Oracle VM Server for SPARC 3.6 and Oracle Solaris 11.4

Security Target

Version 2.3

January 2024

Document prepared by

www.lightshipsec.com

Document History

Version	Date	Description
0.1	27 Oct 2021	Initial draft.
0.2	3 Dec 2021	Updated with developer comments.
0.3	21 Dec 2021	Reverted to Virt_PPv1.1 conformance.
0.4	15 Feb 2022	Updated TOE version, added TD0615, and updated FFC claims.
1.0	3 April 2022	Addressed evaluator ORs.
1.1	9 May 2022	Addressed evaluator ORs.
1.2	14 June 2022	Addressed evaluator ORs.
1.3	7 July 2022	Addressed evaluator ORs.
1.4	6 Sept 2022	Addressed evaluator ORs. Updated TOE version.
1.5	12 Feb 2023	Addressed CBORs.
1.6	22 Feb 2023	Update Table 14
1.7	29 June 2023	Updated TOE version, modified management function claims.
1.8	13 July 2023	Updated FPT_TUD claims.
1.9	9 August 2023	Minor updates.
2.0	25 September 2023	Address CB ORs.
2.1	20 November 2023	Addressed CB ORs.
2.2	10 January 2024	Addressed CB ORs.
2.3	19 January 2024	Updated guidance references.

Table of Contents

List of Tables

1 Introduction

1.1 Overview

1 This Security Target (ST) defines the Oracle Oracle VM Server for SPARC 3.6 and Oracle Solaris 11.4 Target of Evaluation (TOE) for the purposes of Common Criteria (CC) evaluation.

2 Oracle VM Server for SPARC is the SPARC hypervisor virtualization solution for simultaneously running multiple Solaris instances on a single physical domain. A physical domain is the scope of resources that are managed by a single Oracle VM Server for SPARC instance.

1.2 Identification

Table 1: Evaluation identifiers

Target of Evaluation	Oracle VM Server for SPARC 3.6.2.0.57 and Oracle Solaris 11.4.57.0.1.144.3 with IDR 5391
Security Target	Oracle VM Server for SPARC 3.6 and Oracle Solaris 11.4 Security Target, v2.3

1.3 Conformance Claims

3 This ST and the TOE are conformant to the following:

a) CC version 3.1 Release 5

i) CC Part 2 extended

ii) CC Part 3 extended

b) NIAP Protection Profile for Virtualization, Version 1.1 (Base_PP)

c) NIAP PP-Module for Server Virtualization, Version 1.1 (MOD_SV)

d) NIAP Functional Package for SSH, Version 1.0 (PKG_SSH)

e) NIAP Functional Package for TLS, Version 1.1 (PKG_TLS)

f) NIAP PP-Configuration for Virtualization and Server Virtualization Systems, Version 1.0 (VIRT_CFG)

g) NIAP Technical Decisions per Table 2.

Table 2: NIAP Technical Decisions

TD #	Name	NIAP PP/Package
0442	Updated TLS Ciphersuites for TLS Package	PKG_TLS
0469	Modification of test activity for FCS_TLSS_EXT.1.1 test 4.1	PKG_TLS
0499	Testing with pinned certificates	PKG_TLS
0513	CA Certificate loading	PKG_TLS

TD #	Name	NIAP PP/Package
0605	Updates to Certificate Revocation (FIA_X509_EXT.1 for Base Virtualization PP v1.1	Base_PP
0615	Audit generation for hypercalls implemented in HW	Base_PP
0682	Addressing ambiguity in FCS_SSHS_EXT.1 Tests	PKG_SSH
0695	Choice of 128 or 256 bit size in AES-CTR in SSH Functional Package	PKG_SSH
0721	Mapping FTA_TAB.1 to Objective	Base_PP
0726	Corrections to (D)TLSS SFRs in TLS 1.1 FP	PKG_TLS
0732	FCS_SSHS_EXT.1.3 Test 2 Update	PKG_SSH
0739	PKG_TLS_V1.1 has 2 different publication dates	PKG_TLS
0770	TLSS.2 connection with no client cert	PKG_TLS
0777	Clarification to Selections For Auditable Events for FCS_SSH_EXT.1	PKG_SSH
0779	Updated Session Resumption Support in TLS Package V1.1	PKG_TLS

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.
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.

Term	Definition
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 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

Term	Definition
	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.
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, 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.
Physical Platform	The hardware environment on which a VS executes. Physical platform resources include processors, memory, devices, and associated firmware.

Term	Definition
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 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.

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 Solaris 11.4 and is used to provide server virtualization capabilities to users. The TOE is deployed on enterprise-class SPARC server hardware housed in data centers. Administrators 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 The TOE offers a Command line Interface (CLI) over Secure Shell (SSH) via Oracle Solaris to manage the virtualized infrastructure and administer servers running Oracle VM Server for SPARC.

