

# Common Criteria Evaluated Configuration Guide for SUSE LINUX Enterprise Server 15 SP7 (NIAP) - GPOS

February 23, 2026; v1.0



SUSE, SLES, and the SUSE logo are a trademark of SUSE LLC.

atsec and the atsec logo are a trademark of atsec information security GmbH.

IBM, IBM logo, BladeCenter, eServer, iSeries, OS/400, PowerPC, POWER3, POWER4, POWER4+, Power5, Power6, Power7, POWER8, POWER9, POWER10, pSeries, 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 and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Java and all Java-based products are trademarks of Sun Microsystems, Inc., in the United States, 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.

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 or distributed in any form without prior permission provided the copyright notice is retained on all copies. Modified versions of this document may be freely distributed provided that they are clearly identified as such, and this copyright is included intact.

Copyright (c) 2003, 2004 by atsec GmbH, and IBM Corporation or its wholly owned subsidiaries.

Copyright (c) 2020 - 2023 by atsec information security GmbH, and SUSE LLC or its wholly owned subsidiaries.

# Contents


**1 Introduction** **1**
- 1.1 Purpose of this document . . . 1
- 1.2 How to use this document . . . 1
- 1.3 What is a CC compliant system? . . . 2
  - 1.3.1 Hardware requirements . . . 2
  - 1.3.2 Software requirements . . . 3
- 1.4 Requirements for the system's environment . . . 3
  - 1.4.1 Requirements for connectivity . . . 3
  - 1.4.2 Requirements for administrators . . . 4
- 1.5 Requirements for the system's users . . . 4

**2 Installation** **7**
- 2.1 Supported hardware . . . 7
- 2.2 Automated installation process . . . 7
  - 2.2.1 Prerequisites for installation . . . 8
  - 2.2.2 Obtaining of installation images . . . 8
  - 2.2.3 Installation process . . . 9
- 2.3 Additional Configuration . . . 13
  - 2.3.1 x86 Configuration . . . 15
  - 2.3.2 ARM64 System Configuration . . . 15
  - 2.3.3 IBM Z System Configuration . . . 16

**3 System operation** **17**
- 3.1 System startup, shutdown and crash recovery . . . 17
- 3.2 Backup and restore . . . 17
- 3.3 Gaining administrative access . . . 18
  - 3.3.1 Using su . . . 18
  - 3.3.2 Using sudo . . . 18
- 3.4 Installation of additional software . . . 19
- 3.5 Scheduling processes using cron . . . 20
- 3.6 Mounting filesystems . . . 20
- 3.7 Encryption of partitions . . . 22
- 3.8 Secure erasure . . . 23
- 3.9 Configuring password policy . . . 23
- 3.10 Network configuration . . . 23
- 3.11 Managing user accounts . . . 23
  - 3.11.1 Creating users . . . 23
  - 3.11.2 Changing user passwords . . . 24
  - 3.11.3 SSH key-based authentication . . . 24
  - 3.11.4 Changing user properties . . . 25
  - 3.11.5 Locking and unlocking of user accounts . . . 25
  - 3.11.6 Removing users . . . 26

3.11.7 Defining administrative accounts . . . 26
3.12 Using serial terminals . . . 27
3.13 SSH Agent Forwarding . . . 27
3.14 Managing data objects . . . 27
3.14.1 Revoking access . . . 27
3.14.2 SYSV shared memory and IPC objects . . . 27
3.14.3 Posix Message Queues . . . 27
3.15 Configuring object access rights . . . 28
3.16 Setting the system time and date . . . 28
3.17 Configuring time synchronization with NTP . . . 28
3.18 Firewall configuration . . . 29
3.18.1 firewalld auditing of packet filter operations . . . 29
3.19 Screen saver configuration . . . 29
3.20 Update configuration . . . 30
3.20.1 Manual update . . . 30
3.20.2 Automatic update . . . 31
3.21 Cryptographic Support . . . 31
3.21.1 OpenSSL on x86 Architecture . . . 31
3.21.2 OpenSSL on IBM Z Architecture . . . 31
3.21.3 OpenSSL on ARM Architecture . . . 32
3.21.4 SSH Client Configuration . . . 33
3.21.5 SSH Server Configuration . . . 33
3.21.6 Cryptographic key handling . . . 34
3.21.7 Cryptographic key generation and establishment . . . 35
3.22 Certificate signing limitation . . . 35

**4 Monitoring, Logging & Audit 37**
4.1 Reviewing the system configuration . . . 37
4.2 System logging and accounting . . . 38
4.3 Configuring the audit subsystem . . . 39
4.3.1 Intended usage of the audit subsystem . . . 39
4.3.2 Selecting the events to be audited . . . 39
4.3.3 Reading and searching the audit records . . . 39
4.3.4 Starting and stopping the audit subsystem . . . 40
4.3.5 Storage of audit records . . . 40
4.3.6 Remote auditing . . . 41
4.3.7 Reliability of audit data . . . 42
4.4 System configuration variables in */etc/sysconfig* . . . 42

