IBM PowerVM 1060 with VIOS 4.1.1 for Server for POWER10 and HMC for POWER9 Security Target

Version:	**2.0**
Status:	**released**
Last Update:	**2025-07-21**
Classification:	**Public**
Authors:	**atsec corporation**

Revision History

Version	Date	Author(s)	Changes to Previous Revision
1.0	2024-05-31	Trang Huynh	First official draft
2.0	2025-07-21	Trang Huynh, atsec	Public version

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Security Target Identification

Title:	IBM PowerVM 1060 with VIOS 4.1.1 for Server for POWER10 and HMC for POWER9 Security Target
Version:	2.0
Status:	released
Date:	2025-07-21
Sponsor:	International Business Machines Corporation
Developer:	IBM Corporation
Certification Body:	OCSI
Certification ID:	OCSI\CERT\ATS\14\2024
Keywords:	PowerVM, HMC, VIOS, virtualization, hypervisor, partition

1.2 TOE Identification

The TOE is IBM PowerVM 1060 with VIOS 4.1.1 for Server for POWER10 and HMC for Power9.

1.3 TOE Type

The TOE type is a hypervisor with a virtual input/output system and a hardware management console subsystem.

1.4 TOE Overview

The TOE is the IBM Power Virtual Machine (PowerVM) Firmware (FW) 1060 with Virtual Input/Output System (VIOS) 4.1.1 Server for POWER10 and Hardware Management Console (HMC) for POWER9 provided by International Business Machines (IBM) Corporation. The TOE facilitates the sharing of hardware resources by disparate applications (e.g., AIX, Linux). The TOE includes the guidance documentation.

The PowerVM product is based on the concept of a 'hypervisor' that is designed to instantiate 'partitions', each with its own distinct resources, that each appear to their hosted applications as a completely functional underlying platform. These partitions, known as logical partitions (LPARs), are implemented to prevent interference among partitions and to prevent simultaneous sharing of storage and other device resources. VIOS allows partitions access-controlled sharing of individual storage and network devices. The TOE is agnostic to the application running in a LPAR. Both PowerVM and VIOS are configured and managed by the Hardware Management Console (HMC) is a hardware console used for configuration and management of PowerVM and VIOS. HMC provides functionality necessary for administrative personnel to manage the allocation of resources to the configured partitions.

1.4.1 TOE Usage

The TOE is the IBM PowerVM Firmware 1060 with VIOS 4.1.1 Server for POWER10 and HMC for POWER9. While PowerVM performs virtualization of the Central Processing Units (CPUs) and memory space, VIOS performs virtualization of storage and network devices. PowerVM supports assigning individual physical storage or network devices to a partition, but it does not support sharing of physical storage and network devices between partitions. Thus, in a PowerVM-only model, a hardware platform running 100 partitions that all desire access to the Internet would need 100 network devices. By running VIOS in a partition, PowerVM can, for example, assign a small number of physical network devices to the VIOS partition and VIOS can provide virtualized networking to the 100 partitions through controlled sharing of the physical network devices; thus, significantly reducing the number of physical network devices required. A similar VIOS analogy applies to storage devices.

In addition and included as part of the TOE definition, is a directly connected HMC that provides access to the functions necessary to enable administrative personnel to effectively manage the allocation of resources (i.e., processors, memory, and I/O devices) to the configured partitions.

To configure and manage the TOE, one of the following management consoles (MCs) is required.

- Hardware Management Console (HMC)—Hardware console used for configuration and management of PowerVM and VIOS.
- Novalink—Software interface used for virtualization management and configuration for PowerVM.
- Power Virtualization Control (PowerVC)—Advanced virtualization and cloud management offering used for the management and configuration of PowerVM.
- Virtual Hardware Management Console (vHMC)—Software console used for the configuration and management of PowerVM and VIOS.

The TOE was evaluated on the following TOE hardware.

- HMC running on IBM POWER9

1.4.2 TOE security features

The TOE supports the following major security features.

- Auditing
- Cryptographic support
- User data protection
- Identification and authentication
- Security management
- Protection of the TOE security functionality (TSF)
- TOE access
- Trusted path/channels

1.4.3 Non-TOE Hardware and Software

The following non-TOE software abstraction is required for VIOS.

- Open Firmware/Run-Time Abstraction (OF/RTA)—Support for the AIX (including VIOS) and Linux operating systems

1.5 TOE Description

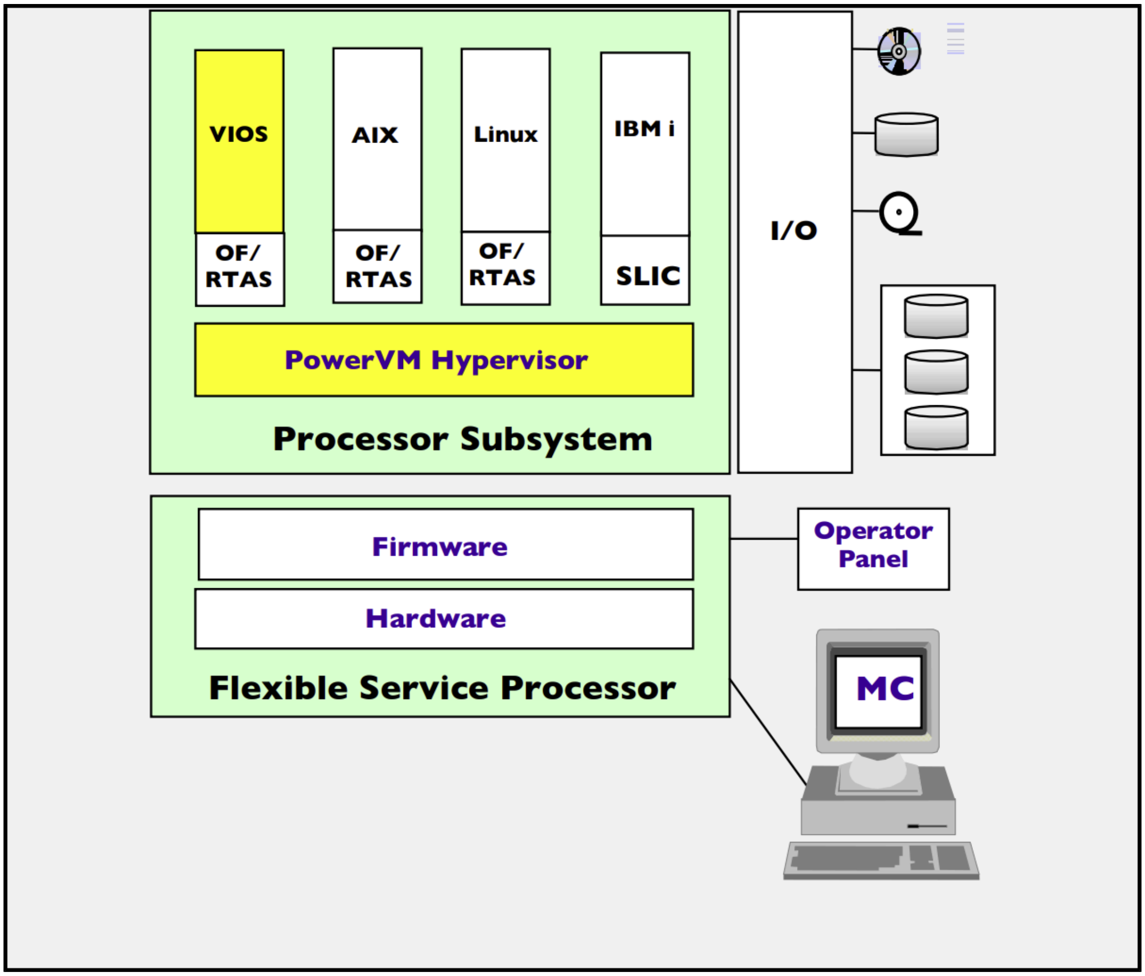
The TOE is the IBM Power Virtual Machine (PowerVM) Firmware (FW) 1060 with Virtual Input/Output System (VIOS) 4.1.1 Server for POWER10 and HMC for POWER9. While the TOE is designed to generally support the entire line of IBM Power Systems products, it has been evaluated and tested on the models shown in section 1.4.1.

Similarly, while the TOE is designed to support multiple storage device types, only virtual Ethernet (vENT) and virtual Small Computer System Interface (vSCSI) devices have been evaluated and tested

1.5.1 TOE Architecture

The TOE comprises the PowerVM Hypervisor (PHYP), VIOS, and HMC as indicated by the yellow highlighting in Figure 1.

Figure 1: TOE architecture



1.5.1.1 Physical Boundaries

Figure 1 identifies the TOE components in the yellow-filled boxes. The other components (e.g. SLIC, and operating systems) are outside the scope of the TOE.

The TOE images are downloadable from the developer’s website over an HTTPS connection. The TOE software is comprised of the following images.

- PowerVM:
  - POWER10: 01ML1060_064_053 (FW1060.10)
- VIOS:
  - Virtual_IO_Server_Base_Install_4.1.1.0_DVD_122024_122024_LCD8298701.iso

- HMC:
  - POWER9: Hardware Management Console version 10.3.1062.1
- The TOE guidance is contained in the document: IBM PowerVM 1060.10 with VIOS 4.1.1 for Power10 and HMC 10.3.1062.1 for Power9 Evaluated Configuration Guide

As indicated earlier, the TOE consists of multiple architectural components. The components expose several interfaces both externally and internally.

The external interfaces include the interfaces to the subject operating in a partition. These include the Hypervisor interfaces as well as the hardware instructions available to applications. There is also an operator panel where basic, non-security related operator functions can be performed by a user with direct physical access to the TOE.

The internal interfaces, specifically those not also available externally, include the Flexible Service Processor (FSP) interface to the Hypervisor.

I/O represents the physical I/O slots either integrated into the hardware drawers or I/O drawers external to the server. The I/O adapters allow for the connection of disk, network, Storage Area Network (SAN), tape, and other individual I/O devices. The physical boundaries can then be broken down into individual logical components. For example, a physical drawer may contain 8 different I/O devices and these individual devices are assigned by the HMC to the configured virtual machines (partitions).

Note that connections to a broad or public network are supported, but they would be treated as resources that can be granted to partitions for operating system use but would not be used by the TOE for its own purposes. Along these lines, while the TOE controls which devices a given partition can access, it does not control or otherwise constrain the nature of those devices. Any functions or connections of those devices are outside the scope of control of the TOE.

1.5.1.2 Logical Boundaries

When assigned to a partition, the logical I/O devices are available to be used by the partition (e.g., disk, network, tape).

This section summarizes the security functions provided by the TOE.

- Auditing
- Cryptographic support
- User data protection
- Identification and authentication
- Security management
- Protection of the TOE security functionality(TSF)
- TOE access
- Trusted path/channels

1.5.1.2.1 Auditing

The TOE provides an audit capability that allows audit records to be generated for security critical events specified in FAU_GEN.1. The audit records are generated through the HMC and VIOS and stored locally as well as transferring to an external audit server protected via SSH. The VIOS records audit events pertaining to connections between the VMs and virtual or physical networking components. The HMC records all audit events listed in FAU_GEN.1.

Audit records are viewable by authorized administrators and are protected from unauthorized modification and deletion.

1.5.1.2.2 Cryptographic Support

The TOE provides cryptographic support using OpenSSL, OpenSSH, and Java cryptographic modules implemented in the TOE.

The TOE uses the OpenSSL version 1.1.1k cryptographic module on the HMC for the following services:

- RSA key generation for X.509 certificates and user and peer authentication
- RSA key generation for SSH user and peer authentication
- HMC trusted updates using published hash

The TOE uses the OpenSSL version 3.1.3 cryptographic module on the Server and the Baseboard Management Controller (BMC) for the following services:

- Server trusted updates using RSA digital signatures

The TOE uses the Java version 17.0.10 cryptographic module for the following services:

- Trusted paths/channels for incoming connections from the remote administrator through HMC GUI to the HMC over HTTPS/TLS 1.2.
- RSA key generation for use by RSA and ECC key establishment schemes in the TLS 1.2 protocol

The TOE uses the OpenSSH version 8.0p1-19.el8.2 cryptographic library for the following services:

- Trusted channel for incoming and outgoing connections to the external audit storage via Secure File Transfer Protocol (SFTP) using the Secure Shell version 2 (SSHv2) protocol.
- Trusted path for incoming connection from the remote administrator through HMC Command Line Interface (CLI) to the HMC using the Secure Shell version 2 (SSHv2) protocol.
- ECC key generation for use by EC-Diffie-Hellman in the SSHv2 protocol.
- RSA key-based authentication

1.5.1.2.3 User Data Protection

Hypervisor

The Hypervisor manages the association of CPUs, memory, and I/O devices, in a relatively static environment, with partitions containing operating system instances. Memory and I/O devices can be assigned to single partitions and when assigned are accessible only by the partition (including OF/ RTAS (Run Time Abstraction Services) and the OS running in the partition). CPUs can also be assigned a single partition, and only that partition (and occasionally the TOE) can use that CPU. CPUs can also be configured to be shared among a collection of partition (shared processor partition or also called micropartitions) and the Hypervisor will save/restore the hardware register state when switching between partitions.

Partitions have no control over the resources they are assigned. The Hypervisor receives the partition management information from the HMC when it is being configured. Once configured, the configured values are continuously enforced.

VIOS

VIOS manages the association of partitions to virtualized storage and network devices and the association of virtualized storage and network devices to physical storage and network devices. Through the HMC, an administrator assigns a set of physical storage and network devices to the VIOS partition. The administrator then creates virtual storage and network devices in VIOS, maps the physical devices to the virtualized devices, and maps the vSCSI (virtual Small Computer Serial Interface) and vENT (virtual Ethernet) to other partitions on the system. These other partitions access the virtualized storage and virtual networking controlled by VIOS. VIOS provides the separation protection between the virtualized storage and virtual network devices so that one partition cannot access another partitions information.

1.5.1.2.4 Identification and authentication

Partitions are implicitly identified by internal numerical identifiers associated with partitions (using internal data structures) as they are defined. Being implicitly identified by the TOE, partitions have no need, nor means, to identify themselves. Furthermore, the identification of a partition is guaranteed by the Hypervisor.

The Hypervisor identifies administrators for configuring and managing partitions and VIOS devices. Administrators use the HMC console to configure and manage the TOE. Administrators can connect to the HMC console via the web-based GUI over HTTPS or via the CLI over SSH. In the evaluated configuration, only local user authentication is supported.

1.5.1.2.5 Security Management

All of the TOE configuration and management occurs via the interface to the HMC console. Administrators can configure and manage the security functions used by the TOE. The TOE supports the separation of management and operational network traffic through separate physical and logical network.

1.5.1.2.6 Protection of the TOE Security Functionality (TSF)

The components of the TOE protect themselves using the domains provided by the POWER processors. The Hypervisor operates in the privileged domain and the partitions, like VIOS, operate in the unprivileged domain. This allows the Hypervisor to protect itself as well as the resources it makes selectively available to the applicable partitions.

Beyond protecting itself and its resources, the TOE is also designed such that when the hardware that supports a partition fails, the other partitions will continue uninterrupted.

Additionally, the TOE provides trusted software updates via a published hash (HMC) and digital signature (Server).

1.5.1.2.7 TOE Access

The TOE provides the capability of displaying of an advisory warning message regarding unauthorized use of the TOE before establishing an administrator session (i.e., the HMC Console).

1.5.1.2.8 Trusted Path/Channels

The TOE provides protected communications between itself and the following external entities.

- Connections using HTTPS/TLS
  - Connection between the remote administrator and the HMC through the HMC GUI
- Connections using SSH
  - Connection between the remote administrator and the HMC through the HMC CLI
  - Connection between the HMC and the external audit storage over SFTP

2 CC Conformance Claim

This Security Target is CC Part 2 extended and CC Part 3 extended. Common Criteria [CC] version 3.1 revision 5 is the basis for this conformance claim.

This Security Target claims exact conformance to the "PP-Configuration for Virtualization and Server Virtualization Systems," version 1.0 [CFG_Virtualization-SV_V1.0]☝. The PP-Configuration includes:

- [PP_BASE_VIRTUALIZATION_V1.1]☝: Protection Profile for Virtualization. Version 1.1 as of 2021-06-14
- [MOD_SV_V1.1]☝: PP-Module for Server Virtualization Systems. Version 1.1 as of 2021-06-14

In addition, this ST claims exact conformance to the following functional packages:

- [PKG_SSH_V1.0]☝: Functional Package for Transport Layer Security (TLS). Version 1.1 as of 2019-03-01.
- [PKG_TLS_V1.1]☝: Functional Package for Secure Shell (SSH). Version 1.0 as of 2021-05-13.

Table 1 below contains the NIAP Technical Decisions (TDs) for the [PP_BASE_VIRTUALIZATION_V1.1]☝ protection profile at the time of the evaluation and a statement of applicability to the evaluation.

Table 1: NIAP Technical Decisions for [PP_BASE_VIRTUALIZATION_V1.1]

TD #	Description	Applicable?	Non-applicability rationale
TD0936	Clarification when CTR_DRBG is Selected for FCS_RBG_EXT.1.2 in PP_BASE_VIRTUALIZATION_V1.1	Yes	
TD0905	Updates to Certificate Revocation (FIA_X509_EXT.1) for Base Virtualization PP v1.1	Yes	
TD0874	Updating FIPS 186-4 to 186-5 in PP_BASE_VIRTUALIZATION_V1.1	Yes	
TD0844	Addition of Assurance Package for Flaw Remediation V1.0 Conformance Claim	No	The ST does not claim conformance to this Assurance Package.
TD0814	Correction to Mixed content in TSS AAs	Yes	
TD0721	Mapping FTA_TAB.1 to Objective	Yes	
TD0615	Audit generation for hypercalls implemented in HW	Yes	

There are no NIAP Technical Decisions (TDs) for [MOD_SV_V1.1]☝.

Table 2 contains the NIAP Technical Decisions (TDs) for the [PKG_TLS_V1.1]☝ functional package at the time of the evaluation and a statement of applicability to the evaluation.