2.2.1 Secure Communications

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

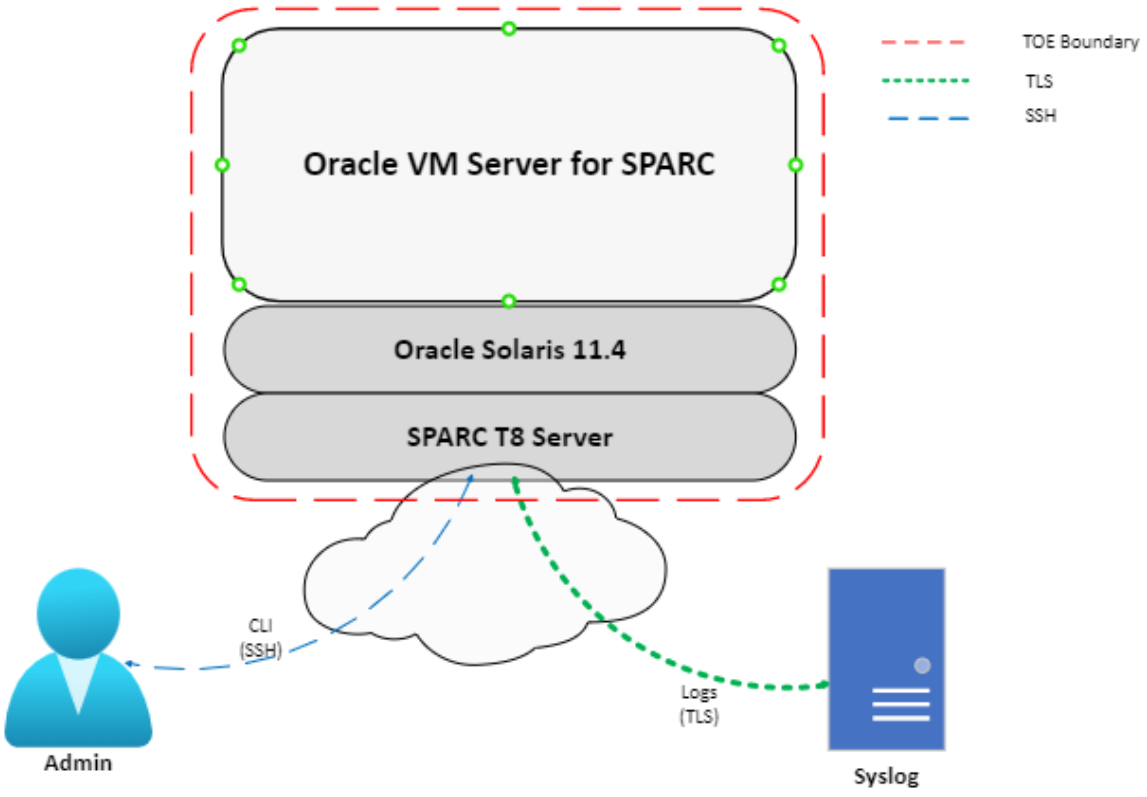

Figure 1: Secure Communication Channels

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

2.4.1 Software

10 The TOE is Oracle VM Server for SPARC 3.6.2.0.57 and Oracle Solaris 11.4.57.0.1.144.3 with IDR 5391, running on the SPARC T8 hardware. The TOE software is installed on the TOE hardware and delivered to the customer by a commercial courier service with a package tracking system.

11 To obtain the CC evaluated version of the TOE, if not already installed, customers may download the TOE software from the Oracle Software Delivery Cloud at https://edelivery.oracle.com/.

12 Note: The SPARC software is bundled with Solaris 11.4.

2.4.2 Guidance Documents

13 The TOE includes the following guidance documents:

a) [CC Guide] Oracle VM Server for SPARC 3.6 and Oracle Solaris 11.4 Common Criteria Guide, v1.5 (PDF)

b) [SPARC] - Oracle VM Server for SPARC 3.6 Documentation Library - https://docs.oracle.com/cd/E93612_01/

c) [T8LIB] – Oracle SPARC T8 information Library - https://docs.oracle.com/en/servers/sparc/t8/index.html

d) [Solaris] – Oracle Solaris 11.4 Information Library - https://docs.oracle.com/cd/E37838_01/

2.4.3 Non-TOE Components

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

a) **Audit Server.** The TOE sends audit events to a syslog server.

2.4.4 CAVP

15 Table 4 identifies the relevant CAVP certificates for the TLS and SSH implementations.

Table 4: CAVP

Algorithm	Standard	Library	CAVP	Hardware/CPU
AES	AES-CBC (as defined in FIPS PUB 197 and NIST SP 800-38A) AES-GCM (as defined in NIST SP 800-38D) AES-CTR (as defined in NIST SP 800-38A) AES-CCM (as defined in NISTP SP 800-38C) AES-KW (as defined in NIST SP 800-38F)	OpenSSL 3.0.8	A4216	SPARC T8 Server / SPARC M8 CPU
ECDSA	FIPS PUB 186-4			
RSA	FIPS PUB 186-4			
HMAC	FIPS PUB 198-1 and FIPS PUB 180-4			
SHS	FIPS PUB 180-4			
KAS-FFC	NIST SP 800-56A			

Algorithm	Standard	Library	CAVP	Hardware/CPU
KDF SSH	N/A – RFC 4253			
Hash_DRBG	NIST SP 800-57	OpenSSL 3.0.8	A4216	
		Oracle Solaris KCF	C1895	

2.5 Excluded Functionality

16 This CC evaluation only covers the functionality identified in section 2.3 when Oracle VM Server for SPARC and Solaris 11.4 are configured in accordance with the [CC Guide]. Generic Solaris 11.4 OS functionality, the LDMD XMPP Management service, and LDMD Migration service are not included in the evaluation and should only be exposed on a dedicated management network in accordance with [CC Guide] and [Solaris].

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 VS 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 VS 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, functions critical to the security of the TOE are by necessity performed by software not produced by the virtualization vendor. Such software may include physical device drivers, and even non-TOE entities such as Host Operating Systems. Since this software has the same or similar privilege level as the VS, vulnerabilities can be exploited by an adversary to compromise the VS and VMs. Where possible, the VS should mitigate the results of potential vulnerabilities or malicious content in third-party code on which it relies. For example, physical device drivers (potentially the Host OS) could be encapsulated within VMs in order to limit the effects of compromise.
T.VMM_ COMPROMISE	The VS 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, by non-TOE software, such as that running in Guest or Helper VMs or on the host platform. This must be prevented to avoid compromising the VS.
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 VS 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 or obtain sensitive information from the TOE. 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 or extract sensitive information. 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.

Identifier	Description
T.WEAK_CRYPTO	To the extent that VMs appear isolated within the VS, 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. Such defeat can result in the compromise of Guest VM data and credentials, and of VS data and credentials, and can enable unauthorized access to the VS or VMs.
T.UNPATCHED_ SOFTWARE	Vulnerabilities in outdated or unpatched software can be exploited by adversaries to compromise the VS or platform.
T.MISCONFIGURATION	The VS 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 VS.
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.NON_MALICIOUS_ USER	The user of the VS is not willfully 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 customer requirements, a VM may need a completely isolated environment with exclusive access to system resources or share some of its resources with other VMs. It must be possible to enforce a security policy that prohibits the transfer of data between VMs through shared devices. When the platform security policy allows the sharing of resources across VM boundaries, the VMM must ensure that all access to those resources is consistent with the policy. The VMM may delegate the responsibility for the mediation of resource sharing to select Service VMs; however in doing so, it remains responsible for mediating access to the Service VMs, and each Service VM must mediate all access to any shared resource that has been delegated to it in accordance with the SSP. Both virtual and physical devices are resources requiring access control. The VMM must enforce access control in accordance with system security policy. Physical devices are platform devices with access mediated via the VMM per the O.VMM_Integrity objective. Virtual devices may include virtual storage devices and virtual network devices. Some of the access control restrictions must be enforced internal to Service VMs, as may be the case for isolating virtual networks. VMMs may also expose purely virtual interfaces. These are VMM specific, and while they are not analogous to a physical device, they are also subject to access control. The VMM must support the mechanisms to isolate all resources associated with virtual networks and to limit a VM's access to only those virtual networks for which it has been configured. The VMM must also support the mechanisms to control the configurations of virtual networks according to the SSP.

