output display to users.

5.3 TOE Security Assurance Requirements

This section defines the Security Assurance Requirements (SARs) for the TOE. The assurance requirements are taken from Protection Profile for Virtualization Version 1.1 with Server Virtualization PP-Module 1.1. The assurance components are summarized in the following table:

Table 9: Assurance Components

Requirement Class	Requirement Component
ASE: Security Target	ASE_CCL.1 Conformance Claims
	ASE_ECD.1 Extended Components Definition
	ASE_INT.1 ST Introduction
	ASE_OBJ.2 Security Objectives
	ASE_REQ.2 Derived Security Requirements
	ASE_TSS.1 TOE Summary Specification
ADV: Development	ADV_FSP.1 Basic Functional Specification
AGD: Guidance Documents	AGD_OPE.1 Operational User Guidance
	AGD_PRE.1 Preparative Procedures

ALC: Lifecycle Support	ALC_CMC.1 Labeling of the TOE
	ALC_CMS.1 TOE CM Coverage
	ALC_TSU_EXT.1 Timely Security Updates
ATE: Tests	ATE_IND.1 Independent Testing – Conformance
AVA: Vulnerability Analysis	AVA_VAN.1 Vulnerability Survey

6 TOE Summary Specification

This chapter describes the following security functions:

- Timely Security Updates
- Security Audit
- Cryptographic Support
- User Data Protection
- Identification and Authentication
- Security Management
- Protection of the TSF
- TOE Access
- Trusted Path/Channels.

6.1 Timely Security Updates

ESXi is a software product. Security updates are released as software updates (installable ISO images) or patches (installed using command line or ESXi Host Client) distributed from VMware's website. The patching mechanism is described in the TOE's operational guidance. Customers are expected to apply updates and patches to address bugs or security issues found after the product releases.

The core ESXi product is in the "esx-base" VIB; other VIBs contain device drivers or additional products. For example, there are VIBs that include NSX or VSAN, both of which are outside the scope of the TOE. All updates and nearly all patches will include an updated "esx-base" VIB.

VMware 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, 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, 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.

6.2 Security Audit

The TOE includes a security audit function for recording security-relevant behavior that occurs. The TSF generates audit records for all audit events listed in Table 6 and Table 7 above. Each audit record includes date, time, applicable subject and object identities, the outcome of the event, and any additional information required by the TOE's conformance claims on a per-event basis.

Specific examples of each audit record can be found in the supplemental administrative guidance.

Audit records are stored on the TOE's file system as flat files. They are protected 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.

The TOE also has the ability to transmit audit data to a remote syslog server using TLS 1.2. In the event of a loss of connectivity to this server, the TOE will generate a local audit record indicating the connectivity loss. Audit events captured during this connection outage will only be recorded locally. Upon restoration of connectivity, the TOE will generate an audit record that is also received by the remote server that an outage has taken place. Events that occurred during this outage are not subsequently transmitted to the remote server and must be reviewed in the local record.

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. Configuration of local audit storage retention does not affect the remote syslog server.

6.3 Cryptographic Support

The TOE uses cryptography to secure data in transit between itself and its operational environment. The keys that are stored by the TOE, how they are generated, what they are used for, how they are output, how they are stored, and how they are erased is discussed in section 9 below.

All TOE cryptographic services are implemented by the OpenSSL cryptographic library, except for random number generation, which is implemented in user space by OpenSSL but in kernel space by Vmkdrbg. The TOE uses VMware OpenSSL FIPS Object Module version 2.0.20-vmw. Vmkdrbg is not separately versioned; the TOE uses the ESXi 7.0 Vmkdrbg. The cryptographic algorithms supplied by the TOE are NIST-validated. The following table identifies the cryptographic algorithms used by the TSF, the associated standards to which they conform, and the NIST certificates that demonstrate that the claimed conformance has been met.

