Oracle Linux 7.6 UEK 5 KVM & Virtualization Manager 4.3

Security Target

Version 2.3

March 2023

Document prepared by

Lightship Security

www.lightshipsec.com

Document History

Version	Date	Description
1.0	7 July 2021	Addressed evaluator ORs
1.1	7 Sep 2021	Remove FIA_X509
1.2	12 Oct 2021	Added FIA_X509 and FCS_TLSC
1.3	28 Oct 2021	Addressed evaluator ORs
1.4	21 Nov 2021	Reverted back to SSHC for log offloading
1.5	26 Jan 2022	Addressed CB ORs
1.6	2 March 2022	Addressed CB ORs
1.7	13 April 2022	Address evaluator ORs
1.8	10 Aug 2022	Address evaluator ORs
1.9	14 Sept 2022	Addressed evaluator ORs
2.0	22 Dec 2022	Updated TOE guidance references
2.1	31 Jan 2023	Addressed evaluator OR.
2.2	13 Feb 2023	Addressed ORs.
2.3	3 March 2023	Updated TOE guidance versions.

Table of Contents

List of Tables

1 Introduction

1.1 Overview

1 This Security Target (ST) defines the Oracle Linux 7.6 UEK 5 KVM & Virtualization Manager 4.3 Target of Evaluation (TOE) for the purposes of Common Criteria (CC) evaluation.

2 Oracle Linux Virtualization Manager (OLVM) is a server virtualization management platform that can manage an Oracle Linux Kernel-based Virtual Machine (KVM) environment. The KVM hypervisor is available on Oracle Linux 7.6 with the Unbreakable Enterprise Kernel Release 5.

1.2 Identification

Table 1: Evaluation identifiers

Target of Evaluation	Oracle Linux 7.6 UEK 5 KVM & Virtualization Manager 4.3.10.4-1.0.21
Security Target	Oracle Linux 7.6 UEK 5 KVM & Virtualization Manager 4.3 Security Target, v2.3

1.3 Conformance Claims

3 This ST supports the following conformance claims:

a) CC version 3.1 revision 4

   i) CC Part 2 extended

   ii) CC Part 3 extended

b) NIAP Protection Profile for Virtualization v1.0 (Base PP)

c) NIAP Extended Package for Server Virtualization v1.0 (SV_EP)

d) NIAP Extended Package for Secure Shell (SSH) v1.0 (SSH_EP)

e) NIAP Technical Decisions per Table 2

Table 2: NIAP Technical Decisions

TD #	Name
0617	TLSC wildcard testing
0598	Expanded AES Modes in FCS_COP for App PP
0568	SFR Rationale
0567	Security Objectives Rationale, SFR Rationale, and Implicitly Satisfied SFRs
0526	Updates to Certificate Revocation (FIA_X509_EXT.1)
0446	Missing selections for SSH

TD #	Name
0443	FPT_VDP_EXT.1 Clarification for Assurance Activity
0432	Corrections to FIA_AFL_EXT.1
0431	Modification to Cipher Suites for TLS
0420	Conflict in FCS_SSHC_EXT.1.1 and FCS_SSHS_EXT.1.1
0363	Access Banner and applicability to programmatic interfaces
0360	AD Server configuration in FMT_MOF_EXT.1
0332	Support for RSA SHA2 host keys
0331	SSH Rekey Testing
0264	Clarification of Auditable Events for FPT_RDM_EXT.1
0250	Hypercall Controls – FPT_HCL_EXT.1 Clarification
0249	Applicability of FTP_ITC_EXT.1
0240	FCS_COP.1.1(1) Platform provided crypto for encryption/decryption
0230	ALC Assurance Activities for Server Virtualization and Base Virtualization PPs
0206	Testing for Non-Existence of Disconnected Virtual Devices
0139	Clarification of testing for FDP_RIP_EXT.2

1.4 Terminology

Table 3: Terminology

Term	Definition
Common Criteria (CC)	Common Criteria for Information Technology Security Evaluation (International Standard ISO/IEC 15408).
Common Criteria Testing Laboratory	Within the context of the Common Criteria Evaluation and Validation Scheme (CCEVS), an IT security evaluation facility, accredited by the National Voluntary Laboratory Accreditation Program (NVLAP) and approved by the NIAP Validation Body to conduct Common Criteria-based evaluations.
Common Evaluation Methodology (CEM)	Common Evaluation Methodology for Information Technology Security Evaluation.
Extended Package (EP)	An implementation-independent set of security requirements for a specific subset of products described by a PP.

Term	Definition
Protection Profile (PP)	An implementation-independent set of security requirements for a category of products.
Security Assurance Requirement (SAR)	A requirement for how the TOE's proper implementation of the SFRs is verified by an evaluator.
Security Functional Requirement (SFR)	A requirement for security enforcement by the TOE.
Security Target (ST)	A set of implementation-dependent security requirements for a specific product.
Target of Evaluation (TOE)	The product under evaluation.
TOE Security Functionality (TSF)	The security functionality of the product under evaluation.
TOE Summary Specification (TSS)	A description of how a TOE satisfies the SFR's in an ST.
Administrator	Administrators perform management activities on the VS. These management functions do not include administration of software running within Guest VMs, such as the Guest OS. Administrators need not be human as in the case of embedded or headless VMs. Administrators are often nothing more than software entities that operate within the VM.
Auditor	Auditors are responsible for managing the audit capabilities of the TOE. An Auditor may also be an Administrator. It is not a requirement that the TOE be capable of supporting an Auditor role that is separate from that of an Administrator.
Domain	A Domain or Information Domain is a policy construct that groups together execution environments and networks by sensitivity of information and access control policy. For example, classification levels represent information domains. Within classification levels, there might be other domains representing communities of interest or coalitions. In the context of a VS, information domains are generally implemented as collections of VMs connected by virtual networks. The VS itself can be considered an Information Domain, as can its Management Subsystem.
Guest Network	See Operational Network.
Guest Operating System (OS)	An operating system that runs within a Guest VM.
Guest VM	A Guest VM is a VM that contains a virtual environment for the execution of an independent computing system. Virtual environments execute mission workloads and implement

Term	Definition
	customer-specific client or server functionality in Guest VMs, such as a web server or desktop productivity applications.
Helper VM	A Helper VM is a VM that performs services on behalf of one or more Guest VMs, but does not qualify as a Service VM—and therefore is not part of the VMM. Helper VMs implement functions or services that are particular to the workloads of Guest VMs. For example, a VM that provides a virus scanning service for a Guest VM would be considered a Helper VM. For the purposes of this document, Helper VMs are considered a type of Guest VM, and are therefore subject to all the same requirements, unless specifically stated otherwise.
Host Operating System (OS)	An operating system onto which a VS is installed. Relative to the VS, the Host OS is part of the Platform.
Hypervisor	The Hypervisor is part of the VMM. It is the software executive of the physical platform of a VS. A Hypervisor's primary function is to mediate access to all CPU and memory resources, but it is also responsible for either the direct management or the delegation of the management of all other hardware devices on the hardware platform.
Hypercall	An API function that allows VM-aware software running within a VM to invoke VMM functionality.
Information Domain	See Domain.
Introspection	A capability that allows a specially designated and privileged domain to have visibility into another domain for purposes of anomaly detection or monitoring.
Libvirt	Libvirt is collection of open source software to manage virtual machines and other virtualization functionality, such as storage and network interface management. These software pieces include an API library, a daemon (libvirtd), and a command line utility (virsh).
Management Network	A network, which may have both physical and virtualized components, used to manage and administer a VS. Management networks include networks used by VS Administrators to communicate with management components of the VS, and networks used by the VS for communications between VS components. For purposes of this document, networks that connect physical hosts for purposes of VM transfer or coordinate, and backend storage networks are considered management networks.
Management Subsystem	Components of the VS that allow VS Administrators to configure and manage the VMM, as well as configure Guest VMs. VMM management functions include VM configuration,

Term	Definition
	virtualized network configuration, and allocation of physical resources.
Operational Network	An Operational Network is a network, which may have both physical and virtualized components, used to connect Guest VMs to each other and potentially to other entities outside of the VS. Operational Networks support mission workloads and customer-specific client or server functionality. Also called a "Guest Network."
Paravirtualized Device	Paravirtualization provides a fast and efficient means of communication for guests to use devices on the host machine. KVM provides paravirtualized devices to virtual machines using the virtio API as a layer between the hypervisor and guest. All virtio devices have two parts: the host device and the guest driver.
Physical Platform	The hardware environment on which a VS executes. Physical platform resources include processors, memory, devices, and associated firmware.
Platform	The hardware, firmware, and software environment into which a VS is installed and executes.
Service VM	A Service VM is a VM whose purpose is to support the Hypervisor in providing the resources or services necessary to support Guest VMs. Service VMs may implement some portion of Hypervisor functionality, but also may contain important system functionality that is not necessary for Hypervisor operation. As with any VM, Service VMs necessarily execute without full Hypervisor privileges—only the privileges required to perform its designed functionality. Examples of Service VMs include device driver VMs that manage access to a physical devices, and name-service VMs that help establish communication paths between VMs.
System Security Policy (SSP)	The overall policy enforced by the VS defining constraints on the behaviour of VMs and users.
User	Users operate Guest VMs and are subject to configuration policies applied to the VS by Administrators. Users need not be human as in the case of embedded or headless VMs, users are often nothing more than software entities that operate within the VM.
Virtual Machine (VM)	A Virtual Machine is a virtualized hardware environment in which an operating system may execute.
Virtual Machine Manager (VMM)	A VMM is a collection of software components responsible for enabling VMs to function as expected by the software executing within them. Generally, the VMM consists of a Hypervisor, Service VMs, and other components of the VS, such as virtual devices, binary translation systems, and

Term	Definition
	physical device drivers. It manages concurrent execution of all VMs and virtualizes platform resources as needed.
Virtualization System (VS)	A software product that enables multiple independent computing systems to execute on the same physical hardware platform without interference from one other. For the purposes of this document, the VS consists of a Virtual Machine Manager (VMM), Virtual Machine (VM) abstractions, a management subsystem, and other components.
KVM	Kernel-based Virtual Machine
NIAP	National Information Assurance Partnership
OLVM	Oracle Linux Virtualization Manager
SME	Subject Matter Expert
TRRT	Technical Rapid Response Team
UEK	Unbreakable Enterprise Kernel
virsh	Command line utility for libvirt

2 TOE Description

2.1 Type

4 The TOE is a hypervisor and virtualization management platform.

2.2 Usage

5 The TOE is bundled with Oracle Linux and is used to provide server virtualization capabilities to users. The TOE would typically be deployed onto enterprise grade hardware housed in data centers and users interact with the TOE via secure remote communication channels.