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 VS—not the integrity of software running inside of Guest VMs or of the physical platform. The overall objective is to ensure the integrity of critical components of a VS. Initial integrity of a VS can be established through mechanisms such as a digitally signed installation or update package, or through integrity measurements made at launch. Integrity is maintained in a running system by careful protection of the VMM from untrusted users and software. For example, it must not be possible for software running within a Guest VM to exploit a vulnerability in a device or hypercall interface and gain control of the VMM. The vendor must release patches for vulnerabilities as soon as practicable after discovery.
O.PLATFORM_ INTEGRITY	The integrity of the VMM depends on the integrity of the hardware and software on which the VMM relies. Although the VS does not have complete control over the integrity of the platform, the VS should as much as possible try to ensure that no users or software hosted by the VS can undermine the integrity of the platform.
O.DOMAIN_ INTEGRITY	While the VS is not responsible for the contents or correct functioning of software that runs within Guest VMs, it is responsible for ensuring that the correct functioning of the software within a Guest VM is not interfered with by other VMs.

O.MANAGEMENT_ ACCESS	VMM management functions include VM configuration, virtualized network configuration, allocation of physical resources, and reporting. Only authorized users (administrators) may exercise management functions. Because of the privileges exercised by the VMM management functions, it must not be possible for the VMM's management components to be compromised without administrator notification. This means that unauthorized users cannot be permitted access to the management functions, and the management components must not be interfered with by Guest VMs or unprivileged users on other networks— including operational networks connected to the TOE. VMMs include a set of management functions that collectively allow administrators to configure and manage the VMM, as well as configure Guest VMs. These management functions are specific to the VS and are distinct from any other management functions that might exist for the internal management of any given Guest VM. These VMM management functions are privileged, with the security of the entire system relying on their proper use. The VMM management functions can be classified into different categories and the policy for their use and the impact to security may vary accordingly. The management functions are distributed throughout the VMM (within the VMM and Service VMs). The VMM must support the necessary mechanisms to enable the control of all management functions according to the system security policy. When a management function is distributed among multiple Service VMs, the VMs must be protected using the security mechanisms of the Hypervisor and any Service VMs involved to ensure that the intent of the system security policy is not compromised. Additionally, since hypercalls permit Guest VMs to invoke the Hypervisor, and often 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 VS must be updated and patched when needed in order to prevent the potential compromise of the VMM, as well as the networks and VMs that it hosts. Identifying and applying needed updates must be a normal part of the operating procedure to ensure that patches are applied in a timely and thorough manner. In order to facilitate this, the VS must support standards and protocols that help enhance the manageability of the VS as an IT product, enabling it to be integrated as part of a manageable network (e.g., reporting current patch level and patchability).
O.VM_ENTROPY	VMs must have access to good entropy sources to support security-related features that implement cryptographic algorithms. For example, in order to function as members of operational networks, VMs must be able to communicate securely with other network entities—whether virtual or physical. They must therefore have access to sources of good entropy to support that secure communication.
O.AUDIT	An audit log must be created that captures accesses to the objects the TOE protects. The log of these accesses, or audit events, must be protected from modification, unauthorized access, and destruction. The audit log must be sufficiently detailed to indicate the date and time of the event, the identify of the user, the type of event, and the success or failure of the event.
O.CORRECTLY_ APPLIED_ CONFIGURATION	The TOE must not apply configurations that violate the current security policy. The TOE must correctly apply configurations and policies to a newly created Guest VM, 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 VS correctly to create the intended security policy.
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.
OE.NON_MALICIOUS_USER	Users are trusted to 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 defined in the Base_PP, PKG_TLS, PKG_SSH, and MOD_SV.

Table 9: Extended Components

Component	Title	Source
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
FCS_TLS_EXT.1	TLS Protocol	PKG_TLS
FCS_TLSC_EXT.1	TLS Client Protocol	PKG_TLS
FCS_SSH_EXT.1	SSH Protocol	PKG_SSH
FCS_SSHS_EXT.1	SSH Protocol – Server	PKG_SSH
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

Component	Title	Source
FDP_VNC_EXT.1	Virtual Networking Components	Base_PP
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
FIA_X509_EXT.1	X.509 Certificate Validation	Base_PP
FIA_X509_EXT.2	X.509 Certificate Authentication	Base_PP
FMT_MOF_EXT.1	Management of Security Functions Behavior	MOD_SV
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
ALC_TSU_EXT.1	Timely Security Updates	Base_PP

5.3 Functional Requirements

Table 10: Summary of SFRs

Requirement	Title	Source	Type
FAU_GEN.1	Audit Data Generation	Base_PP	Mandatory
FAU_SAR.1	Audit Review	Base_PP	Mandatory
FAU_STG.1	Protected Audit Trail Storage	Base_PP	Mandatory
FAU_STG_EXT.1	Off-Loading of Audit Data	Base_PP	Mandatory
FCS_CKM.1	Cryptographic Key Generation	Base_PP	Mandatory
FCS_CKM.2	Cryptographic Key Distribution	Base_PP	Mandatory
FCS_CKM_EXT.4	Cryptographic Key Destruction	Base_PP	Mandatory
FCS_COP.1/Hash	Cryptographic Operation (Hashing)	Base_PP	Mandatory
FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithms)	Base_PP	Mandatory
FCS_COP.1/Sig	Cryptographic Operation (Signature Algorithms)	Base_PP	Mandatory
FCS_COP.1/UDE	Cryptographic Operation (AES Data Encryption/Decryption)	Base_PP	Mandatory
FCS_ENT_EXT.1	Entropy for Virtual Machines	Base_PP	Mandatory
FCS_RBG_EXT.1	Cryptographic Operation (Random Bit Generation)	Base_PP	Mandatory
FCS_TLS_EXT.1	TLS Protocol	PKG_TLS	Selection
FCS_TLSC_EXT.1	TLS Client Protocol	PKG_TLS	Selection
FCS_SSH_EXT.1	SSH Protocol	PKG_SSH	Selection
FCS_SSHS_EXT.1	SSH Protocol – Server	PKG_SSH	Selection
FDP_HBI_EXT.1	Hardware-Based Isolation Mechanisms	Base_PP	Mandatory
FDP_PPR_EXT.1	Physical Platform Resource Controls	Base_PP	Mandatory
FDP_RIP_EXT.1	Residual Information in Memory	Base_PP	Mandatory
FDP_RIP_EXT.2	Residual Information on Disk	Base_PP	Mandatory
FDP_VMS_EXT.1	VM Separation	Base_PP	Mandatory

