SUSE Linux Enterprise Server 15 SP4 Security Target for VPP Compliance

Version:	**1.4**
Status:	**Released**
Last Update:	**2026-01-14**
Classification:	**Released**

Trademarks

SUSE and the SUSE logo are trademarks or registered trademarks of SUSE Linux Products GmbH in Germany, other countries, or both.

Linux is a registered trademark of Linus Torvalds.

UNIX is a registered trademark of The Open Group in the United States and other countries.

IBM, IBM logo, bladecenter, eServer, iSeries, OS/400, POWER3, POWER4, POWER4+, pSeries, System p, POWER5, POWER5+, POWER6, POWER6+, POWER7, POWER7+, System x, System z, S390, xSeries, zSeries, zArchitecture, and z/VM are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Intel, Xeon, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Legal Notice

This document is provided AS IS with no express or implied warranties. Use the information in this document at your own risk.

This document may be reproduced and distributed only in its original entirety without revision.

Copyright ©2026 SUSE LLC

Classification Note

atsec public

Revision History

Revision	Date	Author(s)	Changes to Previous Revision
1.4	2026-01-14	SUSE supported by atsec consultants	Address BSI comments

Copyright ©2026 SUSE LLC

Table of Contents

Copyright ©2026 SUSE LLC

Copyright ©2026 SUSE LLC

Copyright ©2026 SUSE LLC

Copyright ©2026 SUSE LLC

List of Tables

Copyright ©2026 SUSE LLC

1 Introduction

1.1 Security Target Identification

Title:	SUSE Linux Enterprise Server 15 SP4 Security Target for VPP Compliance
Version:	1.4
Status:	Released
Date:	2026-01-14
Sponsor:	SUSE LLC
Developer:	SUSE LLC
Certification Body:	BSI
Certification ID:	BSI-DSZ-CC-1231
Keywords:	SLES, operating system, virtual machine monitor

1.2 TOE Identification

The TOE is SUSE Linux Enterprise Server Version 15 SP4.

1.3 TOE Type

The TOE type is a server virtualization system.

1.4 TOE Overview

The Security Target (ST) serves as the basis for the Common Criteria (CC) evaluation and identifies the Target of Evaluation (TOE), the scope of the evaluation, and the assumptions made throughout. This document will also describe the intended operational environment of the TOE, and the functional and assurance requirements that the TOE meets.

The TOE is the SUSE Linux Enterprise Server 15 SP4, a highly-configurable Linux-based operating system offering virtual machine management support which has been developed to provide a good level of security as required in commercial environments. Details about the supported security functions are outlined in section 1.5. The virtual machine monitor support is integrated into the operating system. The evaluation covers the KVM virtualization technology.

The tested version of the TOE is:

- SLES 15 SP4

1.5 TOE Description

This section provides a general description of the TOE, including physical boundaries, security functions, and relevant TOE documentation and references.

1.5.1 Physical Boundary

The Target of Evaluation is:

- SUSE Linux Enterprise Server version 15 SP4

The TOE is a subset of the distribution provided with SUSE Linux Enterprise Server version 15 SP4. The TOE and its documentation are supplied on ISO images distributed via the SUSE Portal. The TOE includes a package holding the additional user and administrator documentation.

In addition to the installation media, the following documentation is provided:

Copyright ©2026 SUSE LLC

- SUSE Linux Enterprise Server 15 SP4 Evaluated Configuration Guide, Version 2.4
- Manual pages for all applications, configuration files and system calls

Processor	microArch
Intel x86_64	Cascade Lake
AMD x86_64	AMD EPYC 3rd Generation
IBM System Z LPAR	z15
ARM 64 Bit	ARMv8.2-A

Table 1: Hardware platforms

1.5.2 TOE Security Functionality

The TOE provides the security functions conforms to the requirements defined in section 2.

1.5.2.1 Security audit

The TOE generates audit events for all start-up and shut-down functions, and all auditable events as specified by the requirements defined in section 2. Audit events are generated for the following audit functions:

- Start-up and shut-down of the audit functions
- All administrative actions
- Additional events including virtualization-related audit events

Each audit record contains the date and time of the event, type of event, subject and object identity (if applicable), outcome (success or failure) of the event, and auxiliary information.

1.5.2.2 Cryptographic support

The TOE includes the OpenSSL version 1.1.1 cryptographic libraries for performing userspace cryptographic operations. In addition, the Linux kernel crypto API performs the cryptographic operations performed by the kernel. In addition, the TOE uses software noise sources for entropy generation.

The TOE implements SSHv2 for allowing secure remote administration.

1.5.2.3 User data protection

The TOE implements resource assignment to different virtual machines based on the virtual machine configurations. Such resources include physical resources (e.g. devices) as well as virtual resources (e.g. networking support).

1.5.2.4 Identification and authentication

All administrators must be authenticated to the TOE prior to carrying out any actions, including management operations. The TOE supports password-based authentication, authentication based on SSH-keys. The TOE will lock out user accounts after a defined number of unsuccessful password-based authentication attempts to that user account has been met.

Copyright ©2026 SUSE LLC

1.5.2.5 Security Management

The TOE can perform management functions. The administrator has full access to carry-out all management functions offered by the TOE.

1.5.2.6 Protection of the TSF

The TOE implements the following protection of TSF data functions.

- Virtualization support utilizing CPU mechanisms
- Trusted software updates using digital signatures

1.5.2.7 TOE Access

The TOE displays an advisory warning message regarding unauthorized use of the TSF prior to establishment of an administrative user session.

1.5.2.8 Trusted Path/Channels

The TOE offers an SSH server as well as client which uses the SSHv2 protocol allowing remote administration.

1.5.3 TOE Operational Environment

The following environmental components interoperate with the TOE in the evaluated configuration.

Component	Description
Hardware platform	See Table 1
SUSE Customer Center	Server that allows the TOE to download updates

Table 2: TOE operational environment

1.5.4 Product Functionality Excluded from the Scope of the Evaluation

Additional mechanisms and functions that would interfere with the operation of the security functions are disallowed in the evaluated configuration and the Evaluation Configuration Guide provides instructions to the administrator on how to disable them. Note: TOE mechanism which provide additional restrictions to the above claimed security functions are allowed in the evaluated configuration. The following table enumerates mechanisms that are provided with the TOE but which are excluded from the evaluation:

Functions	Exclusion discussion
eCryptFS	eCryptFS is not allowed to be used in the evaluated configuration. The encryption capability provided with this file system is therefore unavailable to any user.
Ext4 file-based encryption	Ext4 file-based encryption is not allowed to be used in the evaluated configuration. The encryption capability provided with this file system is therefore unavailable to any user.
SMACK	The mandatory access control functionality offered by the SMACK LSM is not assessed by the evaluation and disabled in the evaluated configuration.

Copyright ©2026 SUSE LLC

Functions	Exclusion discussion
SELinux	The mandatory access control functionality offered by the SELinux LSM is not assessed by the evaluation and disabled in the evaluated configuration.
Xen	The Xen hypervisor along with all support mechanisms is not subject to evaluation.

Table 3: Non-evaluated functionalities

Note: Packages and mechanisms not covered with security claims and subsequent assessments during the evaluation or disabling the respective functionality in the evaluated configuration result from resource constraints during the evaluation but does not imply that the respective package or functionality is implemented insecurely.

Copyright ©2026 SUSE LLC

2 CC Conformance Claim

This Security Target is CC Part 2 extended and CC Part 3 extended.

This Security Target claims conformance to the following Protection Profiles and PP packages:

- [VPP]: Protection Profile for Virtualization. Version 1.1 as of 2021-06-14; exact conformance.
- [SV]: PP-Module for Server Virtualization Systems. Version 1.1 as of 2021-06-14; exact conformance.
- [SVCONFIG]: PP-Configuration for Virtualization and Server Virtualization Systems. Version 1.0 as of 2021-06-04; exact conformance.
- [SSH]: Functional Package for Secure Shell (SSH). Version 1.0 as of 2021-05-13; exact conformance.

Common Criteria [CC] version 3.1 revision 5 is the basis for this conformance claim.

2.1 Protection Profile Tailoring and Additions

2.1.1 Protection Profile for Virtualization ([VPP])

Table 4 contains the NIAP Technical Decisions (TDs) for this protection profile at the time of the evaluation and a statement of applicability to the evaluation.

NIAP TD	TD description	Applicable?
TD0961	ALC_TSU_EXT.1 in PP_BASE_VIRTUALIZATION_V1.1 has no Evaluation Activities.	Yes
TD0953	Updating FIPS 186-4 to 186-5 in PP_BASE_VIRTUALIZATION_V1.1	Yes
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
TD0844	Addition of Assurance Package for Flaw Remediation V1.0 Conformance Claim	Yes
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

Table 4: NIAP TDs for VPP

2.1.2 PP-Module for Server Virtualization Systems ([SV])

The PP-Module is used in compliance to the PP-Configuration for Virtualization and Server Virtualization Systems Version 1.0 04 June 2021, report number CCEVS-VR-PP-0084 from February 03 2023, version 1.1.

Copyright ©2026 SUSE LLC

2.1.3 PP-Configuration for Virtualization and Server Virtualization Systems ([SVCONFIG])

The PP Protection Profile for Virtualization, Version 1.1, 14 June 2021 (Virtualization PP) is used with the PP-Module for Server Virtualization Systems according to the configuration.

2.1.4 Functional Package for Secure Shell (SSH) ([SSH])

Table 5 contains the NIAP Technical Decisions (TDs) for this PP-Module at the time of the evaluation and a statement of applicability to the evaluation.

NIAP TD	TD description	Applicable?
TD0967	Allowance of Kex-strict in PKG_SSH_V1.0	Yes
TD0909	Updates to FCS_SSH_EXT.1.1 App Note in SSH FP 1.0	Yes
TD0777	TD0777: Clarification to Selections for Auditable Events for FCS_SSH_EXT.1	Yes
TD0732	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

Table 5: NIAP TDs for SSH

Copyright ©2026 SUSE LLC

3 Security Problem Definition

3.1 Threat Environment

3.1.1 Threats countered by the TOE

T.DATA_LEAKAGE

PP Origin: *VPP*

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

PP Origin: *VPP*

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

PP Origin: *VPP*

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

PP Origin: *VPP*

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

PP Origin: *VPP*

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

Copyright ©2026 SUSE LLC

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

PP Origin: *VPP*

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

PP Origin: *VPP*

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

PP Origin: *VPP*

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

PP Origin: *VPP*

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

PP Origin: *VPP*

Copyright ©2026 SUSE LLC

Vulnerabilities in outdated or unpatched software can be exploited by adversaries to compromise the VS or platform.

T.MISCONFIGURATION

PP Origin: *VPP*

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

PP Origin: *VPP*

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

PP Origin: *VPP*

The platform has not been compromised prior to installation of the VS.

A.PHYSICAL

PP Origin: *VPP*

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

PP Origin: *VPP*

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

A.NON_MALICIOUS_USER

PP Origin: *VPP*

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.

Copyright ©2026 SUSE LLC

4 Security Objectives

4.1 Objectives for the TOE

O.VM_ISOLATION

PP Origin: *VPP*

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

PP Origin: *VPP*

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

PP Origin: *VPP*

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.

Copyright ©2026 SUSE LLC

O.DOMAIN_INTEGRITY

PP Origin: *VPP*

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

PP Origin: *VPP*

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

PP Origin: *VPP*

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

Copyright ©2026 SUSE LLC

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

PP Origin: *VPP*

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

PP Origin: *VPP*

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

PP Origin: *VPP*

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

PP Origin: *VPP*

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

PP Origin: *VPP*

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

OE.PHYSICAL

PP Origin: *VPP*

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

OE.TRUSTED_ADMIN

PP Origin: *VPP*

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

OE.NON_MALICIOUS_USER

PP Origin: *VPP*

Copyright ©2026 SUSE LLC

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.