**5 Application Developers 43**

**6 Security guidelines for users 45**
6.1 Online Documentation . . . 45
6.2 Authentication . . . 46
6.3 Password policy . . . 46
6.4 SSH key-based authentication . . . 48
6.5 Access control for files and directories . . . 48
6.5.1 Discretionary Access Control . . . 49
6.6 Data import / export . . . 49
6.7 Screen saver . . . 49

**7 Appendix 51**
7.1 Online Documentation . . . 51

# Chapter 1

# Introduction

## 1.1 Purpose of this document

The SUSE LINUX Enterprise Server (SLES) distribution is designed to provide a secure and reliable operating system for a variety of purposes. Because security requirements obviously depend on the applications and environment, it is not possible to simply certify that the system is "secure", a more precise definition is needed.

The Common Criteria (CC) provides a widely recognized methodology for security certifications. A CC evaluation is fundamentally a two-step process, consisting of defining the "security target" which describes the features that are to be evaluated, and then testing and verifying that the system actually implements these features with a sufficient level of assurance.

This document is a security guide that explains how to set up the evaluated configuration, and provides information to administrators and ordinary users to ensure secure operation of the system. It is intended to be self-contained in addressing the most important issues at a high level, and refers to other existing documentation where more details are needed.

The document primarily addresses administrators, but the section "Security guidelines for users" is intended for ordinary users of the system as well as administrators.

Knowledge of the Common Criteria is not required for readers of this document.

## 1.2 How to use this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 http://www.ietf.org/rfc/rfc2119.txt.

Note that the terms "SHOULD" and "SHOULD NOT" defined in RFC 2119 are avoided in this document to the extent possible. Requirements are either absolute (and marked with MUST and equivalent terms), or entirely optional (in the sense of not affecting required security functions) and marked with RECOMMENDED, MAY or OPTIONAL.

If you follow the requirements in this document when setting up and using the system, your configuration will match the evaluated configuration. Certain configuration options are marked as OPTIONAL and you MAY modify them as needed, but you MUST NOT make other changes, because they will make the system fail to match the evaluated configuration.

Of course, you MUST always use common sense. This document is not a formal specification, and legitimate reasons may exist to modify the system setup in ways not described here if that is necessary for the system to fulfill its intended

purpose. Specifically, applying security patches released by the vendor is strongly RECOMMENDED even though that will cause a deviation from the evaluated configuration.

In cases where the requirements and recommendations in this document conflict with those in other sources (such as the online documentation), the information in this Configuration Guide has higher precedence. You MUST follow the steps described here to reach the evaluated configuration, even if other documentation describes different methods.

The usual convention is used in this guide when referring to manual pages that are included in the software distribution. For example, the notation *ls*(1) means that running the `man -S 1 ls` command will display the manual page for the *ls* command from section one of the installed documentation. In most cases, the `-S` flag and the section number may be omitted from the command, they are only needed if pages with the same name exist in different sections,

## 1.3 What is a CC compliant system?

A system can be considered to be "CC compliant" if it matches an evaluated and certified configuration. This implies various requirements concerning hardware and software, as well as requirements concerning the operating environment, users, and the ongoing operating procedures.

Strictly speaking, an evaluation according to the CC represents the results of investigation of the security properties of the target system according to defined guidelines. It should not be considered as a guarantee for fitness for any specific purpose, but should provide help in deciding the suitability of the system considering how well the intended use fits the described capabilities. It is intended to provide a level of assurance about the security functions that have been examined by a neutral third party.

The software MUST match the evaluated configuration. In the case of an operating system, this also requires that the installed kernel, system, and application software are the same. The documentation (including this guide) will specify permitted variations, such as modifying certain configuration files and settings, and installing software that does not have the capability to affect the security of the system (typically those that do not require root privileges). Please refer to section §3.4 "Installation of additional software" of this guide for more information.

Stated requirements concerning the operating environment MUST be met. Typical requirements include a secure location for the hardware (protected from physical access by unauthorized persons), as well as restrictions concerning permitted network connections.

The operation of the system MUST be in agreement with defined organizational security policies, to ensure that actions by administrators and users do not undermine the system's security.

### 1.3.1 Hardware requirements

The hardware MUST be of one of the following microarchitectures:

**x86 64bit Intel**

- Cascade Lake

**AMD x86_64**

- AMD EPYC 3rd Generation

**SLES on IBM System z based on z/Architecture processors**

- IBM z16

**System based on ARM processors**

- ARMv8.2-A

Running the certified software on other similar hardware may result in an equivalent security level, but the certification does not apply if the hardware is different from that used for the testing processes during the evaluation.

Note, the proper operation of all aspects of the software is only ensured when using the aforementioned hardware systems as several hardware mechanisms which may not be present in other systems are vital for the security of the system.

Please refer to section §2.1 "Supported hardware" for more information about additional hardware supported for use with the evaluated configuration.