6 The TOE is used to provide virtualized instances of services traditionally executed on separate hardware platforms, such as web servers, file servers, and mail servers.

7 OLVM offers a web-based User Interface (UI) which can be used to manage Oracle Linux KVM and virtualized infrastructure. KVM may also be managed via CLI over SSH.

2.2.1 Secure Communications

8 The secure communication protocols within the scope of evaluation are depicted in Figure 1, with the TOE enclosed in green.

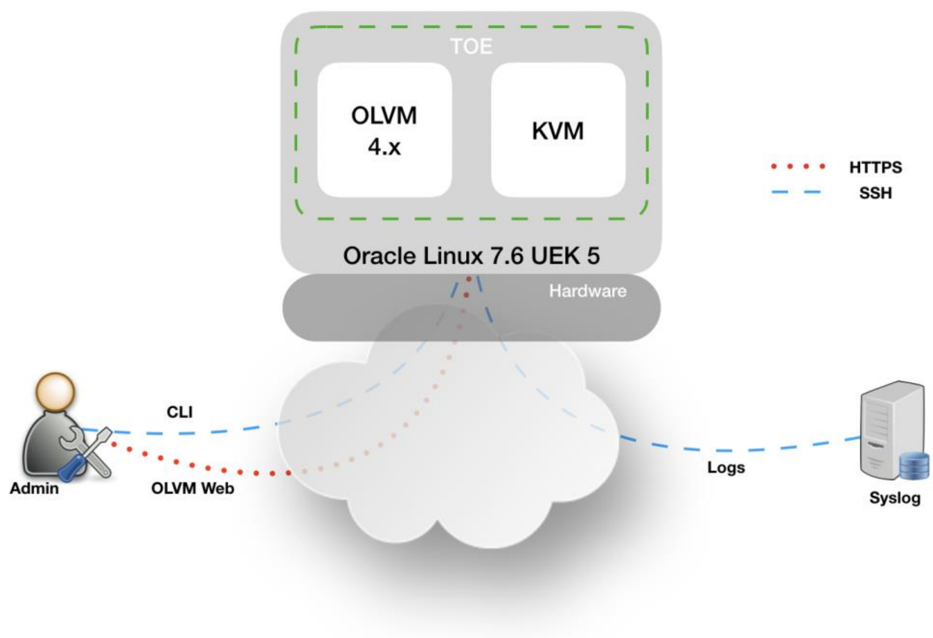

Figure 1: Secure Communication Channels

2.3 Logical Scope / Security Functions

9 The TOE provides the following security functions:

a) **VM Hardware-based Isolation.** The TOE supports isolation mechanisms to constrain a Guest VM's direct access to physical devices.

b) **VM Resource Control.** The TOE enables control of Guest VM access to physical platform resources.

c) **VM Residual Information Clearing.** The TOE ensures that any previous information content in memory or physical disk storage is cleared prior to allocation to a Guest VM.

d) **VM Networking & Separation.** The TOE enables control of mechanisms used to transfer data between Guest VMs, including control of virtual networking components.

e) **VM User Interface.** The TOE indicates to users which VM if any has current input focus and supports unique identification of VMs.

f) **VS Integrity.** The TOE maintains integrity of the virtualization system critical components via measured boot and trusted software updates.

g) **VS Self Protection.** The TOE implements self-protection mechanisms including execution environment mitigations, hardware-assists, hypercall controls, isolation from VMs and controls for removable media.

h) **Protected Communications.** The TOE protects the integrity and confidentiality of communications as noted in section 2.2.1 above.

i) **Secure Administration.** The TOE enables secure management of its security functions, including:

  i) Administrator authentication with passwords

  ii) Configurable password policies

  iii) Role Based Access Control

  iv) Access banners

  v) Management of critical security functions and data

j) **System Monitoring.** The TOE generates audit records and stores them locally and is capable of sending records to a remote audit server. The TOE protects stored audit records and enables their review.

k) **Cryptographic Operations.** The TOE implements a cryptographic module. Relevant Cryptographic Algorithm Validation Program (CAVP) certificates are shown in Table 4.

2.4 Physical Scope

10 The evaluated configuration is the KVM & Virtualization Manager 4.3.10.4-1.0.21 running on Oracle Linux 7.6 UEK 5, tested on the Oracle X7-2 hardware platform with the Intel Xeon Silver 4114 CPU.

2.4.1 Software

11 The TOE is part of the following software:

a) Oracle Linux 7.6 UEK 5
Note: KVM and Virtualization Manager are installed as part of Oracle Linux 7.6 UEK 5

12 The TOE is downloaded by users from the Oracle Software Delivery Cloud at https://edelivery.oracle.com/

2.4.2 Guidance Documents

13 The TOE includes the following guidance documents:

a) [CC Guide] Oracle Linux 7.6 UEK 5 KVM & Virtualization Manager 4.3 Common Criteria Guide, v1.6

b) [OL7-CC] Oracle Linux v7.6 Common Criteria Guidance Document, v0.9

c) Oracle Linux Virtualization Manager: Getting Started Guide https://www.oracle.com/a/ocom/docs/olvm43/olvm-43-gettingstarted.pdf

d) Oracle Linux Virtualization Manager Administration Guide https://www.oracle.com/a/ocom/docs/olvm43/olvm-43-administration.pdf

e) oVirt Administration Guide (upstream OLVM documentation) https://www.ovirt.org/documentation/administration_guide/

f) oVirt Upgrade Guide

https://www.ovirt.org/documentation/upgrade_guide/

g) oVirt Virtual Machine Management Guide

https://www.ovirt.org/documentation/virtual_machine_management_guide/

h) oVirt Introduction to the VM Portal

https://www.ovirt.org/documentation/introduction_to_the_vm_portal/

i) Oracle Linux KVM User's Guide https://docs.oracle.com/en/operating-systems/oracle-linux/kvm-user/

2.4.3 Non-TOE Components

14 The TOE operates with the following components in the environment:

a) **Audit Server.** The TOE is capable of sending audit events to a Syslog server.

b) **Hardware Platforms.** The TOE was tested on the Oracle X7-2 (Intel Xeon Silver 4114) hardware platform.

Table 4: CAVP

Algorithm	Standard	Library	CAVP
AES	AES-CBC (as defined in FIPS PUN 197 and NIST SP 800-38A) AES-CTR (as defined in NIST SP 800-38A)	Oracle Linux 7.6 OpenSSL with AESNI, SHA1 AVX, SHA2 ASM	A1400
RSA	FIPS PUB 186-4	Oracle Linux 7.6 OpenSSL with AESNI, SHA1 AVX, SHA2 ASM	A1400
DH	N/A	Oracle Linux 7.6 OpenSSL with AESNI, SHA1 AVX, SHA2 ASM	A1400
KAS/CVL FFC	NIST Special Publication 800-56A	Oracle Linux 7.6 OpenSSL with AESNI, SHA1 AVX, SHA2 ASM	A1400

Algorithm	Standard	Library	CAVP
HMAC	FIPS PUB 198-1 and FIPS PUB 180-4	Oracle Linux 7.6 OpenSSL with AESNI, SHA1 AVX, SHA2 ASM Oracle Linux 7.6 OpenSSL VPAES and SHA1 SSSE3 Oracle Linux 7.6 OpenSSL with AES and SHA1 assembler	A1400 A1401 A1402
SHS	FIPS PUB 180-4	Oracle Linux 7.6 OpenSSL with AESNI, SHA1 AVX, SHA2 ASM Oracle Linux 7.6 OpenSSL VPAES and SHA1 SSSE3 Oracle Linux 7.6 OpenSSL with AES and SHA1 assembler	A1400 A1401 A1402
HMAC_DRBG	NIST SP 800-57	Oracle Linux 7.6 OpenSSL with AESNI, SHA1 AVX, SHA2 ASM Oracle Linux 7.6 OpenSSL VPAES and SHA1 SSSE3 Oracle Linux 7.6 OpenSSL with AES and SHA1 assembler	A1400 A1401 A1402
CVL SSH v2	KDF 800-135	Oracle Linux 7.6 OpenSSL with AESNI, SHA1 AVX, SHA2 ASM	A1400
CVL TLS v1.2	KDF 800-135	Oracle Linux 7.6 OpenSSL with AESNI, SHA1 AVX, SHA2 ASM	A1400

2.5 Excluded Functionality

15 This CC evaluation only covers the functionality identified in section 2.3 when Oracle Linux 7.6 UEK 5 KVM & Virtualization Manager 4.3.10.4-1.0.21 is configured in accordance with [CC Guide].