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

Table 6: Mapping of security objectives to threats and policies

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.

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

Table 7: Mapping of security objectives for the Operational Environment to assumptions, threats and policies

Copyright ©2026 SUSE LLC

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.

Threat	Rationale for security objectives
T.DATA_LEAKAGE	The threat T.DATA_LEAKAGE is countered by O.VM_ISOLATION as logical separation of VMs and enforcement of domain integrity prevent unauthorized transmission of data from one VM to another. The threat T.DATA_LEAKAGE is countered by O.DOMAIN_INTEGRITY as logical separation of VMs and enforcement of domain integrity prevent unauthorized transmission of data from one VM to another.
T.UNAUTHORIZED_UPDATE	The threat T.UNAUTHORIZED_UPDATE is countered by O.VMM_INTEGRITY as system integrity prevents the TOE from installing a software patch containing unknown and potentially malicious code. Integrity of a Virtualization System can be maintained by ensuring that the only way to modify the VS is through a trusted update process initiated by an authorized Administrator as required by FMT_MOF_EXT.
T.UNAUTHORIZED_MODIFICATION	The threat T.UNAUTHORIZED_MODIFICATION is countered by O.VMM_INTEGRITY as enforcement of VMM integrity prevents the bypass of enforcement mechanisms and auditing ensures that abuse of legitimate authority can be detected. The threat T.UNAUTHORIZED_MODIFICATION is countered by O.AUDIT as enforcement of VMM integrity prevents the bypass of enforcement mechanisms and auditing ensures that abuse of legitimate authority can be detected.
T.USER_ERROR	The threat T.USER_ERROR is countered by O.VM_ISOLATION as 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 threat T.3P_SOFTWARE is countered by O.VMM_INTEGRITY as 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	The threat T.VMM_COMPROMISE is countered by O.VMM_INTEGRITY as 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. The threat T.VMM_COMPROMISE is countered by O.VM_ISOLATION as 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	The threat T.PLATFORM_COMPROMISE is countered by O.PLATFORM_INTEGRITY as 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.

Copyright ©2026 SUSE LLC

Threat	Rationale for security objectives
T.UNAUTHORIZED_ACCESS	The threat T.UNAUTHORIZED_ACCESS is countered by O.MANAGEMENT_ACCESS as ensuring that TSF management functions cannot be executed without authorization prevents untrusted subjects from modifying the behavior of the TOE in an unanticipated manner. Access to management functions must be limited to authorized Administrators as managed through controls required by FMT_MOF_EXT.1.
T.WEAK_CRYPTO	The threat T.WEAK_CRYPTO is countered by O.VM_ENTROPY as acquisition of good entropy is necessary to support the TOE's security-related cryptographic algorithms.
T.UNPATCHED_SOFTWARE	The threat T.UNPATCHED_SOFTWARE is countered by O.PATCHED_SOFTWARE as the ability to patch the TOE software ensures that protections against vulnerabilities can be applied as they become available.
T.MISCONFIGURATION	The threat T.MISCONFIGURATION is countered by O.CORRECTLY_APPLIED_CONFIGURATION as mechanisms to prevent the application of configurations that violate the current security policy help prevent misconfigurations.
T.DENIAL_OF_SERVICE	The threat T.DENIAL_OF_SERVICE is countered by O.RESOURCE_ALLOCATION as the ability of the TSF to ensure the proper allocation of resources makes denial of service attacks more difficult.

Table 8: Sufficiency of objectives countering threats

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.

Assumption	Rationale for security objectives
A.PLATFORM_INTEGRITY	Covering OE.PHYSICAL: If the underlying platform has not been compromised prior to installation of the TOE, its integrity can be assumed to be intact.
A.PHYSICAL	Covering OE.PHYSICAL: If the TOE is deployed in a location that has appropriate physical safeguards, it can be assumed to be physically secure.
A.TRUSTED_ADMIN	Covering OE.TRUSTED_ADMIN: Providing guidance to administrators and ensuring that individuals are properly trained and vetted before being given administrative responsibilities will ensure that they are trusted.
A.NON_MALICIOUS_USER	Covering OE.NON_MALICIOUS_USER: If the organization properly vets and trains users, it is expected that they will be non-malicious.

Copyright ©2026 SUSE LLC

Assumption	Rationale for security objectives
	Covering OE.CONFIG: If the TOE is administered by a non-malicious and non-negligent user, the expected result is that the TOE will be configured in a correct and secure manner.

Table 9: Sufficiency of objectives holding assumptions

Copyright ©2026 SUSE LLC

5 Extended Components Definition

The extended components definitions are defined in the documents specified in Section 2 "CC Conformance Claim".

Copyright ©2026 SUSE LLC

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**.

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 (Refined)		VPP	No	Yes	No	Yes
	FAU_SAR.1 Audit Review		VPP	No	No	No	No
	FAU_STG.1 Protected Audit Trail Storage		VPP	No	No	No	No
	FAU_STG_EXT.1 Off-Loading of Audit Data		VPP	No	No	Yes	Yes
FCS - Cryptographic support	FCS_CKM.1 Cryptographic Key Generation		VPP	No	Yes	No	Yes
	FCS_CKM.2 Cryptographic Key Establishment		VPP	No	No	No	Yes
	FCS_CKM_EXT.4 Cryptographic Key Destruction		VPP	No	No	No	No
	FCS_COP.1/Hash Cryptographic Operation (Hashing)	FCS_COP.1	VPP	No	No	No	Yes
	FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithms)	FCS_COP.1	VPP	No	No	Yes	Yes
	FCS_COP.1/Sig Cryptographic Operation (Signature Algorithms)	FCS_COP.1	VPP	No	Yes	No	Yes
	FCS_COP.1/UDE Cryptographic Operation (AES Data Encryption / Decryption)	FCS_COP.1	VPP	No	No	No	Yes
	FCS_ENT_EXT.1 Entropy for Virtual Machines		VPP	No	No	No	Yes
	FCS_RBG_EXT.1 Random Bit Generation		VPP	No	No	No	Yes
	FCS_SSH_EXT.1 SSH Protocol		SSH	No	No	Yes	Yes

Copyright ©2026 SUSE LLC

Security functional class	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FCS_SSHC_EXT.1 SSH Protocol - Client		SSH	No	No	No	Yes
	FCS_SSHS_EXT.1 SSH Protocol - Server		SSH	No	No	No	Yes
FDP - User data protection	FDP_HBI_EXT.1 Hardware-Based Isolation Mechanisms		VPP	No	No	Yes	Yes
	FDP_PPR_EXT.1 Physical Platform Resource Controls		VPP	No	No	Yes	Yes
	FDP_RIP_EXT.1 Residual Information in Memory		VPP	No	No	No	No
	FDP_RIP_EXT.2 Residual Information on Disk		VPP	No	No	No	No
	FDP_VMS_EXT.1 VM Separation		VPP	No	No	No	Yes
	FDP_VNC_EXT.1 Virtual Networking Components		VPP	No	No	No	No
FIA - Identification and authentication	FIA_AFL_EXT.1 Authentication Failure Handling		VPP	No	No	Yes	Yes
	FIA_UAU.5 Multiple Authentication Mechanisms		VPP	No	No	Yes	Yes
	FIA_UIA_EXT.1 Administrator Identification and Authentication		VPP	No	No	No	No
	FIA_PMG_EXT.1 Password Management		VPP	No	No	No	Yes
FMT - Security management	FMT_MOF_EXT.1 Management of Security Functions Behavior		SV	No	Yes	No	Yes
	FMT_SMO_EXT.1 Separation of Management and Operational Networks		VPP	No	No	No	Yes
FPT - Protection of the TSF	FPT_DVD_EXT.1 Non-Existence of Disconnected Virtual Devices		VPP	No	No	No	No
	FPT_EEM_EXT.1 Execution Environment Mitigations		VPP	No	No	No	Yes
	FPT_HAS_EXT.1 Hardware Assists		VPP	No	No	Yes	No
	FPT_HCL_EXT.1 Hypercall Controls		VPP	No	No	No	No
	FPT_RDM_EXT.1 Removable Devices and Media		VPP	No	No	Yes	Yes
	FPT_TUD_EXT.1 Trusted Update to the Virtualization System		VPP	No	No	No	Yes

Copyright ©2026 SUSE LLC

Security functional class	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FPT_VDP_EXT.1 Virtual Device Parameters		VPP	No	No	No	No
	FPT_VIV_EXT.1 VMM Isolation from VMs		VPP	No	No	No	No
FTA - TOE access	FTA_TAB.1 TOE access banners		VPP	No	No	No	No
FTP - Trusted path/channels	FTP_ITC_EXT.1 Trusted Channel Communications		VPP	No	No	Yes	Yes
	FTP_TRP.1 Trusted Path		VPP	No	No	No	No
	FTP_UIF_EXT.1 User Interface: I/O Focus		VPP	No	No	No	No
	FTP_UIF_EXT.2 User Interface: Identification of VM		VPP	No	No	No	No

Table 10: SFRs for the TOE

6.1.1 Security audit (FAU)

6.1.1.1 FAU_GEN.1 Audit Data Generation (Refined)

PP Origin: *VPP*

Applied TDs: *TD0615*

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

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

b. All administrative actions relevant to claimed SFRs as defined in the Auditable Events Table from the Client and Server PP-Modules

c. Auditable events defined in ~~Table 2~~ *Table 11*

d.
- **no other auditable events**

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~~ *Table 11*

f.
- **no other information**

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	No events specified	
FAU_SAR.1	No events specified	

Copyright ©2026 SUSE LLC

Requirement	Auditable Events	Additional Audit Record Contents
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/Hash	No events specified	
FCS_COP.1/KeyedHash	No events specified	
FCS_COP.1/Sig	No events specified	
FCS_COP.1/UDE	No events specified	
FCS_ENT_EXT.1	No events specified	
FCS_RBG_EXT.1	Failure of the randomization process.	
FCS_SSH_EXT.1	**None**	
	None	
	None	
	None	
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.
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 physical networking components.	VM and virtual or physical networking component identifiers.

Copyright ©2026 SUSE LLC

Requirement	Auditable Events	Additional Audit Record Contents
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_MOF_EXT.1	Attempts to invoke any of the management functions listed in ~~Table 3~~, *Table 12*	Success or failure of attempt Identity or actor
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	**None**	Hypercall interface for which access was attempted.
FPT_HCL_EXT.1	**None**	
FPT_RDM_EXT.1	Connection/disconnection of removable media or device to/from a VM.	VM Identifier, Removable media/device identifier, event description or identifier (connect/disconnect, ejection/insertion, etc.).
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.	
FPT_VDP_EXT.1	No events specified	

Copyright ©2026 SUSE LLC

Requirement	Auditable Events	Additional Audit Record Contents
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	
FIA_PMG_EXT.1	No events specified	
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 11: Auditable Events

TSS Link: *TSS for FAU_GEN.1*

6.1.1.2 FAU_SAR.1 Audit Review

PP Origin: *VPP*

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.

TSS Link: *TSS for FAU_SAR.1*

6.1.1.3 FAU_STG.1 Protected Audit Trail Storage

PP Origin: *VPP*

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.

TSS Link: *TSS for FAU_STG.1*

6.1.1.4 FAU_STG_EXT.1 Off-Loading of Audit Data

PP Origin: *VPP*

Copyright ©2026 SUSE LLC

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: overwrite the oldest audit record**

when the local storage space for audit data is full.