### 1.3.2 Software requirements

The software MUST match the evaluated configuration. In the case of an operating system, this also requires that the installed kernel, system, and application software are the same. The documentation (including this guide) will specify permitted variations, such as modifying certain configuration files and settings, and installing software that does not have the capability to affect the security of the system (typically those that do not require 'root' privileges).

## 1.4 Requirements for the system's environment

The security target covers one or more systems running SLES, networked in a non-hostile network, with a well-managed and non-hostile user community. It is not intended to address the needs of a directly Internet-connected server, or the case where services are to be provided to potentially hostile users.

It is assumed that the value of the stored assets merits moderately intensive penetration or masquerading attacks. It is also assumed that physical controls in place would alert the system authorities to the physical presence of attackers within the controlled space.

You MUST set up the server (or servers) in a physically secure environment, where they are protected from theft and manipulation by unauthorized persons.

You MUST ensure that all connections to peripheral devices and all network connections are protected against tampering, tapping and other modifications. Using the secured protocols of TLSv1.2 or SSHv2 are considered sufficient protection for network connections. All other connections must remain completely within the physically secure server environment.

### 1.4.1 Requirements for connectivity

All components in the network such as routers, switches, and hubs that are used for communication are assumed to pass the user data reliably and without modification. Translations on protocols elements (such as NAT) are allowed as long as those modifications do not lead to a situation where information is routed to somebody other than the intended recipient system. Network and peripheral cabling must be approved for the transmittal of the most sensitive data held by the system.

Any other systems the TOE communicates with MUST be configured and managed under the same management control and operate under the same security policy constraints.

Be aware that information passed to another system leaves the control of the sending system, and the protection of this information against unauthorized access needs to be enforced by the receiving system. If an organization wants to implement a consistent security policy covering multiple systems on a network, organizational procedures MUST ensure that all those systems can be trusted and are configured with compatible security configurations enforcing an organization wide security policy. How to do this is beyond the scope of this Configuration Guide. If you set up a communication link to a system outside your control, please keep in mind that you will not be able to enforce any security policy for any information you pass to such a system over the communication link or in other ways (for example, by using removable storage media).

### 1.4.2 Requirements for administrators

There MUST be one or more competent individuals who are assigned to manage the system and the security of the information it contains. These individuals will have sole responsibility for the following functions: (a) create and maintain roles (b) establish and maintain relationships among roles (c) Assignment and Revocation of users to roles. In addition these individuals (as owners of the entire corporate data), along with object owners will have the ability to assign and revoke object access rights to roles.

The system administrative personnel MUST NOT be careless, willfully negligent, or hostile, and MUST follow and abide by the instructions provided by the administrator documentation.

Every person that has the ability to perform administrative actions by switching to root has full control over the system and could, either by accident or deliberately, undermine the security of the system and bring it into an insecure state. This Configuration Guide provides the basic guidance how to set up and operate the system securely, but is not intended to be the sole information required for a system administrator to learn how to operate Linux securely.

It is assumed, within this Configuration Guide, that administrators who use this guide have a good knowledge and understanding of operating security principles in general and of Linux administrative commands and configuration options in particular. We strongly advise that an organization that wants to operate the system in the evaluated configuration nevertheless have their administrators trained in operating system security principles and SLES security functions, properties, and configuration.

Every organization needs to trust their system administrators not to deliberately undermine the security of the system. Although the evaluated configuration includes audit functions that can be used to make users accountable for their actions, an administrator is able to stop the audit subsystem and reconfigure it such that his actions no longer get audited. Well trained and trustworthy administrators are a key element for the secure operation of the system. This Configuration Guide provides the additional information for system administrators when installing, configuring and operating the system in compliance with the requirements defined in the Security Target for the Common Criteria evaluation.

The above stated assumptions imply that the DAC permissions of system directories, system binary files and their configuration files are left unchanged. Among others, this ensures that only administrators can add new trusted software into the installation.

To ensure the integrity of the system, you MUST schedule periodical reviews of the system operation and system integrity. For example, an integrity verification using the `rpm` tool may be invoked. Another possibility of validating the integrity of the system is the use of `aide`.

## 1.5 Requirements for the system's users

The security target addresses the security needs of cooperating users in a benign environment, who will use the system responsibly to fulfill their tasks.

Note that system availability is *not* addressed in this evaluation, and a malicious user could disable a server through resource exhaustion or similar methods.

The requirements for users specifically include:

- User accounts MUST be assigned only to those users with a need to access the data protected by the system, and who MUST be sufficiently trustworthy not to abuse those privileges. For example, the system cannot prevent data from being intentionally redistributed to unauthorized third parties by an authorized user.
- Users are trusted to accomplish some task or group of tasks within a secure IT environment by exercising complete control over their data.
- All users of the system MUST be sufficiently skilled to understand the security implications of their actions, and MUST understand and follow the requirements listed in section §6 "Security guidelines for users" of this guide. Appropriate training MUST be available to ensure this.