Table 2: NIAP Technical Decisions for [PKG_TLS_V1.1]

TD #	Description	Applicable?	Non-applicability rationale
TD0779	Updated Session Resumption Support in TLS package V1.1	Yes	
TD0770	TLSS.2 connection with no client cert	No	The ST does not claim FCS_TLSS_EXT.2.
TD0739	PKG_TLS_V1.1 has 2 different publication dates	Yes	
TD0726	Corrections to (D)TLSS SFRs in TLS 1.1 FP	Yes	
TD0513	CA Certificate loading	Yes	

TD #	Description	Applicable?	Non-applicability rationale
TD0499	Testing with pinned certificates	Yes	
TD0469	Modification of test activity for FCS_TLSS_EXT.1.1 test 4.1	Yes	
TD0442	Updated TLS Ciphersuites for TLS Package	Yes	

Table 3 contains the NIAP Technical Decisions (TDs) for the [PKG_SSH_V1.0]☝ functional package at the time of the evaluation and a statement of applicability to the evaluation.

Table 3: NIAP Technical Decisions for [PKG_SSH_V1.0]

TD #	Description	Applicable?	Non-applicability rationale
TD0909	Updates to FCS_SSH_EXT.1.1 App Note in SSH FP 1.0	No.	The TOE does not support the "server-sig-algs" extension.
TD0777	Clarification to Selections for Auditable Events for FCS_SSH_EXT.1	Yes	
TD0732	FCS_SSHS_EXT.1.3 Test 2 Update	Yes	
TD0695	Choice of 128 or 256 bit size in AES-CTR in SSH Functional Package.	Yes	
TD0682	Addressing Ambiguity in FCS_SSHS_EXT.1 Tests	Yes	

3 Security Problem Definition

The Security Problem Definition describes the security aspects of the intended environment in which the TOE is to be used and the way it is expected to be employed. The statement of the Security Problem Definition defines the following:

- Threats that the TOE counters
- Assumptions made about the operational environment and the intended method of use for the TOE

3.1 Threat Environment

3.1.1 Threats countered by the TOE

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.

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.

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

3.2.1 Intended usage of the TOE

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.

4 Security Objectives

This section defines the security objectives for the TOE and its supporting environment. The security objectives are intended to counter identified threats and address applicable assumptions.

4.1 Objectives for the TOE

O.VM_ISOLATION

VMs are the fundamental subject of the system. The VMM is responsible for applying the system security policy (SSP) to the VM and all resources. As basic functionality, the VMM must support a security policy that mandates no information transfer between VMs.

The VMM must support the necessary mechanisms to isolate the resources of all VMs. The VMM partitions a platform's physical resources for use by the supported virtual environments. Depending on customer requirements, a VM may need a completely isolated environment with exclusive access to system resources or share some of its resources with other VMs. It must be possible to enforce a security policy that prohibits the transfer of data between VMs through shared devices. When the platform security policy allows the sharing of resources across VM boundaries, the VMM must ensure that all access to those resources is consistent with the policy. The VMM may delegate the responsibility for the mediation of resource sharing to select Service VMs; however in doing so, it remains responsible for mediating access to the Service VMs, and each Service VM must mediate all access to any shared resource that has been delegated to it in accordance with the SSP.

Both virtual and physical devices are resources requiring access control. The VMM must enforce access control in accordance with system security policy. Physical devices are platform devices with access mediated via the VMM per the O.VMM_Integrity objective. Virtual devices may include virtual storage devices and virtual network devices. Some of the access control restrictions must be enforced internal to Service VMs, as may be the case for isolating virtual networks. VMMs may also expose purely virtual interfaces. These are VMM specific, and while they are not analogous to a physical device, they are also subject to access control.

The VMM must support the mechanisms to isolate all resources associated with virtual networks and to limit a VM's access to only those virtual networks for which it has been configured. The VMM must also support the mechanisms to control the configurations of virtual networks according to the SSP.

O.VMM_INTEGRITY

Integrity is a core security objective for Virtualization Systems. To achieve system 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.

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 Objectives for the Operational Environment

OE.CONFIG

TOE administrators will configure the VS correctly to create the intended security policy.

OE.PHYSICAL

Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.

OE.TRUSTED_ADMIN

TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

OE.NON_MALICIOUS_USER

Users are trusted to not be willfully negligent or hostile and use the VS in compliance with the applied enterprise security policy and guidance.

4.3 Security Objectives Rationale

4.3.1 Coverage

The following table provides a mapping of TOE objectives to threats and policies, showing that each objective counters or enforces at least one threat or policy, respectively.

Table 4: Mapping of security objectives to threats and policies

Objective	Threats / OSPs
O.VM_ISOLATION	T.DATA_LEAKAGE T.USER_ERROR T.VMM_COMPROMISE
O.VMM_INTEGRITY	T.UNAUTHORIZED_UPDATE T.UNAUTHORIZED_MODIFICATION T.3P_SOFTWARE T.VMM_COMPROMISE
O.PLATFORM_INTEGRITY	T.PLATFORM_COMPROMISE
O.DOMAIN_INTEGRITY	T.DATA_LEAKAGE
O.MANAGEMENT_ACCESS	T.UNAUTHORIZED_ACCESS
O.PATCHED_SOFTWARE	T.UNPATCHED_SOFTWARE
O.VM_ENTROPY	T.WEAK_CRYPTO
O.AUDIT	T.UNAUTHORIZED_MODIFICATION
O.CORRECTLY_APPLIED_CONFIGURATION	T.MISCONFIGURATION
O.RESOURCE_ALLOCATION	T.DENIAL_OF_SERVICE

The following table provides a mapping of the objectives for the Operational Environment to assumptions, threats and policies, showing that each objective holds, counters or enforces at least one assumption, threat or policy, respectively.

Table 5: Mapping of security objectives for the Operational Environment to assumptions, threats and policies

Objective	Assumptions / Threats / OSPs
OE.CONFIG	A.NON_MALICIOUS_USER
OE.PHYSICAL	A.PLATFORM_INTEGRITY A.PHYSICAL
OE.TRUSTED_ADMIN	A.TRUSTED_ADMIN
OE.NON_MALICIOUS_USER	A.NON_MALICIOUS_USER

4.3.2 Sufficiency

The following rationale provides justification that the security objectives are suitable to counter each individual threat and that each security objective tracing back to a threat, when achieved, actually contributes to the removal, diminishing or mitigation of that threat.

Table 6: Sufficiency of objectives countering threats

Threat	Rationale for Security Objectives
T.DATA_LEAKAGE	Logical separation of VMs and enforcement of domain integrity prevent unauthorized transmission of data from one VM to another. Logical separation of VMs and enforcement of domain integrity prevent unauthorized transmission of data from one VM to another.
T.UNAUTHORIZED_UPDATE	System integrity prevents the TOE from installing a software patch containing unknown and potentially malicious code.
T.UNAUTHORIZED_MODIFICATION	Enforcement of VMM integrity prevents the bypass of enforcement mechanisms and auditing ensures that abuse of legitimate authority can be detected. Enforcement of VMM integrity prevents the bypass of enforcement mechanisms and auditing ensures that abuse of legitimate authority can be detected.
T.USER_ERROR	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	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	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. 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	Platform integrity mechanisms used by the TOE reduce the risk that an attacker can 'break out' of a VM and affect the platform on which the VS is running.
T.UNAUTHORIZED_ACCESS	Ensuring that TSF management functions cannot be executed without authorization prevents untrusted subjects from modifying the behavior of the TOE in an unanticipated manner.
T.WEAK_CRYPTO	Acquisition of good entropy is necessary to support the TOE's security-related cryptographic algorithms.
T.UNPATCHED_SOFTWARE	The ability to patch the TOE software ensures that protections against vulnerabilities can be applied as they become available.
T.MISCONFIGURATION	Mechanisms to prevent the application of configurations that violate the current security policy help prevent misconfigurations.

Threat	Rationale for Security Objectives
T.DENIAL_OF_SERVICE	The ability of the TSF to ensure the proper allocation of resources makes denial of service attacks more difficult.

The following rationale provides justification that the security objectives for the environment are suitable to cover each individual assumption, that each security objective for the environment that traces back to an assumption about the environment of use of the TOE, when achieved, actually contributes to the environment achieving consistency with the assumption, and that if all security objectives for the environment that trace back to an assumption are achieved, the intended usage is supported.

Table 7: Sufficiency of objectives holding assumptions

Assumption	Rationale for Security Objectives
A.PLATFORM_INTEGRITY	If the underlying platform has not been compromised prior to installation of the TOE, its integrity can be assumed to be intact.
A.PHYSICAL	If the TOE is deployed in a location that has appropriate physical safeguards, it can be assumed to be physically secure.
A.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.
A.NON_MALICIOUS_USER	If the organization properly vets and trains users, it is expected that they will be non-malicious. 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.

5 Extended Components Definition

All extended components are taken directly from [PP_BASE_VIRTUALIZATION_V1.1]☝, [MOD_SV_V1.1]☝, [PKG_TLS_V1.1]☝, and [PKG_SSH_V1.0]☝. The following table lists the extended components included in this ST:

Table 8: Extended Components

Functional class	Functional components
Security Audit (FAU)	FAU_STG_EXT.1 Off-Loading of Audit Data
Cryptographic Support (FCS)	FCS_CKM_EXT.4 Cryptographic Key Destruction
	FCS_ENT_EXT.1 Entropy for Virtual Machines
	FCS_HTTPS_EXT.1 HTTPS Protocol
	FCS_RBG_EXT.1/HMC and FCS_RBG_EXT.1/SVR Cryptographic Operation (Random Bit Generation)
	FCS_SSH_EXT.1 SSH Protocol
	FCS_SSHC_EXT.1 SSH Protocol - Client
	FCS_SSHS_EXT.1 SSH Protocol - Server
	FCS_TLS_EXT.1 TLS Protocol
	FCS_TLSS_EXT.1 TLS Server Protocol
User Data Protection (FDP)	FDP_HBI_EXT.1 Hardware-Based Isolation Mechanisms
	FDP_PPR_EXT.1 Physical Platform Resource Controls
	FDP_RIP_EXT.1 Residual Information in Memory
	FDP_RIP_EXT.2 Residual Information on Disk
	FDP_VMS_EXT.1 VM Separation
	FDP_VNC_EXT.1 Virtual Networking Components
Identification and Authentication (FIA)	FIA_AFL_EXT.1 Authentication Failure Handling
	FIA_PMG_EXT.1 Password Management
	FIA_UIA_EXT.1 Administrator Identification and Authentication
	FIA_X509_EXT.1 X.509 Certificate
Security Management (FMT)	FMT_SMO_EXT.1 Separation of Management and OperationalNetworks
	FMT_MOF_EXT.1 Management of Security Function Behaviour
Protection of the TSF (FPT)	FPT_DVD_EXT.1 Non-Existence of Disconnected Virtual Devices
	FPT_EEM_EXT.1 Execution Environment Mitigations
	FPT_HAS_EXT.1 Hardware Assists
	FPT_HCL_EXT.1 Hypercall Controls
	FPT_RDM_EXT.1 Removable Devices and Media
	FPT_TUD_EXT.1 Trusted Updates
	FPT_VDP_EXT.1 Virtual Device Parameters

Functional class	Functional components
	FPT_VIV_EXT.1 VMM Isolation from VMs
Trusted Path/Channel (FTP)	FTP_ITC_EXT.1 Trusted Channel Communications
	FTP_UIF_EXT.1 User Interface:I/O Focus
	FTP_UIF_EXT.2 User Interface: Identification of VM

5.1 Class ALC: Life Cycle Support

This extended assurance component is derived from [PP_BASE_VIRTUALIZATION_V1.1]☝.

5.1.1 Timely Security Updates (ALC_TSU_EXT.1)

Objectives

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

Overview

This extended assurance component is derived from [PP_BASE_VIRTUALIZATION_V1.1]☝.

ALC_TSU_EXT.1.1 - Timely Security Updates

Dependencies: No dependencies.

Developer action elements:

ALC_TSU_EXT.1.1D The developer shall provide a description in the TSS of how timely security updates are made to the TOE.

Content and presentation elements:

ALC_TSU_EXT.1.1C The description shall include the process for creating and deploying security updates for the TOE software/firmware.

ALC_TSU_EXT.1.2C The description shall express the time window as the length of time, in days, between public disclosure of a vulnerability and the public availability of security updates to the TOE.

ALC_TSU_EXT.1.3C The description shall include the mechanisms publicly available for reporting security issues pertaining to the TOE.

Evaluator action elements:

ALC_TSU_EXT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6 Security Requirements

6.1 TOE Security Functional Requirements

The table below summarizes the SFRs for the TOE and the operations performed on the components according to CC part 1. Operations in the SFRs use the following convention:

- Iterations (Iter.) are identified by appending a suffix to the original SFR.
- Refinements (Ref.) added to the text are shown in *italic text*, deletions are shown as ~~strikethrough text~~.
- Assignments (Ass.) are shown in **bold text**.
- Selections (Sel.) are shown in **bold text**.

Table 9: SFRs for the TOE

Security Functional Class	Security Functional Requirement	Base Security Functional Component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
FAU - Security audit	FAU_GEN.1 Audit Data Generation		PP_BASE_VIRTUALIZATION_V1.1	No	Yes	No	Yes
	FAU_SAR.1 Audit Review		PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
	FAU_STG.1 Protected Audit Trail Storage		PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
	FAU_STG_EXT.1 Off-Loading of Audit Data		PP_BASE_VIRTUALIZATION_V1.1	No	No	Yes	Yes
FCS - Cryptographic support	FCS_CKM.1 Cryptographic Key Generation		PP_BASE_VIRTUALIZATION_V1.1	No	No	No	Yes
	FCS_CKM.2 Cryptographic Key Distribution		PP_BASE_VIRTUALIZATION_V1.1	No	No	No	Yes
	FCS_CKM_EXT.4 Cryptographic Key Destruction		PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
	FCS_COP.1/UDE Cryptographic Operation (AES Data Encryption/ Decryption)		PP_BASE_VIRTUALIZATION_V1.1	Yes	No	No	Yes
	FCS_COP.1/HASH Cryptographic Operation (Hashing)		PP_BASE_VIRTUALIZATION_V1.1	Yes	No	No	Yes
	FCS_COP.1/SIG Cryptographic Operation (Signature Algorithms)		PP_BASE_VIRTUALIZATION_V1.1	Yes	No	No	Yes

Security Functional Class	Security Functional Requirement	Base Security Functional Component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FCS_COP.1/KEYEDHASH Cryptographic Operation (Keyed Hash Algorithms)		PP_BASE_VIRTUALIZATION_V1.1	Yes	No	Yes	Yes
	FCS_RBG_EXT.1/HMC Cryptographic Operation (Random Bit Generation) (HMC)		PP_BASE_VIRTUALIZATION_V1.1	Yes	Yes	No	Yes
	FCS_RBG_EXT.1/SVR Cryptographic Operation (Random Bit Generation) (Server)		PP_BASE_VIRTUALIZATION_V1.1	Yes	Yes	No	Yes
	FCS_ENT_EXT.1 Entropy for Virtual Machines		PP_BASE_VIRTUALIZATION_V1.1	No	Yes	No	Yes
	FCS_HTTPS_EXT.1 HTTPS Protocol		PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
	FCS_TLSS_EXT.1 TLS Server Protocol		PKG_TLS_V1.1	No	No	No	Yes
	FCS_TLS_EXT.1 TLS Protocol		PKG_TLS_V1.1	No	No	No	Yes
	FCS_SSH_EXT.1 SSH Protocol		PKG_SSH_V1.0	No	No	Yes	Yes
	FCS_SSHC_EXT.1 SSH Protocol - Client		PKG_SSH_V1.0	No	No	No	Yes
	FCS_SSHS_EXT.1 SSH Protocol - Server		PKG_SSH_V1.0	No	No	No	Yes
FDP - User data protection	FDP_HBI_EXT.1 Hardware-Based Isolation Mechanisms		PP_BASE_VIRTUALIZATION_V1.1	No	No	Yes	Yes
	FDP_PPR_EXT.1 Physical Platform Resource Controls		PP_BASE_VIRTUALIZATION_V1.1	No	No	Yes	Yes
	FDP_RIP_EXT.1 Residual Information in Memory		PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
	FDP_RIP_EXT.2 Residual Information on Disk		PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
	FDP_VMS_EXT.1 VM Separation		PP_BASE_VIRTUALIZATION_V1.1	No	No	No	Yes
	FDP_VNC_EXT.1 Virtual Networking Components		PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No

Security Functional Class	Security Functional Requirement	Base Security Functional Component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
FIA - Identification and authentication	FIA_X509_EXT.1 X.509 Certificate Validation		PP_BASE_VIRTUALIZATION_V1.1	No	No	No	Yes
	FIA_PMG_EXT.1 Password Management		PP_BASE_VIRTUALIZATION_V1.1	No	No	No	Yes
	FIA_UAU.5 Multiple Authentication Mechanisms		PP_BASE_VIRTUALIZATION_V1.1	No	No	Yes	Yes
	FIA_AFL_EXT.1 Authentication Failure Handling		PP_BASE_VIRTUALIZATION_V1.1	No	No	Yes	Yes
	FIA_UIA_EXT.1 Administrator Identification and Authentication		PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
FMT - Security management	FMT_SMO_EXT.1 Separation of Management and Operational Networks		PP_BASE_VIRTUALIZATION_V1.1	No	No	No	Yes
	FMT_MOF_EXT.1 Management of Security Functions Behavior		MOD_SV_V1.1	No	Yes	No	Yes
FPT - Protection of the TSF	FPT_EEM_EXT.1/HMC Execution Environment Mitigations	FPT_EEM_EXT.1	PP_BASE_VIRTUALIZATION_V1.1	Yes	Yes	No	Yes
	FPT_EEM_EXT.1/SVR Execution Environment Mitigations	FPT_EEM_EXT.1	PP_BASE_VIRTUALIZATION_V1.1	Yes	Yes	No	Yes
	FPT_DVD_EXT.1 Non-Existence of Disconnected Virtual Devices		PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
	FPT_HAS_EXT.1 Hardware Assists		PP_BASE_VIRTUALIZATION_V1.1	No	No	Yes	No
	FPT_HCL_EXT.1 Hypercall Controls		PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
	FPT_RDM_EXT.1 Removable Devices and Media		PP_BASE_VIRTUALIZATION_V1.1	No	No	Yes	Yes
	FPT_TUD_EXT.1/HMC Trusted Updates to the Virtualization System	FPT_TUD_EXT.1	PP_BASE_VIRTUALIZATION_V1.1	Yes	Yes	No	Yes