Requirement	Title	Source	Type
FDP_VNC_EXT.1	Virtual Networking Components	Base_PP	Mandatory
FIA_AFL_EXT.1	Authentication Failure Handling	Base_PP	Mandatory
FIA_PMG_EXT.1	Password Management	Base_PP	Selection
FIA_UAU.5	Multiple Authentication Mechanisms	Base_PP	Mandatory
FIA_UIA_EXT.1	Administrator Identification and Authentication	Base_PP	Mandatory
FIA_X509_EXT.1	X.509 Certificate Validation	Base_PP	Selection
FIA_X509_EXT.2	X.509 Certificate Authentication	Base_PP	Selection
FMT_MOF_EXT.1	Management of Security Functions Behavior	MOD_SV	Mandatory
FMT_SMO_EXT.1	Separation of Management and Operational Networks	Base_PP	Mandatory
FPT_DVD_EXT.1	Non-Existence of Disconnected Virtual Devices	Base_PP	Mandatory
FPT_EEM_EXT.1	Execution Environment Mitigations	Base_PP	Mandatory
FPT_HAS_EXT.1	Hardware Assists	Base_PP	Mandatory
FPT_HCL_EXT.1	Hypercall Controls	Base_PP	Mandatory
FPT_RDM_EXT.1	Removable Devices and Media	Base_PP	Mandatory
FPT_TUD_EXT.1	Trusted Updates to the Virtualization System	Base_PP	Mandatory
FPT_VDP_EXT.1	Virtual Device Parameters	Base_PP	Mandatory
FPT_VIV_EXT.1	VMM Isolation from VMs	Base_PP	Mandatory
FTA_TAB.1	TOE Access Banner	Base_PP	Mandatory
FTP_ITC_EXT.1	Trusted Channel Communications	Base_PP	Mandatory
FTP_TRP.1	Trusted Path	Base_PP	Selection
FTP_UIF_EXT.1	User Interface: I/O Focus	Base_PP	Mandatory
FTP_UIF_EXT.2	User Interface: Identification of VM	Base_PP	Mandatory

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 relevant to claimed SFRs as defined in the Auditable Events Table from the Client and Server PP-Modules**]

c) [**Auditable events defined in Table 11**]

d) **[**

- Auditable events defined Table 11 for Selection-Based SFRs,
- Auditable events for the Functional Package for Transport Layer Security (TLS), version 1.1 listed in Table 11,
- Auditable events defined in the audit table for the Functional Package for Secure Shell (SSH), version 1.0,

]

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) **[**

- Additional information defined in Table 11 for Selection-Based SFRs,
- Additional information for the Functional Package for Transport Layer Security (TLS), version 1.1 listed in Table 11,
- Additional information defined in the audit table for the Functional Package for Secure Shell (SSH), version 1.0,

]

Table 11: Auditable Events

Requirement	Auditable Events	Additional Details
FAU_GEN.1	None.	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.	None.
	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.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_COP.1/Sig	None.	None.
FCS_COP.1/UDE	None.	None.
FCS_ENT_EXT.1	None.	None.
FCS_RBG_EXT.1	Failure of the randomization process.	None.
FCS_SSH_EXT.1	[Failure to establish SSH connection]	[Reason for failure and Non-TOE endpoint of attempted connection (IP address)].
	[Establishment of SSH connection]	[Non-TOE endpoint of attempted connection (IP address)].
	[Termination of SSH connection]	[Non-TOE endpoint of attempted connection (IP address)].
	[None]	[None].
FCS_SSHS_EXT.1	None.	None.

Requirement	Auditable Events	Additional Details
FCS_TLSC_EXT.1	Failure to establish a session.	Reason for failure.
	Failure to verify presented identifier.	Presented identifier and reference identifier.
	Establishment/Termination of a TLS session.	Non-TOE endpoint of connection.
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.	VM and physical device identifiers.
	Security policy violations.	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.	VM and virtual or physical networking component identifiers.
	Security policy violations.	Identifier for the security policy that was violated.
	Administrator configuration of inter-VM communications channels between VMs.	VM and virtual or physical networking component identifiers.
FIA_AFL_EXT.1	Unsuccessful login attempts limit is met or exceeded.	Origin of attempt (e.g., IP address).
FIA_UAU.5	None.	None.
FIA_UIA_EXT.1	Administrator authentication attempts.	Provided user identity, origin of the attempt (e.g., console, remote IP address).
	All use of the identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., console, remote IP address).
	[None]	N/A
FIA_PMG_EXT.1	None.	None.

Requirement	Auditable Events	Additional Details
FIA_X509_EXT.1	Failure to validate a certificate.	Reason for failure.
FIA_X509_EXT.2	None.	None.
FMT_MOF_EXT.1	Attempts to invoke any of the management functions listed in Table 12.	Success or failure of attempt Identity of actor
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	[None.]	N/A
	[None.]	None.
FPT_RDM_EXT.1	Connection/disconnection of removable media or device to/from a VM.	VM Identifier, Removable media/device identifier, event description or identifier (connect/disconnect, ejection/insertion, etc.).
	Ejection/insertion of removable media or device from/to an already connected VM.	
FPT_TUD_EXT.1	Initiation of update.	None.
	Failure of signature verification.	None.
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.	User ID and remote source (IP Address) if feasible.
	Termination of the trusted channel.	
	Failures of the trusted path functions.	
FTP_TRP.1	Initiation of the trusted channel.	User ID and remote source (IP address) if feasible.
	Termination of the trusted channel.	

Requirement	Auditable Events	Additional Details
	Failures of the trusted path functions.	
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*] unauthorized 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)

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],
- ECC schemes using [“NIST curves” P-256, P-384, and [no other curves] that meet the following: [FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4],
- FFC Schemes using Diffie-Hellman group 14 that meet the following: [RFC 3526]
- FFC Schemes using safe primes that meet the following: [‘NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes”]

] ~~and specified cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards].~~

FCS_CKM.2 Cryptographic Key Distribution

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

- RSA-based key establishment schemes that meets the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, “Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.2",
- Finite field-based key establishment schemes that meets the following: NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”,
- Key establishment scheme using Diffie-Hellman group 14 that meets the following: RFC 3256

] ~~that meets the following [assignment: list of standards].~~

Application Note: The TOE implements FFC Schemes using "safe-prime" groups (identified in Appendix D of SP 800-56A, Revision 3)

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/Hash Cryptographic Operation (Hashing)

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

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

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

FCS_COP.1/Sig Cryptographic Operation (Signature Algorithms)

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

- RSA schemes using cryptographic key sizes [2048-bit or greater] that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 4]
- ECDSA schemes using ["NIST curves" P-256, P-384, and [no other curves]] that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5]].