TSS Link: *TSS for FAU_STG_EXT.1*

6.1.2 Cryptographic support (FCS)

6.1.2.1 FCS_CKM.1 Cryptographic Key Generation

PP Origin: *VPP*

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

- **RSA schemes using cryptographic key sizes of ~~2048-bit~~ *3072-bit* or greater that meet the following: FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.1**
- **ECC schemes using "NIST curves" P-256, P-384 and P-521 that meet the following: FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.2**
- **FFC schemes using safe primes that meet the following: [NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes"]**

TSS Link: *TSS for FCS_CKM.1*

6.1.2.2 FCS_CKM.2 Cryptographic Key Establishment

PP Origin: *VPP*

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

- **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"**
- **Finite field-based key establishment schemes that meets NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"**

TSS Link: *TSS for FCS_CKM.2*

6.1.2.3 FCS_CKM_EXT.4 Cryptographic Key Destruction

PP Origin: *VPP*

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.

TSS Link: *TSS for FCS_CKM_EXT.4*

Copyright ©2026 SUSE LLC

6.1.2.4 FCS_COP.1/Hash Cryptographic Operation (Hashing)

PP Origin: *VPP*

FCS_COP.1.1/Hash The TSF shall perform [cryptographic hashing] in accordance with a specified cryptographic algorithm

- **SHA-256**
- **SHA-384**
- **SHA-512**

and message digest sizes

- **256 bits**
- **384 bits**
- **512 bits**

that meet the following: FIPS Pub 180-4.

TSS Link: *TSS for FCS_COP.1/Hash*

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

PP Origin: *VPP*

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

TSS Link: *TSS for FCS_COP.1/KeyedHash*

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

PP Origin: *VPP*

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 of ~~2048-bit~~ *3072-bit* or greater that meet the following: FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5**
- **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 6**

TSS Link: *TSS for FCS_COP.1/Sig*

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

PP Origin: *VPP*

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

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

and cryptographic key sizes

- **128-bit key sizes**

Copyright ©2026 SUSE LLC

- **256-bit key sizes.**

TSS Link: *TSS for FCS_COP.1/UDE*

6.1.2.8 FCS_ENT_EXT.1 Entropy for Virtual Machines

PP Origin: *VPP*

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

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

TSS Link: *TSS for FCS_ENT_EXT.1*

6.1.2.9 FCS_RBG_EXT.1 Random Bit Generation

PP Origin: *VPP*

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

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

TSS Link: *TSS for FCS_RBG_EXT.1*

Type of CTR DRBG:

Considering TD0936 the seed data size is appropriate because the used DRBG is a CTR DRBG with DF.

Application Note CTR DRBG:

For the German Schema, the SFR is "translated" into an AIS 20/31 compliant SFR following the FCS_RNG.1 definition. Therefore, this application note states the SFR as part of this application note:

FCS_RNG.1.1: The TSF shall provide a deterministic random number generator conforming to SP800-90A CTR_DRBG with AES-256 core using a derivation function without prediction resistance that implements:

a) DRG2.1: If initialized with a random seed using high-resolution time stamps of the execution time of a fixed set of CPU instructions along with associated memory accesses as seed source, the internal state of the RNG shall have a minentropy of 256 bits.

b) DRG2.2: The DRNG provides forward secrecy.

c) DRG2.3: The DRNG provides backward secrecy.

The TSF shall provide random numbers that meet:

*a) DRG.2.4: The RNG is initialized with a random seed of 384 bits, is reseeded after at most 2**48 generate requests with 256 bits, and has the output property such that 2**19 strings of bit length 128 are mutually different with probability of greater than 1-2**-10.*

b) DRG.2.5: Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A.

6.1.2.10 FCS_SSH_EXT.1 SSH Protocol

PP Origin: *SSH*

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, 5647, 5656, 6668, 8268, 8332** and no other standard.

Copyright ©2026 SUSE LLC

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):**
  - **rsa-sha2-256 (RFC 8332)**
  - **rsa-sha2-512 (RFC 8332)**
  - **ecdsa-sha2-nistp384 (RFC 5656)**
  - **ecdsa-sha2-nistp521 (RFC 5656)**

and no other methods.

FCS_SSH_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than **2^18 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)**
- **aes256-ctr (RFC 4344)**
- **aes128-gcm@openssh.com (RFC 5647)**
- **aes256-gcm@openssh.com (RFC 5647)**

and no other mechanisms.

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

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

and no other mechanisms.

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

- **diffie-hellman-group16-sha512 (RFC 8268)**
- **diffie-hellman-group18-sha512 (RFC 8268)**
- **ecdh-sha2-nistp256 (RFC 5656)**
- **ecdh-sha2-nistp384 (RFC 5656)**
- **ecdh-sha2-nistp521 (RFC 5656)**

and no other mechanisms.

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

- **RFC 4253 (Section 7.2)**
- **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.

Application Note: *If the TOE-attempted rekey is not accepted or completed by the remote peer, the connection is terminated.*

TSS Link: *TSS for FCS_SSH_EXT.1*

Copyright ©2026 SUSE LLC

6.1.2.11 FCS_SSHC_EXT.1 SSH Protocol - Client

PP Origin: *SSH*

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:**
  - **rsa-sha2-256 (RFC 8332)**
  - **rsa-sha2-512 (RFC 8332)**
  - **ecdsa-sha2-nistp384 (RFC 5656)**
  - **ecdsa-sha2-nistp521 (RFC 5656)**

as described in RFC 4251 section 4.1.

TSS Link: *TSS for FCS_SSH_EXT.1*

6.1.2.12 FCS_SSHS_EXT.1 SSH Protocol - Server

PP Origin: *SSH*

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

- **rsa-sha2-256 (RFC 8332)**
- **rsa-sha2-512 (RFC 8332)**
- **ecdsa-sha2-nistp384 (RFC 5656)**
- **ecdsa-sha2-nistp521 (RFC 5656)**

.

TSS Link: *TSS for FCS_SSH_EXT.1*

6.1.3 User data protection (FDP)

6.1.3.1 FDP_HBI_EXT.1 Hardware-Based Isolation Mechanisms

PP Origin: *VPP*

FDP_HBI_EXT.1.1 The TSF shall use

1. **Intel x86: VT-x (CPU, memory), VT-d (PCI-devices), no mechanism (all other devices)**
2. **AMD x86: AMD-V (CPU, memory), AMD-Vi (PCI-devices), no mechanism (all other devices)**
3. **IBM System Z: SIE instruction (CPU, memory), I/O channel (all other devices)**
4. **ARM 64: EL2 mode (CPU, memory), ARM-SMMU (PCI-devices), no mechanism (all other devices)**

to constrain a Guest VM's direct access to the following physical devices: **CPU, memory, PCI-devices, USB devices, UART, RTC, APIC, block devices.**

TSS Link: *TSS for FDP_HBI_EXT.1*

6.1.3.2 FDP_PPR_EXT.1 Physical Platform Resource Controls

PP Origin: *VPP*

FDP_PPR_EXT.1.1 The TSF shall allow an authorized administrator to control Guest VM access to the following physical platform resources: **CPU, memory, PCI-devices (all systems except System Z), USB devices (all systems except System Z)**

Copyright ©2026 SUSE LLC

FDP_PPR_EXT.1.2 The TSF shall explicitly deny all Guest VMs access to the following physical platform resources: **PCI passthrough devices with modifiable option ROM, Memory assigned to host Linux kernel and all software not part of a virtual machine.**

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

TSS Link: *TSS for FDP_PPR_EXT.1*

6.1.3.3 FDP_RIP_EXT.1 Residual Information in Memory

PP Origin: *VPP*

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.

TSS Link: *TSS for FDP_RIP_EXT.1*

6.1.3.4 FDP_RIP_EXT.2 Residual Information on Disk

PP Origin: *VPP*

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.

TSS Link: *TSS for FDP_RIP_EXT.2*

6.1.3.5 FDP_VMS_EXT.1 VM Separation

PP Origin: *VPP*

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.

TSS Link: *TSS for FDP_VMS_EXT.1*

6.1.3.6 FDP_VNC_EXT.1 Virtual Networking Components

PP Origin: *VPP*

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.

TSS Link: *TSS for FDP_VNC_EXT.1*

Copyright ©2026 SUSE LLC

6.1.4 Identification and authentication (FIA)

6.1.4.1 FIA_AFL_EXT.1 Authentication Failure Handling

PP Origin: *VPP*

FIA_AFL_EXT.1.1 The TSF shall detect when **an administrator configurable positive integer within 1 and 2^32 - 1** unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using **username and password**.

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

TSS Link: *TSS for FIA_AFL_EXT.1*

6.1.4.2 FIA_UAU.5 Multiple Authentication Mechanisms

PP Origin: *VPP*

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 **following rules:**

- **Authentication based on username and password: is performed for SSH password-based as well as console login requests with credentials stored by the host OS.**
- **Authentication based on SSH keys: is performed SSH key-based login requests with SSH keys stored by the host OS.**

TSS Link: *TSS for FIA_UAU.5*

6.1.4.3 FIA_UIA_EXT.1 Administrator Identification and Authentication

PP Origin: *VPP*

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.

TSS Link: *TSS for FIA_UIA_EXT.1*

6.1.4.4 FIA_PMG_EXT.1 Password Management

PP Origin: *VPP*

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

Copyright ©2026 SUSE LLC

TSS Link: *TSS for FIA_UIA_EXT.1*

6.1.5 Security management (FMT)

6.1.5.1 FMT_MOF_EXT.1 Management of Security Functions Behavior

PP Origin: *SV*

FMT_MOF_EXT.1.1 The TSF shall be capable of supporting **local, 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~~, *Table 12* based on the following key: X = Mandatory (TOE must provide that function to that role), O = Optional (TOE may or may not provide that function to that role), N = Not Permitted (TOE must not provide that function to that role), S = Selection-Based (TOE must provide that function to that role if the TOE claims a particular selection-based SFR) *Grey/Hyphen - Not supported by the specified role.*

~~Number~~	**Function**	**Admin**	**User**
1	Ability to update the Virtualization System	X	-
2	**Ability to configure Administrator password policy as defined in FIA_PMG_EXT.1**	X	-
3	Ability to create, configure and delete VMs	X	-
4	Ability to set default initial VM configurations	X	-
5	Ability to configure virtual networks including VM	X	-
6	Ability to configure and manage the audit system and audit data	X	-
7	Ability to configure VM access to physical devices	X	-
8	Ability to configure inter-VM data sharing	X	-
9	-	-	-
10	Ability to configure removable media policy	X	-
11	Ability to configure the cryptographic functionality	X	-
12	Ability to change default authorization factors	X	-
13	Ability to enable/disable screen lock	-	-
14	Ability to configure screen lock inactivity timeout	-	-
15	Ability to configure remote connection inactivity timeout	X	-
16	Ability to configure lockout policy for unsuccessful authentication attempts through **limiting number of attempts during a time period**	X	-
17	**Not applicable**	-	-
18	Ability to configure name/address of audit/logging server to which to send audit/logging records	X	-
19	Ability to configure name/address of network time server	X	-

Copyright ©2026 SUSE LLC

[redacted]	Function	Admin	User
20	Ability to configure banner	X	-
21	Ability to connect/disconnect removable devices to/from a VM	X	-
22	Ability to start a VM	X	-
23	Ability to stop/halt a VM	X	-
24	Ability to checkpoint a VM	-	-
25	Ability to suspend a VM	-	-
26	Ability to resume a VM	-	-
27	**Not applicable**	X	-

Table 12: Management functions (SV)

TSS Link: *TSS for FMT_MOF_EXT.1*

6.1.5.2 FMT_SMO_EXT.1 Separation of Management and Operational Networks

PP Origin: *VPP*

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

TSS Link: *TSS for FMT_SMO_EXT.1*

6.1.6 Protection of the TSF (FPT)

6.1.6.1 FPT_DVD_EXT.1 Non-Existence of Disconnected Virtual Devices

PP Origin: *VPP*

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.