Security Functional Class	Security Functional Requirement	Base Security Functional Component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FPT_TUD_EXT.1/SVR Trusted Updates to the Virtualization System	FPT_TUD_EXT.1	PP_BASE_VIRTUALIZATION_V1.1	Yes	Yes	No	Yes
	FPT_TUD_EXT.1/VIOS Trusted Updates to the Virtualization System	FPT_TUD_EXT.1	PP_BASE_VIRTUALIZATION_V1.1	Yes	Yes	No	Yes
	FPT_VDP_EXT.1 Virtual Device Parameters		PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
	FPT_VIV_EXT.1 VMM Isolation from VMs		PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
FTA - TOE access	FTA_TAB.1 TOE Access Banner		PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
FTP - Trusted path/channels	FTP_ITC_EXT.1 Trusted Channel Communications		PP_BASE_VIRTUALIZATION_V1.1	No	No	No	Yes
	FTP_TRP.1 Trusted Path		PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
	FTP_UIF_EXT.1 User Interface: I/O Focus		PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
	FTP_UIF_EXT.2 User Interface: Identification of VM		PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No

6.1.1 Security audit (FAU)

6.1.1.1 FAU_GEN.1 Audit Data Generation

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

a) Start-up and shutdown of 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 ~~2~~ 10;

d)
- **Auditable events defined in Table ~~7~~ 11 ; for Selection-Based SFRs**
- **Auditable events for the Functional Package for Transport Layer Security (TLS), version 1.1 listed in Table ~~3~~ 12 ;**

- **Auditable events defined in in Table ~~the audit table~~ 13 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 ~~2~~ 10;

f)
- **Additional information defined in Table ~~7~~ 11 ;for Selection-Based SFRs,**
- **Additional information for the Functional Package for Transport Layer Security (TLS), version 1.1 listed in Table ~~3~~ 12 ;**
- **Additional information defined in Table ~~the audit table~~ 13 for the Functional Package for Secure Shell (SSH), version 1.0;**

Table 10: Auditable Events for Mandatory Requirements

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	No events specified	
FAU_SAR.1	No events specified	
FAU_STG.1	No events specified	
FAU_STG_EXT.1	Failure of audit data capture due to lack of disk space or pre-defined limit.	
FAU_STG_EXT.1	On failure of logging function, capture record of failure and record upon restart of logging function.	
FCS_CKM.1	No events specified	
FCS_CKM.2	No events specified	
FCS_CKM_EXT.4	No events specified	
FCS_COP.1/UDE	No events specified	
FCS_COP.1/HASH	No events specified	
FCS_COP.1/SIG	No events specified	
FCS_COP.1/KEYEDHASH	No events specified	
FCS_ENT_EXT.1	No events specified	
FCS_RBG_EXT.1/HMC and FCS_RBG_EXT.1/SVR	Failure of the randomization process.	
FDP_HBI_EXT.1	No events specified	
FDP_PPR_EXT.1	Successful and failed VM connections to physical devices where connection is governed by configurable policy.	VM and physical device identifiers.

Requirement	Auditable Events	Additional Audit Record Contents
FDP_PPR_EXT.1	Security policy violations.	Identifier for the security policy that was violated.
FDP_RIP_EXT.1	No events specified	
FDP_RIP_EXT.2	No events specified	
FDP_VMS_EXT.1	No events specified	
FDP_VNC_EXT.1	Successful and failed attempts to connect VMs to virtual and networking components.	VM and virtual networking component identifiers.
FDP_VNC_EXT.1	Security policy violations.	Identifier for the security policy that was violated. VM and virtual or physical networking component identifiers.
FDP_VNC_EXT.1	Administrator configuration of inter-VM communications channels between VMs.	VM and virtual or physical networking component identifiers.
FIA_AFL_EXT.1	Unsuccessful login attempts limit is met or exceeded.	Origin of attempt (e.g., IP address).
FIA_UAU.5	No events specified	
FIA_UIA_EXT.1	Administrator authentication attempts.	Provided user identity, origin of the attempt (e.g., console, remote IP address).
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., console, remote IP address).
FIA_UIA_EXT.1	**None**	Start time and end time of administrator session.
FMT_SMO_EXT.1	No events specified	
FPT_DVD_EXT.1	No events specified	
FPT_EEM_EXT.1	No events specified	
FPT_HAS_EXT.1	No events specified	
FPT_HCL_EXT.1	**Invalid parameter to hypercall detected.**	Hypercall interface for which access was attempted.
FPT_HCL_EXT.1	**Hypercall interface invoked when documented preconditions are not met.**	
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.).
FPT_RDM_EXT.1	Ejection/insertion of removable media or device from/to an already connected VM.	VM Identifier, Removable media/device identifier, event description or identifier (connect/disconnect, ejection/insertion, etc.).
FPT_TUD_EXT.1	Initiation of update.	
FPT_TUD_EXT.1	Failure of signature verification.	

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

Table 11: Auditable Events for Selection-based Requirements

Requirement	Auditable Events	Additional Audit Record Contents
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure. Non-TOE endpoint of connection (IP address) for failures.
FCS_HTTPS_EXT.1	Establishment/Termination of a HTTPS session.	Non-TOE endpoint of connection (IP address).
FIA_PMG_EXT.1	No events specified	
FIA_X509_EXT.1	Failure to validate a certificate.	Reason for failure.
FTP_TRP.1	Initiation of the trusted channel.	User ID and remote source (IP Address) if feasible.
FTP_TRP.1	Termination of the trusted channel.	User ID and remote source (IP Address) if feasible.
FTP_TRP.1	Failures of the trusted path functions.	User ID and remote source (IP Address) if feasible.

Table 12: Auditable Events for TLS Functional Package

Requirement	Auditable Events	Additional Audit Record Contents
FCS_TLSS_EXT.1	Failure to establish a session.	Reason for failure.

Table 13: Auditable Events for SSH Functional Package

Requirement	Auditable Events	Additional Audit Contents
FCS_SSH_EXT.1	**None**	**None**
	None	**None**
	None	**None**
	None	**None**
FCS_SSHC_EXT.1	No events specified.	N/A

Requirement	Auditable Events	Additional Audit Contents
FCS_SSHS_EXT.1	No events specified.	N/A

6.1.1.2 FAU_SAR.1 Audit Review

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

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

6.1.1.3 FAU_STG.1 Protected Audit Trail Storage

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

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

6.1.1.4 FAU_STG_EXT.1 Off-Loading of Audit Data

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

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

6.1.2 Cryptographic support (FCS)

6.1.2.1 FCS_CKM.1 Cryptographic Key Generation

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

- **RSA schemes using cryptographic key sizes [2048-bit or greater] that meet the following: [FIPS PUB 186-5, “Digital Signature Standard (DSS)”, Appendix B.3]**
- **ECC schemes using [“NIST curves” P-256, P-384, and P-521**, **no other curves that meet the following: [FIPS PUB 186-5, “Digital Signature Standard (DSS)”, Appendix B.4]**

.

6.1.2.2 FCS_CKM.2 Cryptographic Key Distribution

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

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

.

6.1.2.3 FCS_CKM_EXT.4 Cryptographic Key Destruction

Origin: PP_BASE_VIRTUALIZATION_V1.1

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.

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

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

- **AES-GCM (as defined in NIST SP 800-38D)**
- **AES-CTR (as defined in NIST SP 800-38A) mode**

and cryptographic key sizes **128-bit key sizes**, **256-bit key sizes**.

6.1.2.5 FCS_COP.1/HASH Cryptographic Operation (Hashing)

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

6.1.2.6 FCS_COP.1/SIG Cryptographic Operation (Signature Algorithms)

Origin: PP_BASE_VIRTUALIZATION_V1.1

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 186-5, “Digital Signature Standard (DSS)”, Section 4]**
- **ECDSA schemes using “NIST curves” P-256, P-384 and P-521 that meet the following: FIPS PUB 186-5, “Digital Signature Standard (DSS)”, Section 5**

6.1.2.7 FCS_COP.1/KEYEDHASH Cryptographic Operation (Keyed Hash Algorithms)

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

6.1.2.8 FCS_RBG_EXT.1/HMC Cryptographic Operation (Random Bit Generation) (HMC)

Origin: PP_BASE_VIRTUALIZATION_V1.1

FCS_RBG_EXT.1.1/HMC The ~~TSF~~ *HMC part of the TSF* shall perform all deterministic random bit generation services in accordance with NIST Special Publication 800-90A using **CTR_DRBG (AES)**

FCS_RBG_EXT.1.2/HMC The deterministic RBG shall be seeded by an entropy source that accumulates entropy from **a software-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.

6.1.2.9 FCS_RBG_EXT.1/SVR Cryptographic Operation (Random Bit Generation) (Server)

Origin: PP_BASE_VIRTUALIZATION_V1.1

FCS_RBG_EXT.1.1/SVR The ~~TSF~~ *Server part of 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/SVR The deterministic RBG shall be seeded by an entropy source that accumulates entropy from **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.

6.1.2.10 FCS_ENT_EXT.1 Entropy for Virtual Machines

Origin: PP_BASE_VIRTUALIZATION_V1.1

FCS_ENT_EXT.1.1 The ~~TSF~~ *Server part of the TSF* shall provide a mechanism to make available to VMs entropy that meets FCS_RBG_EXT.1 through **passthrough access to hardware entropy source** .

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

6.1.2.11 FCS_HTTPS_EXT.1 HTTPS Protocol

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS.

6.1.2.12 FCS_TLSS_EXT.1 TLS Server Protocol

Origin: PKG_TLS_V1.1

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

- **TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288**
- **TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289**
- **TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289**

and also supports functionality for

- **no session resumption or session tickets,**

, and

- **none**

.

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

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

- **RSA with size 2048 bits, and no other sizes,**
- **ECDHE parameters using elliptic curves secp256r1, secp384r1, secp521r1 and no other curves**

.

6.1.2.13 FCS_TLS_EXT.1 TLS Protocol

Origin: PKG_TLS_V1.1

FCS_TLS_EXT.1.1 The product shall implement

- **TLS as a server**

.

6.1.2.14 FCS_SSH_EXT.1 SSH Protocol

Origin: PKG_SSH_V1.0

FCS_SSH_EXT.1.1 The TOE shall implement SSH acting as a **client**, **server** in accordance with that complies with RFCs 4251, 4252, 4253, 4254, **4344**, **5656**, **6668** 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)**

and no other methods.

FCS_SSH_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than **35K bytes** in an SSH transport connection are dropped.

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

- **aes128-ctr (RFC 4344)**

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)**

and no other mechanisms.

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

- **ecdh-sha2-nistp256 (RFC 5656)**
- **ecdh-sha2-nistp384 (RFC 5656)**

and no other mechanisms.

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

- **RFC 5656 (Section 4)**

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 connection time

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

6.1.2.15 FCS_SSHC_EXT.1 SSH Protocol - Client

Origin: PKG_SSH_V1.0

FCS_SSHC_EXT.1.1 The TSF shall authenticate its peer (SSH server) using:

- **using a local database by associating each host name with a public key corresponding to the following list:**
  - **ssh-rsa (RFC 4253)**

as described in RFC 4251 section 4.1.

6.1.2.16 FCS_SSHS_EXT.1 SSH Protocol - Server

Origin: PKG_SSH_V1.0

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

- **ssh-rsa (RFC 4253)**

.

6.1.3 User data protection (FDP)

6.1.3.1 FDP_HBI_EXT.1 Hardware-Based Isolation Mechanisms

Origin: PP_BASE_VIRTUALIZATION_V1.1

FDP_HBI_EXT.1.1 The TSF shall use **slot P1-CO** to constrain a Guest VM's direct access to the following physical devices: **Peripheral Component Interconnect Express (PCIe) slots to which the VMM allows Guest VMs physical access**.

6.1.3.2 FDP_PPR_EXT.1 Physical Platform Resource Controls

Origin: PP_BASE_VIRTUALIZATION_V1.1

FDP_PPR_EXT.1.1 The TSF shall allow an authorized administrator to control Guest VM access to the following physical platform resources: **Peripheral Component Interconnect Express (PCIe) slots to which the VMM is able to control access.**

FDP_PPR_EXT.1.2 The TSF shall explicitly deny all Guest VMs access to the following physical platform resources:

- **PCIe4 4-port NVMe JBOF adapter (FC EJ1X and EJ1Y; CCIN 6B87)**
- **PCIe4 cable adapter (FC EJ24; CCIN 6B92)**

.

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

6.1.3.3 FDP_RIP_EXT.1 Residual Information in Memory

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

6.1.3.4 FDP_RIP_EXT.2 Residual Information on Disk

Origin: PP_BASE_VIRTUALIZATION_V1.1

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.

6.1.3.5 FDP_VMS_EXT.1 VM Separation

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

- **virtual networking**

.

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

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

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

6.1.3.6 FDP_VNC_EXT.1 Virtual Networking Components

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

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

6.1.4 Identification and authentication (FIA)

6.1.4.1 FIA_X509_EXT.1 X.509 Certificate Validation

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

- RFC 5280 certificate validation and certificate path validation
- The certificate path must terminate with a trusted certificate
- The TOE shall validate a certificate path by ensuring the presence of the basicConstraints extension, that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met.
- The TSF shall validate that any CA certificate includes caSigning purpose in the key usage field
- The TSF shall validate revocation status of the certificate using **an OCSP TLS Status Request Extension (OCSP stapling) as specified in RFC 6066** 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.

6.1.4.2 FIA_PMG_EXT.1 Password Management

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

a) Passwords shall be able to be composed of any combination of upper and lower case characters, digits, and the following special characters: **“!”**, **“@”**, **“#”**, **“$”**, **“%”**, **“^”**, **“&”**, **“*”**, **“(“**, **“)”**

b) Minimum password length shall be configurable

c) Passwords of at least 15 characters in length shall be supported

6.1.4.3 FIA_UAU.5 Multiple Authentication Mechanisms

Origin: PP_BASE_VIRTUALIZATION_V1.1

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 **successful local authentication through username and password**

6.1.4.4 FIA_AFL_EXT.1 Authentication Failure Handling

Origin: PP_BASE_VIRTUALIZATION_V1.1

FIA_AFL_EXT.1.1 The TSF shall detect when

- **an administrator configurable positive integer between 3 and 50**

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

FIA_AFL_EXT.1.2 When the defined number of unsuccessful authentication attempts has been met, the TSF shall: **prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password or PIN until the unlocking of the account 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"** .

6.1.4.5 FIA_UIA_EXT.1 Administrator Identification and Authentication

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

6.1.5 Security management (FMT)

6.1.5.1 FMT_SMO_EXT.1 Separation of Management and Operational Networks

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

6.1.5.2 FMT_MOF_EXT.1 Management of Security Functions Behavior

Origin: MOD_SV_V1.1

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

FMT_MOF_EXT.1.2 The TSF shall be capable of performing the following management functions, [controlled by an Administrator or User as shown in Table ~~3~~ 14, 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 14: Server Virtualization 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**	S	N
3	Ability to create, configure and delete VMs	X	-
4	Ability to set default initial VM configurations	X	N
5	Ability to configure virtual networks including VM	X	-
6	Ability to configure and manage the audit system and audit data	X	N
7	Ability to configure VM access to physical devices	X	-
8	Ability to configure inter-VM data sharing	X	-
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
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
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	-
20	Ability to configure banner	X	N

6.1.6 Protection of the TSF (FPT)

6.1.6.1 FPT_EEM_EXT.1/HMC Execution Environment Mitigations

Origin: PP_BASE_VIRTUALIZATION_V1.1

FPT_EEM_EXT.1.1/ HMC The ~~TSF~~ *HMC part of the TSF* shall take advantage of execution environment-based vulnerability mitigation mechanisms supported by the Platform such as:

- **No mechanisms**

6.1.6.2 FPT_EEM_EXT.1/SVR Execution Environment Mitigations

Origin: PP_BASE_VIRTUALIZATION_V1.1

FPT_EEM_EXT.1.1/ SVR The ~~TSF~~ *Server part of the TSF* shall take advantage of execution environment-based vulnerability mitigation mechanisms supported by the Platform such as:

- **Memory execution protection (e.g. Data Execution Protection (DEP))**
- **Stack buffer overflow protection**
- **Heap corruption detection**

6.1.6.3 FPT_DVD_EXT.1 Non-Existence of Disconnected Virtual Devices

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

6.1.6.4 FPT_HAS_EXT.1 Hardware Assists

Origin: PP_BASE_VIRTUALIZATION_V1.1

FPT_HAS_EXT.1.1 The VMM shall use **Power Instruction Set Architecture (ISA) Privilege States** to reduce or eliminate the need for binary translation.

FPT_HAS_EXT.1.2 The VMM shall use **Power ISA Privilege States, Hardware Page Tables (HPTs), Translation Control Entries (TCEs)** to reduce or eliminate the need for shadow page tables.

6.1.6.5 FPT_HCL_EXT.1 Hypercall Controls

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

6.1.6.6 FPT_RDM_EXT.1 Removable Devices and Media

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

FPT_RDM_EXT.1.2 The TSF shall enforce the following rules when **Removable Media Device (RMD)** are switched between information domains, then

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

6.1.6.7 FPT_TUD_EXT.1/HMC Trusted Updates to the Virtualization System

Origin: PP_BASE_VIRTUALIZATION_V1.1

FPT_TUD_EXT.1.1/HMC The ~~TSF~~ *HMC part of 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/HMC The ~~TSF~~ *HMC part of the TSF* shall provide administrators the ability to manually initiate updates to TOE firmware/software and **no other update mechanism** .