Table 10: Cryptographic Functions

Functions	Libraries	Standards	CAVP Certificates
FCS_CKM.1 Cryptographic Key Generation			
ECC key pair generation (NIST curves P-256, P-384, P-521)	OpenSSL	FIPS PUB 186-4	A1292

Functions	**Libraries**	**Standards**	**CAVP Certificates**
RSA key pair generation (2048-bit, 3072-bit)	OpenSSL	FIPS PUB 186-4	A1292
FCS_CKM.2 Cryptographic Key Establishment			
ECC based key establishment	OpenSSL	NIST SP 800-56A Rev. 3	A1292
RSA based key establishment	OpenSSL	RFC 8017 section 7.2	Vendor affirmed
FCS_COP.1/Hash Cryptographic Operation (Hashing)			
SHA-1, SHA-256, SHA-384, SHA-512	OpenSSL	FIPS PUB 180-4	A1292
FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithms)			
HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	OpenSSL	FIPS PUB 198-1 FIPS PUB 180-4	A1292
FCS_COP.1/Sig Cryptographic Operation (Signature Algorithms)			
RSA (2048-bit or greater)	OpenSSL	FIPS PUB 186-4, Section 4	A1292
ECDSA (P-256, P-384, P-521)	OpenSSL	FIPS PUB 186-4, Section 5	A1292
FCS_COP.1/UDE Cryptographic Operation (Encryption/Decryption)			
AES-GCM, AES-CBC, AES-CTR (128, 256 bits)	OpenSSL	GCM as defined in NIST SP 800-38D CBC as defined in NIST SP 800-38A CTR as defined in NIST SP 800-38A	A1292
FCS_RBG_EXT.1 Random Bit Generation			
CTR_DRBG(AES)	OpenSSL	NIST SP 800-90A NIST SP 800-57	A1292
CTR_DRBG(AES)	Vmkdrbg		A2118

The TOE generates RSA and ECDSA keys in support of TLS. The TSF generates ECC keys using P-256, P-384, and P-521. These keys are generated in support of the ECDHE key establishment schemes that are used for TLS communications. To ensure sufficient key strength, the TOE also implements DRBG functionality for key generation, using the AES-CTR_DRBG.

The TOE's hash function is used in support of digital signature and keyed-hash message authentication (HMAC) functions.

The TOE's HMAC functions support the following key lengths, hash functions, block sizes, and output MAC lengths:

- HMAC-SHA-1: key and digest length 160 bits, SHA-1 hash on block size of 512 bits
- HMAC-SHA-256: key and digest length 256 bits, SHA-256 hash on block size of 512 bits
- HMAC-SHA-384: key and digest length 384 bits, SHA-384 hash on block size of 1024 bits

- HMAC-SHA-512: key and digest length 512 bits, SHA-512 hash on block size of 1024 bits

The TOE relies on a combination of physical hardware and software entropy sources. The proprietary Entropy Analysis Report (EAR) describes how the TSF extracts random data from hardware- and software-based sources to ensure that an amount of entropy that is at least equal to the strength of the generated keys is present (i.e., at least 256 bits when the largest supported keys are generated) when seeding the DRBG for key generation purposes. The random numbers from the DRBG can then be obtained from the /dev/random pseudo-device, which is used to seed OpenSSL.

In addition, the TSF includes passthrough access to the TOE's physical entropy source (RDSEED) so that guest VMs can independently acquire platform entropy for their own purposes. As VMs are isolated from each other, there is no sharing of entropy between VMs because each VM accesses the entropy source independently.

The TOE uses TLS 1.2 for client and server communications. In the case where the TOE acts as a TLS server, all other TLS versions are rejected. The 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_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

The TLS server implementation supports the following TLS cipher suites:

- TLS_RSA_WITH_AES_128_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

When ECDHE cipher suites are used, the TSF uses secp256r1, secp384r1, and secp521r1 for key establishment. When RSA cipher suites are used, the TSF uses 2048 bit keys.