TSS Link: *TSS for FPT_DVD_EXT.1*

6.1.6.2 FPT_EEM_EXT.1 Execution Environment Mitigations

PP Origin: *VPP*

FPT_EEM_EXT.1.1 The TSF shall take advantage of execution environment-based vulnerability mitigation mechanisms supported by the Platform such as: **Address space randomization, Memory execution protection, Stack buffer overflow protection**

TSS Link: *TSS for FPT_EEM_EXT.1*

6.1.6.3 FPT_HAS_EXT.1 Hardware Assists

PP Origin: *VPP*

FPT_HAS_EXT.1.1 The VMM shall use

a) Intel x86: VT-x

Copyright ©2026 SUSE LLC

b) AMD x86: AMD-V

c) IBM System Z: SIE instruction

d) ARM 64: EL2 mode

to reduce or eliminate the need for binary translation.

FPT_HAS_EXT.1.2 The VMM shall use

a) Intel x86: EPT

b) AMD x86: EPT

c) IBM System Z: SIE instruction

d) ARM 64: EL2 mode

to reduce or eliminate the need for shadow page tables.

TSS Link: *TSS for FPT_HAS_EXT.1*

6.1.6.4 FPT_HCL_EXT.1 Hypercall Controls

PP Origin: *VPP*

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.

TSS Link: *TSS for FPT_HCL_EXT.1*

6.1.6.5 FPT_RDM_EXT.1 Removable Devices and Media

PP Origin: *VPP*

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

FPT_RDM_EXT.1.2 The TSF shall enforce the following rules when **virtual removable media** are switched between information domains, then **the Administrator has granted explicit access for the media or device to be connected to the receiving domain**

TSS Link: *TSS for FPT_RDM_EXT.1*

6.1.6.6 FPT_TUD_EXT.1 Trusted Update to the Virtualization System

PP Origin: *VPP*

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

FPT_TUD_EXT.1.2 The TSF shall provide administrators the ability to manually initiate updates to TOE firmware/software and **automatic updates.**

FPT_TUD_EXT.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a **digital signature mechanism not using certificates** prior to installing those updates.

TSS Link: *TSS for FPT_TUD_EXT.1*

6.1.6.7 FPT_VDP_EXT.1 Virtual Device Parameters

PP Origin: *VPP*

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

Copyright ©2026 SUSE LLC

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.

TSS Link: *TSS for FPT_VDP_EXT.1*

6.1.6.8 FPT_VIV_EXT.1 VMM Isolation from VMs

PP Origin: *VPP*

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.

TSS Link: *TSS for FPT_VIV_EXT.1*

6.1.7 TOE access (FTA)

6.1.7.1 FTA_TAB.1 TOE access banners

PP Origin: *VPP*

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.

TSS Link: *TSS for FTA_TAB.1*

6.1.8 Trusted path/channels (FTP)

6.1.8.1 FTP_ITC_EXT.1 Trusted Channel Communications

PP Origin: *VPP*

FTP_ITC_EXT.1.1 The TSF shall use

- **SSH as conforming to the Functional Package for Secure Shell**

and **certificate-based authentication of the remote peer, 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), separation of management and operational networks (if selected in FMT_SMO_EXT.1), SSH peer** 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.

TSS Link: *TSS for FTP_ITC_EXT.1*

6.1.8.2 FTP_TRP.1 Trusted Path

PP Origin: *VPP*

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].

Copyright ©2026 SUSE LLC

FTP_TRP.1.1 The TSF shall permit [remote administrators] to initiate communication via the trusted path.

FTP_TRP.1.1 The TSF shall require the use of the trusted path for [[all remote administration actions]].

TSS Link: *TSS for FTP_TRP.1*

6.1.8.3 FTP_UIF_EXT.1 User Interface: I/O Focus

PP Origin: *VPP*

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

TSS Link: *TSS for FTP_UIF_EXT.1*

6.1.8.4 FTP_UIF_EXT.2 User Interface: Identification of VM

PP Origin: *VPP*

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

TSS Link: *TSS for FTP_UIF_EXT.2*

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.

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/Hash	O.MANAGEMENT_ACCESS, O.VMM_INTEGRITY
FCS_COP.1/KeyedHash	O.MANAGEMENT_ACCESS, O.VMM_INTEGRITY
FCS_COP.1/Sig	O.MANAGEMENT_ACCESS, O.VMM_INTEGRITY

Copyright ©2026 SUSE LLC

Security functional requirements	Objectives
FCS_COP.1/UDE	O.MANAGEMENT_ACCESS, O.VMM_INTEGRITY
FCS_ENT_EXT.1	O.DOMAIN_INTEGRITY, O.VM_ENTROPY
FCS_RBG_EXT.1	O.DOMAIN_INTEGRITY, O.MANAGEMENT_ACCESS, O.VM_ENTROPY, O.VMM_INTEGRITY
FCS_SSH_EXT.1	O.MANAGEMENT_ACCESS
FCS_SSHC_EXT.1	O.MANAGEMENT_ACCESS
FCS_SSHS_EXT.1	O.MANAGEMENT_ACCESS
FDP_HBI_EXT.1	O.PLATFORM_INTEGRITY
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
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_UAU.5	O.MANAGEMENT_ACCESS
FIA_UIA_EXT.1	O.MANAGEMENT_ACCESS
FIA_PMG_EXT.1	O.MANAGEMENT_ACCESS
FMT_MOF_EXT.1	O.MANAGEMENT_ACCESS, O.VMM_INTEGRITY
FMT_SMO_EXT.1	O.MANAGEMENT_ACCESS
FPT_DVD_EXT.1	O.PLATFORM_INTEGRITY, O.VM_ISOLATION

Copyright ©2026 SUSE LLC

Security functional requirements	Objectives
FPT_EEM_EXT.1	O.DOMAIN_INTEGRITY, O.PLATFORM_INTEGRITY, O.VM_ISOLATION, O.VMM_INTEGRITY
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	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
FTP_ITC_EXT.1	O.MANAGEMENT_ACCESS
FTP_TRP.1	O.MANAGEMENT_ACCESS
FTP_UIF_EXT.1	O.DOMAIN_INTEGRITY
FTP_UIF_EXT.2	O.DOMAIN_INTEGRITY

Table 13: Mapping of security functional requirements to security objectives

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.

Security objectives	Rationale
O.VM_ISOLATION	[VPP]FAU_GEN.1: Audit events can report attempts to breach isolation. [VPP]FCS_CKM_EXT.4: Requires cryptographic key destruction to protect domain data in shared storage. [VPP]FDP_PPR_EXT.1: Requires support for reducing attack surface through disabling access to unneeded physical platform resources. [VPP]FDP_RIP_EXT.1: Ensures that domain data is cleared from memory before memory is re-allocated. [VPP]FDP_RIP_EXT.2: Ensures that domain data is cleared from physical storage upon re-allocation of the storage.

Copyright ©2026 SUSE LLC

Security objectives	Rationale
	[VPP]FDP_VMS_EXT.1: Ensures that authorized data transfers between VMs are done securely. [VPP]FDP_VNC_EXT.1: Ensures that network traffic is visible only to VMs configured to be that network. [VPP]FPT_DVD_EXT.1: Ensures that VMs can access only those virtual devices that they are configured to access. [VPP]FPT_EEM_EXT.1: Requires that the TOE use security mechanisms supported by the physical platform. [VPP]FPT_HAS_EXT.1: Requires that the TOE use platform-supported virtualization assists to reduce attack surface. [VPP]FPT_VDP_EXT.1: Requires validation of parameter data passed to the hardware abstraction by untrusted VMs. [VPP]FPT_VIV_EXT.1: Ensures that untrusted VMs cannot invoke privileged code without proper hypervisor mediation.
O.VMM_INTEGRITY	[VPP]FAU_GEN.1: Audit events can report potential integrity breaches and attempts. [VPP]FCS_CKM.1: Requires generation of asymmetric keys for protection of integrity measures. [VPP]FCS_COP.1/Hash: Ensures proper functioning of cryptographic algorithms used to protect data integrity. [VPP]FCS_COP.1/KeyedHash: Ensures proper functioning of cryptographic algorithms used to protect data integrity. [VPP]FCS_COP.1/Sig: Ensures proper functioning of cryptographic algorithms used to protect data integrity. [VPP]FCS_COP.1/UDE: Ensures proper functioning of cryptographic algorithms used to protect data integrity. [VPP]FCS_RBG_EXT.1: Requires that the TOE has access to high-quality entropy for cryptographic purposes. [VPP]FDP_PPR_EXT.1: Requires support for reducing attack surface through disabling access to unneeded physical platform resources. [VPP]FDP_VMS_EXT.1: Ensures that authorized data transfers between VMs are done securely. [VPP]FDP_VNC_EXT.1: Ensures that network traffic is visible only to VMs configured to be that network. [VPP]FPT_EEM_EXT.1: Requires that the TOE use security mechanisms supported by the physical platform. [VPP]FPT_HAS_EXT.1: Requires that the TOE use platform-supported virtualization assists to reduce attack surface. [VPP]FPT_HCL_EXT.1: Requires that Hypercall parameters be validated. [VPP]FPT_VDP_EXT.1: Requires validation of parameter data passed to the hardware abstraction by untrusted VMs. [VPP]FPT_VIV_EXT.1: Ensures that untrusted VMs cannot invoke privileged code without proper hypervisor mediation.

Copyright ©2026 SUSE LLC

Security objectives	Rationale
	[SV]FMT_MOF_EXT.1: Integrity of a Virtualization System can be maintained by ensuring that the only way to modify the VS is through a trusted update process initiated by an authorized Administrator as required by FMT_MOF_EXT.1.
O.PLATFORM_INTEGRITY	[VPP]FDP_HBI_EXT.1: Requires that the TOE use platform- supported mechanisms for access to physical devices. [VPP]FDP_PPR_EXT.1: Requires support for reducing attack surface through disabling access to unneeded physical platform resources. [VPP]FDP_VMS_EXT.1: Ensures that authorized data transfers between VMs are done securely. [VPP]FDP_VNC_EXT.1: Ensures that network traffic is visible only to VMs configured to be that network. [VPP]FPT_DVD_EXT.1: Ensures that VMs cannot access virtual devices that they are not configured to access. [VPP]FPT_EEM_EXT.1: Requires that the TOE use security mechanisms supported by the physical platform. [VPP]FPT_HAS_EXT.1: Requires that the TOE use platform- supported virtualization assists to reduce attack surface. [VPP]FPT_HCL_EXT.1: Requires that Hypercall parameters be validated. [VPP]FPT_VDP_EXT.1: Requires validation of parameter data passed to the hardware abstraction by untrusted VMs. [VPP]FPT_VIV_EXT.1: Ensures that untrusted VMs cannot invoke privileged code without proper hypervisor mediation.
O.DOMAIN_INTEGRITY	[VPP]FCS_CKM_EXT.4: Requires cryptographic key destruction to protect domain data in shared storage. [VPP]FCS_ENT_EXT.1: Requires that domains have access to high-quality entropy for cryptographic purposes. [VPP]FCS_RBG_EXT.1: Requires that the TOE has access to high-quality entropy for cryptographic purposes. [VPP]FDP_RIP_EXT.1: Ensures that domain data is cleared from memory before memory is re-allocated to another domain. [VPP]FDP_RIP_EXT.2: Ensures that domain data is cleared from physical storage upon re-allocation of the storage to another domain. [VPP]FDP_VMS_EXT.1: Ensures that authorized data transfers between domains are done securely. [VPP]FDP_VNC_EXT.1:Ensures that network traffic is visible only to VMs configured to be that network. [VPP]FPT_EEM_EXT.1: Requires that the TOE use security mechanisms supported by the physical platform. [VPP]FPT_HAS_EXT.1: Requires that the TOE use platform-supported virtualization assists to reduce attack surface. [VPP]FPT_RDM_EXT.1: Requires support for rules for switching removeable media between domains to reduce the chance of data spillage.