FPT_TUD_EXT.1.3/HMC The ~~TSF~~ *HMC part of the TSF* shall provide means to authenticate firmware/ software updates to the TOE using a **published hash** prior to installing those updates.

6.1.6.8 FPT_TUD_EXT.1/SVR Trusted Updates to the Virtualization System

Origin: PP_BASE_VIRTUALIZATION_V1.1

FPT_TUD_EXT.1.1/SVR The ~~TSF~~ *Server part of 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/SVR The ~~TSF~~ *Server part of the TSF* shall provide administrators the ability to manually initiate updates to TOE firmware/software and **no other update mechanism** .

FPT_TUD_EXT.1.3/SVR The ~~TSF~~ *Server part of 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.

6.1.6.9 FPT_TUD_EXT.1/VIOS Trusted Updates to the Virtualization System

Origin: PP_BASE_VIRTUALIZATION_V1.1

FPT_TUD_EXT.1.1/VIOS The ~~TSF~~ *VIOS part of 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/ VIOS The ~~TSF~~ *VIOS part of the TSF* shall provide administrators the ability to manually initiate updates to TOE firmware/software and **no other update mechanism**.

FPT_TUD_EXT.1.3/ VIOS The ~~TSF~~ *VIOS part of the TSF* shall provide means to authenticate firmware/ software updates to the TOE using a **published hash** prior to installing those updates.

6.1.6.10 FPT_VDP_EXT.1 Virtual Device Parameters

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

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

6.1.6.11 FPT_VIV_EXT.1 VMM Isolation from VMs

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

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

6.1.7 TOE access (FTA)

6.1.7.1 FTA_TAB.1 TOE Access Banner

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

6.1.8 Trusted path/channels (FTP)

6.1.8.1 FTP_ITC_EXT.1 Trusted Channel Communications

Origin: PP_BASE_VIRTUALIZATION_V1.1

FTP_ITC_EXT.1.1 The TSF shall use

- **TLS as conforming to the Functional Package for Transport Layer Security**
- **TLS/HTTPS as conforming to FCS_HTTPS_EXT.1**
- **SSH as conforming to the Functional Package for Secure Shell**

and

- **non-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)**
- **no other capabilities**

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

6.1.8.2 FTP_TRP.1 Trusted Path

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

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

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

6.1.8.3 FTP_UIF_EXT.1 User Interface: I/O Focus

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

6.1.8.4 FTP_UIF_EXT.2 User Interface: Identification of VM

Origin: PP_BASE_VIRTUALIZATION_V1.1

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

6.2 Security Functional Requirements Rationale

6.2.1 Coverage

The following table provides a mapping of SFR to the security objectives, showing that each security functional requirement addresses at least one security objective.

Table 15: Mapping of security functional requirements to security objectives

Security Functional Requirements	Objectives
FAU_GEN.1	O.AUDIT, O.MANAGEMENT_ACCESS, O.VM_ISOLATION, O.VMM_INTEGRITY
FAU_SAR.1	O.AUDIT
FAU_STG.1	O.AUDIT
FAU_STG_EXT.1	O.AUDIT
FCS_CKM.1	O.MANAGEMENT_ACCESS, O.VMM_INTEGRITY
FCS_CKM.2	O.MANAGEMENT_ACCESS
FCS_CKM_EXT.4	O.DOMAIN_INTEGRITY, O.RESOURCE_ALLOCATION, O.VM_ISOLATION
FCS_COP.1/UDE	O.MANAGEMENT_ACCESS
FCS_COP.1/HASH	O.MANAGEMENT_ACCESS
FCS_COP.1/SIG	O.MANAGEMENT_ACCESS
FCS_COP.1/KEYEDHASH	O.MANAGEMENT_ACCESS
FCS_RBG_EXT.1/HMC	O.DOMAIN_INTEGRITY, O.MANAGEMENT_ACCESS, O.VMM_INTEGRITY
FCS_RBG_EXT.1/SVR	O.DOMAIN_INTEGRITY, O.MANAGEMENT_ACCESS, O.VM_ENTROPY, O.VMM_INTEGRITY
FCS_ENT_EXT.1	O.DOMAIN_INTEGRITY, O.VM_ENTROPY
FDP_HBI_EXT.1	O.PLATFORM_INTEGRITY
FIA_X509_EXT.1	O.MANAGEMENT_ACCESS
FCS_HTTPS_EXT.1	O.MANAGEMENT_ACCESS
FDP_PPR_EXT.1	O.PLATFORM_INTEGRITY, O.VM_ISOLATION, O.VMM_INTEGRITY
FDP_RIP_EXT.1	O.DOMAIN_INTEGRITY, O.RESOURCE_ALLOCATION, O.VM_ISOLATION
FDP_RIP_EXT.2	O.DOMAIN_INTEGRITY, O.RESOURCE_ALLOCATION, O.VM_ISOLATION
FIA_PMG_EXT.1	O.MANAGEMENT_ACCESS
FIA_UAU.5	O.MANAGEMENT_ACCESS

Security Functional Requirements	Objectives
FMT_SMO_EXT.1	O.MANAGEMENT_ACCESS
FPT_EEM_EXT.1/HMC	O.DOMAIN_INTEGRITY, O.PLATFORM_INTEGRITY, O.VM_ISOLATION, O.VMM_INTEGRITY
FPT_EEM_EXT.1/SVR	O.DOMAIN_INTEGRITY, O.PLATFORM_INTEGRITY, O.VM_ISOLATION, O.VMM_INTEGRITY
FDP_VMS_EXT.1	O.CORRECTLY_APPLIED_CONFIGURATION, O.DOMAIN_INTEGRITY, O.PLATFORM_INTEGRITY, O.VM_ISOLATION, O.VMM_INTEGRITY
FDP_VNC_EXT.1	O.DOMAIN_INTEGRITY, O.PLATFORM_INTEGRITY, O.VM_ISOLATION, O.VMM_INTEGRITY
FIA_AFL_EXT.1	O.MANAGEMENT_ACCESS
FIA_UIA_EXT.1	O.MANAGEMENT_ACCESS
FPT_DVD_EXT.1	O.PLATFORM_INTEGRITY, O.VM_ISOLATION
FPT_HAS_EXT.1	O.DOMAIN_INTEGRITY, O.PLATFORM_INTEGRITY, O.VM_ISOLATION, O.VMM_INTEGRITY
FPT_HCL_EXT.1	O.PLATFORM_INTEGRITY, O.VMM_INTEGRITY
FPT_RDM_EXT.1	O.DOMAIN_INTEGRITY
FPT_TUD_EXT.1/HMC	O.PATCHED_SOFTWARE
FPT_TUD_EXT.1/SVR	O.PATCHED_SOFTWARE
FPT_TUD_EXT.1/VIOS	O.PATCHED_SOFTWARE
FPT_VDP_EXT.1	O.DOMAIN_INTEGRITY, O.PLATFORM_INTEGRITY, O.VM_ISOLATION, O.VMM_INTEGRITY
FPT_VIV_EXT.1	O.PLATFORM_INTEGRITY, O.VM_ISOLATION, O.VMM_INTEGRITY
FTA_TAB.1	O.MANAGEMENT_ACCESS
FMT_MOF_EXT.1	O.MANAGEMENT_ACCESS, O.VMM_INTEGRITY
FTP_ITC_EXT.1	O.MANAGEMENT_ACCESS

Security Functional Requirements	Objectives
FTP_TRP.1	O.MANAGEMENT_ACCESS
FTP_UIF_EXT.1	O.DOMAIN_INTEGRITY
FTP_UIF_EXT.2	O.DOMAIN_INTEGRITY
FCS_TLSS_EXT.1	O.MANAGEMENT_ACCESS
FCS_TLS_EXT.1	O.MANAGEMENT_ACCESS
FCS_SSH_EXT.1	O.MANAGEMENT_ACCESS
FCS_SSHC_EXT.1	O.MANAGEMENT_ACCESS
FCS_SSHS_EXT.1	O.MANAGEMENT_ACCESS

6.2.2 Sufficiency

The following rationale provides justification for each security objective for the TOE, showing that the security functional requirements are suitable to meet and achieve the security objectives.

Table 16: Security objectives for the TOE rationale

Security Objectives	Rationale
O.VM_ISOLATION	[PP_BASE_VIRTUALIZATION_V1.1]☝ FAU_GEN.1 defines the auditable events that must be generated to diagnose the cause of unexpected system behavior. [PP_BASE_VIRTUALIZATION_V1.1]☝ FCS_CKM_EXT.4 requires cryptographic keys in volatile memory to be destroyed or rendered unrecoverable. [PP_BASE_VIRTUALIZATION_V1.1]☝ FDP_PPR_EXT.1 defines physical platform resources that Guest VM are allowed and disallowed access. [PP_BASE_VIRTUALIZATION_V1.1]☝ FDP_RIP_EXT.1 requires deallocation of physical memory prior to allocating to a Guest VM. [PP_BASE_VIRTUALIZATION_V1.1]☝ FDP_RIP_EXT.2 requires deallocation of physical disk storage prior to allocating to a Guest VM. [PP_BASE_VIRTUALIZATION_V1.1]☝ FDP_VMS_EXT.1 defines the mechanisms for transfering data between Guest VMs. It also defines a policy prohibiting data sharing between Guest VMs. [PP_BASE_VIRTUALIZATION_V1.1]☝ FDP_VNC_EXT.1 requires virtual networking components to connect VMs to each other and to physical networks. [PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_DVD_EXT.1 prevents Guest VMs from accessing virtual device interfaces not present in the VM's current hardware configuration. [PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_EEM_EXT.1/HMC defines the execution-based vulnerability mitigation mechanisms supported by the platform. [PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_EEM_EXT.1/SVR defines the execution-based vulnerability mitigation mechanisms supported by the platform.

Security Objectives	Rationale
	[PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_HAS_EXT.1 defines hardware-based virtualization assists to reduce or eliminate binary translation or shadow page tables. [PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_VDP_EXT.1 ensures that the VMM is not vulnerable to compromise through malformed data passed to the virtual device interface from a Guest OS. [PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_VIV_EXT.1 ensures software running in one VM does not degrade or disrupt other VMs, the VMM, or the platform. It also ensures that without the VMM, a Guest M cannot invoke platform code with equal or higher privilege than the VMM.
O.VMM_INTEGRITY	[PP_BASE_VIRTUALIZATION_V1.1]☝ FAU_GEN.1 defines the auditable events that must be generated to diagnose the cause of unexpected system behavior. [PP_BASE_VIRTUALIZATION_V1.1]☝ FCS_CKM.1 defines the asymmetric algorithm used by the TOE. [PP_BASE_VIRTUALIZATION_V1.1]☝ FCS_RBG_EXT.1/HMC defines the random bit generator used for create the asymmetric keys. [PP_BASE_VIRTUALIZATION_V1.1]☝ FCS_RBG_EXT.1/SVR defines the random bit generator used for Guest VMs. [PP_BASE_VIRTUALIZATION_V1.1]☝ FDP_PPR_EXT.1 defines physical platform resources that Guest VM are allowed and disallowed access. [PP_BASE_VIRTUALIZATION_V1.1]☝ FDP_VMS_EXT.1 defines the mechanisms for transferring data between Guest VMs. It also defines a policy prohibiting data sharing between Guest VMs. [PP_BASE_VIRTUALIZATION_V1.1]☝ FDP_VNC_EXT.1 requires virtual networking components to connect VMs to each other and to physical networks. [PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_EEM_EXT.1/HMC defines the execution-based vulnerability mitigation mechanisms supported by the platform. [PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_EEM_EXT.1/SVR defines the execution-based vulnerability mitigation mechanisms supported by the platform. [PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_HAS_EXT.1 defines hardware-based virtualization assists to reduce or eliminate binary translation or shadow page tables. [PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_HCL_EXT.1 ensures the integrity of the VMM by protecting the attack service exposed to untrusted Gust VMs through Hypercalls. [PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_VDP_EXT.1 ensures that the VMM is not vulnerable to compromise through the processing of malformed data passed to the virtual device interface from a Guest OS. [PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_VIV_EXT.1 ensures that the software running within a Guest VM cannot compromise other VMs, the VMM, or the platform. [MOD_SV_V1.1]☝ FMT_MOF_EXT.1 requires related management functions are performed.

Security Objectives	**Rationale**
O.PLATFORM_INTEGRITY	[PP_BASE_VIRTUALIZATION_V1.1]☝ FDP_HBI_EXT.1 requires that Guest VM's direct access to particular physical devices are constrained. [PP_BASE_VIRTUALIZATION_V1.1]☝ FDP_PPR_EXT.1 defines physical platform resources that Guest VM are allowed and disallowed access. [PP_BASE_VIRTUALIZATION_V1.1]☝ FDP_VMS_EXT.1 defines the mechanisms for transferring data between Guest VMs. It also defines a policy prohibiting data sharing between Guest VMs. [PP_BASE_VIRTUALIZATION_V1.1]☝ FDP_VNC_EXT.1 requires virtual networking components to connect VMs to each other and to physical networks. [PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_DVD_EXT.1 prevents Guest VMs from accessing virtual device interfaces not present in the VM's current hardware configuration. [PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_EEM_EXT.1/HMC defines the execution-based vulnerability mitigation mechanisms supported by the platform. [PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_EEM_EXT.1/SVR defines the execution-based vulnerability mitigation mechanisms supported by the platform. [PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_HAS_EXT.1 defines hardware-based virtualization assists to reduce or eliminate binary translation or shadow page tables. [PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_HCL_EXT.1 ensures the integrity of the VMM by protecting the attack service exposed to untrusted Gust VMs through Hypercalls. [PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_VDP_EXT.1 ensures that the VMM is not vulnerable to compromise through the processing of malformed data passed to the virtual device interface from a Guest OS. [PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_VIV_EXT.1 ensures that the software running within a Guest VM cannot compromise other VMs, the VMM, or the platform.
O.DOMAIN_INTEGRITY	[PP_BASE_VIRTUALIZATION_V1.1]☝ FCS_CKM_EXT.4 requires cryptographic keys in volatile memory to be destroyed or rendered unrecoverable. [PP_BASE_VIRTUALIZATION_V1.1]☝ FCS_ENT_EXT.1 ensures that entropy of one VM does not affect other VM's on the same platform. [PP_BASE_VIRTUALIZATION_V1.1]☝ FCS_RBG_EXT.1/HMC defines the random bit generator used for create the asymmetric keys. [PP_BASE_VIRTUALIZATION_V1.1]☝ FCS_RBG_EXT.1/SVR defines the random bit generator used for guest VMs. [PP_BASE_VIRTUALIZATION_V1.1]☝ FDP_RIP_EXT.1 requires deallocation of physical memory prior to allocating to a Guest VM. [PP_BASE_VIRTUALIZATION_V1.1]☝ FDP_RIP_EXT.2 requires deallocation of physical disk storage prior to allocating to a Guest VM. [PP_BASE_VIRTUALIZATION_V1.1]☝ FDP_VMS_EXT.1 defines the mechanisms for transferring data between Guest VMs. It also defines a policy prohibiting data sharing between Guest VMs.

Security Objectives	Rationale
	[PP_BASE_VIRTUALIZATION_V1.1]☝ FDP_VNC_EXT.1 requires virtual networking components to connect VMs to each other and to physical networks. [PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_EEM_EXT.1/HMC defines the execution-based vulnerability mitigation mechanisms supported by the platform. [PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_EEM_EXT.1/SVR defines the execution-based vulnerability mitigation mechanisms supported by the platform. [PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_HAS_EXT.1 defines hardware-based virtualization assists to reduce or eliminate binary translation or shadow page tables. [PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_RDM_EXT.1 ensures controls are implemented for the transfer of virtual and physical removable data and their devices between information domains. [PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_VDP_EXT.1 ensures that the VMM is not vulnerable to compromise through the processing of malformed data passed to the virtual device interface from a Guest OS. [PP_BASE_VIRTUALIZATION_V1.1]☝ FTP_UIF_EXT.1 requires user input focus. [PP_BASE_VIRTUALIZATION_V1.1]☝ FTP_UIF_EXT.2 ensures unique identification of a VM's output display to users.
O.MANAGEMENT_ACCESS	[PP_BASE_VIRTUALIZATION_V1.1]☝ FAU_GEN.1 defines the auditable events that must be generated to diagnose the cause of unexpected system behavior. [PP_BASE_VIRTUALIZATION_V1.1]☝ FCS_CKM.1, FCS_CKM.2, FCS_COP.1/UDE, FCS_COP.1/HASH, FCS_COP.1/SIG, FCS_COP.1/KEYEDHASH, FCS_RBG_EXT.1/HMC, and FCS_RBG_EXT.1/SVR define the cryptographic operations and key lifecycle activity used to support the establishment of protected communications. [PP_BASE_VIRTUALIZATION_V1.1]☝ FIA_X509_EXT.1 defines how the TSF validates X.509 certificates as part of establishing protected communications. [PP_BASE_VIRTUALIZATION_V1.1]☝ FIA_PMG_EXT.1 ensures that password-based administrator login is properly implemented. [PP_BASE_VIRTUALIZATION_V1.1]☝ FCS_HTTPS_EXT.1 defines the trusted communication protocols for communication between the TOE and another trusted IT entity. [PKG_TLS_V1.1]☝ FCS_TLS_EXT.1 and FCS_TLSS_EXT.1 define the trusted communication protocols for communication between the TOE and another trusted IT entity. [PKG_SSH_V1.0]☝ FCS_SSH_EXT.1, FCS_SSHC_EXT.1 and FCS_SSHS_EXT.1 define the trusted communication protocols for communication between the TOE and another trusted IT entity.