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

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

- AES Key Wrap (KW) (as defined in NIST SP 800-38F),
- AES-GCM (as defined in NIST SP 800-38D),
- AES-CCM (as defined in NIST SP 800-38C),
- AES-CBC (as defined in FIPS PUB 197, and NIST SP 800-38A) mode,
- AES-CTR (as defined in NIST SP 800-38A) mode

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

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 [Hash_DRBG (any)]

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

FCS_TLS_EXT.1 TLS Protocol

FCS_TLS_EXT.1.1 The product shall implement [

- TLS as a client

].

FCS_TLSC_EXT.1 TLS Client Protocol

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

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246,
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246,
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,
- TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,
- TLS_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,
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288

] and also supports functionality for [none].

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

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

FCS_SSH_EXT.1 SSH Protocol

FCS_SSH_EXT.1.1 The TOE shall implement SSH acting as a [server] in accordance with that complies with RFCs 4251, 4252, 4253, 4254, [4344, 5647, 5656, 6668, 8268] and [*no other standard*].

FCS_SSH_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods: [

- "password" (RFC 4252),
- "publickey" (RFC 4252): [
  - ssh-rsa (RFC 4253),
  - ecdsa-sha2-nistp256 (RFC 5656),
  - ecdsa-sha2-nistp384 (RFC 5656)]

] and no other methods.

FCS_SSH_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [*262,144 bytes*] in an SSH transport connection are dropped.

FCS_SSH_EXT.1.4 The TSF shall protect data in transit from unauthorized disclosure using the following mechanisms: [

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

] and no other mechanisms.

FCS_SSH_EXT.1.5 The TSF shall protect data in transit from modification, deletion, and insertion using: [

- hmac-sha2-256 (RFC 6668),
- hmac-sha2-512 (RFC 6668),
- implicit

] and no other mechanisms.

FCS_SSH_EXT.1.6 The TSF shall establish a shared secret with its peer using: [

- diffie-hellman-group14-sha256 (RFC 8268)

] and no other mechanisms.

FCS_SSH_EXT.1.7 The TSF shall use SSH KDF as defined in [

- RFC 4253 (Section 7.2),

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

FCS_SSH_EXT.1.8 The TSF shall ensure that [

- a rekey of the session keys,

] occurs when any of the following thresholds are met:

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

FCS_SSHS_EXT.1 SSH Protocol – Server

FCS_SSHS_EXT.1.1 The TSF shall authenticate itself to its peers (SSH Client) using: [

- ssh-rsa (RFC 4253),
- ecdsa-sha2-nistp256 (RFC 5656),
- ecdsa-sha2-nistp384 (RFC 5656)

].

5.3.3 User Data Protection (FDP)

FDP_HBI_EXT.1 Hardware-Based Isolation Mechanisms

FDP_HBI_EXT.1.1 The TSF shall use [[*logical domains*]] to constrain a Guest VM's direct access to the following physical devices: [[*CPU, memory, 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: [*CPU, memory, PCI Bus*].

FDP_PPR_EXT.1.2 The TSF shall explicitly deny all Guest VMs access to the following physical platform resources: [[*Integrated Lights Out Management (ILOM)*]].

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 to zeros upon 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 by default enforce a policy prohibiting sharing of data between Guest VMs.

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

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

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 Authentication Failure Handling

FIA_AFL_EXT.1.1 The TSF shall detect when [

- an administrator-configurable positive integer within a [*1 - 15*]]

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

FIA_AFL_EXT.1.2 When the defined number of unsuccessful authentication attempts has been met, the TSF shall: [prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password or PIN until [*an account unlock*] is taken by an Administrator, 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_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*].

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.

FIA_X509_EXT.1 X.509 Certificate Validation

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

- RFC 5280 certificate validation and certificate path validation
- The certificate path must terminate with a trusted certificate
- The TOE shall validate a certificate path by ensuring the presence of the basicConstraints extension, that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met.
- The TSF shall validate that any CA certificate includes caSigning purpose in the key usage field
- The TSF shall validate revocation status of the certificate using [a CRL as specified in RFC8603] with [no exceptions].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing Purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the EKU field.

- OCSP certificates presented for OCSP responses shall have the OCSP Signing Purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the EKU field.

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

FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [TLS].

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

5.3.5 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	Admin	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~~	O	O

Number	Function	Admin	User
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	N	N
14	Ability to configure screen lock inactivity timeout	N	N
15	Ability to configure remote connection inactivity timeout	X	N
16	Ability to configure lockout policy for unsuccessful authentication attempts through [limiting number of attempts during a time period]	X	N
17	[Not applicable]	N	N
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	N
23	Ability to stop/halt a VM	X	N
24	Ability to checkpoint a VM	N	N
25	Ability to suspend a VM	N	N
26	Ability to resume a VM	N	N
27	[Not applicable]	N	N

]

FMT_SMO_EXT.1 Separation of Management and Operational Networks

FMT_SMO_EXT.1.1 The TSF shall support the separation of management and operational network traffic through [separate physical networks, separate logical networks].

5.3.6 Protection of the TSF (FPT)

FPT_DVD_EXT.1 Non-Existence of Disconnected Virtual Devices

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

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) Address space randomization,

b) Memory execution protection (e.g., DEP),

c) Stack buffer overflow protection,

d) Heap corruption detection].

FPT_HAS_EXT.1 Hardware Assists

FPT_HAS_EXT.1.1 The VMM shall use [*None*] to reduce or eliminate the need for binary translation.

FPT_HAS_EXT.1.2 The VMM shall use [*None*] to reduce or eliminate the need for shadow page tables.

FPT_HCL_EXT.1 Hypercall Controls

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

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 [*PCI devices*] are switched between information domains, then [

e) 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 [automatic updates].

FPT_TUD_EXT.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a [digital signature mechanism not using certificates] 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 VMs

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.

5.3.7 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.