It is part of your responsibility as a system administrator to verify that these requirements are met, and to be available to users if they need your help in maintaining the security of their data.

# Chapter 2

# Installation

The evaluation covers a fresh installation of SLES 15, on one of the supported hardware platforms as defined in section §1.3.1 "Hardware requirements" of this guide.

The evaluated configuration MUST be the only operating system installed on the server.

## 2.1 Supported hardware

You MAY attach the following peripherals without invalidating the evaluation results. Other hardware MUST NOT be installed in or attached to the system.

- Any storage devices and backup devices supported by the operating system (this includes hard disks, CD-ROM drives and tape drives).
- All Ethernet network adapters supported by the operating system. Modems, ISDN and other WAN adapters are not part of the evaluated environment.
- Any printers supported by the operating system.
- Operator console consisting of a keyboard, video monitor, and optionally mouse. Additionally, you MAY directly attach supported serial terminals (see section §3.12 "Using serial terminals" of this guide), but *not* modems, ISDN cards, or other remote access terminals.

Note, the IBM Crypto Express cards MUST NOT be installed. This restriction stems from resource constraints during the evaluation and is by no means an indication that either the hardware or the associated software drivers are improperly implemented.

## 2.2 Automated installation process

This section describes the detailed steps to be performed when installing the SLES operating system on the target server.

The installation process is fully automated, except for configuration options the administrator must provide, like supplying the network configuration or user names and passwords for administrative users.

All settings listed here are REQUIRED unless specifically declared otherwise.

### 2.2.1 Prerequisites for installation

It is RECOMMENDED that you disconnect all network connections until the post-install system configuration is finished. You MAY use a network if required for the installation (for example when using a NFS file server instead of CD-ROMs). If you do use a network, you MUST ensure that this network is secure, for example by directly connecting the new system to a standalone NFS server with no other network connections.

You will need the following components to install a system in the evaluated configuration as explained in the following sections:

- The target system that will be installed, refer to section §1.3.1 "Hardware requirements" of this guide for the list of supported hardware. The target system REQUIRES at least one local hard drive that will be erased and repartitioned for use by the evaluated configuration.
- The availability of the **SUSE LINUX Enterprise Server 15 SP7** applicable for the chosen hardware system. Note that you MUST use the 64bit version marked as *x86_64* for the Intel based hardware. Please note that no re-spin image must be used instead the standard SLES image MUST be used.
- A static IP address if you are intending to attach the target system to a network; DHCP should not be used. In addition, you will need to configure the netmask, gateway, and DNS server list manually. When a static IP address shall be used during installation, the boot option of

  ```
  netsetup=1
  ```

  must be used as otherwise the installer will default to DHCP.
- A method to make the contents of the ISO images containing SLES available to the target system, including ensuring the possibility to boot the boot image provided with the re-spin of the SLES ISO images. The methods include the storing of the ISO images on a USB device and booting from it, using TFTP and NFS, using HTTP-based distribution of images, etc. The possible installation methods are explained in the SLES 15 SP7 deployment guide provided at https://documentation.suse.com/en-us/sles/15-SP7/.

  After obtaining the ISO images, you MUST perform the integrity verification of the downloaded files with the keys provided on the SUSE security web sites https://www.suse.com/support/security/keys/.

### 2.2.2 Obtaining of installation images

You MUST download the standard ISO images from the SUSE web site on a separate Internet-connected computer, and either burn it on a USB device, or make the contents available as outlined in the SLES 15 SP7 deployment guide provided at https://documentation.suse.com/en-us/sles/15-SP7/. The download to the following platform-specific images is https://www.suse.com/download/sles/.

Please note that if the image you want to download does not appear on the page, please log in to your account first.

**SUSE Linux Enterprise Server 15 SP7 for IBM Z System**

- Filename: SLE-15-SP7-Full-s390x-GM-Media1.iso
- SHA256 Checksum: 052b4627a811afd39d9ae27da5727364760b9ea6159fc0848d0600b4419d1f23

**SUSE Linux Enterprise Server 15 SP7 for Intel 64/AMD64**

- Filename: SLE-15-SP7-Full-x86_64-GM-Media1.iso
- SHA256 Checksum: cd9430420726219f9c73aa7a389b43057e1d9c610d279b7f71d3687c06d8e66c

**SUSE Linux Enterprise Server 15 SP7 for ARM64**

- Filename: SLE-15-SP7-Full-aarch64-GM-Media1.iso
- SHA256 Checksum: 9a136f0bff5473e82ec1ebe34e8fbac1bdc0cdfd128b3219b4e2ab2df6654d44

You MUST use **SUSE Linux Enterprise Server 15 SP7**. Make sure that you are using the appropriate version for your platform, refer to section §1.3.1 "Hardware requirements" of this guide for the list of supported hardware and the corresponding version needed.