Security Objectives	Rationale
	[PP_BASE_VIRTUALIZATION_V1.1]☝ FIA_UAU.5 provides mechanisms that prevent untrusted users from accessing the TSF and FIA_AFL_EXT.1 prevents brute force authentication attempts. FIA_UIA_EXT.1 requires authentication of administrators before performing management functions. [PP_BASE_VIRTUALIZATION_V1.1]☝ FMT_SMO_EXT.1 ensures that there's separation between management and operation network traffic. [MOD_SV_V1.1]☝ FMT_MOF_EXT.1 requires related management functions are performed. [PP_BASE_VIRTUALIZATION_V1.1]☝ FTP_ITC_EXT.1 defines the trusted communications channels supported by the TOE. [PP_BASE_VIRTUALIZATION_V1.1]☝ FTP_TRP.1 provides one or more secure remote interfaces for management of the TSF and FTA_TAB.1 provides actionable warnings against misuse of these interfaces.
O.PATCHED_SOFTWARE	[PP_BASE_VIRTUALIZATION_V1.1]☝ FPT_TUD_EXT.1/HMC, FPT_TUD_EXT.1/SVR and FPT_TUD_EXT.1/VIOS enforce integrity of software updates.
O.VM_ENTROPY	[PP_BASE_VIRTUALIZATION_V1.1]☝ FCS_ENT_EXT.1 ensures that entropy of one VM does not affect other VM's on the same platform. [PP_BASE_VIRTUALIZATION_V1.1]☝ FCS_RBG_EXT.1/SVR defines the random bit generator used for Guest VMs.
O.AUDIT	[PP_BASE_VIRTUALIZATION_V1.1]☝ FAU_GEN.1 defines the auditable events that must be generated to diagnose the cause of unexpected system behavior. [PP_BASE_VIRTUALIZATION_V1.1]☝ FAU_SAR.1 ensures audit records are readable to administrators. [PP_BASE_VIRTUALIZATION_V1.1]☝ FAU_STG.1 ensures that the audit trail is protected from unauthorized deletion and modification. [PP_BASE_VIRTUALIZATION_V1.1]☝ FAU_STG_EXT.1 requires transfer of audit data to an external IT entity over a protected communication channel.
O.CORRECTLY_APPLIED_CONFIGURATI	[PP_BASE_VIRTUALIZATION_V1.1]☝ FDP_VMS_EXT.1 defines the mechanisms for transferring data between Guest VMs. It also defines a policy prohibiting data sharing between Guest VMs.
O.RESOURCE_ALLOCATION	[PP_BASE_VIRTUALIZATION_V1.1]☝ FCS_CKM_EXT.4 requires cryptographic keys in volatile memory to be destroyed or rendered unrecoverable. [PP_BASE_VIRTUALIZATION_V1.1]☝ FDP_RIP_EXT.1 requires deallocation of physical memory prior to allocating to a Guest VM. [PP_BASE_VIRTUALIZATION_V1.1]☝ FDP_RIP_EXT.2 requires deallocation of physical disk storage prior to allocating to a Guest VM.

6.2.3 Security Requirements Dependency Analysis

The following table demonstrates the dependencies of the SFRs modeled in [PP_BASE_VIRTUALIZATION_V1.1]☝, [MOD_SV_V1.1]☝, [PKG_TLS_V1.1]☝ and [PKG_SSH_V1.0]☝, and how the SFRs for the TOE resolve those dependencies.

Table 17: TOE SFR dependency analysis

Security Functional Requirement	Dependencies	Resolution
FAU_GEN.1	FPT_STM.1	The PP explicitly excludes the SFR of FPT_STM.1. Due to claiming exact conformance, this SFR is excluded from the ST as well.
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1
FAU_STG.1	FAU_GEN.1	FAU_GEN.1
FAU_STG_EXT.1	FAU_GEN.1	FAU_GEN.1
	FTP_ITC_EXT.1	FTP_ITC_EXT.1
FCS_CKM.1	FCS_COP.1	FCS_COP.1/SIG
	FCS_CKM.2	FCS_CKM.2
	FCS_CKM.4	The SFR of FCS_CKM_EXT.4 is the replacement for FCS_CKM.4 defined by the PP. This SFR enhances FCS_CKM.4
FCS_CKM.2	FCS_CKM.1	FCS_CKM.1
	FCS_CKM.4	The SFR of FCS_CKM_EXT.4 is the replacement for FCS_CKM.4 defined by the PP. This SFR enhances FCS_CKM.4
FCS_CKM_EXT.4	FCS_CKM.1	FCS_CKM.1
FCS_COP.1/UDE	FCS_CKM.1	FCS_CKM.2 The symmetric cipher operation is employed as part of the trusted channel as specified by FTP_ITC_EXT.1. The trusted channel uses a key agreement schema defined by FCS_CKM.2 to derive the symmetric session keys. Hence, the dependency for key generation is met by the SFR claiming the key agreement mechanism.
	FCS_CKM.4	The SFR of FCS_CKM_EXT.4 is the replacement for FCS_CKM.4 defined by the PP. This SFR enhances FCS_CKM.4
FCS_COP.1/HASH	FCS_CKM.1	The hashing operation does not require any key material.
	FCS_CKM.4	The SFR of FCS_CKM_EXT.4 is the replacement for FCS_CKM.4 defined by the PP. This SFR enhances FCS_CKM.4
FCS_COP.1/SIG	FCS_CKM.1	FCS_CKM.1
	FCS_CKM.4	The SFR of FCS_CKM_EXT.4 is the replacement for FCS_CKM.4 defined by the PP. This SFR enhances FCS_CKM.4

Security Functional Requirement	Dependencies	Resolution
FCS_COP.1/KEYEDHASH	FCS_CKM.1	The keyed hash cipher operation is employed as part of the trusted channel as specified by FTP_ITC_EXT.1. The trusted channel uses a key agreement schema defined by FCS_CKM.2 to derive the keyed hash keys. Hence, the dependency for key generation is met by the SFR claiming the key agreement mechanism.
	FCS_CKM.4	The SFR of FCS_CKM_EXT.4 is the replacement for FCS_CKM.4 defined by the PP. This SFR enhances FCS_CKM.4
FCS_RBG_EXT.1/HMC	FCS_COP.1	The SFR FCS_COP.1 are covered by FCS_COP.1/UDE, FCS_COP.1/HASH, FCS_COP.1/SIG, FCS_COP.1/KEYEDHASH defined by the PP. This SFR enhances FCS_COP.1
FCS_RBG_EXT.1/SVR	FCS_COP.1	The SFR FCS_COP.1 are covered by FCS_COP.1/UDE, FCS_COP.1/HASH, FCS_COP.1/SIG, FCS_COP.1/KEYEDHASH defined by the PP. This SFR enhances FCS_COP.1
FCS_ENT_EXT.1	FCS_RBG_EXT.1	FCS_RBG_EXT.1/SVR
FDP_HBI_EXT.1	FDP_VMS_EXT.1	FDP_VMS_EXT.1
FIA_X509_EXT.1	FPT_STM.1	The PP explicitly excludes the SFR of FPT_STM.1. Due to claiming exact conformance, this SFR is excluded from the ST as well.
FCS_HTTPS_EXT.1	FCS_TLSS_EXT.1	FCS_TLSS_EXT.1
FDP_PPR_EXT.1	FDP_HBI_EXT.1	FDP_HBI_EXT.1
	FMT_SMR.1	The PP explicitly excludes the SFR of FMT_SMR.1. Due to claiming exact conformance, this SFR is excluded from the ST as well.
FDP_RIP_EXT.1	No dependencies	
FDP_RIP_EXT.2	No dependencies	
FIA_PMG_EXT.1	FIA_UIA_EXT.1	FIA_UIA_EXT.1
FIA_UAU.5	No dependencies	
FMT_SMO_EXT.1	No dependencies	
FPT_EEM_EXT.1/HMC	No dependencies	
FPT_EEM_EXT.1/SVR	No dependencies	
FDP_VMS_EXT.1	No dependencies	
FDP_VNC_EXT.1	FDP_VMS_EXT.1	FDP_VMS_EXT.1

Security Functional Requirement	Dependencies	Resolution
	FMT_SMR.1	The PP excludes the SFR of FMT_SMR.1. Due to claiming exact conformance, this SFR is excluded from the ST as well.
FIA_AFL_EXT.1	FIA_UIA_EXT.1	FIA_UIA_EXT.1
FIA_UIA_EXT.1	FIA_UAU.5	FIA_UAU.5
FPT_DVD_EXT.1	FPT_VDP_EXT.1	FPT_VDP_EXT.1
FPT_HAS_EXT.1	No dependencies	
FPT_HCL_EXT.1	FMT_SMR.1	The PP explicitly excludes the SFR of FMT_SMR.1. Due to claiming exact conformance, this SFR is excluded from the ST as well.
FPT_RDM_EXT.1	FDP_VMS_EXT.1	FDP_VMS_EXT.1
FPT_TUD_EXT.1/HMC	FCS_COP.1	FCS_COP.1/SIG
FPT_TUD_EXT.1/SVR	FCS_COP.1	FCS_COP.1/SIG
FPT_TUD_EXT.1/VIOS	FCS_COP.1	FCS_COP.1/SIG
FPT_VDP_EXT.1	FPT_VIV_EXT.1	FPT_VIV_EXT.1
FPT_VIV_EXT.1	FDP_PPR_EXT.1	FDP_PPR_EXT.1
	FDP_VMS_EXT.1	FDP_VMS_EXT.1
FTA_TAB.1	No dependencies	
FMT_MOF_EXT.1	No dependencies	
FTP_ITC_EXT.1	FAU_STG_EXT.1	FAU_STG_EXT.1
FTP_TRP.1	No dependencies	
FTP_UIF_EXT.1	No dependencies	
FTP_UIF_EXT.2	No dependencies	
FCS_TLSS_EXT.1	FCS_CKM.2	FCS_CKM.2
	FCS_COP.1	FCS_COP.1/UDE FCS_COP.1/HASH FCS_COP.1/KEYEDHASH FCS_COP.1/SIG
	FCS_RBG_EXT.1	FCS_RBG_EXT.1/HMC FCS_RBG_EXT.1/SVR
	FIA_X509_EXT.1	FIA_X509_EXT.1
FCS_TLS_EXT.1	FCS_CKM.2	FCS_CKM.2
	FCS_COP.1	FCS_COP.1/UDE FCS_COP.1/HASH FCS_COP.1/KEYEDHASH FCS_COP.1/SIG
	FCS_RBG_EXT.1	FCS_RBG_EXT.1/HMC FCS_RBG_EXT.1/SVR

Security Functional Requirement	Dependencies	Resolution
	FIA_X509_EXT.1	FIA_X509_EXT.1
FCS_SSH_EXT.1	FCS_CKM.1	FCS_CKM.1
	FCS_CKM.2	FCS_CKM.2
	FCS_COP.1	FCS_COP.1/UDE FCS_COP.1/HASH FCS_COP.1/KEYEDHASH FCS_COP.1/SIG
	FCS_RBG_EXT.1	FCS_RBG_EXT.1/HMC FCS_RBG_EXT.1/SVR
	FIA_X509_EXT.1	FIA_X509_EXT.1
	FPT_STM.1	The PP explicitly excludes the SFR of FPT_STM.1. Due to claiming exact conformance, this SFR is excluded from the ST as well.
FCS_SSHC_EXT.1	FCS_CKM.1	FCS_CKM.1
	FCS_CKM.2	FCS_CKM.2
	FCS_COP.1	FCS_COP.1/UDE FCS_COP.1/HASH FCS_COP.1/KEYEDHASH FCS_COP.1/SIG
	FCS_RBG_EXT.1	FCS_RBG_EXT.1/HMC FCS_RBG_EXT.1/SVR
	FIA_X509_EXT.1	FIA_X509_EXT.1
	FPT_STM.1	The PP explicitly excludes the SFR of FPT_STM.1. Due to claiming exact conformance, this SFR is excluded from the ST as well.
FCS_SSHS_EXT.1	FCS_CKM.1	FCS_CKM.1
	FCS_CKM.2	FCS_CKM.2
	FCS_COP.1	FCS_COP.1/UDE FCS_COP.1/HASH FCS_COP.1/KEYEDHASH FCS_COP.1/SIG
	FCS_RBG_EXT.1	FCS_RBG_EXT.1/HMC
	FIA_X509_EXT.1	FIA_X509_EXT.1
	FPT_STM.1	The PP explicitly excludes the SFR of FPT_STM.1. Due to claiming exact conformance, this SFR is excluded from the ST as well.

6.2.4 Internal consistency and mutual support of SFRs

6.3 Security Assurance Requirements

The security assurance requirements (SARs) for the TOE are defined in the PP_BASE_VIRTUALIZATION_V1.1 protection profile.

The following table shows the SARs, and the operations performed on the components according to CC part 3: iteration (Iter.), refinement (Ref.), assignment (Ass.) and selection (Sel.).

Table 18: SARs

Security Assurance Class	Security Assurance Requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
ASE Security Target evaluation	ASE_CCL.1 Conformance claims	PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
	ASE_ECD.1 Extended components definition	PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
	ASE_INT.1 ST introduction	PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
	ASE_OBJ.2 Security objectives	PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
	ASE_REQ.2 Derived security requirements	PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
	ASE_SPD.1 Security problem definition	PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
	ASE_TSS.1 TOE summary specification	PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
ADV Development	ADV_FSP.1 Basic functional specification	PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
AGD Guidance documents	AGD_OPE.1 Operational user guidance	PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
	AGD_PRE.1 Preparative procedures	PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
ALC Life-cycle support	ALC_CMC.1 Labelling of the TOE	PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
	ALC_CMS.1 TOE CM coverage	PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No

Security Assurance Class	Security Assurance Requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
	ALC_TSU_EXT.1	PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
ATE Tests	ATE_IND.1 Independent testing - conformance	PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No
AVA Vulnerability assessment	AVA_VAN.1 Vulnerability survey	PP_BASE_VIRTUALIZATION_V1.1	No	No	No	No

6.4 Security Assurance Requirements Rationale

SAR rationales are provided by the PPs to which this ST conforms. Section 2 contains the list of PPs.

7 TOE Summary Specification

7.1 TOE Security Functionality

The TSS page numbers in Table 19 provide a quick index to each SFR's TSS entry in Table 20 of the next section.

Table 19: TSS index

SFR	TSS Page
FAU_GEN.1	62
FAU_SAR.1	63
FAU_STG.1	63
FAU_STG_EXT.1	63
FCS_CKM.1	64
FCS_CKM.2	65
FCS_CKM_EXT.4	65
FCS_COP.1/UDE	66
FCS_COP.1/Hash	66
FCS_COP.1/Sig	67
FCS_COP.1/KeyedHash	67
FCS_RBG_EXT.1/HMC	67
FCS_RBG_EXT.1/SVR	68
FCS_ENT_EXT.1	68
FCS_HTTPS_EXT.1	80
FCS_SSH_EXT.1	81
FCS_SSHC_EXT.1	83
FCS_SSHC_EXT.1	83
FCS_TLS_EXT.1	79
FCS_TLSS_EXT.1	79
FDP_HBI_EXT.1	68
FDP_PPR_EXT.1	69
FDP_RIP_EXT.1	70
FDP_RIP_EXT.2	71
FDP_VMS_EXT.1	71
FDP_VNC_EXT.1	71
FIA_UAU.5	72
FIA_AFL_EXT.1	72

SFR	TSS Page
FIA_AFL_EXT.1	81
FIA_UIA_EXT.1	72
FIA_X509_EXT.1	80
FMT_MOF_EXT.1	73
FMT_SMO_EXT.1	73
FPT_DVD_EXT.1	74
FPT_EEM_EXT.1/HMC FPT_EEM_EXT.1/SVR	74
FPT_HAS_EXT.1	74
FPT_HCL_EXT.1	75
FPT_RDM_EXT.1	76
FPT_TUD_EXT.1/HMC FPT_TUD_EXT.1/SVR FPT_TUD_EXT.1/VIOS	76
FPT_VDP_EXT.1	77
FPT_VIV_EXT.1	77
FTA_TAB.1	78
FTP_ITC_EXT.1	78
FTP_TRP.1	80
FTP_UIF_EXT.1	78
FTP_UIF_EXT.2	78

7.1.1 TOE SFR compliance rationale

Table 20 provides the rationale for how the TOE complies with each of the SFRs in Section 6.1. Table 20 uses the following abbreviations.

- Summary--Description of how the TOE meets the SFR
- AA--Assurance Activity
- N/A--Not Applicable
- Resp--Response

Table 20: TOE SFR compliance rationale

TOE SFRs	TOE SFR compliance rationale
FAU_GEN.1 Audit Record Generation	**Objective(s):** O.AUDIT **Summary:** The TOE provides a general facility to collect data required for auditing. This function is provided by the HMC collects and records system audit data. The audit record contains at minimum the contents described below.