5.3.8 Trusted Path/Channel (FTP)

FTP_ITC_EXT.1 Trusted Channel Communication

FTP_ITC_EXT.1.1 The TSF shall use [

- TLS as conforming to the Functional Package for Transport Layer Security,
- SSH as conforming to the Functional Package for Secure Shell

] and [

- certificate-based authentication of the remote peer,

] to provide a trusted communication channel between itself, and

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

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 – Conformance
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)

6.1.1 Audit Data Generation (FAU_GEN.1)

29 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.

30 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.

31 Refer to section 3.2.4.3 of the Oracle VM Server for SPARC 3.6 and Oracle Solaris 11.4 Common Criteria Guide for material details on audit record formats.

32 By default, when audit records fill the available disk space, the system tracks the number of dropped audit records. A warning is issued when one percent of available disk space remains.

6.1.2 Audit Review (FAU_SAR.1)

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

6.1.3 Protected Audit Trail Storage (FAU_STG.1)

34 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)

35 The TOE forwards logs to a Syslog server in real-time via TLS as described in FCS_TLSC_EXT.1.

6.2 Cryptographic Support (FCS)

36 The TOE employs OpenSSL 3.0.8 (CAVP A4216) to provide the services described below.

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

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

a) RSA – 2048, 3072

b) ECDSA – P-256, P-384

c) FFC – Safe Primes

d) Diffie-Hellman group 14

38 The TOE supports the following key establishment schemes:

a) RSA based schemes

b) FFC based schemes / safe primes

c) Diffie-Hellman group 14

39 Table 14 identifies the scheme being used by each service.

Table 14: Key Generation/Establishment Mapping

Scheme	Usage	SFR	Service
RSA	Key Generation Key Establishment	FCS_TLSC_EXT.1	Logs
	Key Generation	FCS_SSHS_EXT.1	Remote Administration
ECDSA	Key Generation	FCS_SSHS_EXT.1	Remote Administration
FFC (safe primes)	Key Generation Key Establishment	FCS_TLSC_EXT.1	Logs
FFC - DH Group 14	Key Establishment	FCS_SSHS_EXT.1	Remote Administration

40 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.

41 For RSA-based key establishment, the TOE acts as a sender for TLS.

6.2.2 Key Destruction (FCS_CKM_EXT.4)

42 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.

43 For volatile memory, destruction is executed by removal of power to the memory.

44 For non-volatile memory the destruction consists of the invocation of an interface provided by the underlying platform that instructs the underlying platform to destroy the abstraction that represents the key.

Table 15: Key Destruction

Key	Generator / Initiator	Storage	Destruction
TLS Session Keys[1] (FCS_TLSC_EXT.1.1)	OpenSSL	Volatile	Removal of power

[1] Since the TOE only implements a TLS client without mutual authentication there are no persistent private keys within the scope of the TOE.

Key	Generator / Initiator	Storage	Destruction
SSH Private Keys (FCS_SSHS_EXT.1.4)	OpenSSL	Persistent	OS delete
		Volatile	Removal of power
SSH Session Keys (FCS_SSHS_EXT.1.3)	OpenSSL	Volatile	Removal of power
ZFS KEK	ZFS	Persistent	OS delete
ZFS DEK	ZFS	Persistent	Rendered unrecoverable upon destruction of the KEK.

6.2.3 Cryptographic Operation - Hashing (FCS_COP.1/Hash)

45 The TOE supports Cryptographic hashing services conforming to FIPS Pub 180-4. The hashing algorithms are used for HMAC services and digital signature verification for trusted updates.

46 The following hashing algorithms supported: SHA-1, SHA-256, SHA-384, and SHA-512.

47 The message digest sizes supported are: 160 bits, 256 bits, 384 bits, and 512 bits.

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

48 The TOE supports keyed hash algorithms: HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 used in TLS, SSH.

49 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-384	1024 bits	384 bits	384 bits
HMAC-SHA-512	1024 bits	512 bits	512 bits

6.2.5 Cryptographic Operation – Signature Algorithms (FCS_COP.1/Sig)

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

b) ECDSA digital signature algorithm conforming to FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5. The ECDSA supported curves are: P-256 and P-384.

51 Digital signatures are used in TLS (RSA) and SSH (RSA and ECDSA).

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

52 The TOE implements AES-CBC-128, AES-CBC-256, AES-GCM-128, and AES-GCM-256 in support of TLS. AES-CBC-128, AES-CBC-256, AES-GCM-128, AES-GCM-256, AES-CTR-128, and AES-CTR-256 are supported for SSH.

53 For ZFS, the TOE relies on a Data Encryption Key (DEK) which is wrapped using a Key Encryption Key (KEK). The TOE supports AES-GCM-128, AES-CCM-128, AES-GCM-256, and AES-CCM-256 for the DEK. AES-KW-128 and AES-KW-256 are supported for the KEK (depending on the size of the DEK it is encrypting).

6.2.7 Entropy for Virtual Machines (FCS_ENT_EXT.1)

54 The TOE provides a hardware-based entropy noise source to guest domains as a paravirtualized device, exposed as a hardware RNG. Guest VMs access this hardware device using the API described in Chapter 25 of the UltraSPARC Virtual Machine Specification document: https://sun4v.github.io/downloads/hypervisor-api-3.0draft7.pdf.

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 affect the entropy acquired by another VM.

6.2.8 Random Bit Generation (FCS_RBG_EXT.1)

56 The TOE leverages Hash_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.

57 The TOE implements the following DRBGs:

- Oracle Solaris Kernel Cryptographic Framework – Hash_DRBG
- OpenSSL 3.0.8 – Hash_DRBG

6.2.9 TLS Client Protocol (FCS_TLSC_EXT.1)

58 The TOE operates as a TLS client with the external syslog server.

59 The TOE 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_RSA_WITH_AES_256_CBC_SHA

c) TLS_RSA_WITH_AES_128_CBC_SHA256

d) TLS_RSA_WITH_AES_256_CBC_SHA256

e) TLS_RSA_WITH_AES_128_GCM_SHA256

f) TLS_RSA_WITH_AES_256_GCM_SHA384

g) TLS_DHE_RSA_WITH_AES_128_CBC_SHA256

h) TLS_DHE_RSA_WITH_AES_256_CBC_SHA256

i) TLS_DHE_RSA_WITH_AES_128_GCM_SHA256

j) TLS_DHE_RSA_WITH_AES_256_GCM_SHA384

60 The TLS client is capable of negotiating ciphersuites that include RSA and DHE key agreement schemes. Supported RSA key sizes are 2048 and 3072 bits. Safe primes are supported for DHE.