Copyright ©2026 SUSE LLC

Security objectives	Rationale
	[VPP]FPT_VDP_EXT.1: Requires validation of parameter data passed to the hardware abstraction by untrusted VMs. [VPP]FTP_UIF_EXT.1: Ensures that users are able to determine the domain with the current input focus. [VPP]FTP_UIF_EXT.2: Ensures that users can know the identity of any VM that they can access.
O.MANAGEMENT_ACCESS	[VPP]FAU_GEN.1: Audit events report attempts to access the management subsystem. [VPP]FCS_CKM.1: Requires generation of asymmetric keys for trusted communications channels. [VPP]FCS_CKM.2: Requires establishment of cryptographic keys for trusted communications channels. [VPP]FCS_COP.1/Hash: Ensures proper functioning of cryptographic algorithms used to implement access controls. [VPP]FCS_COP.1/KeyedHash: Ensures proper functioning of cryptographic algorithms used to implement access controls. [VPP]FCS_COP.1/Sig: Ensures proper functioning of cryptographic algorithms used to implement access controls. [VPP]FCS_COP.1/UDE: Ensures proper functioning of cryptographic algorithms used to implement access controls. [VPP]FCS_RBG_EXT.1: Requires that the TOE has access to high-quality entropy for cryptographic purposes. [VPP]FIA_AFL_EXT.1: Requires that the TOE detect failed authentication attempts for Administrator access. [VPP]FIA_PMG_EXT.1: Ensures that password-based administrator login is properly implemented. [VPP]FIA_UAU.5: Ensures that strong mechanisms are used for Administrator authentication. [VPP]FIA_UIA_EXT.1: Requires that Administrators be successfully authenticated before performing management functions. [VPP]FMT_SMO_EXT.1: Requires that the TOE support having separate management and operational networks. [VPP]FTP_ITC_EXT.1: Ensures that trusted communications channels are implemented using good cryptography. [VPP]FTP_TRP.1: Ensures that certain communications use a trusted path. [VPP]FTA_TAB.1: Displays advisory notice and consent warning message regarding use of the TOE to Administrators. [SV]FMT_MOF_EXT.1: Access to management functions must be limited to authorized Administrators as managed through controls required by FMT_MOF_EXT.1. [SSH]FCS_SSH_EXT.1: Ensures that SSH trusted communications channels are implemented properly. [SSH]FCS_SSHC_EXT.1: Ensures that SSH client trusted communications channels are implemented properly.

Copyright ©2026 SUSE LLC

Security objectives	Rationale
	[SSH]FCS_SSHS_EXT.1: Ensures that SSH server trusted communications channels are implemented properly.
O.PATCHED_SOFTWARE	[VPP]FPT_TUD_EXT.1: Requires support for product updates.
O.VM_ENTROPY	[VPP]FCS_ENT_EXT.1: Requires that domains have access to high-quality entropy for cryptographic purposes. [VPP]FCS_RBG_EXT.1: Requires that the TOE has access to high-quality entropy for cryptographic purposes.
O.AUDIT	[VPP]FAU_GEN.1: Requires reporting of audit events. [VPP]FAU_SAR.1: Requires support for Administrator review of audit records. [VPP]FAU_STG.1: Requires protection of stored audit records. [VPP]FAU_STG_EXT.1: Requires support for protected transmission of audit records off the TOE.
O.CORRECTLY_APPLIED_CONFIGURATION	[VPP]FDP_VMS_EXT.1: Ensures that data sharing between VMs is turned off by default.
O.RESOURCE_ALLOCATION	[VPP]FCS_CKM_EXT.4: Requires cryptographic key destruction to ensure residual data in shared storage is unrecoverable. [VPP]FDP_RIP_EXT.1: Ensures that domain data is cleared from memory before memory is re-allocated. [VPP]FDP_RIP_EXT.2: Ensures that domain data is cleared from storage upon re-allocation of the storage.

Table 14: Security objectives for the TOE rationale

6.3 Security Assurance Requirements

The security assurance requirements (SARs) for the TOE are defined in the VPP protection profile; and defined in CC assurance package.

6.3.1 ALC Life-cycle support

6.3.1.1 ALC_TSU_EXT.1 Timely Security Updates

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.

Copyright ©2026 SUSE LLC

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.

6.4 Security Assurance Requirements Rationale

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

Copyright ©2026 SUSE LLC

7 TOE Summary Specification

7.1 TSS Security Assurance Evaluation Activity

7.1.1 Timely security updates (ALC_TSU_EXT.1)

SUSE generally does not disclose, discuss, or confirm security issues until a full investigation is complete and any necessary patches or releases are available. Further details about the security release process can be obtained at https://www.suse.com/support/security/ Once an issue has been confirmed and a patch has been made available, references containing technical details on the patches are made available and Common Vulnerabilities and Exposures (CVEs), etc. are released.

SUSE distributes information about security issues in its products through its "SUSE Update Advisory" page. (https://www.suse.com/support/update/)

Potential security vulnerabilities can be reported by following the procedures on the "SUSE Security Contacts" page (https://www.suse.com/support/security/contact/). This includes sending an email to "security@suse.com" or "security@suse.de" and includes the ability to encrypt information using the SUSE Security Team PGP key.

7.2 TOE Security Functionality

7.2.1 Audit

7.2.1.1 FAU_GEN.1 Audit Data Generation (Refined)

PP Origin: *VPP*

PP Origin: *SSH*

SFR Link: *FAU_GEN.1*

Audit events are generated for the following audit functions.

- Start-up and shut-down of the audit functions
- All administrative actions relevant to claimed SFRs as defined in the Auditable Events Table from the Client and Server PP-Modules
- Other auditable events specified in SFR

The Lightweight Audit Framework (LAF) is designed to be an audit system for Linux compliant with the requirements from Common Criteria. LAF is able to intercept all system calls as well as retrieving audit log entries from privileged user space applications. The subsystem allows configuring the events to be actually audited from the set of all events that are possible to be audited. Those events are configured in a specific configuration file and then the kernel is notified to build its own internal structure for the events to be audited.

Audit functionality

The Linux kernel implements the core of the LAF functionality. It gathers all audit events, analyzes these events based on the audit rules and forwards the audit events that are requested to be audited to the audit daemon executing in user space.

Audit events are generated in various places of the kernel. In addition, a user space application can create audit records which needs to be fed to the kernel for further processing.

The audit functionality of the Linux kernel is configured by user space applications which communicate with the kernel using a specific netlink communication channel. This netlink channel is also to be used by applications that want to send an audit event to the kernel.

Copyright ©2026 SUSE LLC

The kernel netlink interface is usable only by applications possessing the following capabilities:

- CAP_AUDIT_CONTROL: Performing management operations like adding or deleting audit rules, setting or getting auditing parameters;
- CAP_AUDIT_WRITE: Submitting audit records to the kernel which in turn forwards the audit records to the audit daemon.

Based on the audit rules, the kernel decides whether an audit event is discarded or to be sent to the user space audit daemon for storing it in the audit trail. The kernel sends the message to the audit daemon again using the above mentioned netlink communication channel. The audit daemon writes the audit records to the audit trail. An internal queuing mechanism is used for this purpose. When the queue does not have sufficient space to hold an audit record the TOE switches into single user mode, is halted, all processes are stopped that generate audit records, or the audit daemon executes an administrator-specified notification action depending on the configuration of the audit daemon. This ensures that audit records do not get lost due to resource shortage and the administrator can backup and clear the audit trail to free disk space for new audit logs.

Access to audit data by normal users is prohibited by the discretionary access control function of the TOE, which is used to restrict the access to the audit trail and audit configuration files to the system administrator only.

The system administrator can define the events to be audited from the overall events that the Lightweight Audit Framework using simple filter expressions. This allows for a flexible definition of the events to be audited and the conditions under which events are audited. The system administrator is also able to define a set of user IDs for which auditing is active or alternatively a set of user IDs that are not audited.

The system administrator can select files to be audited by adding them to a watch list that is loaded into the kernel.

The audit trail is stored in files that are readable by the root user only.

Audit trail

An audit record consists of one or more lines of text containing fields in a "keyword=value" tagged format. The following information is contained in all audit record lines:

- Type: indicates the source of the event, such as SYSCALL, PATH, USER_LOGIN, or LOGIN
- Timestamp: Date and time the audit record was generated
- Audit ID: unique numerical event identifier
- Login ID ("auid"), the user ID of the user authenticated by the system (regardless if the user has changed his real and / or effective user ID afterwards)
- Effective user ID: the effective user ID of the process at the time the audit event was generated
- Success or failure (where appropriate)
- Process ID of the subject that caused the event (PID)

This information is followed by event specific data. In some cases, such as SYSCALL event records involving file system objects, multiple text lines will be generated for a single event, these all have the same time stamp and audit ID to permit easy correlation.

The audit trail is stored in ASCII text. The TOE provides tools for managing ASCII files that can be used for post-processing of audit data. These tools include:

- less - reads the ASCII audit data
- ausearch - allows selective extraction of records from the audit trail using defined selection criteria

Copyright ©2026 SUSE LLC

- sort - The audit records are listed in chronological order by default. The sort utility can be used together with ausearch to use a different sorting order.

The audit trail is stored in files which are accessible by root only.

7.2.1.2 FAU_SAR.1 Audit Review

PP Origin: *VPP*

SFR Link: *FAU_SAR.1*

Administrators can read the audit record with a text editor considering the audit data is ASCII as well as with provided tools like ausearch. The audit log is read-accessible to administrators only.

7.2.1.3 FAU_STG.1 Protected Audit Trail Storage

PP Origin: *VPP*

SFR Link: *FAU_STG.1*

The audit log is read-accessible to administrators only using file access permissions on the directory used for storage as well as on the audit log files.

7.2.1.4 FAU_STG_EXT.1 Off-Loading of Audit Data

PP Origin: *VPP*

SFR Link: *FAU_STG_EXT.1*

By using the service of audisp-remote, the audit daemon can replicate the audit logs to a remote audit server. The audisp-remote tool is capable of transmitting the data through a SSH-protected communication channel as defined by TSS_FTP_ITC_EXT.1.

When audit data is stored locally, the policy of the audit daemon allows the specification that in case the audit trail becomes full, the oldest audit entries are overwritten with new audit entries.

7.2.2 Cryptography

The security features that use cryptography in this ST are the following.

- SSH
- Trusted update

The cryptographic modules used to implement the above security features are the following.

- OpenSSL (user space)
- libgcrypt (support for GPG)

The integrity verification of updates is performed using GPG using libgcrypt.

All other mechanisms rely on OpenSSL for the cryptographic primitives.

Table 15 lists the algorithms discussed in the following subsections.