16 The TOE contains a REST API. Access to this interface requires valid administrator credentials. The REST API is not used in the evaluated configuration.

3 Security Problem Definition

3.1 Threats

Table 5: Threats

Identifier	Description
T.DATA_LEAKAGE	It is a fundamental property of VMs that the domains encapsulated by different VMs remain separate unless data sharing is permitted by policy. For this reason, all Virtualization Systems shall support a policy that prohibits information transfer between VMs. It shall be possible to configure VMs such that data cannot be moved between domains from VM to VM, or through virtual or physical network components under the control of the VS. When VMs are configured as such, it shall not be possible for data to leak between domains, neither by the express efforts of software or users of a VM, nor because of vulnerabilities or errors in the implementation of the VMM or other VS components. If it is possible for data to leak between domains when prohibited by policy, then an adversary on one domain or network can obtain data from another domain. Such cross-domain data leakage can, for example, cause classified information, corporate proprietary information, or personally identifiable information to be made accessible to unauthorized entities.
T.UNAUTHORIZED_ UPDATE	It is common for attackers to target outdated versions of software containing known flaws. This means it is extremely important to update Virtualization System software as soon as possible when updates are available. But the source of the updates and the updates themselves must be trusted. If an attacker can write their own update containing malicious code they can take control of the VS.
T.UNAUTHORIZED_ MODIFICATION	System integrity is a core security objective for Virtualization Systems. To achieve system integrity, the integrity of each VMM component must be established and maintained. Malware running on the platform must not be able to undetectably modify Virtualization System components while the system is running or at rest. Likewise, malicious code running within a virtual machine must not be able to modify Virtualization System components.
T.USER_ERROR	If a Virtualization System is capable of simultaneously displaying VMs of different domains to the same user at the same time, there is always the chance that the user will become confused and unintentionally leak information between domains. This is especially likely if VMs belonging to different domains are indistinguishable. Malicious code may also attempt to interfere with the user's ability to distinguish between domains. The VS must take measures to minimize the likelihood of such confusion.

Identifier	Description
T.3P_SOFTWARE	In some VS implementations, critical functions are by necessity performed by software not produced by the virtualization vendor. Such software may include Host Operating Systems and physical device drivers. Vulnerabilities in this software can be exploited by an adversary and result in VMM compromise. Where possible, the VS should mitigate the results of potential vulnerabilities or malicious content in third-party code.
T.VMM_ COMPROMISE	The Virtualization System is designed to provide the appearance of exclusivity to the VMs and is designed to separate or isolate their functions except where specifically shared. Failure of security mechanisms could lead to unauthorized intrusion into or modification of the VMM, or bypass of the VMM altogether. This must be prevented to avoid compromising the Virtualization System.
T.PLATFORM_ COMPROMISE	The VS must be capable of protecting the platform from threats that originate within VMs and operational networks connected to the VS. The hosting of untrusted—even malicious—domains by the VS cannot be permitted to compromise the security and integrity of the platform on which the VS executes. If an attacker can access the underlying platform in a manner not controlled by the VMM, the attacker might be able to modify system firmware or software—compromising both the Virtualization System and the underlying platform.
T.UNAUTHORIZED_ ACCESS	Functions performed by the management layer include VM configuration, virtualized network configuration, allocation of physical resources, and reporting. Only certain authorized system users (administrators) are allowed to exercise management functions. Virtualization Systems are often managed remotely over communication networks. Members of these networks can be both geographically and logically separated from each other, and pass through a variety of other systems which may be under the control of an adversary, and offer the opportunity for communications to be compromised. An adversary with access to an open management network could inject commands into the management infrastructure. This would provide an adversary with administrator privilege on the platform, and administrative control over the VMs and virtual network connections. The adversary could also gain access to the management network by hijacking the management network channel.
T.WEAK_CRYPTO	To the extent that VMs appear isolated within the Virtualization System, a threat of weak cryptography may arise if the VMM does not provide good entropy to support security-related features that depend on entropy to implement cryptographic algorithms. For example, a random number generator keeps an estimate of the number of bits of noise in the entropy pool. From this entropy pool random numbers are created. Good random numbers are essential to implementing strong cryptography. Cryptography implemented using poor random numbers can be defeated by a sophisticated adversary.

Identifier	Description
T.UNPATCHED_ SOFTWARE	Vulnerabilities in outdated or unpatched software can be exploited by adversaries to compromise the Virtualization System or platform.
T.MISCONFIGURATION	The Virtualization System may be misconfigured, which could impact its functioning and security. This misconfiguration could be due to an administrative error or the use of faulty configuration data.
T.DENIAL_OF_ SERVICE	A VM may block others from system resources (e.g. system memory, persistent storage, and processing time) via a resource exhaustion attack.

3.2 Assumptions

Table 6: Assumptions

Identifier	Description
A.PLATFORM_ INTEGRITY	The platform has not been compromised prior to installation of the Virtualization System.
A.PHYSICAL	Physical security commensurate with the value of the TOE and the data it contains is assumed to be provided by the environment.
A.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance.
A.COVERT_ CHANNELS	If the TOE has covert storage or timing channels, then for all VMs executing on that TOE, it is assumed that relative to the IT assets to which they have access, those VMs will have assurance sufficient to outweigh the risk that they will violate the security policy of the TOE by using those covert channels
A.NON_MALICIOUS_ USER	The user of the VS is not wilfully negligent or hostile and uses the VS in compliance with the applied enterprise security policy and guidance. At the same time, malicious applications could act as the user, so requirements which confine malicious applications are still in scope.

3.3 Organizational Security Policies

18 None defined.

4 Security Objectives

4.1 Security Objectives for the TOE

Table 7: Security Objectives for the TOE

Identifier	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 the use case, a VM may require 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 sharing of particular resources 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. Devices, whether virtual or physical, are resources requiring access control. The VMM must enforce access control in accordance to 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.

Identifier	Description
O.VMM_INTEGRITY	Integrity is a core security objective for Virtualization Systems. To achieve system integrity, the integrity of each VMM component must be established and maintained. This objective concerns only the integrity of the Virtualization System—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 Virtualization System. 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 is capable of undermining 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.

Identifier	Description
O.MANAGEMENT_ ACCESS	VMM management functions include VM configuration, virtualized network configuration, allocation of physical resources, and reporting. Only certain authorized system users (administrators) are allowed to 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 virtualization system, 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 might be 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 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.

Identifier	Description
O.PATCHED_ SOFTWARE	The Virtualization System 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	The purpose of audit is to capture and protect data about what happens on a system so that it can later be examined to determine what has happened in the past.
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.
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 8: Security Objectives for the Operational Environment

Identifier	Description
OE.CONFIG	TOE administrators will configure the Virtualization System 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.

Identifier	Description
OE.COVERT_ CHANNELS	If the TOE has covert storage or timing channels, then for all VMs executing on that TOE, it is assumed that those VMs will have sufficient assurance relative to the IT assets to which they have access, to outweigh the risk that they will violate the security policy of the TOE by using those covert channels.
OE.NON_MALICIOUS _USER	Users are trusted to be not wilfully negligent or hostile and use the VS in compliance with the applied enterprise security policy and guidance.

5 Security Requirements

5.1 Conventions

20 This document uses the following font conventions to identify the operations defined by the CC:

a) **Assignment.** Indicated with italicized text.

b) **Refinement.** Indicated with bold text and strikethroughs.

c) **Selection.** Indicated with underlined text.

d) **Assignment within a Selection:** Indicated with italicized and underlined text.

e) **Iteration.** Indicated by adding a string starting with "/" (e.g. "FCS_COP.1/Hash").

21 **Note:** Selection and assignment operations performed within the Security Target are denoted within brackets []. Operations shown without brackets are reproduced from the Protection Profile.

5.2 Extended Components Definition

22 The following extended components are specified by the claimed PP/Eps, which do not provide a formal definition of each component.

Table 9: Extended Components

Component	Title	Source
FMT_MOF_EXT.1	Management of Security Functions	SV_EP
FCS_SSHS_EXT.1	SSH Protocol – Server	SSH_EP
FCS_SSHC_EXT.1	SSH Protocol – Client	SSH_EP
FAU_STG_EXT.1	Off-Loading of Audit Data	Base PP
FCS_CKM_EXT.4	Cryptographic Key Destruction	Base PP
FCS_ENT_EXT.1	Entropy for Virtual Machines	Base PP
FCS_RBG_EXT.1	Cryptographic Operation (Random Bit Generation)	Base PP
FDP_HBI_EXT.1	Hardware-Based Isolation Mechanisms	Base PP
FDP_PPR_EXT.1	Physical Platform Resource Controls	Base PP
FDP_RIP_EXT.1	Residual Information in Memory	Base PP
FDP_RIP_EXT.2	Residual Information on Disk	Base PP
FDP_VMS_EXT.1	VM Separation	Base PP
FDP_VNC_EXT.1	Virtual Networking Components	Base PP