TOE SFRs	TOE SFR compliance rationale	
	AA	The evaluator shall check the TSS and ensure that it lists all of the auditable events and provides a format for audit records. Each audit record format type shall be covered, along with a brief description of each field. The evaluator shall check to make sure that every audit event type mandated by the PP-Configuration is described in the TSS.
	Resp	The HMC component of the TOE generates audit records the start-up and shutdown of the audit functions and all other auditable events listed in 10, 11, 12, and 13. Several record types are generated. In general, the information recorded in the audit records contains: • user name, e.g., root • process ID, e.g., 1658 • client name, e.g., IBM.ServiceRMd • client PID, e.g., 2895 • hmcSessionID, e.g., none • timestamp, e.g., 2023-09-19 18:43:03.246 • thread ID, e.g., Thread-874--2052162 • return code, e.g., returned 488, • error message, e.g., VIOLOG: Error handle is either invalid or corrupt
FAU_SAR.1 Audit Review	**Objective(s):** O.AUDIT **Summary:** The TSF provides administrators the capability to read all information from the audit records. Administrators can view the audit records via the HMC console.	
	AA	None
	Resp	N/A
FAU_STG.1 Protected Audit Trail Storage	**Objective(s):** O.AUDIT **Summary:** The HMC component of the TOE generates audit records. Administrators must log on to the HMC in order to review audit records. The TOE protects the audit trail from unauthorized users.	
	AA	The evaluator shall ensure that the TSS describes how the audit records are protected from unauthorized modification or deletion. The evaluator shall ensure that the TSS describes the conditions that must be met for authorized deletion of audit records.
	Resp	The HMC component of the TOE generates audit records. The audit records are stored on a external server. The HMC Administrator, an authorized user, can upload the records to the external audit server. Unauthorized users cannot login to the HMC and therefore cannot modify or delete them and cannot upload the records to the external audit server. No user including administrative users can modify or delete the local audit files; they can only view the them.
FAU_STG_EXT.1 Off-Loading of Audit Data	**Objective(s):** O.AUDIT **Summary:** The HMC transfers the audit records to external audit storage using SFTP (via SSH)	

<table>
<tr><th>TOE SFRs</th><th colspan="2">TOE SFR compliance rationale</th></tr>
<tr><td></td><td>AA</td><td>The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided.</td></tr>
<tr><td></td><td>Resp</td><td>The HMC component of the TOE provides commands to offload audit log data to external audit storage via SFTP.
<pre>
    1) cpfile -t vpp -l r -f file-name -o export -h host-
name -u user-ID [--passwd password] [--repopasswd <password>]
    2) cpfile -t vpp -l r -f file-name -o import -h host-
name -u user-ID [--passwd password] [--repopasswd <password>]
</pre></td></tr>
<tr><td></td><td>AA</td><td>The evaluator shall examine the TSS to ensure it describes what happens when the local audit data store is full.</td></tr>
<tr><td></td><td>Resp</td><td>When the local audit data store is full, the TOE overwrites previous audit records starting with the oldest records. There is no scenario in which the local audit data store becomes full. The TOE enforces log rotation policies that automatically manage disk usage. When a log file reaches a specified size threshold, it is archived and a new log file is created. A maximum number of archived log files is maintained; once this limit is reached, the oldest file is deleted. This ensures that the total disk space used by log data remains bounded and does not grow indefinitely.</td></tr>
<tr><td>FCS_CKM.1
Cryptographic Key Generation</td><td colspan="2"><b>Objective(s):</b> O.MANAGEMENT_ACCESS, O.VMM_INTEGRITY
<b>Summary:</b> See description below.</td></tr>
<tr><td></td><td>AA</td><td>The evaluator shall ensure that the TSS identifies the key sizes supported by the TOE. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme.</td></tr>
<tr><td></td><td>Resp</td><td>Table 21 lists the cryptographic modules and, for each module, the asymmetric algorithms, key sizes, curves, standards, and usages used to satisfy the ST claims. (SP800-56B Key Establishment is not claimed.)
<p><b>Table 21: Asymmetric key generation, verification, and establishment algorithms</b></p>
<table>
<tr><th>Module</th><th>Algorithm</th><th>Capabilities</th><th>Standard</th><th>Usage</th></tr>
<tr><td rowspan="3">Java cryptographic library</td><td>RSA KeyGen</td><td>Modulo: 2048</td><td>FIPS186-5</td><td rowspan="3">HTTPS server;
HTTPS client;
TLS server;
authentication</td></tr>
<tr><td>RSA key establishment</td><td>Modulo: 2048</td><td>RFC8017</td></tr>
<tr><td>Elliptic curve-based key establishment (KAS-ECC-SSC Sp800-56Ar3)</td><td>Curves: P-256, P-384, P-521</td><td>SP800-56A-Rev3</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>TOE SFRs</th><th colspan="2">TOE SFR compliance rationale</th></tr>
<tr><td></td><td></td><td><table>
<tr><td rowspan="2">OpenSSH</td><td>Elliptic curve-based key establishment (KAS-ECC-SSC Sp800-56Ar3)</td><td>Curves:
P-256, P-384</td><td>SP800-56A-Rev3</td><td rowspan="2">SSH client;
SSH server;
key pair generation, authentication</td></tr>
<tr><td>RSA KeyGen</td><td>Modulo: 2048</td><td>FIPS186-5</td></tr>
<tr><td>OpenSSL</td><td>RSA KeyGen</td><td>Modulo: 2048</td><td>RFC8017</td><td>key pair generation, certificate management</td></tr>
</table></td></tr>
<tr><td rowspan="3">FCS_CKM.2
Cryptographic Key Distribution</td><td colspan="2"><b>Objective(s):</b> O.MANAGEMENT_ACCESS
<b>Summary:</b> The HMC supports RSA key establishment with 2048-bit keys as used in TLS and EC-Diffie-Hellman with P-256 and P-384 as used in SSH.</td></tr>
<tr><td>AA</td><td>The evaluator shall ensure that the supported key establishment schemes correspond to the key generation schemes identified in FCS_CKM.1.1. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme.</td></tr>
<tr><td>Resp</td><td>The TOE supports the following key establishment scheme:
• RSA-based key establishment schemes according to 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"
• ECC Key establishment scheme using EC-Diffie-Hellman with curves P-256 and P-384, P-521 that meets the following: SP800-56A-Rev3
RSA-based key establishment is used in TLS sessions when ciphersuites with RSA key exchange are negotiated.
EC-Diffie-Hellman key establishment is used in SSH client and SSH server and TLS sessions when ciphersuites with ECDHE key exchange are negotiated.</td></tr>
<tr><td rowspan="3">FCS_CKM_EXT.4
Cryptographic Key Destruction</td><td colspan="2"><b>Objective(s):</b> O.DOMAIN_INTEGRITY, O.RESOURCE_ALLOCATION, O.VM_ISOLATION
<b>Summary:</b> For key and key materials, they are destroyed upon powering off of the TOE system.</td></tr>
<tr><td>AA</td><td>The evaluator shall check to ensure the TSS lists each type of key and its origin and location in memory or storage. The evaluator shall verify that the TSS describes when each type of key is cleared.</td></tr>
<tr><td>Resp</td><td>For HMC, the TOE stores keys and certificates used for TLS communication, and keys used for SSH communication at the following locations:
• HMC pkcs12 file where certificate and key is attached: /etc/httpd/conf.d/hmcserverKey.pem
• HMC key: /etc/httpd/conf.d/hmcserverKey.key
• HMC certificate: /etc/httpd/conf.d/hmcserverKey.crt</td></tr>
</table>

TOE SFRs	TOE SFR compliance rationale	
		The TOE stores keys used for SSH communication at the following locations: • SSH Key pairs for the host machine: /etc/ssh/ssh_host_rsa_key and ssh_host_rsa_key.pub • SSH User-created keys: ~/.ssh/id_rsa and id_rsa.pub The administrator guidance provide detailed instructions on how to delete keys and certificates stored in the TOE. For the Server component, all cryptographic keys, including RSA and ECDSA keys for digital signature generation and verification, RSA and Elliptic Curve keys used for key establishment by TLS 1.2, AES-256-GCM keys, and HMAC-SHA-384 keys used by TLS 1.2, are stored in RAM and are deleted when the TOE powers off.
FCS_COP.1/UDE Cryptographic Operation (AES Data Encryption/Decryption)	**Objective(s):** O.MANAGEMENT_ACCESS **Summary:** The TOE performs encryption and decryption using the following algorithms: • AES-GCM with 256-bit key size as specified in NIST SP 800-38D • AES-CTR with 128-bit key size as specified in NIST SP 800-38A	
	AA	None.
	Resp	N/A
FCS_COP.1/HASH Cryptographic Operation (Hashing)	**Objective(s):** O.MANAGEMENT_ACCESS **Summary:** The TOE supports the hash functions listed below according to FIPS PUB 180-4 "Secure Hash Standard": • SHA-1 used by ssh-rsa in SSH authentication, and for published hash verification of HMC image updates. • SHA-256 used by hmac-sha-256 for data integrity in SSH and digital signatures of certificates in TLS. Used in RSA digital signature verification of Server image updates using RSA with SHA-256. Also used by ecdh-sha2-nistp256 for SSH key exchange. Finally used for published hash verification of VIOS image updates. • SHA-384 used by hmac-sha-384 for data integrity in TLS. Also used by ecdh-sha2-nistp384 for SSH key exchange.	
	AA	The evaluator shall check that the association of the hash function with other TSF cryptographic functions (for example, the digital signature verification function) is documented in the TSS.
	Resp	The TOE supports cryptographic hashing services conforming to FIPS PUB 180-4. The TOE supports the following hash algorithms: SHA-1, SHA-256, and SHA-384. The message digest sizes supported are: 160 bits, 256 bits, and 384. The hashing algorithms are used for authentication, digital signature services, HMAC services and image update verifications. SHA-1 is used by the ssh-rsa as an SSH authentication method and in published hash verification of HMC image updates. SHA-256 is used by RSA for digital signature verification of certificates in the Server component of the TOE, it is used for data integrity in SSH using HMAC-SHA-256, it is used in RSA digital signature verification of Server image updates, it is also used by ecdh-sha2-nistp256 for SSH key exchange, finally used for published hash verification of VIOS image updates. SHA-384 is used by HMAC-SHA-384 in the TLS ciphersuites TLS_RSA_WITH_AES_256_GCM_SHA348,

TOE SFRs	TOE SFR compliance rationale	
		TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, and TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384, and it is used by ecdh-sha2-nistp384 for SSH key exchange. Updates to the HMC are performed via the CLI. The update mechanism is described in the guidance document. The HMC validates the TOE's update image by verifying the hash for the HMC update driver via the sha1sum command performed manually by an authorized administrator. Updates to the Server are performed through the HMC GUI. The update mechanism is described in the guidance document.
FCS_COP.1/SIG Cryptographic Operation (Signature Algorithms)	**Objective(s):** O.MANAGEMENT_ACCESS **Summary:** The TOE provides cryptographic signature generation and verification using the following schemes compliant with FIPS 186-5. • RSA schemes with 2048-bit key size according to FIPS PUB 186-5. • ECDSA schemes using “NIST curves” P-256, P-384 and P-521 according to FIPS PUB 186-5 The signatures for SSH can be generated and verified via the CLI on the HMC. For TLS, they are generated and verified as part of the TLS session establishment.	
	AA	None.
	Resp	N/A
FCS_COP.1/ KEYEDHASH Cryptographic Operation (Keyed Hash Algorithms)	**Objective(s):** O.MANAGEMENT_ACCESS **Summary:** The TOE provides HMAC-SHA-384 that is used in TLS 1.2 ciphersuites TLS_RSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, and TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384. It also uses HMAC-SHA-256 in SSH for data integrity.	
	AA	The evaluator shall examine the TSS to ensure that it specifies the following values used by the HMAC function: key length, hash function used, block size, and output MAC length used.
	Resp	The TOE provides HMAC-SHA-384 that is used in TLS 1.2 ciphersuites TLS_RSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, and TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384. The key length is 384-bits, the blocksize is 1024 bits, the output MAC length is 384 bits. It also provides HMAC-SHA-256 that is used in SSH. The key length is 256-bits, the blocksize is 512-bits and the output MAC length is 256-bits.
FCS_RBG_EXT.1/ HMC Cryptographic Operation (Random Bit Generation) (HMC)	**Objective(s):** O.DOMAIN_INTEGRITY, O.MANAGEMENT_ACCESS, O.VM_ENTROPY, O.VMM_INTEGRITY **Summary:** The TOE implements its own deterministic random bit generator (DRBG) functionality. It implements CTR_DRBG in OpenSSL as the default DRBG. Whenever OpenSSH generates keys, it will use OpenSSL's CTR_DRBG. More details are provided in the proprietary Entropy Assessment Report.	
	AA	None
	Resp	N/A

TOE SFRs	TOE SFR compliance rationale	
FCS_RBG_EXT.1/ SVR Cryptographic Operation (Random Bit Generation) (Server)	**Objective(s):** O.DOMAIN_INTEGRITY, O.MANAGEMENT_ACCESS, O.VM_ENTROPY, O.VMM_INTEGRITY **Summary:** The TOE uses Java which implements the Hash_DRBG with SHA-256 in SecureRandom. More details are provided in the proprietary Entropy Assessment Report.	
	AA	None
	Resp	N/A
FCS_ENT_EXT.1 Entropy for Virtual Machines	**Objective(s):** O.DOMAIN_INTEGRITY, O.VM_ENTROPY **Summary:** The Server uses the underlying IBM Power10 processor hardware entropy source which is based upon Ring Oscillators to create entropy. A separate entropy report has been provided from IBM describing this entropy source in great technical detail, including the design, assumptions, theoretical models, statistical results and health tests. In addition, a Public Use Document has been provided which summarizes the high-level details of the entropy source. The DARN instruction is used by a caller (e.g. API) of the entropy source to obtain entropy. The entropy source provides 32 bits of entropy per 64 bits provided by the DARN instruction.	
	AA	The evaluator shall verify that the TSS describes how the TOE provides entropy to Guest VMs, and how to access the interface to acquire entropy or random numbers. The evaluator shall verify that the TSS describes the mechanisms for ensuring that one VM does not affect the entropy acquired by another.
	Resp	The PowerVM Hypervisor provides entropy to Guest VMs by exposing the Power10 hardware DARN (Deliver A Random Number) instruction through hypervisor calls. For IBM i partitions, the random(uint64&) hypervisor call returns a 64-bit random value sourced directly from the DARN instruction. For RPA partitions, the H_Random hypervisor call provides the same functionality. Each Guest VM accesses entropy independently via its own hypervisor call context. The hypervisor ensures isolation between VMs by preventing shared state or reuse of entropy values across partitions. The DARN instruction is a hardware-based random number generator that provides high-quality entropy, and its output is not influenced by software state or other VMs.
FDP_HBI_EXT.1 Hardware-Based Isolation Mechanisms	**Objective(s):** O.PLATFORM_INTEGRITY **Summary:** See description below.	
	AA	The evaluator shall ensure that the TSS provides evidence that hardware-based isolation mechanisms are used to constrain VMs when VMs have direct access to physical devices, including an explanation of the conditions under which the TSF invokes these protections.
	Resp	HMC users can create VMs from the HMC GUI and the underlying hypervisor which provides features to have them isolated. The VMs are listed as independent isolated machines from the HMC GUI. The components of the TOE protect themselves using the domains provided by the PowerVM processors. The Hypervisor operates in the privileged domain and the partitions, like VIOS, operate in the unprivileged domain. This allows the Hypervisor to protect itself as well as the resources it makes selectively available to the applicable partitions. For example, when a partition references memory, they can only reference the

TOE SFRs	TOE SFR compliance rationale	
		memory though a page table that translates a virtual real address. The page table is in physical real memory and cannot be accessed by VMs. The physical cores (processors) on the server have registers that can only be changed when running in the hypervisor privileged state. An example of this protection is whenever a virtual processor is dispatched to run on a physical core, the hypervisor sets a register to the location of the page table in use by that VM. These features of the PowerVM servers are always active.
FDP_PPR_EXT.1 Physical Platform Resource Controls	**Objective(s):** O.VM_ISOLATION, O.VMM_INTEGRITY, O.PLATFORM_INTEGRITY **Summary:** HMC users can assign resources to VMs from HMC GUI and underlying hypervisor provides feature to have them isolated. These assigned resources are listed as independent resources of VMs on HMC GUI.	
	AA	The evaluator shall examine the TSS to determine that it describes the mechanism by which the VMM controls a Guest VM's access to physical platform resources. This description shall cover all of the physical platforms allowed in the evaluated configuration by the ST. It should explain how the VMM distinguishes among Guest VMs, and how each physical platform resource that is controllable (that is, listed in the assignment statement in the first element) is identified to an Administrator. The evaluator shall ensure that the TSS describes how the Guest VM is associated with each physical resource, and how other Guest VMs cannot access a physical resource without being granted explicit access. For TOEs that implement a robust interface (other than just "allow access" or "deny access"), the evaluator shall ensure that the TSS describes the possible operations or modes of access between a Guest VM's and physical platform resources. If physical resources are listed in the second element, the evaluator shall examine the TSS and operational guidance to determine that there appears to be no way to configure those resources for access by a Guest VM. The evaluator shall document in the evaluation report their analysis of why the controls offered to configure access to physical resources can't be used to specify access to the resources identified in the second element (for example, if the interface offers a drop-down list of resources to assign, and the denied resources are not included on that list, that would be sufficient justification in the evaluation report).
	Resp	Each Guest VM is managed from the HMC GUI. The individual Guest VMs are isolated by the underlying hypervisor which ensures the physical resources assigned to the Guest VM are isolated. There are some PCIe devices that are hubs with multiple devices behind each hub. These are also defined as cable cards and cannot be assigned to partitions. These are: • PCIe4 4-port NVMe JBOF adapter (FC EJ1X and EJ1Y; CCIN 6B87) • PCIe4 cable adapter (FC EJ24; CCIN 6B92) The HMC shows these devices in the system list of Physical I/O adapters, which appears in the HMC as NVMe JBOF Card. The HMC does not allow these devices to be assigned to partitions. In PowerVM, internal devices are identified by a token called the Dynamic Reconfiguration Connector (DRC), which is used by the Hypervisor and PowerVM partitions.