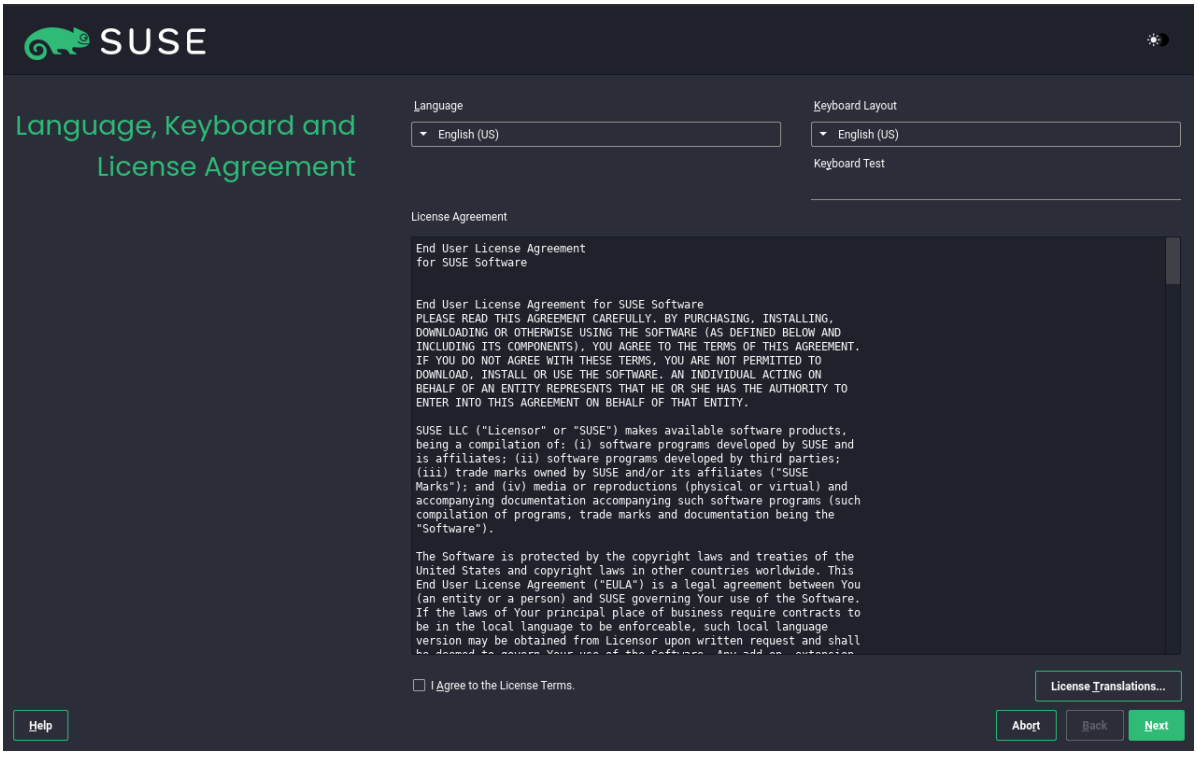
You MUST verify that the SHA256 checksums of the SLES 15 SP7 ISO image files are correct. It is RECOMMENDED to check the checksums including the signature shown on the SUSE web site by performing the steps outlined on the Security web page at http://www.suse.com/security/download-verification.html.

### 2.2.3 Installation process

The preparation of the initial boot and the way how the boot prompt is accessed depends on the chosen hardware and configuration. Please see the architecture specific SLES guidance documentation to set up the boot environment.

Please note that any of the following illustrations are considered to support the explanation. The displayed information may be slightly different.

The following illustration shows an example of how to boot directly from SLES USB flash drive. If a different boot mechanism is used, this illustration may not be applicable.