SFR	Algorithm	Capabilities	Usage
FCS_CKM.1	RSA KeyGen	3072, 4096	SSH mutual authentication
	ECC KeyGen	P-384, P-521, P-256	SSH key establishment and mutual authentication
	FFC schema KeyGen	safe primes compliant to SP800-56A rev. 3	SSH key establishment

Copyright ©2026 SUSE LLC

SFR	Algorithm	Capabilities	Usage
FCS_CKM.2	ECC Key Establishment (KAS-ECC)	P-256, P-384, P-521	SSH key establishment
	FFC Key Establishment (KAS-FFC)	safe primes compliant to SP800-56A rev. 3	SSH key establishment
FCS_COP.1/UDE	AES-CTR, AES-GCM	128-bit, 256-bit	SSH client / server
FCS_COP.1/Hash	SHA-256, SHA-384, SHA-512		Trusted update SSH client / server
FCS_COP.1/Sig	RSA SigGen/SigVer	3072, 4096 with: SHA-256, SHA-384, SHA-512	Trusted update SSH client / server
	ECDSA SigGen/SigVer	P-256, P-384, P-521 with: SHA-256, SHA-384, SHA-512	SSH client / server
FCS_COP.1/KeyedHash	HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512		SSH client / server
FCS_RBG_EXT.1	CTR_DRBG	AES	SSH client / server

Table 15: Cryptographic algorithm table

7.2.2.1 FCS_CKM.1 Cryptographic Key Generation

PP Origin: *VPP*

SFR Link: *FCS_CKM.1*

The TOE supports generation of 3072-bit, and 4096-bit RSA keys conforming to FIPS PUB 186-5 Digital Signature Standard (DSS), Appendix A.1. The TOE supports the generation of RSA keys for use SSH sessions.

The TOE supports NIST curves P-256, P-384, and P-521 for key generation conforming to "FIPS PUB 186-5 Digital Signature Standard (DSS)", Appendix. SSH sessions use these curves for ECDH key establishment and for ECDSA-based client authentication.

Please refer to Table 15 for details.

7.2.2.2 FCS_CKM.2 Cryptographic Key Establishment

PP Origin: *VPP*

SFR Link: *FCS_CKM.2*

The TOE supports cryptographic key establishment using the following schemes.

- Elliptic curve-based key establishment with NIST curves P-256, P-384, and P-521 as specified in NIST SP 800-56A rev 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"
- Finite field-based key establishment schemes that meets NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" using safe primes as defined in SP800-56A rev 3.

For SSH support, ECDH-based key establishment is supported.

Copyright ©2026 SUSE LLC

For SSH support, DH-based key establishment is supported.

Please refer to Table 15 for details.

7.2.2.3 FCS_CKM_EXT.4 Cryptographic Key Destruction

PP Origin: *VPP*

SFR Link: *FCS_CKM_EXT.4*

Ephemeral cryptographic key material is held in RAM and is zeroized as soon as it is not required any more by overwriting the memory location with zeros.

The following keys are managed by OpenSSL on behalf of the respective protected communication channels:

- Ephemeral symmetric keys: The keys are derived during the cryptographic handshake from the Diffie-Hellman operation and stay in memory until rekey or destruction of the connection.
- Ephemeral MAC keys: The keys are derived during the cryptographic handshake from the Diffie-Hellman operation and stay in memory until rekey or destruction of the connection.
- Ephemeral DH/ECDH keys: The keys are generated by OpenSSL using its DRBG during the cryptographic handshake from the Diffie-Hellman operation and stay in memory until the handshake is completed.
- Ephemeral representation of asymmetric keys: The keys are read from files during startup of the application handling the cryptographic connection and stay in memory until the application terminates.

All ephemeral keys are held by OpenSSL in cipher handles which contains the memory buffer holding the keys. When the cipher handle is released, the key memory is zeroized.

For non-volatile memory, the life of a SSH key pairs is indefinite. A user or administrator can manually destroy them. To erase long-term key material held in files the TOE provides the tool fstrim. After a deletion of a file with sensitive data, this tool uses the SSD TRIM command to inform the SSD to discard unused blocks bypassing wear leveling. In addition, the tool shred is available that overwrites files multiple times with random data. This tool can be used for HDD to delete data.

7.2.2.4 FCS_COP.1/Hash Cryptographic Operation - Hashing

PP Origin: *VPP*

SFR Link: *FCS_COP.1/Hash*

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

The following hashing algorithms supported: SHA-256, SHA-384, and SHA-512. The message digest sizes supported are: 256 bits, 384 bits, and 512 bits.

Please refer to Table 15 for details.

7.2.2.5 FCS_COP.1/KeyedHash Cryptographic Operation - Keyed-Hash Message Authentication (Refined)

PP Origin: *VPP*

SFR Link: *FCS_COP.1/KeyedHash*

The TOE supports keyed-hash message authentication conforming to FIPS Pub 198-1 "The Keyed-Hash Message Authentication Code" and FIPS Pub 180-4 "Secure Hash Standard" with the following algorithms

Copyright ©2026 SUSE LLC

- HMAC-SHA-256: keysize 256 bits, output size 256 bits, block size 512 bits
- HMAC-SHA-384: keysize 384 bits, output size 384 bits, block size 1024 bits
- HMAC-SHA-512: keysize 512 bits, output size 512 bits, block size 1024 bits

Please refer to Table 15 for details.

7.2.2.6 FCS_COP.1/Sig Cryptographic Operation - Signing (Refined)

PP Origin: *VPP*

SFR Link: *FCS_COP.1/Sig*

The TOE provides cryptographic signature generation and verification in accordance with the following cryptographic algorithms.

- RSA digital signature algorithm conforming to FIPS Pub 186-5, "Digital Signature Standard (DSS)", Section 5. The RSA key sizes supported are: 3072 bits, and 4096-bit.
- Elliptical curve digital signature algorithm conforming to FIPS Pub 186-5, "Digital Signature Standard (DSS)", Section 6. The TOE supports curves P-256, P-384, and P-521.

Please refer to Table 15 for details.

7.2.2.7 FCS_COP.1/UDE Cryptographic Operation - AES Data Encryption / Decryption

PP Origin: *VPP*

SFR Link: *FCS_COP.1/UDE*

The TOE supports AES encryption using 128-bit or 256-bit keys in the following modes.

- CTR as specified in NIST SP 800-38A
- GCM as specified in NIST SP 800-38D

Please refer to Table 15 for details.

7.2.2.8 FCS_ENT_EXT.1 Entropy for Virtual Machines

PP Origin: *VPP*

SFR Link: *FCS_ENT_EXT.1*

The Linux host kernel offers a virtio-rng interface to its entropy source of /dev/random to virtual machines allowing them to pull entropy from the host. The virtio-rng interface ensures that multiple virtual machines are separated from each other such that the generation requests are isolated and cannot interfere with each other.

7.2.2.9 FCS_RBG_EXT.1 Random Bit Generation

PP Origin: *VPP*

SFR Link: *FCS_RBG_EXT.1*

The TOE uses a CTR_DRBG(AES) to generate random bits. The DRBG is seeded by the userspace software-based noise source of the Jitter RNG version 3.4.1. The noise source provides data with a minimum of 384 bits of entropy to initially seed the DRBG and with 256 bits of entropy to reseed the DRBG. The data from the noise source is given to the DRBG seed/reseed function.

Please refer to Table 15 for details.

7.2.2.10 FCS_SSH_EXT.1 SSH Protocol

PP Origin: *SSH*

SFR Link: *FCS_SSH_EXT.1*

Copyright ©2026 SUSE LLC

The TOE provides the Secure Shell Protocol Version 2 (SSH v2.0) to allow users from a remote host to establish a secure connection and perform a logon to the TOE.

The following table documents implementation details concerning the OpenSSH implementation's compliance to the relevant standards. It addresses areas where the standards permit different implementation choices such as optional features.

Reference	Description	Implementation Details
RFC 4253 chapter 5	Compatibility with old SSH versions	The OpenSSH implementation is capable of interoperating with clients and servers using the old 1.x protocol. That functionality is explicitly disabled in the evaluated configuration, it permits protocol version 2.0 exclusively.
RFC 4253 section 6.2	Compression	OpenSSH supports the OPTIONAL "zlib" compression method.
RFC 4253 section 6.3	Encryption	The ciphers supported in the evaluated configuration are listed in FCS_SSH_EXT.1 for the SSH protocol.
RFC 4252 chapter 7	Public Key Authentication Method: "publickey"	This REQUIRED authentication method is supported by OpenSSH but can be disabled by the administrator of the OpenSSH daemon.
RFC 4252 chapter 8	Password Authentication Method: "password"	This SHOULD authentication method is supported by OpenSSH but can be disabled by the administrator of the OpenSSH daemon.
RFC 4252 chapter 8	Password change request and setting new password	The OpenSSH implementation supports the optional password change mechanism in the evaluated configuration.
RFC 4252 chapter 9	Host-Based Authentication: "hostbased"	This OPTIONAL authentication method is disabled in the evaluated configuration.

Table 16: SSH implementation notes

The TOE supports the generation of RSA, FFC as well as ECC key pairs. These key pairs are used by OpenSSH for the host keys as well as for the per-user keys. When a user registers his public key with the user he wants to access on the server side, a key-based authentication can be performed instead of a password-based authentication. The key generation mechanism uses the random number generator of the underlying cryptographic library. The evaluated configuration permits the import of externally-generated key pairs.

The TOE supports the following security functions of the SSH v2.0 protocol:

- Establishing a secure communication channel using the following cryptographic functions provided by the SSH v2.0 protocol:
  - Encryption as defined in section 6.3 of [RFC4253] - the DH/ECDH forming the basis for the key agreement and thus the symmetric / MAC keys are generated using the random number generator of the underlying cryptographic library;
  - Diffie-Hellman key agreement used in conjunction with the key derivation function as defined in section 7.2 of [RFC4253] supplemented by [RFC5656] chapter 4;
  - The keyed hash function for integrity protection as defined in section 6.4 of [RFC4253].

  Note: The protocol supports more cryptographic algorithms than the ones listed above. Those other algorithms are not covered by this evaluation and should be disabled or not used when running the evaluated configuration.

Copyright ©2026 SUSE LLC

- Performing user authentication using the standard password-based authentication method the TOE provides for users (password authentication method as defined in chapter 5 of [RFC4252]).
- Performing user authentication using a RSA, or ECDSA key-based authentication method (public key authentication method as defined in chapter 5 of [RFC4252]).
- Checking the integrity of the messages exchanged and close down the connection in case an integrity error is detected.

The OpenSSH applications of sshd, ssh and ssh-keygen use the OpenSSL random number generator seeded by the getrandom system call to generate cryptographic keys. OpenSSL provides different DRNGs depending whether the FIPS 140-2 mode is enabled in the system.

The cryptographic implementations ensure that sensitive data is appropriately zeroized before releasing the associated memory.

The TOE supports the following authentication mechanisms with SSH:

- Password-based authentication
- Key-based authentication with RSA keys and ECC keys.

The TOE maintains a counter for each SSH packet which is increased by the number of received bytes. If the counter reaches the threshold of 262144 bytes, the connection is closed.

After processing at most 2^30 bytes covering both sent and received data or the last re-key is more than 1 hour ago, the TOE initiates a re-keying, when the option RekeyLimit is set appropriately.

The SSH implementation supports the following ciphers:

- Symmetric ciphers: aes128-ctr, aes256-ctr, AES128 GCM, AES256 GCM
- Asymmetric ciphers: rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521
- MAC: hmac-sha2-256, hmac-sha2-512, AES-GCM
- Key agreement: diffie-hellman-group16-sha512, diffie-hellman-group18-sha512, ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521

The aforementioned statements apply equally to the SSH client and server implementation.

7.2.2.11 FCS_SSHC_EXT.1 SSH Protocol - Client

PP Origin: *SSH*

SFR Link: *FCS_SSHC_EXT.1*

The TOE provides an SSH client which supports the SSH server authentication using RSA, EdDSA and ECDSA.

7.2.2.12 FCS_SSHS_EXT.1 SSH Protocol - Server

PP Origin: *SSH*

SFR Link: *FCS_SSHS_EXT.1*

The TOE provides an SSH server which supports the SSH client authentication using RSA, EdDSA and ECDSA.