Component	Title	Source
FIA_AFL_EXT.1	Authentication Failure Handling	Base PP
FIA_PMG_EXT.1	Password Management	Base PP
FIA_UIA_EXT.1	Administrator Identification and Authentication	Base PP
FMT_MSA_EXT.1	Default Data Sharing Configuration	Base PP
FMT_SMO_EXT.1	Separation of Management and Operational Networks	Base PP
FPT_DVD_EXT.1	Non-Existence of Disconnected Virtual Devices	Base PP
FPT_EEM_EXT.1	Execution Environment Mitigations	Base PP
FPT_HAS_EXT.1	Hardware Assists	Base PP
FPT_HCL_EXT.1	Hypercall Controls	Base PP
FPT_RDM_EXT.1	Removable Devices and Media	Base PP
FPT_TUD_EXT.1	Trusted Updates to the Virtualization System	Base PP
FPT_VDP_EXT.1	Virtual Device Parameters	Base PP
FPT_VIV_EXT.1	VMM Isolation from VMs	Base PP
FTP_ITC_EXT.1	Trusted Channel Communications	Base PP
FTP_UIF_EXT.1	User Interface: I/O Focus	Base PP
FTP_UIF_EXT.2	User Interface: Identification of VM	Base PP
FCS_HTTPS_EXT.1	HTTPS Protocol	Base PP
FCS_TLSS_EXT.1	TLS Server Protocol	Base PP

5.3 Functional Requirements

Table 10: Summary of SFRs

Requirement	Title	Type	Source
FAU_GEN.1	Audit Data Generation	Mandatory	Base PP
FAU_SAR.1	Audit Review	Mandatory	Base PP
FAU_STG.1	Protected Audit Trail Storage	Mandatory	Base PP
FAU_STG_EXT.1	Off-Loading of Audit Data	Mandatory	Base PP
FCS_CKM.1	Cryptographic Key Generation	Mandatory	Base PP
FCS_CKM.2	Cryptographic Key Establishment	Mandatory	Base PP
FCS_CKM_EXT.4	Cryptographic Key Destruction	Mandatory	Base PP
FCS_COP.1(1)	Cryptographic Operation (AES Data Encryption/Decryption)	Mandatory	Base PP
FCS_COP.1(1)/SSH	Cryptographic Operation – Encryption/Decryption (Refined)	Selection	Base PP
FCS_COP.1(2)	Cryptographic Operation (Hashing)	Mandatory	Base PP
FCS_COP.1(3)	Cryptographic Operation (Signature Algorithms)	Mandatory	Base PP
FCS_COP.1(4)	Cryptographic Operation (Keyed Hash Algorithms)	Mandatory	Base PP
FCS_ENT_EXT.1	Entropy for Virtual Machines	Mandatory	Base PP
FCS_RBG_EXT.1	Cryptographic Operation (Random Bit Generation)	Mandatory	Base PP
FDP_HBI_EXT.1	Hardware-Based Isolation Mechanisms	Mandatory	Base PP
FDP_PPR_EXT.1	Physical Platform Resource Controls	Mandatory	Base PP
FDP_RIP_EXT.1	Residual Information in Memory	Mandatory	Base PP
FDP_RIP_EXT.2	Residual Information on Disk	Mandatory	Base PP
FDP_VMS_EXT.1	VM Separation	Mandatory	Base PP
FDP_VNC_EXT.1	Virtual Networking Components	Mandatory	Base PP
FIA_AFL_EXT.1/OLVM	Authentication Failure Handling	Mandatory	Base PP

Requirement	Title	Type	Source
FIA_AFL_EXT.1/SSH	Authentication Failure Handling	Mandatory	Base PP
FIA_PMG_EXT.1	Password Management	Selection	Base PP
FIA_UAU.5	Multiple Authentication Mechanisms	Mandatory	Base PP
FIA_UIA_EXT.1	Administrator Identification and Authentication	Mandatory	Base PP
FMT_MOF_EXT.1	Management of Security Functions	Mandatory	SV_EP
FMT_MSA_EXT.1	Default Data Sharing Configuration	Mandatory	Base PP
FMT_SMO_EXT.1	Separation of Management and Operational Networks	Mandatory	Base PP
FPT_DVD_EXT.1	Non-Existence of Disconnected Virtual Devices	Mandatory	Base PP
FPT_EEM_EXT.1	Execution Environment Mitigations	Mandatory	Base PP
FPT_HAS_EXT.1	Hardware Assists	Mandatory	Base PP
FPT_HCL_EXT.1	Hypercall Controls	Mandatory	Base PP
FPT_RDM_EXT.1	Removable Devices and Media	Mandatory	Base PP
FPT_TUD_EXT.1	Trusted Updates to the Virtualization System	Mandatory	Base PP
FPT_VDP_EXT.1	Virtual Device Parameters	Mandatory	Base PP
FPT_VIV_EXT.1	VMM Isolation from VMs	Mandatory	Base PP
FTA_TAB.1	TOE Access Banner	Mandatory	Base PP
FTP_ITC_EXT.1	Trusted Channel Communications	Mandatory	Base PP
FTP_TRP.1	Trusted Path	Selection	Base PP
FTP_UIF_EXT.1	User Interface: I/O Focus	Mandatory	Base PP
FTP_UIF_EXT.2	User Interface: Identification of VM	Mandatory	Base PP
FCS_HTTPS_EXT.1	HTTPS Protocol	Selection	Base PP
FCS_TLSS_EXT.1	TLS Server Protocol	Selection	Base PP
FCS_SSH_EXT.1	SSH Protocol	Selection	SSH_EP
FCS_SSHC_EXT.1	SSH Protocol – Client	Selection	SSH_EP

Requirement	Title	Type	Source
FCS_SSHS_EXT.1	SSH Protocol – Server	Selection	SSH_EP

5.3.1 Security Audit (FAU)

FAU_GEN.1 Audit Data Generation

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 audit functions;

b) All administrative actions;

c) *Specifically defined auditable events in Table 11*.

d) [no other information]

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 11*.

f) [no other information]

Table 11: Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	Startup and shutdown of audit functions.	None.
FAU_SAR.1	None.	None.
FAU_STG.1	None.	None.
FAU_STG_EXT.1	Failure of audit data capture due to lack of disk space or pre-defined limit. On failure of logging function, capture record of failure and record upon restart of logging function.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM_EXT.4	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FCS_COP.1(1)	None.	None.
FCS_COP.1(2)	None.	None.
FCS_COP.1(3)	None.	None.
FCS_COP.1(4)	None.	None.
FCS_ENT_EXT.1	None.	None.
FCS_RBG_EXT.1	Failure of the randomization process.	No additional information.
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session. Establishment/Termination of a HTTPS session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
FCS_TLSS_EXT.1	Failure to establish a TLS Session. Establishment/Termination of a TLS session.	Reason for failure. Non-TOE endpoint of connection (IP address).
FDP_HBI_EXT.1	None.	None.
FDP_PPR_EXT.1	Successful and failed VM connections to physical devices where connection is governed by configurable policy. Security policy violations.	VM and physical device identifiers. Identifier for the security policy that was violated.
FDP_RIP_EXT.1	None.	None.
FDP_RIP_EXT.2	None.	None.
FDP_VMS_EXT.1	None.	None.
FDP_VNC_EXT.1	Successful and failed attempts to connect VMs to virtual and physical networking components. Security policy violations. Administrator configuration of inter-VM communications channels between VMs.	VM and virtual or physical networking component identifiers. Identifier for the security policy that was violated.
FIA_PMG_EXT.1	None.	None.
FIA_UAU.5	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FIA_UIA_EXT.1	Administrator authentication attempts. All use of the identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g.,console, remote IP address).
FMT_MSA_EXT.1	None.	None.
FMT_SMO_EXT.1	None.	None.
FPT_DVD_EXT.1	None.	None.
FPT_EEM_EXT.1	None.	None.
FPT_HAS_EXT.1	None.	None.
FPT_HCL_EXT.1	Attempts to access disabled hypercall interfaces. Security policy violations.	Interface for which access was attempted. Identifier for the security policy that was violated.
FPT_RDM_EXT.1	Connection/disconnection of removable media or device to/from a VM. Ejection/insertion of removable media or device from/to an already connected VM.	VM Identifier, Removable media/device identifier, event description or identifier (connect/disconnect, ejection/insertion, etc.)
FPT_TUD_EXT.1	Initiation of update. Failure of signature verification.	No additional information.
FPT_VDP_EXT.1	None.	None.
FPT_VIV_EXT.1	None.	None.
FTA_TAB.1	None.	None.
FTP_ITC_EXT.1	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions.	User ID and remote source (IP Address) if feasible.
FTP_TRP.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	User ID and remote source (IP address) if feasible.

Requirement	Auditable Events	Additional Audit Record Contents
FTP_UIF_EXT.1	None.	None.
FTP_UIF_EXT.2	None.	None.

FAU_SAR.1 Audit Review

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.

FAU_STG.1 Protected Audit Trail Storage

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 modifications to the stored audit records in the audit trail.

FAU_STG_EXT.1 Off-Loading of Audit Data

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 [drop new audit data] when the local storage space for audit data is full.

5.3.2 Cryptographic Support

FCS_CKM.1 Cryptographic Key Generation

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 [2048-bit or greater] that meet the following*: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3];*
- FFC schemes using cryptographic key sizes [2048-bit or greater] that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1]].

FCS_CKM.2 Cryptographic Key Establishment

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

- RSA-based key establishment schemes that meets the following: NIST Special Publication 800-56B, “Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography”;
- Finite field-based key establishment schemes that meets the following: NIST Special Publication 800-56A, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”].

FCS_CKM_EXT.4 Cryptographic Key Destruction

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.