TOE SFRs	TOE SFR compliance rationale	
		There are virtual DRCs and physical DRCs which are categorized as follows: • Virtual Processor DRCs: The DRC values for virtual processor are of the form 0x1000xxxx where 10 indicates virtual processor and xxxx is the logical processor index for the specific partition. The first virtual processor assigned to any partition would be 0x10000000, the second would be 0x10000001 and so on. These values are not unique across the server (ex. The first virtual processor for every partition is 0x10000000). The hypervisor manages the mapping of logical processors to physical resources. • Virtual Memory DRCs: Memory is assigned to partition in Logical Memory Block (LMB) sizes which can be 128MB, 256MB, 1024MB, 2048MB or 4096MB. Customer can choose the LMB size on a server basis. When assigning memory to partitions from the HMC, the allowed values are in multiples of the LMB size. So, if the LMB size is 256MB and 2 GB is assigned to the partition, then 8 LMBs would be assigned to the partition. The DRC values for virtual processor are of the form 0x8000xxxx where 80 indicates virtual processor and xxxx is the logical index of the LMB for the specific partition. The first logical LMB assigned to any partition would be 0x80000000, the second would be 0x80000001 and so on. These values are not unique across the server (ex. The first logical LMB for every partition is 0x80000000). The hypervisor manages the mapping of logical LMBs to physical resources. • Virtual Device DRCs: Like virtual processors and virtual memory DRCs are virtual devices. The DRC values for virtual devices are of the form 0x30ddxxxx where 30 indicates virtual device, dd indicates the type of device and xxxx is the logical index of the device for the specific partition. The first logical device of a given type assigned to any partition would be 0x30dd0000, the second would be 0x30dd0001 and so on. These values are not unique across the server (ex. The first logical device for every partition is 0x30dd0000). The hypervisor manages the mapping of virtual devices to resources. • PCI device DRCs: The DRC values for physical PCI resource are of the form 0x21ttssbb where tt is the subtype of the devices, ss is the slot number and bb is the bus number. Unlike virtual devices, these DRC values are global across the server.
FDP_RIP_EXT.1 Residual Information in Memory	**Objective(s):** O.VM_ISOLATION, O.DOMAIN_INTEGRITY, O.RESOURCE_ALLOCATION **Summary:** See description below.	
	AA	The evaluator shall ensure that the TSS documents the process used for clearing physical memory prior to allocation to a Guest VM, providing details on when and how this is performed. Additionally, the evaluator shall ensure that the TSS documents the conditions under which physical memory is not cleared prior to allocation to a Guest VM and describes when and how the memory is cleared.
	Resp	The total memory installed on the server is divided into multiple Logical Memory Blocks (LMBs). The size of the LMB is set at server power on and can be 128MB, 256MB, 1024MB, 2048MB or 4096MB. Individual LMBs are assigned to the hypervisor or to individual VMs. Each LMB is owned by one and only one partition at any moment in time. At server boot, all of the LMBs are zeroed by both the hardware and firmware. When VMs

TOE SFRs	TOE SFR compliance rationale	
		are created, these LMBs are assigned to the individual VMs based on the configuration set by the administrator. PowerVM does support the ability to dynamically remove LMBs from one VM and assign the LMBs to other VMs. When memory is removed from a VM, the hypervisor ensures there are no entries in the page table that point to the memory being removed. The hypervisor also zeros the content of the memory in both the caches and in the memory to ensure the receiving partition is not able to retrieve the previous content of the memory.
FDP_RIP_EXT.2 Residual Information on Disk	**Objective(s):** O.VM_ISOLATION, O.DOMAIN_INTEGRITY, O.RESOURCE_ALLOCATION **Summary:** Residual information on physical disk storage can be cleared manually by the administrators issuing CLI commands.	
	AA	The evaluator shall ensure that the TSS documents how the TSF ensures that disk storage is zeroed upon allocation to Guest VMs. Also, the TSS must document any conditions under which disk storage is not cleared prior to allocation to a Guest VM. Any file system format and metadata information needed by the evaluator to perform the below test shall be made available to the evaluator, but need not be published in the TSS.
	Resp	Physical disk storage is cleared prior to allocating to a guest VM. This must be performed by the administrator manually. The procedure for doing this is provided in the Evaluated Configuration Guide.
FDP_VMS_EXT.1 VM Separation	**Objective(s):** O.VM_ISOLATION, O.VMM_INTEGRITY, O.PLATFORM_INTEGRITY, O.DOMAIN_INTEGRITY, O.CORRECTLY_APPLIED_CONFIGURATION **Summary:** VMs can be part of different network which will not allow the communication between them. If VMs are in same network then it can transfer data to one another. The administrator must explicitly enable VM to VM communication either by configuring a virtual ethernet connection. By default, there are no network connections enabled between partitions. PowerVM does not provide any features that allow sharing of memory contents between VMs. The administrator can copy a file from one VM to another using scp.	
	AA	The evaluator shall examine the TSS to verify that it documents all inter-VM communications mechanisms (as defined above), and explains how the TSF prevents the transfer of data between VMs outside of the mechanisms listed in FDP_VMS_EXT.1.1.
	Resp	The administrator must explicitly enable VM to VM communication either by configuring a virtual ethernet connection. By default, there are no network connections enabled between partitions. PowerVM does not provide any features that allow sharing of memory contents between VMs.
FDP_VNC_EXT.1 Virtual Networking Components	**Objective(s):** O.VM_ISOLATION, O.VMM_INTEGRITY, O.PLATFORM_INTEGRITY, O.DOMAIN_INTEGRITY, **Summary:** See description below.	
	AA	The evaluator shall examine the TSS (or a proprietary annex) to verify that it describes the mechanism by which virtual network traffic is ensured to be visible only to Guest VMs configured to be on that virtual network.

TOE SFRs	TOE SFR compliance rationale	
	Resp	The administrator has the ability to configure virtual switches, virtual LANs and physical network devices to allow/prevent the flow of data across virtual/physical network. Each VM has a vNIC providing vSwitching (VLAN tagging and packet forwarding) for network traffic isolation.
FIA_UAU.5 Multiple Authentication Mechanisms	**Objective(s):** O.VM_ISOLATION, O.VMM_INTEGRITY, O.PLATFORM_INTEGRITY, O.DOMAIN_INTEGRITY, **Summary:** An administrative user can login to the TOE via the HMC console using username and password (for the HTTPS connections) and username and password and public key (for SSH connections). The HMC console is available via the GUI using HTTPS and the Command Line Interface (CLI) using SSH.	
	AA	None
	Resp	N/A
FIA_AFL_EXT.1 Authentication Failure Handling	**Objective(s):** O.MANAGEMENT_ACCESS **Summary:** The HMC console supports authentication failure lock mechanism. It also provides commands that can be used to set the maximum login attempts (i.e., max_login_attempts) and the login timeout (i.e., login_suspend_time values) for username and password login attempts. Additionally, the TOE supports the capability to unlock the account by another administrator when maximum attempt of login are reached.	
	AA	None
	Resp	N/A
FIA_UIA_EXT.1 Administrator Identification and Authentication	**Objective(s):** O.MANAGEMENT_ACCESS **Summary:** The HMC supports GUI access to the server via HTTPS. This login method is via username and password. The HMC supports SSH access from the CLI to the server using username and password. If the correct pair (username, password) is used, then a successful login will occur.	
	AA	The evaluator shall examine the TSS to determine that it describes the logon process for each logon method (local, remote (HTTPS, SSH, etc.)) supported for the product. This description shall contain information pertaining to the credentials allowed/used, any protocol transactions that take place, and what constitutes a “successful logon.” The evaluator shall examine the operational guidance to determine that any necessary preparatory steps (e.g., establishing credential material such as pre-shared keys, tunnels, certificates) to logging in are described. For each supported login method, the evaluator shall ensure the operational guidance provides clear instructions for successfully logging on. If configuration is necessary to ensure the services provided before login are limited, the evaluator shall determine that the operational guidance provides sufficient instruction on limiting the allowed services.
	Resp	Login to HMC requires using remote methods, which includes GUI (through HTTPS) or SSH. On successful login from GUI, HMC dashboard will be presented. On successful login from CLI, HMC command prompt will be displayed. In order to log in to the TOE via GUI, password authentication is required, while for CLI, both password and public key authentication can be used. Incorrect user name or password and incorrect private key, will not allow access to the TOE.

TOE SFRs	TOE SFR compliance rationale	
FMT_SMO_EXT.1 Separation of Management and Operational Networks	**Objective(s):** O.MANAGEMENT_ACCESS **Summary:** The TOE provides separation of management and operational networks as described below.	
	AA	The evaluator shall examine the TSS to verify that it describes how management and operational traffic is separated.
	Resp	The TOE consists of two hardware components: the Hardware Management Console (HMC) and the Power Server. The HMC is directly connected to the Power Server via a dedicated private network interface, ensuring secure and isolated communication for management functions. The HMC also connects to a separate management network via a second interface, typically accessed through a secure web browser session. While the Power Server uses a single physical Network Interface Card (NIC) to handle both management and operational traffic, logical separation is enforced through the use of virtual NICs and a virtual switch. Each Guest VM and the HMC are assigned distinct virtual NICs, which are connected to a virtual switch configured to enforce strict traffic isolation. Management traffic is logically separated from operational traffic using VLAN tagging and access control policies within the virtual switch. This ensures that Guest VMs cannot access or interfere with management functions.
FMT_MOF_EXT.1 Management of Security Functions Behavior	**Objective(s):** O.VMM_INTEGRITY, O.MANAGEMENT_ACCESS **Summary:** The TOE provides local and remote administration of the TOE through the HMC console.	
	AA	The evaluator shall examine the TSS and Operational Guidance to ensure that it describes which security management functions require Administrator privilege and the actions associated with each management function. The evaluator shall verify that for each management function and role specified in the FMT_MOF_EXT.1.1 Server Virtualization Management Functions Table (Table 3), the defined role is able to perform all mandatory functions as well as all optional or selection-based functions claimed in the ST.
	Resp	The TOE provides remote administration of the TOE through the HMC console. The HMC console can be accessed via the GUI or the CLI via SSH. The following management functions require administrator privilege: • update the Virtualization System • configure Administrator password policy as defined in FIA_PMG_EXT.1 • create, configure and delete VMs • set default initial VM configurations • configure virtual networks including VM • configure and manage the audit system and audit data • configure VM access to physical devices • configure inter-VM data sharing • configure removable media policy • configure the cryptographic functionality

TOE SFRs	TOE SFR compliance rationale	
		• change default authorization factors • configure remote connection inactivity timeout • configure lockout policy for unsuccessful authentication attempts through limiting number of attempts during a time period • configure name/address of audit/logging server to which to send audit/logging records • configure name/address of network time server • configure banner Detailed instructions on how to perform each of the above management functions are provided in the administrator guidance.
FPT_EEM_EXT.1/HMC, FPT_EEM_EXT.1/SVR Execution Environment Mitigations	**Objective(s):** O.VM_ISOLATION, O.VMM_INTEGRITY, O.PLATFORM_INTEGRITY, O.DOMAIN_INTEGRITY **Summary:** For the HMC part of the TOE, the platform provides no execution environment-based vulnerability mitigation mechanisms. For the Server part of the TOE, the platform provides the following mitigation mechanisms: • Memory execution protection (e.g., Data Execution Protection (DEP)) • Stack buffer overflow protection • Heap corruption detection	
	AA	The evaluator shall examine the TSS to ensure that it states, for each platform listed in the ST, the execution environment-based vulnerability mitigation mechanisms used by the TOE on that platform. The evaluator shall ensure that the lists correspond to what is specified in FPT_EEM_EXT.1.1.
	Resp	The TOE (HMC portion) runs in a closed and restricted environment and does not require hardware assists and memory-handling extensions from the platform. The TOE (Server portion) relies on the following execution environmentbased vulnerability mitigation mechanisms supported by the Platform: • Memory execution protection (e.g., Data Execution Protection (DEP)) • Stack buffer overflow protection • Heap corruption detection
FPT_DVD_EXT.1 Non-Existence of Disconnected Virtual Devices	**Objective(s):** O.VM_ISOLATION, O.PLATFORM_INTEGRITY, **Summary:** Virtual devices are configured on a per VM basis such that they are only accessible to the VM that owns that virtual device. A non-existent or disconnected virtual device access through a hypervisor call would fail with an error return code. The device's interface is documented in the TSS under FPT_VDP_EXT.1.	
	AA	None.
	Resp	N/A
FPT_HAS_EXT.1 Hardware Assists	**Objective(s):** O.VM_ISOLATION, O.VMM_INTEGRITY, O.PLATFORM_INTEGRITY, O.DOMAIN_INTEGRITY,	

TOE SFRs	TOE SFR compliance rationale	
	Summary: The VMM uses the Power ISA Privilege States to reduce or eliminate the need for binary translation. In addition, the VMM uses Power ISA Privilege States, Hardware Page Tables (HPTs), and Translation Control Entries (TCEs) to reduce the need for shadow page tables.	
	AA	The evaluator shall examine the TSS to ensure that it states, for each platform listed in the ST, the hardware assists and memory-handling extensions used by the TOE on that platform. The evaluator shall ensure that these lists correspond to what is specified in the applicable FPT_HAS_EXT component.
	Resp	The VMM uses Power Instruction Set Architecture (ISA) Privilege States to reduce or eliminate the need for binary translation. The VMM uses Power ISA Privilege States, Hardware Page Tables (HPTs), Translation Control Entries (TCEs) to reduce or eliminate the need for shadow page tables. Memory Address Translation is used to enforce memory access between VM's. CPU execution utilizes nested page tables and other hardware isolation mechanisms to ensure proper VM separation. Hypervisor execution ensures only the hypervisor has access to privileged instructions/memory regions.
FPT_HCL_EXT.1 Hypercall Controls	**Objective(s):** O.VM_ISOLATION, O.PLATFORM_INTEGRITY, **Summary:** The TOE does not provide a Hypercall interface where specific interfaces are configurable. It does provide predefined OS profiles that are used for supporting various operating systems.	
	AA	The evaluator shall examine the TSS (or proprietary TSS Annex) to ensure that all hypercall functions are documented at the level necessary for the evaluator to run the below test. Documentation for each hypercall interface must include: how to invoke the interface, parameters and legal values, and any conditions under which the interface can be invoked (e.g., from guest user mode, guest privileged mode, during guest boot only).
	Resp	The following Hypervisor Calls documentation presents a proprietary list of hypercall functions. The categories of the hcall interface are outlined below: • [SLIC_HCALLS_LIST] SLIC Hypervisor Calls List • [RPA_HCALLS_LIST] RPA Hypervisor Calls List • [RPA_HIDDEN_HCALLS_LIST] RPA Hidden Hypervisor Calls List • [PFW_PHYP_HIDDEN_HCALLS_LIST] PFW PowerVM Hidden Hypervisor Calls List • [PLATFORM_HCALLS_LIST] Platform Hypervisor Calls List Regarding the invocation of hypercalls, the Power hardware allows hypercall to be executed only when it is in the privileged (kernel) state. Since the IBM i operating system is a proprietary product, only IBM can write code to execute in the privileged state. For testing purposes only, a special version of IBM i is available internally that allows hypercalls to be tested. For RPA calls, the Linux operating system allows kernel extensions executed in the privileged state to invoke the RPA hypercall. All RPA hypervisor calls are common to Linux, AIX and VIOS running on Power hardware.

TOE SFRs	TOE SFR compliance rationale	
FPT_RDM_EXT.1 Removable Devices and Media	**Objective(s):** O.DOMAIN_INTEGRITY, **Summary:** The TOE supports removable devices and media. A list of media devices attached with Partition can be viewed from the HMC GUI.	
	AA	The evaluator shall examine the TSS to ensure it describes the association between the media or devices supported by the TOE and the actions that can occur when switching information domains.
	Resp	Removable devices and media can be assigned to the VM using the HMC GUI. Removable devices are assigned by the HMC to one and only one VM at time. To change the ownership of a device, e.g., from one VM to another, requires an overt action on the HMC by an authorized administrator. The VMs themselves cannot transfer the ownership of physical resources.
FPT_TUD_EXT.1/HMC, FPT_TUD_EXT.1/SVR, FPT_TUD_EXT.1/VIOS Trusted Updates to the Virtualization System	**Objective(s):** O.PATCHED_SOFTWARE, **Summary:** The HMC provides trusted software update mechanisms for updating the system firmware. Updates to the HMC have a SHA-1 hash associated with them, while updates to the VIOS have a SHA-256 hash associated with them. The Server provides software update mechanisms for updating the system software. Updates to the Server use an RSA digital signature using SHA-256 and a key size of 2048 bits.	
	AA	The evaluator shall verify that the TSS describes all TSF software update mechanisms for updating the system software. Updates to the TOE either have a hash associated with them or are signed by an authorized source. The evaluator shall verify that the description includes either a digital signature or published hash verification of the software before installation and that installation fails if the verification fails. The evaluator shall verify that the TSS describes the method by which the digital signature or published hash is verified to include how the candidate updates are obtained, the processing associated with verifying the update, and the actions that take place for both successful and unsuccessful verification. If digital signatures are used, the evaluator shall also ensure the definition of an authorized source is contained in the TSS. If the ST author indicates that a certificate-based mechanism is used for software update digital signature verification, the evaluator shall verify that the TSS contains a description of how the certificates are contained on the device. The evaluator also ensures that the TSS (or administrator guidance) describes how the certificates are installed/updated/selected, if necessary.
	Resp	The HMC images have a SHA-1 published hash value associated with them. There are pre-validation checks of the installations like certificate validation, version, architecture and size as part of the system software update process for HMC. The software installation processes will begin as soon as all pre-validations are successful. The system will boot up with a newer version if the process is successful. In the unlikely event that a pre-validation fails, the installations will be terminated. In other words after successful pre-validation during the installation processes if the binary installers fails then the system will expect manual reboot from admin and HMC will try to boot with partial installation. The VIOS images have a SHA-256 published hash value associated with them.

TOE SFRs	TOE SFR compliance rationale	
		Similarly, the Server implements an RSA SHA-256 digital signature verification with a key size of 2048 bits. There are pre-validation checks of the installations like certificate validation, version, architecture and size as part of the system software update process. The software installation process will begin as soon as all pre-validations are successful. The system will boot up with a newer version if the process is successful. In the unlikely event that a pre-validation fails, the installations will be terminated. In other words after successful pre-validation during the installation process if the binary installers fails then the system will expect manual reboot from admin and the Server will try to boot with partial installation. The firmware image digest is generated using OpenSSL. The digest is then sent to the signing server to perform the exponent operation. The signing server is an x86 based box (not associated with PowerVM) with a 4769 crypto card attached. The signed image is loaded onto the BMC. The BMC firmware runs on a service processor which is a separate chip that is located within a PowerVM system. The BMC will verify the signature using OpenSSL after the image is loaded.
FPT_VDP_EXT.1, Virtual Device Parameters	**Objective(s):** O.VM_ISOLATION, O.VMM_INTEGRITY, O.PLATFORM_INTEGRITY **Summary:** See description below.	
	AA	The evaluator shall examine the TSS to ensure it lists all virtual devices accessible by the guest OS. The TSS, or a separate proprietary document, must also document all virtual device interfaces at the level of I/O ports or PCI Bus interfaces - including port numbers (absolute or relative to a base), port name, address range, and a description of legal input values. The TSS must also describe the expected behavior of the interface when presented with illegal input values. This behavior must be deterministic and indicative of parameter checking by the TSF. The evaluator must ensure that there are no obvious or publicly known virtual I/O ports missing from the TSS. There is no expectation that evaluators will examine source code to verify the “all” part of the evaluation activity.
	Resp	Virtual devices can be assigned to VMs from the HMC. The management console maintains a list of all the possible device types and all the virtual devices assigned to individual VMs. These includes virtual ethernet, virtual disk, virtual Small Computer System Interface (SCSI), virtual fiber channel (FC), virtual optical. The developer also provided a separate proprietary document named "Power Architecture® Platform Requirements (PAPR)" ([PAPR]☝) containing description of all virtual device interfaces at the level of I/O ports or PCI Bus interfaces that are not allowed. A partition, after HMC configuration to allow communication with the virtual or physical device can make hypervisor calls to access the devices. When making the hypervisor call the hypervisor performs validation over all input values, specific to the action and virtual device. When an invalid input value is identified, the hypervisor will fail the hypervisor call and a return code will be returned to the user.
FPT_VIV_EXT.1 VMM Isolation from VMs	**Objective(s):** O.VM_ISOLATION, O.VMM_INTEGRITY, O.PLATFORM_INTEGRITY **Summary:** See description below.	