61 The TOE leverages X.509 certificates for establishing reference identifiers. DNS Name and IP address in the Subject Alternative Name (SAN) are supported in the evaluated configuration. Wildcards are supported. Certificate pinning is not supported.

6.2.10 SSH Protocol (FCS_SSH_EXT.1)

62 The TOE protects remote administrator communications via SSH with the following characteristics:

a) Public key (ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384) and password-based authentication is supported.

b) If the SSH packets are greater than 256KB they are automatically dropped.

c) Supported encryption algorithms: aes128-ctr, aes256-ctr, aes128-cbc, aes256-cbc, aes128-gcm@openssh.com, aes256-gcm@openssh.com

d) Supported MAC algorithms: hmac-sha2-256, hmac-sha2-512 and implicit (when using @openssh.com encryption).

e) Supported shared secret algorithms: diffie-hellman-group14-sha256.

f) Supported KDFs as per RFC 4253 (Section 7.2).

g) Connection re-keys occur after 1 hour of connection time, or after an aggregate of 1 gig of data has been transmitted or received (whichever occurs first).

63 **Note**: Password-based authentication is performed as per RFC 4252, section 8 (https://www.ietf.org/rfc/rfc4252.html#section-8). The TOE sends SSH_MSG_USERAUTH_REQUEST messages with the method set to “password”.

6.3 User Data Protection (FDP)

6.3.1 Hardware-Based Isolation Mechanisms (FDP_HBI_EXT.1)

64 Guest VMs do not have direct or unmediated access to physical resources. When a VM is provisioned, it is assigned to a logical domain which provides access to the allocated CPU, memory, and PCI devices.

65 Additional information on hypervisor architecture, privilege and isolation mechanisms can be found under Chapters 1.2 and 1.3 can be found in the following document: https://sun4v.github.io/downloads/hypervisor-api-3.0draft7.pdf.

6.3.2 Physical Platform Resource Controls (FDP_PPR_EXT.1)

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

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

a) CPU

b) Memory

c) PCI Bus

68 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 by assigning the PCI end-device to the corresponding PCI bus ID, and then assigning the bus ID to the domain. Additional details on creating a Root domain and assigning PCI buses can be found at: https://docs.oracle.com/cd/E93612_01/html/E93617/rootdomainwithpcibuses.html.

69 CPU and memory are not allocated via PCI, but are assigned to the logical domain. TOE configuration also does not allow for ILOM to be assigned to PCI buses or domains.

6.3.3 Residual Information in Memory (FDP_RIP_EXT.1)

70 The TOE clears memory prior to allocation to a Guest VM. Memory is cleared by the hypervisor as it is being allocated to a guest. There are no conditions under which newly allocated memory would not get cleared.

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

6.3.4 Residual Information on Disk (FDP_RIP_EXT.2)

72 The TOE makes use of virtual disks for VM storage. Virtual disks are zeroed upon creation. A VM may be attached to a shared virtual disk, in which case, the disk is not zeroed prior to allocation.

73 The TOE only supports ZFS storage. 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)

74 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.

6.3.6 Virtual Networking Components (FDP_VNC_EXT.1)

75 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)

76 The TOE allows the administrator to configure a login policy that locks a user's account after a configured number of failed authentication attempts. A locked account may be unlocked by the administrator or configured to unlock after a defined time period.

6.4.2 Password Management (FIA_PMG_EXT.1)

77 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 "!", "@", "#", "$", "%", "^", "&", "*", "(", ")".

78 The minimum password length is configurable by the administrator.

6.4.3 Multiple Authentication Mechanisms (FIA_UAU.5)

79 The TOE supports the following authentication mechanisms:

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

6.4.4 Administrator Identification and Authentication (FIA_UIA_EXT.1)

80 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.

81 Administrators must be successfully authenticated before being able to perform any management and configuration activities on the TOE. 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.

6.4.5 X.509 Certificates (FIA_X509_EXT.1, FIA_X509_EXT.2)

82 When an X.509 certificate is presented, the TOE verifies the certificate path, and certification validation process by verifying the following rules:

a) RFC 5280 certificate validation and certificate path validation.

b) The certificate path must terminate with a trusted CA certificate.

c) The OS shall validate a certificate path by ensuring the presence of the basicConstraints extension, that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met.

d) The TSF shall validate that any CA certificate includes caSigning purpose in the key usage field

83 The OS shall validate the extendedKeyUsage field according to the following rules:

a) Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.

b) Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the EKU field.

84 A Security Administrator must configure the TOE to use CRL for revocation checking. When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall not accept the certificate.

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

86 The TOE uses X.509v3 certificates for TLS connections. All X.509 certificates are maintained within the trust store, found under the `/etc/certs/CA` directory.

6.5 Security Management (FMT)

6.5.1 Management of Security Functions Behaviour (FMT_MOF_EXT.1)

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

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

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

6.6 Protection of the TSF (FPT)

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

89 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)

90 The TOE makes use of the Solaris-provided environment-based vulnerability mitigation mechanisms:

a) Address space randomization

b) Memory execution protection

c) Stack buffer overflow protection

d) Heap corruption detection

91 In addition, the Data Execution Prevention (DEP) feature prevents an application or service from executing code in a non-executable memory region.

92 For the configuration and management domain, the above mechanisms are provided by Solaris. User input that may impact the operation, configuration, or deployment of guest VMs can only be executed from this space. The TOE enforces vulnerability mitigation mechanisms for guest VMs.

6.6.3 Hardware Assists (FPT_HAS_EXT.1)

93 The TOE does not perform binary translations or use shadow page tables. Each guest VM has an independent set of physical memory translations managed by the Hypervisor.

6.6.4 Hypercall Controls (FPT_HCL_EXT.1)

94 Hypercalls are enabled by default and cannot be disabled. The TOE supports the hypercalls as documented in the following document: https://sun4v.github.io/downloads/hypervisor-api-3.0draft7.pdf. A summary of

supported hypercalls is presented in Table A.2 of section A5. Chapters 11-27 and 31 describe the supported hypercalls in detail.