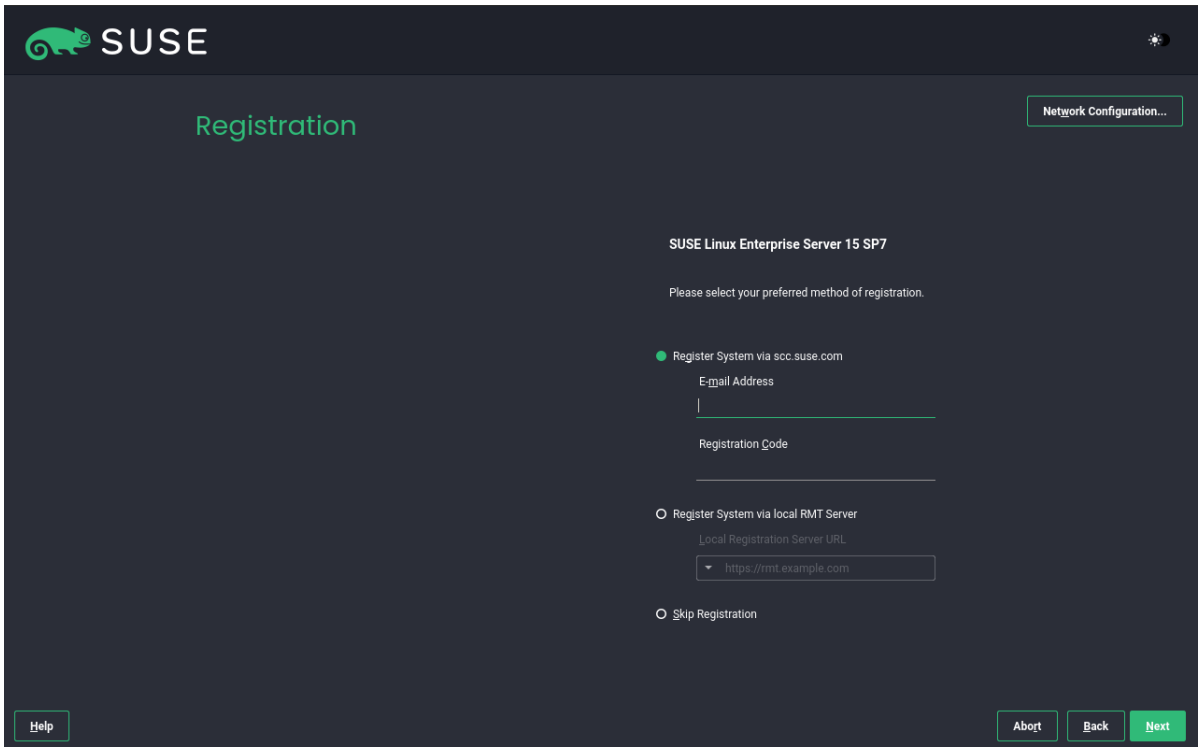

When the installer asks for the installation language, you MUST select "English" as this is the only language supported for the evaluated configuration. Further you can adjust the keyboard layout to suit your layout. You MUST read the licence agreement carefully and check the checkbox if you agree to it.

After you have agreed to the licence agreement, it is RECOMMENDED that you register your product either via scc.suse.com or via a local RMT server.

Please note, that if you skip the registration at this point you SHOULD register your product after the post-install system configuration is finished. This may be done using the following command

```
SUSEConnect -r <registration code> -e <email address>
```

During the registration a prompt is shown offering updated repositories. You MUST NOT enable those repositories during installation to receive the latest updated.

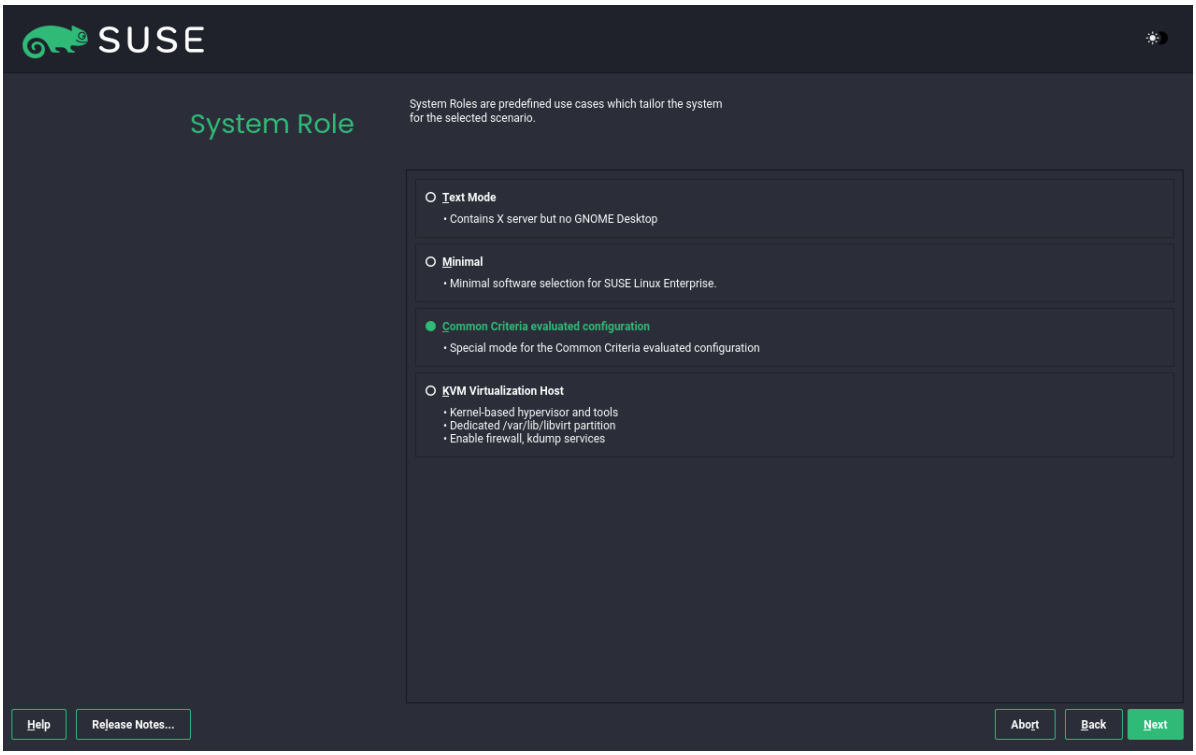



In the extension and module selection, the two modules

- Basesystem Module
- Server Application Module

MUST be selected. Additional packages MAY be selected, however, the same restrictions as stated in §3.4 "Installation of additional software" apply.