TOE SFRs	TOE SFR compliance rationale	
	AA	The evaluator shall verify that the TSS (or a proprietary annex to the TSS) describes how the TSF ensures that guest software cannot degrade or disrupt the functioning of other VMs, the VMM or the platform. And how the TSF prevents guests from invoking higher-privilege platform code, such as the examples in the note.
	Resp	PowerVM allows partitions to be configured with fixed amounts of processing resource and memory that is allocated to the specific VMs. Also, physical I/O devices can also be assigned to specific VMs. Additionally, the TOE supports ensures that an attempt to update virtual firmware or virtual BIOS cannot cause the VMM to be modified.
FTA_TAB.1 Access Banner	**Objective(s):** O.MANAGEMENT_ACCESS **Summary:** The HMC displays a banner, if set by the user on the GUI login page. The CLI displays a banner as an advisory note. For the GUI, the user will see a welcome message when they login to the GUI. Instructions to configure the banner is provided in guidance document.	
	AA	None
	Resp	N/A
FTP_ITC_EXT.1 Trusted Channel Communications	**Objective(s):** O.MANAGEMENT_ACCESS O.AUDIT **Summary:** The TOE provides trusted channels for various communications. The TOE provides a remote administrator access to the TOE via the HMC GUI over HTTPS or via the CLI over SSH. The TOE provides connection for the remote audit server over TLS and SFTP (over SSH).	
	AA	The evaluator will review the TSS to determine that it lists all trusted channels the TOE uses for remote communications, including both the external entities and remote users used for the channel as well as the protocol that is used for each.
	Resp	The HMC component uses remote communication through the trusted channel for the GUI using HTTPS and TLS 1.2 and the CLI using SSH. The TOE communicates to the external audit server via SFTP (over SSH).
FTP_UIF_EXT.1 User Interface: I/O Focus	**Objective(s):** O.DOMAIN_INTEGRITY **Summary:** The TOE provides access to the virtual machine's terminal window via the HMC GUI or HMC CLI. In order to access the terminal window, the user must be properly authenticated. Instructions on how to access the I/O interfaces are provided in the guidance document.	
	AA	The evaluator shall ensure that the TSS lists the supported user input devices.
	Resp	The TOE provides access to the virtual machine's terminal window via the HMC GUI (via HTTPS/TLS) or the HMC CLI (via SSH) after the user has identified and authenticated.
FTP_UIF_EXT.2 User Interface: Identification of VM	**Objective(s):** O.DOMAIN_INTEGRITY **Summary:** When the VM terminal is launched from the HMC, the HMC displays the name of the partition. Instructions for displaying the partitions is provided in the guidance document.	

TOE SFRs	TOE SFR compliance rationale	
	AA	The evaluator shall ensure that the TSS describes the mechanism for identifying VMs to the user, how identities are assigned to VMs, and how conflicts are prevented.
	Resp	The HMC displays the name of partition when its terminal window is launched. This can be done by the user opening the virtual terminal of VM using this menu option from HMC GUI. System resources -> VM -> Console -> Open Terminal Window. The name of VM along with System name, is clearly displayed at top of the pop-up terminal window which helps users to identify the VM terminal. When user creates a VM on HMC, default name is assigned consisting of timestamp and a random integer. Users can override this default value with their choice of name. Before VM is created, the TOE checks for the duplicate name. Once validation is successful, then only VM is created to prevent conflict in the name.
FCS_TLS_EXT.1 TLS Protocol	**Objective(s):** O.MANAGEMENT_ACCESS, O.AUDIT **Summary:** The TOE implements the TLS protocol as a server for protecting communication paths. The TLS protocol is implemented by the Java cryptographic module. The Java cryptographic module implements TLS services for the HMC. The Java cryptographic module implements HTTPS/TLS services for the connection between the remote administrator and the HMC through the HMC GUI.	
	AA	None
	Resp	None.
FCS_TLSS_EXT.1 TLS Server Protocol	**Objective(s):** O.AUDIT O.MANAGEMENT_ACCESS **Summary:** The Java cryptographic library implements TLS 1.2 server functionality to support for the connection between the remote administrator and the HMC through the HMC GUI.	
	AA	The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the cipher suites supported are specified. The evaluator shall check the TSS to ensure that the cipher suites specified include those listed for this component.
	Resp	The Java cryptographic module implements the TLS protocol version 1.2 compliant with RFC5246. The TOE enables only the following cipher suites in TLS when acting as a server: • TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288 and TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 and TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289.
	AA	The evaluator shall verify that the TSS contains a description of the denial of old SSL and TLS versions consistent relative to selections in FCS_TLSS_EXT.1.2.
	Resp	In the evaluated configuration, the TOE only supports version 1.2 of the TLS protocol. The TOE denies connections from clients requesting invalid or previous versions of the protocol including SSL v2.0, SSL v3.0, TLS v1.0 and TLS v1.1.
	AA	The evaluator shall verify that the TSS describes the key agreement parameters of the server's Key Exchange message.

TOE SFRs	TOE SFR compliance rationale	
	Resp	The Java cryptographic module uses the key exchange methods based on the cipher suites supported by the TOE in the evaluated configuration, which are RSA with key size 2048 bits and ECDHE with NIST curves P-256, P-384, and P-512.
FCS_HTTPS_EXT.1 HTTPS Protocol	**Objective(s):** O.MANAGEMENT_ACCESS O.AUDIT **Summary:** The TOE supports HTTPS (over TLS) for the connection between the remote administrator and the HMC through the HMC GUI.	
	AA	The evaluator shall check the TSS to ensure that it is clear on how HTTPS uses TLS to establish an administrative session, focusing on any client authentication required by the TLS protocol vs. security administrator authentication which may be done at a different level of the processing stack.
	Resp	The TOE implements the HTTPS protocol to connect the remote administrator to the HMC GUI and between the HMC component and Server component. The TOE uses HTTPS over the TLS protocol complaint with RFC2818.
FTP_TRP.1 Trusted Path	**Objective(s):** O.MANAGEMENT_ACCESS **Summary:** Remote administration is performed via the HMC GUI using HTTPS over TLS 1.2. In addition, remote administration can be performed via SSH.	
	AA	The evaluator shall examine the TSS to determine that the methods of remote TOE administration are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST.
	Resp	The remote administration is performed using the HMC GUI. The HMC GUI connection is protected by HTTPS/TLS 1.2. In addition, remote administration can be performed from the HMC CLI whose connection is protected by SSH. Both of these protocols are claimed in other SFRs of the ST.
FIA_X509_EXT.1 X.509 Certificate Validation	**Objective(s):** O.MANAGEMENT_ACCESS **Summary:** The TOE performs X.509 certification validation as described below.	
	AA	The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place. The evaluator ensures the TSS also provides a description of the certificate path validation algorithm. The evaluator shall examine the TSS to confirm that it describes the behavior of the TOE when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel. If the requirement that the administrator is able to specify the default action, then the evaluator shall ensure that the operational guidance contains instructions on how this configuration action is performed.
	Resp	The TOE performs X.509 certificate validation and certificate path validation in accordance to RFC 5280 via OCSP when a new certificate is imported. By design, certificates cannot be deleted and must be replaced by another certificate. Certificate validation, which is provided by the OpenSSL and Java cryptographic libraries, is performed as follows:

TOE SFRs	TOE SFR compliance rationale	
		• Certificate validation and certificate path validation conforms to RFC5280. • 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 TOE will validate revocation status of the certificate using an OCSP TLS Status Request Extension (OCSP stapling) as specified in RFC 6066 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. Also, the TOE will only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE. The certificate path must terminate with a trusted certificate. The TOE will reject a certificate if it is found to be invalid. Also, the TOE will reject a certificate with unknown revocation status if the validation through OCSP stapling is not successful. X.509 certificate begin and end dates will be validated while importing the certificate into the HMC. If invalid, the certificate will not be imported. Certificate validation via OCSP must be performed manually by an authorized administrators by issuing the HMC CLI commands provided in the guidance document. TOE can generate a Certificate Signing Request (CSR) and receive the corresponding CA certificate response file. Via the HMC, the administrator can validate the certificate attributes. The TOE supports importing the certificate stored in a USB or stored in a local path on the HMC.
FIA_PMG_EXT.1 Password Management	**Objective(s):** O.MANAGEMENT_ACCESS **Summary:** The TOE provides the password management capabilities for administrative passwords as specified in FIA_PMG_EXT.1.	
	AA	None
	Resp	N/A.
FCS_SSH_EXT.1 SSH Protocol	**Objective(s):** O.MANAGEMENT_ACCESS	

TOE SFRs	TOE SFR compliance rationale	
	Summary: Access to the CLI of the HMC console is provided by SSH where username and password and public key login is supported. Audit logs are transferred from the HMC to the external audit storage using SSH.	
	AA	The evaluator shall ensure that the selections indicated in the ST are consistent with selections in this and subsequent components. Otherwise, this SFR is evaluated by activities for other SFRs. The evaluator shall check to ensure that the authentication methods listed in the TSS are identical to those listed in this SFR component; and, ensure if password-based authentication methods have been selected in the ST then these are also described; and, ensure that if keyboard-interactive is selected, it describes the multifactor authentication mechanisms provided by the TOE The evaluator shall check that the TSS describes how “large packets” are detected and handled. The evaluator will check the description of the implementation of SSH in the TSS to ensure the encryption algorithms supported are specified. The evaluator will check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component. The evaluator will check the description of the implementation of SSH in the TSS to ensure the hashing algorithms supported are specified. The evaluator will check the TSS to ensure that the hashing algorithms specified are identical to those listed for this component. The evaluator will check the description of the implementation of SSH in the TSS to ensure the shared secret establishment algorithms supported are specified. The evaluator will check the TSS to ensure that the shared secret establishment algorithms specified are identical to those listed for this component. The evaluator will check the description of the implementation of SSH in the TSS to ensure the KDFs supported are specified. The evaluator will check the TSS to ensure that the KDFs specified are identical to those listed for this component. The evaluator shall check the TSS to ensure that if the TOE enforces connection rekey or termination limits lower than the maximum values that these lower limits are identified. In cases where hardware limitation will prevent reaching data transfer threshold in less than one hour, the evaluator shall check the TSS to ensure it contains: a. An argument describing this hardware-based limitation and b. Identification of the hardware components that form the basis of such argument. For example, if specific Ethernet Controller or Wi-Fi radio chip is the root cause of such limitation, these subsystems shall be identified.
	Resp	The TOE implements the SSHv2 client and SSHv2 server protocol. The SSH client is used to protect the transfer of audit logs to external audit log storage. SSH is compliant with RFC 4344. The TOE supports a maximum packet length of 35K bytes and if the packet is greater than 35K bytes, the packet is dropped/discarded. It is also compliant with the following.

TOE SFRs	TOE SFR compliance rationale	
		• RFC 4251 supporting SSH architecture • RFC 4252 supporting password-based authentication • RFC 4253 supporting transport layer protocol • RFC 6668 supporting HMACs with SHA-2 • RFC 5656 supporting ECDH with NIST curve nistp256 and nistp384 and SSH KDF The TOE's SSH implementation also supports key-based and password-based authentication methods as per RFC 4252. The TOE supports the PP-specified transport public key authentication algorithm ssh-rsa (i.e., RSA signatures with PKCS1 1.5 and SHA-1) and rejects all others. The TOE supports the PP-specified encryption algorithm aes128-ctr and rejects all others. The TOE supports the PP-specified data integrity algorithm hmac-sha2-256 and rejects all others. The TOE supports the key exchange/establishment methods ecdh-sha2-nistp256 ecdh-sha2-nistp384, and rejects all others. The TOE supports SSH KDF in accordance with RFC 5656 (section 4). The TOE will ensure a connection termination occurs when one hour connection time, no more than one gigabyte of transmitted data, or no more than one gigabyte of received data.
FCS_SSHC_EXT.1 SSH Client Protocol	**Objective(s):** O.MANAGEMENT_ACCESS **Summary:** See description of the SSH protocol above. The HMC contains a local database storing SSH public keys for the HMC users.	
	AA	None
	Resp	N/A
FCS_SSHS_EXT.1 SSH Server Protocol	**Objective(s):** O.MANAGEMENT_ACCESS **Summary:** See description of the SSH protocol above.	
	AA	None
	Resp	N/A

7.1.2 TOE Security Assurance Requirement

7.1.2.1 Timely Security Updates (ALC_TSU_EXT.1)

IBM products including the TOE follows Product Security Incident Response Team (PSIRT) process for timeline to analyze, fix and publish the fixes as per https://www.ibm.com/trust/security-vulnerability-management.

Fixes are made available to users through Fix Central at https://www.ibm.com/support/fixcentral/.

The process of creating and deploying security updates is same as regular product update, where user updates HMC with images available at Fix Central.

IBM's CVE fix time window for on-prem products is not fixed per medium or high severity, but rather based on the specific product and its affected versions. IBM generally addresses vulnerabilities based on their criticality, with more critical issues receiving priority. The specific remediation steps, including fixes and workarounds, are detailed in IBM security bulletins.

Reporting security issues: All security issues are reported using Notification on ibm.com. User can subscribe to product they would like to receive notification here https://www.ibm.com/systems/support/myview/subscription/css.wss

8 Abbreviations, Terminology, and References

8.1 Abbreviations

AES
Advanced Encryption Standard

AIX
IBM's Advanced Interactive eXecutive operating system

CA
Certificate Authority

CBC
Cipher Block Chaining

CLI
Command Line Interface

CRL
Certificate Revocation List

DNS
Domain Name System

ECD
Extended Components Definition

FSP
Flexible Service Processor

GUI
Graphical User Interface

HMC
Hardware Management Console

HTTPS
Hypertext Transfer Protocol Secure

IBM i
IBM i operating system

Linux
IBM Linux operating system

LPAR
Logical Partition

MC
Management Console

OCSP
Online Certficate Status Protocol

OF/RTA
Open Firmware/Run-Time Abstraction

OpenSSH
Open Secure Socket

OpenSSL
Open Secure Socket Layer

Operator Panel
Front panel controls on the physical server

PHYP
PowerVM Hypervisor

PowerVC
POWER Virtual Control

PowerVM
POWER Virtual Machine

RFC
Request for Comments

RSA
Rivest-Shamir-Adleman

SCSI
Small Computer System Interface

SFTP
Secure File Transfer Protocol

SLIC
System Licensed Internal Code

SSH
Secure Shell

OpenSSL
Open Secure Socket Layer

TLS
Transport Layer Security

TOE
Target of Evaluation

TSF
TOE Security Function

vENT
Virtual Ethernet

vHMC
Virtual HMC

VIOS
Virtual Input/Output System

vSCSI
Virtual SCSI

8.2 References

CC **Common Criteria for Information Technology Security Evaluation**
Version 3.1R5
Date April 2017
Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R5.pdf

Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R5.pdf
Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R5.pdf

CFG_Virtualization-SV_V1.0 **PP-Configuration for Virtualization and Server Virtualization Systems**
Version 1.0
Date 2021-06-04
Location https://www.niap-ccevs.org/protectionprofiles/458

MOD_SV_V1.1 **PP-Module for Server Virtualization Version 1.1**
Version 1.1
Date 2021-06-14
Location https://www.niap-ccevs.org/protectionprofiles/458

PAPR **Power Architecture® Platform Requirements (PAPR)**
Version 10.60
Date 2024
Location https://files.openpower.foundation/s/XFgfMaqLMD5Bcm8

PFW_PHYP_HIDDEN_HCALLS_LIST **PFW PowerVM Hidden Hypervisor Calls List**
Author(s) Scott Mayes, IBM
Version 1
Date 2024-02-13

PKG_SSH_V1.0 **Functional Package for SSH Version 1.0**
Version 1.0
Date 2021-05-13
Location https://www.niap-ccevs.org/protectionprofiles/459

PKG_TLS_V1.1 **Functional Package for SSH Version 1.0**
Version 1.1
Date 2019-03-01
Location https://www.niap-ccevs.org/protectionprofiles/439

PLATFORM_HCALLS_LIST **Platform Hypervisor Calls List**
Author(s) Scott Mayes, IBM
Version 1
Date 2024-02-13

PP_BASE_VIRTUALIZATION_V1.1 **Protection Profile for Virtualization Version 1.1**
Version 1.1
Date 2021-06-14
Location https://www.niap-ccevs.org/protectionprofiles/458

RPA_HCALLS_LIST **RPA Hypervisor Calls List**
Author(s) IBM
Version 1
Date 2024-10-21

RPA_HIDDEN_HCALLS_LIST **RPA Hidden Hypervisor Calls List**
Author(s) IBM
Version 1

	Date	2024-10-21
SLIC_HCALLS_LIST	**SLIC Hypervisor Calls List**	
	Author(s)	Pete Heyrman, IBM
	Version	1
	Date	2024-10-21