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

FCS_COP.1.1(1) The TSF shall perform [*encryption and decryption*] in accordance with a specified cryptographic algorithm [AES-CBC (as defined in FIPS PUB 197, and NIST SP 800-38A) mode]

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

FCS_COP.1(1)/SSH Cryptographic Operation - Encryption/Decryption (Refined)

FCS_COP.1.1(1)/SSH The SSH software shall [invoke platform-provided] encryption/decryption services for data in accordance with a specified cryptographic algorithm AES-CTR (as defined in NIST SP 800-38A) mode and cryptographic key sizes [128-bit, 256-bit].

Application Note: This SFR was changed by TD0240.

FCS_COP.1(2) Cryptographic Operation (Hashing)

FCS_COP.1.1(2) The TSF shall perform [*cryptographic hashing*] in accordance with a specified cryptographic algorithm [SHA-1, SHA-256, SHA-512]and **message digest** sizes [160, 256, 512 bits] that meet the following: [FIPS PUB 180-4, “Secure Hash Standard”].

FCS_COP.1(3) Cryptographic Operation (Signature Algorithms)

FCS_COP.1.1(3) 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]

FCS_COP.1(4) Cryptographic Operation (Keyed Hash Algorithms)

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

FCS_ENT_EXT.1 Entropy for Virtual Machines

FCS_ENT_EXT.1.1 The TSF shall provide a mechanism to make available to VMs entropy that meets FCS_RBG_EXT.1 through [virtual device interface].

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

FCS_RBG_EXT.1 Cryptographic Operation (Random Bit Generation)

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

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by anentropy source that accumulates entropy from [a software-based noise source **and** 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.

FCS_HTTPS_EXT.1 HTTPS Protocol

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

FCS_TLSS_EXT.1 TLS Server Protocol

FCS_TLSS_EXT.1.1 The TSF shall implement [TLS 1.2 (RFC 5246)] supporting the following cipher suites: [

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246].

FCS_TLSS_EXT.1.2 The TSF 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 TSF shall [perform RSA key establishment with key size [2048 bits, 3072 bits]; generate DiffieHellman parameters of size [2048 bits, 3072 bits]].

Application Note: This SFR was changed by TD0431.

FCS_SSH_EXT.1 SSH Protocol

FCS_SSH_EXT.1.1 The SSH software shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254 and [6668] as a [client, server]

FCS_SSHC_EXT.1 SSH Protocol – Client

FCS_SSHC_EXT.1.1 The SSH client shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, and [none]

Application Note: This SFR was changed by TD0420.

FCS_SSHC_EXT.1.2 The SSH client shall ensure that, as described in RFC4253, packets greater than [*262144*] bytes in an SSH transport connection are dropped.

FCS_SSHC_EXT.1.3 The SSH software shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: aes128-ctr, aes256-ctr, [aes128-cbc, aes256-cbc].

Application Note: This SFR was changed by TD0446

FCS_SSHC_EXT.1.4 The SSH client shall ensure that the SSH transport implementation uses [rsa-sha2-512] and [no other public key algorithms] as its public key algorithm(s) and rejects all other public key algorithms.

Application Note: This SFR was changed by TD0332.

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

Application Note: This SFR was changed by TD0446

FCS_SSHC_EXT.1.6 The SSH client shall ensure that [diffie-hellman-group14-sha1] and [no other methods] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHC_EXT.1.7 The SSH server shall ensure that the SSH connection be rekeyed after [no more than 2^{28} packets have been transmitted] using that key.

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

FCS_SSHS_EXT.1 SSH Protocol – Server

FCS_SSHS_EXT.1.1 The SSH server shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, and [password-based].

FCS_SSHS_EXT.1.2 The SSH server shall ensure that, as described in RFC4253, packets greater than [*262144*] bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.3 The SSH server shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [aes128-ctr, aes256-ctr, aes128-cbc, aes256-cbc].

Application Note: This SFR was changed by TD0446.

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

Application Note: This SFR was changed by TD0332.

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

FCS_SSHS_EXT.1.6 The SSH server shall ensure that [diffie-hellman-group14-sha1] and [no other methods] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHS_EXT.1.7 The SSH server shall ensure that the SSH connection be rekeyed after [no more than 2^{28} packets have been transmitted] using that key.

Application Note: The test number 1 for this SFR was changed by TD0331.

5.3.3 User Data Protection (FDP)

FDP_HBI_EXT.1 Hardware-Based Isolation Mechanisms

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

FDP_PPR_EXT.1 Physical Platform Resource Controls

FDP_PPR_EXT.1.1 The TSF shall allow an authorized administrator to control Guest VM access to the following physical platform resources: [*Network Adapter (Physical NIC), CPU, memory*].

FDP_PPR_EXT.1.2 The TSF shall explicitly deny all Guest VMs access to the following physical platform resources: [no physical platform resources].

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

FDP_RIP_EXT.1 Residual Information in Memory

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.

FDP_RIP_EXT.2 Residual Information on Disk

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

FDP_VMS_EXT.1 VM Separation

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 allow Administrators to configure these mechanisms to [enable, disable] the transfer of data between Guest VMs.

FDP_VMS_EXT.1.3 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.

FDP_VNC_EXT.1 Virtual Networking Components

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.

5.3.4 Identification and Authentication (FIA)

FIA_AFL_EXT.1/OLVM Authentication Failure Handling

FIA_AFL_EXT.1.1/OLVM 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 a [password]

Application Note: This SFR was changed by TD0432

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

Application Note: This SFR was changed by TD0432

FIA_AFL_EXT.1/SSH Authentication Failure Handling

FIA_AFL_EXT.1.1/SSH The TSF shall detect when [

- an administrator configurable positive integer within a [*1 - 999*]] unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using a [password]

Application Note: This SFR was changed by TD0432

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

Application Note: This SFR was changed by TD0432

FIA_PMG_EXT.1 Password Management

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.

FIA_UAU.5 Multiple Authentication Mechanisms

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

- [local] authentication based on username and password,
- [local] authentication based on an SSH public key credential]

to support Administrator authentication.

FIA_UAU.5.2 The TSF shall authenticate any **Administrator’s** claimed identity according to the [*SSH CLI first performs the public key-based authentication which is followed by the username and password authentication if the public key authentication was unsuccessful, OLVM Web GUI authentication is based on username and password*].

Application Note: This SFR is altered by TD0360.

FIA_UIA_EXT.1 Administrator Identification and Authentication

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.

5.3.5 Security Management (FMT)

FMT_MOF_EXT.1 Management of Security Functions Behavior

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 12, based on the following key:

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

O = Optional (TOE may or may not 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 12: Management Functions

Number	Function	Administrator	User
1	Ability to update the Virtualization System	X	N
2	Ability to configure Administrator password policy as defined in FIA_PMG_EXT.1	X	N
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
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
8	Ability to configure inter-VM data sharing	X	N
9	Ability to enable/disable VM access to Hypercall functions	X	N
10	Ability to configure removable media policy	X	N
11	Ability to configure the cryptographic functionality	X	N
12	Ability to change default authorization factors	X	N
~~13~~	~~Ability to enable/disable screen lock~~	~~O~~	~~O~~
~~14~~	~~Ability to configure screen lock inactivity timeout~~	~~O~~	~~O~~
15	Ability to configure remote connection inactivity timeout	X	N

Number	Function	Administrator	User
16	Ability to configure lockout policy for unsuccessful authentication attempts through [limiting number of attempts during a time period]	X	N
~~17~~	~~Ability to configure name/address of directory server to bind with~~	~~S~~	~~O~~
18	Ability to configure name/address of audit/logging server to which to send audit/logging records	X	N
19	Ability to configure name/address of network time server	X	N
20	Ability to configure banner	X	N
21	Ability to connect/disconnect removable devices to/from a VM	X	N
22	Ability to start a VM	X	O
23	Ability to stop/halt a VM	X	O
24	Ability to checkpoint a VM	X	N
25	Ability to suspend a VM	X	O
26	Ability to resume a VM	X	O

Application Note: This SFR is altered by TD0360.

5.3.6 Security Management (FMT)

FMT_MSA_EXT.1 Default Data Sharing Configuration

FMT_MSA_EXT.1.1 The TSF shall by default enforce a policy prohibiting sharing of data between Guest VMs using [virtual networking].

FMT_MSA_EXT.1.2 The TSF shall allow Administrators to specify alternative initial configuration values to override the default values when a Guest VM is created.

FMT_SMO_EXT.1 Separation of Management and Operational Networks

FMT_SMO_EXT.1.1 The TSF shall support the configuration of separate management and operational networks through [logical means].

5.3.7 Protection of the TSF (FPT)

FPT_DVD_EXT.1 Non-Existence of Disconnected Virtual Devices

FPT_DVD_EXT.1.1 The TSF shall limit a Guest VM's access to virtual devices to those that are present in the VM's current virtual hardware configuration.

FPT_EEM_EXT.1 Execution Environment Mitigations

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

a) <u>Address space randomization</u>

b) <u>Stack buffer overflow protection</u>].

FPT_HAS_EXT.1 Hardware Assists

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 [*Extended Page Tables (EPT)*] to reduce or eliminate the need for shadow page tables.

FPT_HCL_EXT.1 Hypercall Controls

FPT_HCL_EXT.1.1 The TSF shall provide a Hypercall interface for Guest VMs to use to invoke functionality provided by the VMM.

FPT_HCL_EXT.1.2 The TSF shall allow administrators to configure any VM's Hypercall interface to disable access to individual functions, all functions, or groups of functions.

Application Note: This SFR was changed by TD0250

FPT_HCL_EXT.1.3 The TSF shall permit exceptions to the configuration of the following Hypercall interface functions: [*all functions*].