7.2.3 User data protection

7.2.3.1 FDP_HBI_EXT.1 Hardware-Based Isolation Mechanisms

PP Origin: *VPP*

SFR Link: *FDP_HBI_EXT.1*

Copyright ©2026 SUSE LLC

The TOE uses isolation mechanisms offered by the underlying hardware to separate virtual machines from each other. The following mechanisms are in use:

- Intel x86: VT-x (CPU, memory), VT-d (PCI-devices), no mechanism (all other devices)
- AMD x86: AMD-V (CPU, memory), AMD-Vi (PCI-devices), no mechanism (all other devices)
- IBM System Z: SIE instruction (CPU, memory), I/O channel (all other devices)
- ARM 64: EL2 mode (CPU, memory), ARM-SMMU (PCI-devices), no mechanism (all other devices)

The hardware support is always invoked.

7.2.3.2 FDP_PPR_EXT.1 Physical Platform Resource Controls

PP Origin: *VPP*

SFR Link: *FDP_PPR_EXT.1*

The TOE allows the administrator to control the Guest VM's access to all physical resources, covering CPU, memory, PCI-devices (all systems except System Z), USB devices (all systems except System Z).

Irrespective to the memory mappings that can be configured by administrators, the memory used by the hypervisor (the host Linux kernel) and the VMM (all user space applications apart from the virtual machines) can never be assigned to virtual machines.

When a VMM (QEMU) is instantiated by the TOE, all resource configurations are applied. This is achieved by the following concept: a single virtual machine is represented as an independent process in the host Linux operating system. This application is granted access to the different resources by regular Linux device assignments. In addition, by using AppArmor policies, the processes resembling different virtual machines with their resources are separated from each other. This implies that a virtual machine is limited to the resources assigned to it.

The platform resources are identified by an administrator as follows:

- CPU: As it is permissible to assign physical CPU cores to virtual machines, the administrator utilizes the host platform's CPU topology to identify the applicable CPUs. Per default, only virtual CPUs are assigned backed by Linux threads in the host which implies that there are no CPU resources to be selected by the administrator.
- Memory: The administrator only selects the amount of memory without selecting particular physical sections of the host memory.
- PCI-devices: The PCI device is identified by its unique PCI device ID.
- USB devices: The USB device is identified by its unique set of USB bus ID, USB device number and USB device ID.

The administrator is only permitted to assign physical devices to virtual machines where the administrator verified that those devices do not offer the means to update the potentially present PCI BIOS also know as option ROM as this BIOS may be executed during device enumeration on x86 platforms.

7.2.3.3 FDP_RIP_EXT.1 Residual Information in Memory

PP Origin: *VPP*

SFR Link: *FDP_RIP_EXT.1*

The memory used by the TOE for itself as well as virtual machines is cleared when the memory is reassigned to a different purpose. The only deviation applies to cryptographic sensitive data like keys where the used cryptographic library of OpenSSL ensures the memory is cleared at the time the memory is not needed any more. The clearing is implemented by overwriting the relevant memory locations with zeros.

Copyright ©2026 SUSE LLC

7.2.3.4 FDP_RIP_EXT.2 Residual Information on Disk

PP Origin: *VPP*

SFR Link: *FDP_RIP_EXT.2*

The disk space assigned to virtual machines is maintained as files by the host system. When new disk space is assigned to a virtual machine, a new file must be created whose content is automatically zeroized by the TOE during creation of that file.

7.2.3.5 FDP_VMS_EXT.1 VM Separation

PP Origin: *VPP*

SFR Link: *FDP_VMS_EXT.1*

The TOE allows the administrator to configure virtual networking. Such virtual networking is implemented by defining one or more bridge networks to which zero or more virtual network interfaces from virtual machines can be assigned. It is possible that virtual network interfaces from different virtual machines are assigned which implies that they are able to communicate using that bridge network. The administrator is allowed to configure zero or more physical network interfaces to be assigned to a bridge network allowing the communication of virtual machines with external entities.

Different bridge networks are not allowed to communicate with each other within the TOE. Of course, it is possible that different bridge networks may have access to common remote entities which then allow communication between the bridge networks. This behavior, however, cannot be controlled by the TOE as it follows standard networking practices. If an administrator does not want to allow any communication between bridge networks, the administrator must ensure that either no remote accesses are possible or that the remote systems are separated from each other.

Apart from bridge networks, administrators are able to assign PCI network devices directly to guest VMs.

7.2.3.6 FDP_VNC_EXT.1 Virtual Networking Components

PP Origin: *VPP*

SFR Link: *FDP_VNC_EXT.1*

See the TSS for FDP_VMS_EXT.1.

7.2.4 Identification and authentication

7.2.4.1 FIA_AFL.1 Authentication Failure Handling

PP Origin: *VPP*

SFR Link: *FIA_AFL_EXT.1*

The TOE will detect when an administrator configurable integer within 1 and 2^32 - 1 unsuccessful authentication attempts for authentication based on username and password attempts as well as SSH-based authentication attempts have been met. Once the specified number of unsuccessful authentication attempts for an account has been met, the TOE will lock out the account. This account must then be unlocked by the administrator.

7.2.4.2 FIA_UAU.5 Multiple Authentication Mechanisms

PP Origin: *VPP*

SFR Link: *FIA_UAU.5*

The TOE supports local authentication based on username/password.

Copyright ©2026 SUSE LLC

For password-based authentication, the user account contains a username and a password. A random salt is created for the password which is used to derive a SHA2-512 hash value that is stored in the /etc/shadow file. When a user logs into the system, the TOE uses the entered password and the randomly generated salt and compares this with the stored value. If they match, then the user is granted access to the system. If the values do not match, then the user is not granted access.

The TOE supports local authentication with SSH-keys. The public key is stored with the SSH server that a user wants to use for the key-based authentication. Similarly, an SSH client authenticates the remote SSH server's public key against his local database of public keys. SSH-key-based authentication is specified in RFC4252.

7.2.4.3 FIA_UIA_EXT.1 Administrator Identification and Authentication

PP Origin: *VPP*

SFR Link: *FIA_UIA_EXT.1*

Administrators have first to log on via SSH using their unprivileged accounts and one of the mechanisms specified by FIA_UAU.5 and then use the su or sudo commands to assume the root user identity. Thus, such users are subject to all identification and authentication constraints normal users are subjected, too.

7.2.4.4 FIA_PMG_EXT.1 Password Management

PP Origin: *VPP*

SFR Link: *FIA_PMG_EXT.1*

When configuring passcodes, the TOE enforces a proper passcode quality rule as defined in FIA_PMG_EXT.1. The enforcement is provided by means of well-chosen PAM configuration.

7.2.5 Security management

7.2.5.1 FMT_MOF_EXT.1 Management of Security Functions Behavior

PP Origin: *SV*

SFR Link: *FMT_MOF_EXT.1*

The TOE supports the following roles: Administrator and User. The Administrator is a member of the local admin group or an applied configuration profile, and the User is an unprivileged account.

Administration can take place either locally by allowing administrators to log onto the TOE, or remotely using the libvirtd administrative interface.

The Administrator has access to the management functions enumerated in FMT_MOF_EXT.1.

Access to the administrator-only management functions is restricted by means of appropriate permission bits applied to the configuration files that hold the respective configuration data. In addition, access to the libvirtd administrative interface is restricted to users passing a successful authentication using bi-directional SSH key-based authentication.

7.2.5.2 FMT_SMO_EXT.1 Separation of Management and Operational Networks

PP Origin: *VPP*

SFR Link: *FMT_SMO_EXT.1*

Access to the administrative interfaces is either provided via SSH to access the local console or SSH to access the remote management libvirtd interface. This implies that SSH provides the trusted channel following FTP_ITC_EXT.1.

Copyright ©2026 SUSE LLC

Operational network traffic, i.e. network traffic from / to the virtual machines, is separated by using the following mechanisms:

- For communication that remains within the physical perimeter of the system, Linux network bridges are used to establish network communication. The administrator can configure zero or more network bridges and assign zero or more virtual machines to these bridges. All virtual machines assigned to one bridge can communicate with each other. Virtual machines on dissimilar bridges cannot communicate with each other.
- For communication with entities that are physically external to the perimeter of the system, the administrator can assign zero or more physical network interfaces to the aforementioned bridges. This implies that virtual machines can communicate with external entities via the physical network interfaces. To separate the external traffic of different virtual machines that are assigned to different bridges, different physical network interfaces must be assigned to the bridges by administrators. Note that the network separation the TOE can enforce only applies up to the point the traffic leaves the physical interfaces. External entities may still be able to relay that data and thus may allow network communication between the virtual machines after all. This issue, however, is a standard networking issue the administrator must solve with mechanisms external to the TOE.

7.2.6 Protection of the TSF

7.2.6.1 FPT_DVD_EXT.1 Non-Existence of Disconnected Virtual Devices

PP Origin: *VPP*

SFR Link: *FPT_DVD_EXT.1*

Virtual machines can only access and interact with virtual / physical resources that are mapped to it by the TOE. The mapping is established by using MMIO registers that are mapped into the guest's address space as well as message signaled interrupts (MSI) which are routed to the guest. The TOE only maps the MMIO registers and MSI to guest VMs that belong to devices that are assigned to the guest VM by an administrator.

Completely virtualized devices are backed with a state within the TOE memory that is only created when the virtual device is instantiated and subsequently mapped to the intended virtual machine. Thus, if a virtual machine is not assigned a virtualized device, it does not exist in the first place.

7.2.6.2 FPT_EEM_EXT.1 Execution Environment Mitigations

PP Origin: *VPP*

SFR Link: *FPT_EEM_EXT.1*

The TOE always randomizes process address memory locations with 11 bits (stack) and 28 bits (text segment) of entropy.

The TOE also uses the hardware support of setting a no-execute bit on memory that is intended to be read/written. Conversely, memory that is intended to be executed is set read-only. This ensures that the hardware detects and denies improper use of memory.

The TOE protects all TOE binaries from stack-based and heap-based buffer overflow attacks using:

- ASLR to randomize the locations of the stack, preventing attackers from jumping to specific data that has been written to the stack.
- Stack canaries to detect if the stack has been overwritten when returning from a function.

Copyright ©2026 SUSE LLC

7.2.6.3 FPT_HAS_EXT.1 Hardware Assists

PP Origin: *VPP*

SFR Link: *FPT_HAS_EXT.1*

See the TSS description for FDP_HBI_EXT.1 outlining the used hardware virtualization mechanisms.

In addition, the TOE uses the following hardware mechanisms for memory address translation:

- Intel x86: EPT
- AMD x86: EPT
- IBM System Z: SIE instruction
- ARM 64: EL2 mode

7.2.6.4 FPT_HCL_EXT.1 Hypercall Controls

PP Origin: *VPP*

SFR Link: *FPT_HCL_EXT.1*

A proprietary documentation is provided outlining all available hypercalls including their operation (i.e. how the hypercalls are invoked, parameters, legal values, additional information).

7.2.6.5 FPT_RDM_EXT.1 Removable Devices and Media

PP Origin: *VPP*

SFR Link: *FPT_RDM_EXT.1*

The TOE is defined to only support virtual removable devices and media such as CDs considering the hardware platforms are all server systems that are not given physical access to regular users and thus guest VMs. The assignment of virtual removable devices and media is achieved using the administrative interfaces offered by the TOE to configure virtual machines.

The TOE supports the assignment of files that are represented as virtual removable media to virtual machines. The supported types of virtual removable media are: block devices with different protocols (VirtIO, SCSI), CD devices.