6.6.5 Removable Devices and Media (FPT_RDM_EXT.1)

95 The TOE Administrator controls access to removable media, whether physical or virtual, by means of explicit configuration. Access to physical or virtual media is controlled by allocating the PCI bus assigned to that domain. Removable physical media applies to USB storage devices, floppy drives and physical CD-ROM (or DVD) 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)

96 The TOE software is packaged with Solaris updates. The Oracle Solaris Image Packaging System (IPS) is a framework that enables the following tasks:

a) List and search software packages

b) Install, update, and remove software

c) Upgrade to a new Oracle Solaris operating system release

97 The IPS interface allows administrators to restrict which packages can be installed.

98 Oracle Solaris software is distributed in IPS packages. IPS packages are stored in IPS package repositories, which are populated by IPS publishers. IPS packages are installed into Oracle Solaris images. The Text Installer image will have pkg.oracle.com/solaris/release by default. Most customers don't deploy with that - they use Automated Installer instead. Then publishers are in the install image configuration file. These point to an internal copy of the repository inside the customer’s own network.

99 Oracle Solaris provides the *pkg update* command (and other *pkg* commands) to check for updates to itself and installed applications (software packages).

100 X.509 certificates are used as a container in which public keys used to verify the digital signatures of software packages are stored. Software packages/updates are bundled with publisher metadata that includes a 2048-bit RSA digital signature of the package data. These signatures are generated using Oracle-issued private keys. Package signatures are verified prior to package installation. Failure to validate a package signature will prevent the package from being installed. Package installation takes place only after successful validation of the package signature. Public keys used to verify package signatures from authorized sources are stored in the TOE’s trust store within X.509 certificates. By default, the TOE comes with Oracle public keys pre-installed under /etc/certs/CA.

6.6.7 Virtual Device Parameters (FPT_VDP_EXT.1)

101 Parameters passed from Guest VMs to virtual device interfaces are thoroughly validated and all illegal values 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.

102 All devices are exposed as PCI devices where presence of appropriate PCI identifying information determines presence of a device. Details on PCI bus

interfaces can be found at the following link: https://docs.oracle.com/cd/E93612_01/html/E93617/rootdomainwithpcibuses.html.

103 The following virtual devices are supported in the evaluated configuration:

- Virtual Disk (vDisk)
- VLAN (vnet / vswitch)
- Virtual SCSI HBA (vHBA)

104 See the SPARC proprietary document for a list of ports, parameters, and legal values.

6.6.8 VMM Isolation from VMs (FPT_VIV_EXT.1)

105 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.

106 The TOE has no BIOS, but instead provides platform firmware which is updated from the ILOM or control domain. The ILOM is a separate part of the system with its own CPU and memory, inside the physical chassis, running its own embedded OS. Only this control domain has the ability to reconfigure platform resources (CPU, memory, PCI).

6.7 TOE Access (FTA)

6.7.1 TOE Access Banner (FTA_TAB.1)

107 Access banners may be configured for the SSH CLI.

6.8 Trusted Path/Channel (FTP)

6.8.1 Trusted Channel Communications (FTP_ITC_EXT.1)

108 The TOE implements the following trusted channels:

a) TLS for syslog

b) SSH for the CLI

6.8.2 Trusted Path (FTP_TRP.1)

109 The implements the following trusted paths:

a) SSH for the CLI

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

110 The TOE supports keyboard over SSH for user input devices.

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

111 VMs are assigned a unique name when they are created. A VM cannot be created using an existing name. This name is displayed to users of the VM in the CLI, in which the VM is running.

Rationale

6.9 Conformance Claim Rationale

112 The following rationale is presented with regards 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, MOD_SV, PKG_SSH, PKG_TLS and subsequent TDs defined in Table 2. No additional requirements have been specified.

6.10 Security Objectives Rationale

113 Table 17 provides a coverage mapping between the security objectives, threats, OSPs and assumptions:

Table 17: Security Objectives Mapping

	T.DATA_LEAKAGE	T.UNAUTHORIZED_UPDATE	T.UNAUTHORIZED_MODIFICATION	T.USER_ERROR	T.3P_SOFTWARE	T.VMM_COMPROMISE	T.PLATFORM_COMPROMISE	T.UNAUTHORIZED_ACCESS	T.WEAK_CRYPTO	T.UNPATCHED_SOFTWARE	T.MISCONFIGURATION	T.DENIAL_OF_SERVICE	A.NON_MALICIOUS_USER	A.PLATFORM_INTEGRITY	A.PHYSICAL	A.TRUSTED_ADMIN
O.VM_ISOLATION	X															
O.DOMAIN_INTEGRITY	X															
O.VMM_INTEGRITY		X	X		X	X										
O.AUDIT			X													
O.VM_ISOLATION				X		X										
O.PLATORM_INTEGRITY							X									

	T.DATA_LEAKAGE	T.UNAUTHORIZED_UPDATE	T.UNAUTHORIZED_MODIFICATION	T.USER_ERROR	T.3P_SOFTWARE	T.VMM_COMPROMISE	T.PLATFORM_COMPROMISE	T.UNAUTHORIZED_ACCESS	T.WEAK_CRYPTO	T.UNPATCHED_SOFTWARE	T.MISCONFIGURATION	T.DENIAL_OF_SERVICE	A.NON_MALICIOUS_USER	A.PLATFORM_INTEGRITY	A.PHYSICAL	A.TRUSTED_ADMIN
O.MANAGEMENT_ACCESS								X								
O.VM_ENTROPY									X							
O.PATCHED_SOFTWARE										X						
O.CORRECTLY_APPLIED_CONFIGURATION											X					
O.RESOURCE_ALLOCATION												X				
OE.NON_MALICIOUS_USER													X			
OE.CONFIG													X			
OE.PHYSICAL														X	X	
OE.TRUSTED_ADMIN																X

114 Table 18 provides the justification to show that the security objectives are suitable to address the security problem.

Table 18: 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.

Threat, Assumption, or OSP	Security Objective	Rationale
T.UNAUTHORIZED _UPDATE	O.VMM_INTEGRITY	System integrity prevents the TOE from installing a software patch containing unknown and potentially malicious code. Integrity of a Virtualization System can be maintained by ensuring that the only way to modify the VS is through a trusted update process initiated by an authorized Administrator as required by FMT_MOF_EXT.
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.
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.

Threat, Assumption, or OSP	Security Objective	Rationale
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. Access to management functions must be limited to authorized Administrators as managed through controls required by FMT_MOF_EXT.1.
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.
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.PHYSICAL	If the underlying platform has not been compromised prior to installation of the TOE, its integrity can be assumed to be intact.

Threat, Assumption, or OSP	Security Objective	Rationale
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.

6.11 Security Requirements Rationale

6.11.1 SAR Rationale

115 All security requirements are drawn directly from the claimed Base_PP, MOD_SV, PKG_SSH, and PKG_TLS and are consistent with the principle of exact conformance.