FPT_HCL_EXT.1.4 The TSF shall validate the parameters passed to the hypercall interface prior to execution of the VMM functionality exposed by that interface.

Application Note: Access to the hypercall interface is enabled by default and cannot be disabled.

FPT_RDM_EXT.1 Removable Devices and Media

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 [Virtual: *CD/DVD(ISO), Floppy Drive, Physical: USB Storage Device*] are switched between information domains, then [

c) the Administrator has granted explicit access for the media or device to be connected to the receiving domain].

FPT_TUD_EXT.1 Trusted Updates to the Virtualization System

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] prior to installing those updates.

FPT_VDP_EXT.1 Virtual Device Parameters

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.

FPT_VIV_EXT.1 VMM Isolation from VM's

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.

5.3.8 TOE Access (FTA)

FTA_TAB.1 TOE Access Banner

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.

Application Note: This SFR was changed by TD0363.

5.3.9 Trusted Path/Channel (FTP)

FTP_ITC_EXT.1 Trusted Channel Communication

FTP_ITC_EXT.1.1 The TSF shall use [

- TLS as conforming to [FCS_TLSS_EXT.1],
- TLS/HTTPS as conforming to FCS_HTTPS_EXT.1,
- SSH as conforming to the Extended Package for Secure Shell]

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 selected EP),
  - no other capabilities]

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.

FTP_TRP.1 Trusted Path

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*.

FTP_UIF_EXT.1 User Interface: I/O Focus

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

FTP_UIF_EXT.2 User Interface: Identification of VM

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

5.4 Assurance Requirements

5.4.1 Summary of Requirements

24 The TOE security assurance requirements are summarized in Table 13.

Table 13: Assurance Requirements

Assurance Class	Components	Description
Security Target Evaluation	Class ASE	As per ASE activities defined in [CEM] plus the TSS assurance activities defined for any SFRs claimed by the TOE.
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative Procedures
Life-Cycle Support	ALC_CMC.1	Labeling of the TOE
	ALS_CMS.1	TOE CM Coverage
	ALC_TSU_EXT.1	Timely Security Updates
Tests	ATE_IND.1	Independent Testing – Sample
Vulnerability Assessment	AVA_VAN.1	Vulnerability Survey

5.4.2 Timely Security Updates (ALC_TSU_EXT.1)

25 Oracle's timely security update methodology is published here: https://www.oracle.com/corporate/security-practices/assurance/vulnerability/security-fixing.html

26 Oracle's security vulnerability reporting procedures for users are published here: https://www.oracle.com/corporate/security-practices/assurance/vulnerability/reporting.html

27 Oracle's security alerts are published here: https://www.oracle.com/security-alerts/

6 TOE Summary Specification

28 The following describes how the TOE fulfils each SFR included in section 5.

6.1 Security Audit (FAU)

29 The auditing subsystem of Oracle Linux 7.6 UEK 5 KVM & Virtualization Manager 4.3 keeps a record of how the system is being used.

30 The TOE must be configured in accordance with [CC Guide] to ensure the events and information listed below are generated.

6.1.1 Audit Data Generation (FAU_GEN.1)

31 The TOE leverages the Lightweight Audit Framework (LAF) audit system.

32 Audit events are generated for the following audit functions:

a) Start-up and shut-down of the audit functions;

b) All administrative actions;

c) Audit events identified in Table 11.

33 Each audit record contains the following information:

a) Date and time of the event, type of event, subject identity (if applicable), and outcome (success or failure) of the event.

34 Logs that are generated by the TOE follow the type and format identified in the following link: https://access.redhat.com/articles/4409591.

35 Once the audit files are full, the administrator will be notified.

36 The TOE will stop logging locally and continue to send log records to the remote log server when configured in accordance with [CC Guide].

6.1.2 Audit Review (FAU_SAR.1)

37 The TOE provides the capability for users to read the audit records.

6.1.3 Protected Audit Trail Storage (FAU_STG.1)

38 The audit trail is stored in files which are only accessible by administrators. Only administrators may delete these files.

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

39 The TOE forwards logs to a Syslog server via SSH as described in FCS_SSHC_EXT.1. When the local audit data store is full, the TOE stops logging locally and continues to send log records to the remote log server.

6.2 Cryptographic Support (FCS)

40 The TOE employs the OpenSSL cryptographic module to provide the services described below.

6.2.1 Key Generation/Establishment (FCS_CKM.1 & FCS_CKM.2)

2 The TOE supports the following asymmetric cryptographic key generation algorithms:

a) RSA – 2048 and 3072

b) FFC – 2048 and 3072

3 The TOE supports the following key establishment schemes:

a) RSA based schemes

b) FFC based schemes

c) Diffie-Hellman group 14

4 Table 14 below identifies the scheme being used by each service.

Table 14: Key Generation/Establishment Mapping

Scheme	Usage	SFR	Service
RSA	Key Generation Key Establishment	FCS_TLSS_EXT.1	Remote Administration
	Key Generation	FCS_SSHS_EXT.1	Remote Administration
FFC	Key Generation Key Establishment	FCS_TLSS_EXT.1	Remote Administration
FFC - DH Group 14	Key Establishment	FCS_SSHS_EXT.1 FCS_SSHC_EXT.1	Remote Administration Logs

41 In the event of a decryption error, the TOE only logs/outputs aggregate generic error messages and does not reveal the particular error that occurred.

42 For RSA-based key establishment, the TOE acts as a recipient for TLS.

6.2.2 Key Destruction (FCS_CKM_EXT.4)

43 Table 15 identifies the TOE relevant cryptographic keys and related destruction information. The Generator/Initiator column indicates the entity that causes the key to enter volatile memory.

44 For volatile memory, the destruction shall be executed by a single overwrite consisting of zeroes when the keys are no longer needed.

45 For non-volatile memory the destruction consists of the invocation of an interface provided by the OS that logically addresses the storage location of the key and performs a single overwrite consisting of zeroes.

Table 15: Key Destruction

Key	Generator / Initiator	Storage	Destruction
TLS Private Keys (FCS_TLSS_EXT.1.1)	OpenSSL	Non-Volatile	Single overwrite consisting of zeroes.
TLS Session Keys (FCS_TLSS_EXT.1.1)	OpenSSL	Volatile	

Key	Generator / Initiator	Storage	Destruction
SSH Private Keys (FCS_SSHS_EXT.1.4) (FCS_SSHC_EXT.1.4)	OpenSSH	Non-Volatile	
		Volatile	
SSH Session Keys (FCS_SSHS_EXT.1.3) (FCS_SSHC_EXT.1.3)	OpenSSH	Volatile	

6.2.3 Encryption/Decryption (FCS_COP.1.1(1))

46 The TOE implements AES-CBC-128 and AES-CBC-256 in support of TLS and SSH.

6.2.4 Encryption/Decryption for SSH (FCS_COP.1.1(1)/SSH)

47 The TOE implements AES-CTR-128 and AES-CTR-256 in support of SSH. The SSH application relies on the platform for this functionality.

6.2.5 Hashing (FCS_COP.1(2))

48 The TOE supports Cryptographic hashing services conforming to FIPS Pub 180-4. The hashing algorithms are used for signature services and HMAC services.

49 The following hashing algorithms supported: SHA-1, SHA-256 and SHA-512.

50 The message digest sizes supported are: 160 bits, 256 bits and 512 bits.

6.2.6 Signature Algorithms (FCS_COP.1(3))

51 The TOE provides Cryptographic signature generation and verification in accordance with the following cryptographic algorithms:

a) RSA digital signature algorithm conforming to FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 4.

b) The RSA key sizes supported are: 2048 and 3072 bits.

6.2.7 Keyed Hash Algorithms (FCS_COP.1(4))

52 The TOE support keyed hash algorithms: HMAC-SHA-1, HMAC-SHA-256 and HMAC-SHA-512 used in TLS, SSH.

53 The characteristics of the HMACs used in the TOE are given in Table 16.

Table 16: HMAC Characteristics

Algorithm	Block Size	Key Size	Digest Size
HMAC-SHA-1	512 bits	160 bits	160 bits
HMAC-SHA-256	512 bits	256 bits	256 bits
HMAC-SHA-512	1024 bits	512 bits	512 bits

6.2.8 Entropy for Virtual Machines (FCS_ENT_EXT.1)

54 The VirtIO RNG (random number generator) is a paravirtualized device that is exposed as a hardware RNG device to the guest. This effectively allows a host to inject entropy into a guest via several means: The default mode uses the host's /dev/urandom, but a physical HW RNG device or EGD (Entropy Gathering Daemon) source can also be used. The Evaluated Configuration uses the default /dev/random which itself is fed by both high-speed hardware and software noise sources.

55 The methods described in FDP_HBI_EXT.1 ensure isolation between VMs (and their paravirtualized devices). Further, the TOE makes use of multiple entropy sources (as described in the proprietary Entropy Assessment Report), including hardware-based sources that cannot be influenced by software running on the host or VMs. Hence, one VM cannot not affect the entropy acquired by another VM.

6.2.9 Random Bit Generation (FCS_RBG_EXT.1)

56 The TOE leverages HMAC_DRBG (any) seeded by an entropy source that accumulates entropy from software and hardware noise sources 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.

6.2.10 HTTPS Protocol (FCS_HTTPS_EXT.1)

57 The OLVM web GUI is accessed via an HTTPS connection using the TLS implementation described by FCS_TLSS_EXT.1. The TOE does not use HTTPS in a client capacity. The TOE’s HTTPS protocol complies with RFC 2818.