In the next step you MUST select the "Common Criteria evaluated configuration" to install the SUSE Linux Enterprise Server in the evaluated configuration.




This configuration contains some restrictions to the available configuration options. And asks for the a password for device encryption during installation. The password MUST be compliant to the requirements as stated in §6.3.

The next step is the partitioning. You MAY modify the following settings at this point:

- The partition layout may be changed to suit the organizational needs. In particular, you MAY change the partitions -- however you MUST NOT change the partition setup for */boot* if the full disk encryption configuration is booted. In addition you MAY change the used filesystems to either EXT4, BTRFS, or XFS. You MUST NOT use any other filesystem. VFAT MAY be automatically selected for */boot* or */boot/efi* where you MUST NOT change that selection. If you change settings, you MUST ensure that the partitions are formatted -- the default ensures the formatting of the partitions.

  You MAY select next to full disk encryption also encryption only for specific partitions. See §3.7 for additional information on the encryption of partitions.

  The suggested partitioning schema does not configure partitions for */tmp* or */var/tmp*. The automated configuration mounts *tmpfs* partitions to these directories to avoid having temporary files stored on disk. You MAY create a partition for either directory. The automated installation will skip setting up the *tmpfs* partition if the directories are already mount points. It is RECOMMENDED that you specify the mount options of *nosuid* and *nodev* for partitions mounted at either */tmp* or */var/tmp*.

  You MAY alter the partitioning setup at runtime, provided that only the allowed file system types listed in the above paragraph are used.

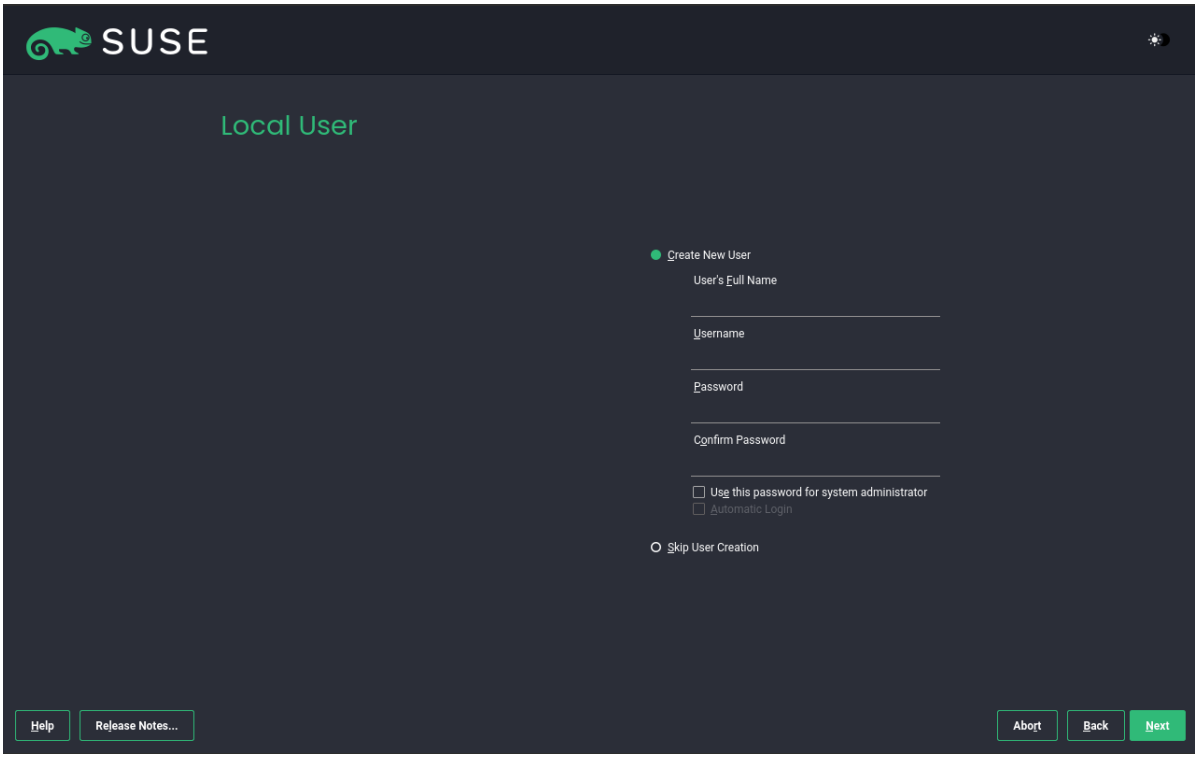
- If you plan to utilize the system to host virtual machines and the disk devices serving disk space to the virtual machines shall be provided with regular files, it is RECOMMENDED that you consider the location of these files at this point. These files are potentially very large. The default location chosen by `libvirtd` is */var/lib/libvirt/images/*. You MAY modify the partitioning layout to grant space for the files used as disk device backends for virtual machines.