As part of certificate validation in the establishment of TLS connectivity, the TOE validates the reference identifier of a presented server certificate. This is done through validation of the Common Name (CN) in the subject field of the certificate or the Subject Alternative Name (SAN). The TSF expects the SAN to contain a case-insensitive DNS name or IP address. Wildcards are supported for SANs that are DNS names. Certificate pinning is not supported.

If the TOE is prompted with a session using an old SSL or TLS version, the connection is rejected if the peer cannot use a newer version. During the server key exchange, the key agreement parameters are passed. These include the public base and modulus used for establishing the communication before authentication.

The TOE implements TLS as a client for remote syslog communications and TLS as a server for remote administration using HTTPS. The TOE's implementation of HTTPS conforms to RFC 2818.

6.4 User Data Protection

ESXi supports communication between VMs through virtual networking, which the guest accesses via a virtual network interface controller (vNIC). A virtual machine has no network connections unless explicitly configured. An administrator may configure the 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 above mechanism when expressly enabled by an authorized Administrator.

The ESXi hypervisor 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.

Documentation describes the following devices as configurable for physical access by a virtual machine. Access requires both a global (host) configuration and a per-VM configuration, and is applied at configuration time. Access resolves to allow/deny only (no fine-grained controls), which is logged to the audit system.

- USB devices: a virtual machine may exchange USB packets with a host-connected USB device. No such devices are initially configured. USB devices are identified to the administrator via VendorID and ProductID. This allows the administrator to determine the devices they are granting a Guest VM access to.
  Security policy: global allow, per-VM default-deny, mutually exclusive access.
- Network adapter: a virtual machine may exchange Ethernet packets with the physical network when configured with a virtual switch that joins to the physical network. Network adapters are identified to the administrator via a physical NIC label that is referenced in the virtual switch associated with the adapter. This allows the administrator to determine the devices they are granting a Guest VM access to.
  Security policy: per-vSwitch default-deny, per-VM default-deny, non-exclusive access.

In the evaluated configuration, the TOE and its operational environment will be configured during initial setup in such a manner that no Guest VM access to PCI passthrough devices or raw device mappings to storage logical unit numbers (LUNs) will be permitted.

For those physical devices that can be configured for access by a Guest VM, audit records indicating access are generated when an administrator adds or removes physical device access through virtual machine configuration. Audit records indicating denial are generated when an attempt is made to violate mutual exclusion.

Documentation describes all other virtual devices as having no access to physical hardware. A selection of devices follows.

- Network (using a virtual switch not connected to a physical network): All network packets are routed to a destination by the virtual switch. When such packets transit the physical network, the hypervisor encapsulates the packets (for example, using VLANs).
- Storage (using a disk type other than RDM or physical CD-ROM/DVD): All storage commands are implemented by the hypervisor as I/O to VMDK files or ISO images. Additionally, access to ISO images is read-only.
- CPU + Memory: Protected as specified for FDP_HBI_EXT.1.
- Serial, Parallel: ESXi implements a complete virtual serial and parallel device, respectively.
- USB: many virtual USB devices are fully emulated with no connection to physical devices. For example, the virtual USB keyboard has no connection to a physical keyboard.

Unless noted above as configurable for direct physical access, the hypervisor denies virtual machines all access to physical devices.

Traffic traversing a virtual network is visible only to Guest VMs that are configured by an Administrator to be members of that virtual network.

Pages allocated to kernel threads are zeroed out at allocation time. How pages are zeroed out for virtual machines and userspace applications is determined by the global `Mem.MemEagerZero` advanced option, and in the case of virtual machines, also by the per-VM `sched.mem.eagerZero` option:

- When `Mem.MemEagerZero` is set to 0 (the default), pages are zeroed when they are allocated to virtual machines and userspace applications. While this prevents exposing information from virtual machines to other clients, previous content can stay present in memory for a long time if the memory is not re-used.
- For more immediate content destruction, when `Mem.MemEagerZero` is set to 1, pages are zeroed when a userspace application exits. For virtual machines, such pages are zeroed when the virtual machine powers off, when its pages are migrated, or when virtual machine memory is reclaimed. Also, for virtual machines only, one can obtain this behavior by setting the per-VM option `sched.mem.eagerZero = TRUE`. This option is set during initial configuration via the supported VIM API.