58 RFC 2818 specifies HTTP over TLS. The majority of RFC 2818 is spent on discussing practices for validating endpoint identities and how connections must be setup and torn down. The TOE web GUI operates on an explicit port designed to natively speak TLS: it does not attempt STARTTLS or similar multi-protocol negotiation which is described in section 2.3 of RFC 2818. The web server TLS implementation attempts to send closure Alerts prior to closing a connection in accordance with section 2.2.2 of RFC 2818.

6.2.11 TLS Server Protocol (FCS_TLSS_EXT.1)

59 The TOE operates as a TLS server for the OLVM web GUI.

60 The server only allows TLS protocol version 1.2 (rejecting any other protocol version) and is restricted to the following ciphersuites:

a) TLS_RSA_WITH_AES_128_CBC_SHA

b) TLS_DHE_RSA_WITH_AES_128_CBC_SHA256

c) TLS_DHE_RSA_WITH_AES_256_CBC_SHA256

61 The TLS server is capable of negotiating ciphersuites that include RSA and DHE key agreement schemes. Supported RSA key sizes are 2048 and 3072 bits. Supported DHE key agreement parameters are 2048 and 3072 bits.

6.2.12 SSH Protocol – Client (FCS_SSHC_EXT.1)

62 The TOE operates as an SSH client for the trusted channel with the Audit Server.

63 The TOE supports public key-based (rsa-sha2-512) authentication.

64 The TOE examines the size of each received SSH packet. If the packet is greater than 262144 bytes, it is automatically dropped.

65 The TOE utilizes AES-CTR-128, AES-CTR-256, AES-CBC-128 and AES-CBC-256 for SSH encryption.

66 SSH transport implementation uses SSH-RSA as its public key algorithms.

67 The TOE provides data integrity for SSH connections via HMAC-SHA1, HMAC-SHA2-256 and HMAC-SHA2-512.

68 The TOE supports diffie-hellman-group14-sha1 for SSH key exchanges.

69 The TOE will re-key SSH connections after no more that 2^{28} packets have been transmitted using that key.

6.2.13 SSH Protocol – Server (FCS_SSHS_EXT.1)

70 The TOE CLI is remotely accessed using the SSH implementation.

71 The TOE supports password-based or public key (rsa-sha2-256 and rsa-sha2-512) for client authentication.

72 The TOE supports the following host key algorithms for the SSH server:

- ssh-rsa
- rsa-sha2-256
- rsa-sha2-512

73 The TOE examines the size of each received SSH packet. If the packet is greater than 262144 bytes, it is automatically dropped.

74 The TOE utilizes AES-CTR-128, AES-CTR-256, AES-CBC-128 and AES-CBC-256 for SSH encryption.

75 The TOE provides data integrity for SSH connections via HMAC-SHA1, HMAC-SHA2-256 and HMAC-SHA2-512.

76 The TOE supports diffie-hellman-group14-sha1 for SSH key exchanges.

77 The TOE will re-key SSH connections after no more that 2^{28} packets have been transmitted using that key.

6.3 User Data Protection (FDP)

6.3.1 Hardware-Based Isolation Mechanisms (FDP_HBI_EXT.1)

78 The TOE leverages the instruction sets Intel VT-x to enforce CPU isolation between the host and virtual machines. Intel VT-d is used 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.

6.3.2 Physical Platform Resource Controls (FDP_PPR_EXT.1)

79 The TOE uses VT-x 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.

80 The VMM distinguishes between VMs as follows - after a VM is created within KVM, it is registered as a domain within libvirt. Domains (VMs) are identified (distinguished) by an ID number and alpha-numeric name.

81 Physical devices that may be made available to VMs by an administrator are:

a) CPU

b) Memory

c) Network Adapter (Physical NIC)

82 When a VM is created or edited by an administrator, the above devices are either added/configured (allowed) or not added/configured (denied) to the VM.

6.3.3 Residual Information in Memory (FDP_RIP_EXT.1)

83 The Linux kernel clears memory prior to allocation to a Guest VM. The process is performed as part of standard kernel memory management when allocating memory to a userspace process.

84 There are no conditions where memory clearing is not performed.

6.3.4 Residual Information on Disk (FDP_RIP_EXT.2)

85 The TOE makes use of virtual disks for VM storage. Virtual disks are zeroed upon creation. Virtual disks are described in “13 Virtual Disks” of [OVIRT], chapter 13.1.

86 A VM may be attached to a shared virtual disk, in which case, the disk is not zeroed prior to allocation.

87 The TOE supports NFS, iSCSI, and FC storage types. The V5 format is used for storing domain and volume metadata. V5 metadata encompasses domain and volume metadata as follows:

- Domain Metadata
    - File storage domains store domain metadata in files.
    - Block storage domains store domain metadata in Logical Volume Management (LVM) Volume Group (VG) tags.
- Volume Metadata
    - File storage domains store volume metadata in files.
    - Block storage domains store volume metadata in metadata Logical Volumes (LVs).

6.3.5 VM Separation (FDP_VMS_EXT.1)

88 The TOE 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 other virtual machines or the external network. Configuration of virtual networking is described in “3.5 Setting up Networking for KVM Guests” of [KVM] and “2.3 Networks” of [OLVM].

89 A Guest VM cannot access the data of another Guest VM, or transfer data to another Guest VM other than through the mechanisms described in FDP_VMS_EXT.1.1 when expressly enabled by an authorized Administrator. There are no design or implementation flaws that permit the above mechanisms to be bypassed or defeated, or for data to be transferred through undocumented mechanisms. This claim does not apply to covert channels or architectural side-channels.

6.3.6 Virtual Networking Components (FDP_VNC_EXT.1)

90 Configuration of virtual networking is described in “3.5 Setting up Networking for KVM Guests” of [KVM] and “2.3 Networks” of [OLVM].

91 Traffic traversing a virtual network is visible only to Guest VMs that are configured by an Administrator to be members of that virtual network. There are no design or implementation flaws that permit the virtual networking configuration to be bypassed or defeated, or for data to be transferred through undocumented mechanisms. This claim does not apply to covert channels or architectural side-channels.

6.4 Identification and Authentication (FIA)

6.4.1 Authentication Failure Handling (FIA_AFL_EXT.1.1/SSH & FIA_AFL_EXT.1.2/SSH)

92 The TOE will detect when an administrator configurable integer within 1-999 unsuccessful authentication attempts for authentication based on username and password occur related to authentication on local console and password-based authentication via SSH v2 protocol. (FIA_AFL_EXT.1.1/SSH)

93 Once the specified number of unsuccessful authentication attempts for an account has been met, the OS shall disable the account, and prevent the user from accessing the TOE. (FIA_AFL_EXT.1.2/SSH)

6.4.2 Authentication Failure Handling (FIA_AFL_EXT.1.1/OLVM & FIA_AFL_EXT.1.2/OLVM)

94 The TOE will detect when an administrator configurable integer within 1-10 unsuccessful authentication attempts for authentication based on username and password occur related to OLVM authentication. (FIA_AFL_EXT.1.1/OLVM)

95 Once the specified number of unsuccessful authentication attempts for an account 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. (FIA_AFL_EXT.1.2/OLVM)

6.4.3 Password Management (FIA_PMG_EXT.1)

96 The TOE supports the local definition of users with corresponding passwords. The passwords can be composed of any combination of upper and lower case letters, numbers, and special characters "!", "@", "#", "$", "%", "^", "&", "*", "(", ")".

97 The minimum password length is settable by the administrator.

6.4.4 Multiple Authentication Mechanisms (FIA_UAU.5)

98 The TOE supports the following authentication mechanisms:

a) **OLVM Web.** HTTPS username and passwords

b) **SSH CLI.** Username and password combination and SSH public-keys

6.4.5 Administrator Identification and Authentication (FIA_UIA_EXT.1)

99 SSH CLI first performs the public key-based authentication which is followed by the username and password authentication if the public key authentication was unsuccessful. OLVM Web GUI authentication is based on username and password.

100 Administrators must be successfully authenticated before being granted access to any TOE functionality. Authentication is successful when the correct username and password combination are provided. Public key-based authentication (SSH CLI) requires the public key be added to the authorized key store.

6.5 Security Management (FMT)

6.5.1 Default Data Sharing Configuration (FMT_MSA_EXT.1)

101 Guest VMs are not connected to any virtual networks by default. The TOE administrator can create alternative default settings for virtual networking.

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

102 Administrators can establish separate management and operational networks using virtual networking.

6.6 Protection of the TSF (FPT)

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

103 Guest VMs only have access to the virtual devices that they have been explicitly configured to use.

6.6.2 Execution Environment Mitigations (FPT_EEM_EXT.1)

104 KVM makes use of the Oracle Linux provided environment-based vulnerability mitigation mechanisms:

a) Address space randomization

b) Stack buffer overflow protection

The Data Execution Prevention (DEP) feature prevents an application or service from executing code in a non-executable memory region. Hardware-enforced DEP works in conjunction with XD (Execute Disable – Intel) bit on compatible CPUs. Oracle Linux does not emulate the XD bit in software for CPUs that do not implement the XD bit in hardware.

6.6.3 Hardware Assists (FPT_HAS_EXT.1)

105 KVM supports the following hardware assists:

a) **Intel CPUs.** Intel VT-x and Extended Page Tables (used as hardware assists for binary translation).