7.2.6.6 FPT_TUD_EXT.1 Trusted Update to the Virtualization System

PP Origin: *VPP*

SFR Link: *FPT_TUD_EXT.1*

The TOE allows the user to check for and install updates using the zypper application. Using zypper, updates can be manually queried, downloaded and installed as well as automatically be installed. When an update is initiated, the TOE downloads the update package and performs the RSA 4096-bit digital signature verification. If the verification is successful, the TOE installs the update. If the verification is unsuccessful, the TOE terminates the updates process.

The public key used for the digital signature is copied onto the system via the initial installation process from the installation media. The public key is maintained as part of the RPM database that is tracking the installed software packages. Updates to that public key are installed via a software package update which in is subject to signature verification before installation like all other packages.

7.2.6.7 FPT_VDP_EXT.1 Virtual Device Parameters

PP Origin: *VPP*

SFR Link: *FPT_VDP_EXT.1*

Copyright ©2026 SUSE LLC

The entire configuration definition including the specification of all virtual devices and their parameters that can be made available to guest VMs is given in the documentation provided at https://libvirt.org/format.html.

The settings and configurations for each given device are documented at https://libvirt.org/formatdomain.html. During load time of the configuration, the TOE verifies that the configuration is appropriate. If inconsistencies are detected in the configuration, the respective virtual machine cannot be instantiated.

7.2.6.8 FPT_VIV_EXT.1 VMM Isolation from VMs

PP Origin: *VPP*

SFR Link: *FPT_VIV_EXT.1*

Virtual machines are executed as unprivileged applications as part of the Linux host system. In addition to the regular process isolation, virtual machine processes are subject to a AppArmor-based confinement to prevent access to resources of the host system or resources of other virtual machines. Each virtual machine is instantiated with its private copy of the virtual machine monitor provided with the QEMU binary.

The virtual machine has no capability to invoke host platform specific interfaces whose use have system-wide effects. This implies that a virtual machine guest cannot invoke an SMI (x86) or access the BIOS interfaces. Access to the BIOS is denied by means of memory management: the MMIO and PIO registers to interact with the system BIOS are not mapped to the virtual machine.

Details about the virtual machine separation and isolation is provided in a proprietary documentation.

7.2.7 TOE access

7.2.7.1 FTA_TAB.1 Default TOE access banners

PP Origin: *VPP*

SFR Link: *FTA_TAB.1*

The TOE will display an advisory warning message regarding unauthorized use of the TOE prior to establishing an administrative user session.

7.2.8 Trusted path/channels

7.2.8.1 FTP_ITC_EXT.1 Trusted Channel Communications

PP Origin: *VPP*

SFR Link: *FTP_ITC_EXT.1*

The TOE uses SSHv2 as conforming to FCS_SSH_EXT.1 to provide a trusted channel between itself and authorized IT entities.

7.2.8.2 FTP_TRP.1 Trusted Path

PP Origin: *VPP*

SFR Link: *FTP_TRP.1*

The TOE provides a trusted path using the cryptographic network protocol of SSHv2 specified in this ST between itself and remote administrators that provides assured identification of its endpoints. This trusted path based on SSHv2 is used to allow remote administrators to securely access the TOE for administration.

Copyright ©2026 SUSE LLC

7.2.8.3 FTP_UIF_EXT.1 User Interface: I/O Focus

PP Origin: *VPP*

SFR Link: *FTP_UIF_EXT.1*

The TOE allows access to the guest VM console or graphical interface by means of a VNC channel which is tunneled through the SSHv2 connection used to access the host. That VNC channel is established by libvirtd when a user connects to the intended virtual machine. The VNC communication channel is created on-demand only when the virtual machine is accessed. Thus, the calling user is able to unambiguously identify the accessed guest VM console or graphical interface.

7.2.8.4 FTP_UIF_EXT.2 User Interface: Identification of VM

PP Origin: *VPP*

SFR Link: *FTP_UIF_EXT.2*

See TSS for FTP_UIF_EXT.1 for details how the console of a guest VM is identified.

In addition to the guest VM's console, the VM can be accessed using networking links. As the TOE only supports TCP/IP, a virtual machine is identified by its unique IP address(es) configured by administrators.

Copyright ©2026 SUSE LLC

8 Abbreviations, Terminology, and References

8.1 Abbreviations

ACE
Access Control Entry

AES
Advanced Encryption Standard

app
Application

API
Application Programming Interface

ASLR
Address Space Layout Randomization

BSD
Berkeley Software Distribution

BSM
Basic Security Module

CA
Certificate Authority

CBC
Cipher Block Chaining

CC
Common Criteria

CCM
Counter with CBC-MAC

CEM
Common Evaluation Methodology

CIFS
Common Internet File System

CMC
Certificate Management over CMS

CMS
Cryptographic Message Syntax

CSP
Critical Security Parameters

CTR
Counter Mode Block Chaining

CVE
Common Vulnerabilities and Exposures

DAR
Data At Rest

DEK
Data Encryption Key

Copyright ©2026 SUSE LLC

DEP
Data Execution Prevention

DNS
Domain Name System

DRBG
Deterministic Random Bit Generator

DSS
Digital Signature Standard

ECC
Elliptic Curve Cryptography

ECDH
Elliptic Curve Diffie-Hellman

ECDHE
ECDH Ephemeral

EKU
extendedKeyUsage

EST
Enrollment over Secure Transport

GCM
Galois/Counter Mode

GID
Group Identifier

GPOS
General Purpose Operating System

HCI
Host Controller Interface

HMAC
Keyed-hash Message Authentication Code

HTTPS
Hypertext Transfer Protocol Secure

ID
Identifier or Identity

IP
Internet Protocol

KAS
Key Agreement Scheme

KEK
Key Encryption Key

MAC
Message Authentication Code

OCSP
Online Certificate Status Protocol

OID
Object Identifier

Copyright ©2026 SUSE LLC

OS
Operating System

PBKDF
Password-Based Key Derivation Function

PII
Personally Identifiable Information

PIN
Personal Identification Number

PKI
Public Key Infrastructure

POSIX
Portable Operating System Interface

PP
Protection Profile

RA
Registration Authority

RBG
Random Bit Generator

ROM
Read Only Memory

RSA
Rivest-Shamir-Adleman

SAN
Subject Alternative Name

SAR
Security Assurance Requirement

SCEP
Simple Certificate Enrollment Protocol

SEP
Secure Enclave Processor

SFR
Security Functional Requirement

SHA
Secure Hash Algorithm

SMB
Server Message Block

SoC
System on a Chip

ST
Security Target

TLS
Transport Layer Security

TOE
Target of Evaluation

Copyright ©2026 SUSE LLC

TSF
TOE Security Functionality

TSFI
TSF Interface

TSS
TOE Summary Specification

UDID
Unique Device Identifier

UID
User Identifier

UUID
Universally Unique Identifier

XTS
XEX-based tweaked-codebook mode with ciphertext stealing

8.2 Terminology

This section contains definitions of technical terms that are used with a meaning specific to this document. Terms defined in the [CC] are not reiterated here, unless stated otherwise.

Administrator
An administrator is responsible for management activities, including setting policies that are applied by the enterprise on the operating system. This administrator could be acting remotely through a management server, from which the system receives configuration policies. An administrator can enforce settings on the system which cannot be overridden by non-administrator users.

API
A specification of routines, data structures, object classes, and variables that allows an application to make use of services provided by another software component, such as a library. APIs are often provided for a set of libraries included with the platform.

app
Software that runs on a platform and performs tasks on behalf of the user or owner of the platform, as well as its supporting documentation.

AppArmor
Linux kernel LSM module that is able to implement additional restrictions for executables. This LSM is unused in the evaluated configuration.

ASLR
An anti-exploitation feature which loads memory mappings into unpredictable locations. ASLR makes it more difficult for an attacker to redirect control to code that they have introduced into the address space of a process.

CC
Common Criteria for Information Technology Security Evaluation.

CEM
Common Evaluation Methodology for Information Technology Security Evaluation.

Credential
Data that establishes the identity of a user, e.g. a cryptographic key or password.

Copyright ©2026 SUSE LLC

CSP
: Information that is either user or system defined and is used to operate a cryptographic module in processing encryption functions including cryptographic keys and authentication data, such as passwords, the disclosure or modification of which can compromise the security of a cryptographic module or the security of the information protected by the module.

DAR Protection
: Countermeasures that prevent attackers, even those with physical access, from extracting data from non-volatile storage. Common techniques include data encryption and wiping.

DEP
: An anti-exploitation feature of modern operating systems executing on modern computer hardware, which enforces a non-execute permission on pages of memory. DEP prevents pages of memory from containing both data and instructions, which makes it more difficult for an attacker to introduce and execute code.

Developer
: An entity that writes OS software. For the purposes of this document, vendors and developers are the same.

General Purpose Operating System
: A class of OSes designed to support a wide-variety of workloads consisting of many concurrent applications or services. Typical characteristics for OSes in this class include support for third-party applications, support for multiple users, and security separation between users and their respective resources. General Purpose Operating Systems also lack the real-time constraint that defines Real Time Operating Systems (RTOS). RTOSes typically power routers, switches, and embedded devices.

Host-based Firewall
: A software-based firewall implementation running on the OS for filtering inbound and outbound network traffic to and from processes running on the OS.

OS
: Software that manages physical and logical resources and provides services for applications.

PII
: Any information about an individual maintained by an agency, including, but not limited to, education, financial transactions, medical history, and criminal or employment history and information which can be used to distinguish or trace an individual's identity, such as their name, social security number, date and place of birth, mother's maiden name, biometric records, etc., including any other personal information which is linked or linkable to an individual.

PP
: An implementation-independent set of security requirements for a category of products.

SAR
: A requirement to assure the security of the TOE.

Sensitive Data
: Sensitive data may include all user or enterprise data or may be specific application data such as PII, emails, messaging, documents, calendar items, and contacts. Sensitive data must minimally include credentials and keys. Sensitive data shall be identified in the OS's TSS by the ST author.

SFR
: A requirement for security enforcement by the TOE.

Copyright ©2026 SUSE LLC

ST
A set of implementation-dependent security requirements for a specific product.

TOE
The product under evaluation. In this case, the Operating System and its supporting documentation.

TSF
The security functionality of the product under evaluation.

TSS
A description of how a TOE satisfies the SFRs in a ST.

User
A user is subject to configuration policies applied to the operating system by administrators. On some systems under certain configurations, a normal user can temporarily elevate privileges to that of an administrator. At that time, such a user should be considered an administrator.

8.3 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

RFC4252 **The Secure Shell (SSH) Authentication Protocol**
Date January 2006
Location http://tools.ietf.org/html/rfc4252

RFC4253 **The Secure Shell (SSH) Transport Layer Protocol**
Date January 2006
Location http://tools.ietf.org/html/rfc4253

RFC5656 **Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer**
Date December 2009
Location http://tools.ietf.org/html/rfc5656

SSH **Functional Package for Secure Shell (SSH)**
Version 1.0
Date 2021-05-13
Location https://www.niap-ccevs.org/Profile/Info.cfm?PPID=459&id=459

SV **PP-Module for Server Virtualization Systems**
Version 1.1
Date 2021-06-14
Location https://www.niap-ccevs.org/Profile/Info.cfm?PPID=458&id=458

Copyright ©2026 SUSE LLC

SVCONFIG	**PP-Configuration for Virtualization and Server Virtualization Systems**	
	Version	1.0
	Date	2021-06-04
	Location	https://www.niap-ccevs.org/MMO/PP/CFG_Virtualization-SV_v1.0.pdf
VPP	**Protection Profile for Virtualization**	
	Version	1.1
	Date	2021-06-14
	Location	https://www.niap-ccevs.org/Profile/Info.cfm?PPID=456&id=456

Copyright ©2026 SUSE LLC