Setting `Mem.MemEagerZero` to 1 overrides any per-VM `sched.mem.eagerZero` setting.

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.

Physical disk storage is cleared (zeroed) 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).

The VMM leverages the VT-x (with Extended Page Tables) processor instructions as a hardware mechanism to enforce CPU isolation between the host and virtual machines. VMkernel employs VT-d (IOMMU) to enforce hardware isolation of physical PCI devices between host and virtual machines in any configuration where the virtual machine has direct access to physical devices. These mechanisms are always enabled and cannot be disabled.

6.5 Identification and Authentication

ESXi implements username and password-based authentication for all remote interfaces to the TOE.

Account lockout (by default 15 minutes) is supported for access through all remote channels using username/password.

For the authentication implemented in the evaluated configuration, the following password policy is implemented by default for the evaluated TOE configuration:

- Password length of at least 7 characters.
- Must contain at least one character from three of the following classes:
  - Uppercase letter
  - Lowercase letter
  - Number
  - Special character, such as ampersand (&), hash key (#), and percent sign (%).

If connecting remotely, a username and password is transmitted securely over TLS. If the credentials match valid entries and have sufficient permissions, the user is successfully logged in to the TOE and given Administrator privileges. The user also receives a session token (via HTTP cookie). The token may be used in lieu of re-authenticating during subsequent connections provided the login session has not expired.

X.509 certificates are validated for TLS connection establishment and validation of trusted updates.

During execution of X.509 certificate validation, validation occurs in compliance with RFC 5280 as well as to the rules defined in Section 5.2.4.5. The TOE maintains a repository for certificates and selects the appropriate certificate based upon the criteria identified within Section 5.2.4.5 (e.g. extendedKeyUsage field). This information is checked for validity as well as against CRL for revocation before determining if the certificate should be accepted and the session established.

Validation occurs on two paths.

- TLS client: Validation is performed by the TLS client (remote syslog), which checks for trusted roots in the system's configured TLS trust store. Validation is an optional configuration for remote syslog. Most validations are mandatory; CRLs and some checks on the root certificate are configurable. 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. Terminating the channel on validation failure is not configurable.

- Trusted update: Validation is performed by the update mechanism immediately prior to applying the update. Validation is mandatory when the system is in SecureBoot mode. If validation fails, the update is not applied and an error is returned. No connection is established during the validity check of the certificate.

6.6 Security Management

The TOE has one Administrator role that can perform all administrative functions. The TOE has several management interfaces; the management functions supported by the TOE and the management interfaces on which those functions can be performed are shown in the table below. The ESXi Host Client is a GUI front-end for the VIM API; all functions executed on this interface call underlying VIM API functions that can also be invoked directly.

Table 11: TSF Management Functions by Interface

Function	VIM API	ESXi Host Client	ESXCLI
Ability to update the Virtualization System	X	X	X
Ability to configure Administrator password policy as defined in FIA_PMG_EXT.1	X	X	
Ability to create, configure and delete VMs	X	X	
Ability to set default initial VM configurations	X	X	
Ability to configure virtual networks including VM	X	X	
Ability to configure and manage the audit system and audit data	X	X	X
Ability to configure VM access to physical devices	X	X	
Ability to configure inter-VM data sharing	X	X	
Ability to enable/disable VM access to Hypercall functions			
Ability to configure removable media policy	X	X	
Ability to configure the cryptographic functionality	X	X	X
Ability to change default authorization factors	X	X	X

Ability to enable/disable screen lock			
Ability to configure screen lock inactivity timeout			
Ability to configure remote connection inactivity timeout	X	X	
Ability to configure lockout policy for unsuccessful authentication attempts through limiting number of attempts during a time period	X	X	
Ability to configure name/address of directory server to bind with			
Ability to configure name/address of audit/logging server to which to send audit/logging records	X	X	X
Ability to configure name/address of network time server	X	X	
Ability to configure banner	X	X	
Ability to connect/disconnect removable devices to/from a VM	X	X	
Ability to start a VM	X	X	
Ability to stop/halt a VM	X	X	
Ability to checkpoint a VM	X	X	
Ability to suspend a VM	X	X	
Ability to resume a VM	X	X	