6.6.4 Hypercall Controls (FPT_HCL_EXT.1)

106 Hypercalls are enabled by default and cannot be disabled. KVM supports the following hypercalls:

a) KVM_HC_VAPIC_POLL_IRQ - Trigger guest exit so that the host can check for pending interrupts on reentry.

b) KVM_HC_KICK_CPU - Hypercall used to wakeup a vCPU from halt (HLT) state.

c) KVM_HC_CLOCK_PAIRING - Hypercall used to synchronize host and guest clocks.

d) KVM_HC_SEND_IPI - Send Inter-processor Interrupt (IPIs) to multiple vCPUs.

107 The parameters and legal values for the above hypercalls are documented at the following location: https://www.kernel.org/doc/html/latest/virt/kvm/x86/hypercalls.html Where not explicitly implied, the legal values are within the bounds of the data type.

6.6.5 Removable Devices and Media (FPT_RDM_EXT.1)

108 The TOE 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 floppies and virtual optical device images (e.g. ISO images). ISO images are presented read-only (no write access is permitted).

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

109 The TOE software is delivered and installed using Red Hat Packages (RPMs).

110 An Oracle PGP Public Key is used to verify the RPM during installation. The public key is installed on the system at the time of installation. The TOE leverages 2048 bit RSA digital signature mechanism for signing and verification of packages/updates. SHA-256 used for integrity verification. If the signature verification is successful, then the RPM package is installed. Otherwise it fails the installation. The administrator must download the RPM from the Oracle download center.

111 Note that the TOE leverages the digital signature mechanism of the Oracle public key that is used to verify the RPM rather than claiming an entire PKI for code signing.

112 To obtain updates, the OS pulls the latest update lists from Oracle servers nightly and either installs new RPMs automatically or informs the administrator about the presence of update RPMs, depending on the system configuration. The installation of these updates follows the signature verification procedure discussed above.

6.6.7 Virtual Device Parameters (FPT_VDP_EXT.1)

113 KVM implements many core devices for virtual machines as software. These emulated hardware devices are crucial for virtualizing operating systems. Emulated devices are virtual devices which exist entirely in software.

114 In addition, KVM provides emulated drivers. These form a translation layer between the virtual machine and the Linux kernel (which manages the source device). The device level instructions are completely translated by the KVM hypervisor. Any device of the same type (storage, network, keyboard, or mouse) that is recognized by the Linux kernel can be used as the backing source device for the emulated drivers.

115 The following virtual devices are supported:

- CPUs
- Intel i440FX host PCI bridge
- PIIX3 PCI to ISA bridge
- PS/2 mouse and keyboard

- EvTouch USB graphics tablet
- PCI UHCI USB controller and a virtualized USB hub
- Emulated serial ports
- EHCI controller, virtualized USB storage and a USB mouse
- USB 3.0 xHCI host controller
- Storage drivers
- PCI IDE (CD/DVD-ROM)
- Floppy disk driver
- HDA sound device (intel-hda)
- Cirrus CLGD 5446 PCI VGA card
- Standard VGA graphics card with Bochs VESA extension
- Intel E1000 network adapter
- Realtek 8139 network adapter
- Intel 6300 ESB PCI watchdog device
- iBase 700 ISA watchdog device
- Virtio-net (network device)
- Virtio-block (block device)
- Virtio-scsi (controller device)
- Clock source
- Virtio-serial (serial device)
- Virtio-balloon (balloon device)
- Virtio-rng (Random Number generator)
- QXL driver

116 Additional details on the above paravirtualized and emulated devices can be found at: Virtualized Hardware Device documentation.

117 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. Thorough testing and architectural design reviews have been conducted to ensure the accuracy of these claims, and there are no known design or implementation flaws that bypass or defeat the security of the virtual device interfaces.

118 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. See the separate document “Oracle KVM Virtual Devices” for a list of virtual devices, ports, and legal values.

6.6.8 VMM Isolation from VMs (FPT_VIV_EXT.1)

119 Software running in a VM is not able to degrade or disrupt the functioning of other VMs, the VMM, or the Platform. There are no design or implementation flaws that bypass or defeat VM isolation.

6.7 Security Management (FMT)

6.7.1 Management of Security Functions Behaviour (FMT_MOF_EXT.1)

120 The TOE is capable of performing the management functions marked with an X or O in Table 12.

6.8 TOE Access (FTA)

6.8.1 TOE Access Banner (FTA_TAB.1)

121 Access banners may be configured for both the SSH CLI and OLVM Web interfaces.

6.9 Trusted Path/Channel (FTP)

6.9.1 Trusted Channel Communications (FTP_ITC_EXT.1)

122 The TOE implements the following trusted channels:

a) TLS/HTTPS for the OLVM web interface

b) SSH for the CLI

c) SSH for Syslog

6.9.2 Trusted Path (FTP_TRP.1)

123 The implements the following trusted paths:

a) TLS/HTTPS for the OLVM web interface

b) SSH for the CLI

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

124 The TOE supports keyboard and pointer (mouse, trackball etc.) user input devices.

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

125 VMs are assigned a unique name when they are created. This name is displayed to users of the VM in the title bar of the Remote Viewer window in which the VM is running.

7 Rationale

7.1 Conformance Claim Rationale

126 The following rationale is presented with regard to the PP conformance claims:

a) **TOE type.** As identified in section 2.1, the TOE is a server virtualization management platform, consistent with the Base PP.

b) **Security problem definition.** As shown in section 3, the threats, OSPs and assumptions are reproduced directly from the Base PP.

c) **Security objectives.** As shown in section 4, the security objectives are reproduced directly from the Base PP.

d) **Security requirements.** As shown in section 5, the security requirements are reproduced directly from the Base PP, SV_EP and SSH_EP. No additional requirements have been specified.

7.2 Security Objectives Rationale

127 All security objectives are drawn directly from the NIAP Virtualization Base PP. Table 17 reproduces the rationale from the NIAP Virtualization Base PP.

Table 17: Security Objectives Rationale

Threat, Assumption, or OSP	Security Objective	Rationale
T.DATA_LEAKAGE	O.VM_ISOLATION O.DOMAIN_INTEGRITY	Logical separation of VMs and enforcement of domain integrity prevent unauthorized transmission of data from one VM to another.
T.UNAUTHORIZED _UPDATE	O.VMM_INTEGRITY	System integrity prevents the TOE from installing a software patch containing unknown and potentially malicious code
T.UNAUTHORIZED _MODIFICATION	O.VMM_INTEGRITY O.AUDIT	Enforcement of VMM integrity prevents the bypass of enforcement mechanisms and auditing ensures that abuse of legitimate authority can be detected.
T.USER_ERROR	O.VM_ISOLATION	Isolation of VMs includes clear attribution of those VMs to their respective domains which reduces the likelihood that a user inadvertently inputs or transfers data meant for one VM into another.

Threat, Assumption, or OSP	Security Objective	Rationale
T.3P_SOFTWARE	O.VMM_INTEGRITY	The VMM integrity mechanisms include environment-based vulnerability mitigation and potentially support for introspection and device driver isolation, all of which reduce the likelihood that any vulnerabilities in third-party software can be used to exploit the TOE.
T.VMM_COMPROMISE	O.VMM_INTEGRITY O.VM_ISOLATION	Maintaining the integrity of the VMM and ensuring that VMs execute in isolated domains mitigate the risk that the VMM can be compromised or bypassed.
T.PLATFORM _COMPROMISE	O.PLATFORM_INTEGRITY	Platform integrity mechanisms used by the TOE reduce the risk that an attacker can 'break out' of a VM and affect the platform on which the VS is running.
T.UNAUTHORIZED _ACCESS	O.MANAGEMENT_ACCESS	Ensuring that TSF management functions cannot be executed without authorization prevents untrusted subjects from modifying the behaviour of the TOE in an unanticipated manner.
T.WEAK_CRYPTO	O.VM_ENTROPY	Acquisition of good entropy is necessary to support the TOE's security-related cryptographic algorithms.
T.UNPATCHED _SOFTWARE	O.PATCHED_SOFTWARE	The ability to patch the TOE software ensures that protections against vulnerabilities can be applied as they become available.
T.MISCONFIGURATION	O.CORRECTLY_APPLIED _CONFIGURATION	Mechanisms to prevent the application of configurations that violate the current security policy help prevent misconfigurations.
T.DENIAL_OF _SERVICE	O.RESOURCE_ALLOCATION	The ability of the TSF to ensure the proper allocation of resources makes denial of service attacks more difficult.

Threat, Assumption, or OSP	Security Objective	Rationale
A.COVERT _CHANNELS	OE.COVERT_CHANNELS	It is expected that any data contained within VMs is commensurate with the security provided by the TOE, which includes any vulnerabilities due to the potential presence of covert storage and/or timing channels.
A.NON_MALICIOUS _USER	OE.NON_MALICIOUS_USER	If the organization properly vets and trains users, it is expected that they will be non-malicious.
	OE.CONFIG	If the TOE is administered by a non-malicious and non-negligent user, the expected result is that the TOE will be configured in a correct and secure manner.
A.PLATFORM _INTEGRITY	OE.PLATFORM_INTEGRITY	If the underlying platform has not been compromised prior to installation of the TOE, its integrity can be assumed to be intact.
A.PHYSICAL	OE.PHYSICAL	If the TOE is deployed in a location that has appropriate physical safeguards, it can be assumed to be physically secure.
A.TRUSTED_ADMIN	OE.TRUSTED_ADMIN	Providing guidance to administrators and ensuring that individuals are properly trained and vetted before being given administrative responsibilities will ensure that they are trusted.

7.3 Security Requirements Rationale

129 All security requirements are drawn directly from the claimed Base PP and extended packages consistent with the principle of exact conformance.