The TSF enforces separation of data sharing between Guest VMs and between management and operational networks. By default, data sharing between Guest VMs is prohibited per FDP_VMS_EXT.1, but this can be administratively enabled through the use of virtual networking. To ensure that the management and operational networks of the TOE remain separated, an administrator can configure separate networks for management and operation. This can either be done 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. Management traffic over the management network, whether physical or logical, is always handled through trusted channels that use TLS or TLS/HTTPS.

6.7 Protection of the TSF

Software running in a VM is not able to degrade or disrupt the functioning of other VMs, the VMM, or the Platform. 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. VMware ESXi 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.

While ESXi does contain 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.

A proprietary annex to this ST provides information on the hypercall interfaces, including all applicable functions, parameters, legal values, configuration settings, and how the functions are called. Administrators of the TOE have the ability to disable these hypercall functions.

The following virtual devices are within the scope of evaluation:

- Network controllers
  - E1000e
  - VMXNET 3
- 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)

A proprietary annex to the ST provides information on virtual device interfaces, including the response to illegal values.

Most devices are exposed as PCI devices where presence of appropriate PCI identifying information determines presence of a device. Some devices also have IO ports, either well known or relative to a base.

Parameters passed from Guest VMs to virtual device interfaces are thoroughly validated and all illegal values (as specified in the TSS) are rejected. Additionally, parameters passed from Guest VMs to virtual device interfaces are not able to degrade or disrupt the functioning of other VMs, the VMM, or the Platform.

The ESXi VMM uses VT-x to reduce the use of binary translation. It also uses EPT to eliminate the need for shadow page tables.

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. These mechanisms assist in prevention of unintended machine code execution within the environment.

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).

Audit records are generated when a virtual machine is associated with a particular media, and when the association is removed.

The TOE has a method for Administrators to securely update the TOE software via a supported management interface. It should be noted that only an Administrator can perform this action.

Candidate updates are obtained from VMware’s official website, 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. Certificates used by the update process are shipped as part of the TOE software and stored in `/usr/share/certs`. Certificates are updated only as a result of a validated software update. 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.

6.8 TOE Access

An authorized administrator can define and modify a banner that will be displayed prior to allowing a user to log in.

6.9 Trusted Path/Channels

The TOE offers services over multiple interfaces. A remote administrator may access the TOE using the VIM API, the ESXi Host Client (Web UI), or ESXCLI. Each of these interfaces use TLS/HTTPS. Therefore, all remote administrative sessions with the TOE use a trusted path; the TOE does not accept administrative actions over any remote path other than those listed here.

The HTTPS channel serves some static content (including the ESXi Host Client and some API metadata for API versioning) prior to accepting authentication credentials. All administrative actions require valid authentication credentials prior to executing the operation.

The TOE makes limited use of external services. In the evaluated configuration, the only remote service is syslog. The TOE may be configured to connect to a remote syslog server over TLS.

When using the VIM API, 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. The ESXCLI interface is not used to manipulate VMs so the notion of uniquely identifying VMs is not applicable to that interface.

When using the ESXi Host Client, 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.

7 Protection Profile Claims

This ST is conformant to the Protection Profile for Virtualization, Version 1.1, 14 June 2021 (Virtualization PP), PP-Module for Server Virtualization Systems, Version 1.1, 14 June 2021 (SV Module), and Functional Package for Transport Layer Security (TLS) Version 1.1, 01 March 2019, along with all applicable errata and interpretations from the certificate issuing scheme.

All claimed SFRs are defined in the Virtualization PP, SV Module, and TLS Package. All mandatory SFRs are claimed. Optional and objective SFRs are claimed at the TOE developer's discretion. Selection-based SFR claims are consistent with the selections made in the mandatory SFRs that prompt their inclusion.

8 Rationale

This Security Target includes by reference the Base Virtualization PP's Security Problem Definition, Security Objectives, and Security Assurance Requirements. The Security Target does not add, remove, or modify any of these items. Security Functional Requirements have been reproduced with the Protection Profile operations completed. All selections, assignments, and refinements made on the claimed Security Functional Requirements have been performed in a manner that is consistent with what is permitted by the Virtualization PP and SV Module. The proper set of selection-based requirements have been claimed based on the selections made in the mandatory requirements. Consequently, the claims made by this Security Target are sufficient to address the TOE's security problem. Rationale for the sufficiency of the TOE Summary Specification is provided below.

8.1 TOE Summary Specification Rationale

This section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions meet the TOE security requirements. Each description includes rationale indicating which requirements the corresponding security functions satisfy. The combined security functions work together to satisfy all of the security requirements. The security functions described in Section 6 are necessary for the TSF to enforce the required security functionality. The security functions are mapped to security requirements through the functional class of the requirements. Specifically, this mapping is as follows:

- Security Audit: all requirements that begin with 'FAU'
- Cryptographic Support: all requirements that begin with 'FCS'
- User Data Protection: all requirements that begin with 'FDP'
- Identification and Authentication: all requirements that begin with 'FIA'
- Security Management: all requirements that begin with 'FMT'
- Protection of the TSF: all requirements that begin with 'FPT'
- TOE Access: all requirements that begin with 'FTA'
- Trusted Path/Channels: all requirements that begin with 'FTP'

9 Key Lifecycle

The following table lists the keys used by the TSF along with their method of generation, their use, what mechanism (if any) exists to output them, where they are stored, and how they are erased when no longer needed.

Table 12: Key Usage

Key	Key Type	Use	Generation/ Input	Output	Storage	Zeroization/Cleared
RSA Private Key	2048 - 3072 bits	Signature generation, decryption, key transport (TLS) (user space)	Internally generated using openssl	Output via API in plaintext	Non-volatile (state.tgz)	Only regeneration of key will replace the current private key. [1]
RSA Public Key	2048 - 3072 bits	Signature verification, encryption, Key transport (TLS) (user space)	Internally generated using openssl	Output via API in plaintext (in the form of cert)	Non-volatile (state.tgz)	Only regeneration of cert will replace the current public key. [1]
RSA Private Key	2048 - 3072 bits	Signature generation, decryption, key transport (TLS) (user space)	Internally generated using openssl	Never	Volatile	[2] Cleared after use by overwrite with zeroes.
RSA Public Key	2048 - 3072 bits	Signature verification, encryption, Key transport (TLS) (user space)				
AES Key	128, 256 bits	Encryption, Decryption (user space)				
AES GCM Key	128, 256 bits	Encryption, Decryption, MAC Generation and Verification (user space)				
HMAC key	160, 256, 384, 512	Message Authentication				

Key	Key Type	Use	Generation/ Input	Output	Storage	Zeroization/Cleared
	bits	(user space)				
CTR_DRBG CSPs	V (128 bits); Key (128, 190, 256 bits); Entropy Input	Random Number Generation (user space)				
CTR_DRBG CSPs	AES Key (256 bits)	Random Number Generation (kernel space)	Internally generated using vmkdrbg	Never	Volatile	DRBG output is cleared after use with overwrite, DRBG state data (seed) is cleared when volatile memory is unpowered

[1] Key destruction for the non-volatile RSA key state is performed by overwriting the old state.tgz content with zeroes when it is replaced.

[2] Key destruction is performed automatically by API calls for keys when the key is released by an application using OpenSSL. Destruction is by zeroization (overwriting with zeroes). An unplanned application exit may result in residual keys remaining in memory. When these residual keys are cleared is determined by how userspace memory zeroization is configured. See Section 6.4, "User Data Protection", for more information.