- You MAY modify the boot loader options such as setting a password. However, you MUST NOT modify the boot loader type, the boot loader location and the selected boot loader option "Linux - CC evaluated configuration" which is the default selection.

You are now asked for the time zone applicable for the system. You MAY select the appropriate time zone. In addition, you MAY configure an NTP server at this point by selecting "Change" in the box "Date and Time".




Afterwards a new local user can be created which is shown in the following illustration.



After all options are configured as desired, the installation process is invoked by clicking "Install". The partitions are generated and formatted and the packages are installed.

After the installation process completes, the system reboots into the post-installation phase.

The system is automatically configured. Intervention by the administrator is not needed.

You MUST ensure that the configuration result shows a successful configuration. If at least one step in the configuration fails you MUST NOT proceed at the current configuration state as the system is in an unknown state at this time.

After finishing the installation, the user created during the installation process should already be part of the *trusted* group to allow access to the root identity and privilege. If the user is not part of the *trusted* group the following command can be used to add this user to the *trusted* group where <USER> refers to the user name of the user added previously:

```
usermod -G trusted -a <USER>
```

**IBM Z System installation**

The installation on IBM Z System may involve some considerations on how to access the installation system. Usually, the IBM Z System machines are headless systems where the z/VM console is accessed remotely. To install the previously downloaded image a VPN connection to the HMC z16 environment is required. Further for FTP server, where the s390x ISO image is mounted, a username and password needs to be set.

Note: In the evaluated configuration SUSE Linux Enterprise 15 SP7 MUST be installed in LPAR mode. An overview of the installation steps to follow is shown below, additional information is available at https://documentation.suse.com/sles/15-SP7/html/SLES-all/cha-zseries.html.

The first step of the SUSE Linux Enterprise 15 SP7 installation is to load from a remote media. To do so enter the FTP credentials including host name and filepath. In the next step select the [...]/suse.ins (not [...]/susehmc.ins) and proceed. After providing the password for the user the image will be loaded and should succeed without any problems.

The next step is to access the HMC operating system output and to start the installation, choosing as source the network and as protocol FTP. Note: Do not use the OSI Layer 2 support.

Afterwards, the network interface must be configured to connect to the FTP server. Therefore The IP address, gateway and DNS server must be set, the search domain is empty. Enter the IP address of the FTP server and the media directory as well as the password and username to access the FTP server.

Continue by selecting no proxy. The installation system is now loaded. Select SSH installation and define a password as only SSH is supported. When selecting the option to access the system via SSH, the SSH keys are generated and the SSH server is started. This method is the suggested access method as the graphical YaST is started in case the administrator client machine hosts an X11 windowing system and the administrator enables X11 forwarding with the SSH client. Now connect over the VPN via SSH to the installation system.

After booting and selecting the access method for installation, the boot image waits for the administrator to log in and start "yast.ssh" on the command line. YaST will automatically invoke the auto-installation process.

After selecting SUSE Linux Enterprise Server 15 SP7 and accepting to the License Agreement select "Configure ZFCP Disks" and follow the steps described in §2.2.3. Afterwards you can connect to the system via SSH.

Please note that for later configuration the tool YaST MUST NOT be used.

## 2.3 Additional Configuration

After finishing the installation process as described in the previous chapter. You MUST follow the configurations based on the platform stated in this chapter and you MAY install the following packages using the zypper command if you want to enable remote auditing, automatic and manual updates with the SUSE Customer Center (SCC) and firewall configurations:

```
zypper install audit-audispd-plugins-3.0.6-150400.4.16.1
yast2-online-update-4.7.0-150700.1.1
yast2-online-update-configuration-4.7.0-150700.1.1
firewalld-2.0.1-150600.3.5.1
```

In addition, the following update packages, which are available via the update mechanism after the system is registered with the update channels documented in §3.20 "Update configuration", MUST be installed:

```
zypper in openssh-9.6p1-150600.6.34.1 openssh-server-9.6p1-150600.6.34.1
openssh-clients-9.6p1-150600.6.34.1 openssh-helpers-9.6p1-150600.6.34.1
openssh-common-9.6p1-150600.6.34.1
```

In the evaluated configuration FIPS mode MUST be enabled, to verify if FIPS mode is activated execute the following command:

```
fips-mode-setup --check
```

If the result states "FIPS mode is disabled", the following configuration MUST be executed to enable it:

1. Install crypto-policies module

   ```
   zypper in crypto-policies
   ```

2. Enable fips mode using the command

   ```
   fips-mode-setup --enable
   ```

After the reboot verify that the system is in FIPS mode using the above stated command

```
fips-mode-setup --check
```

Linux Kernel

You MUST install the kernel-default package as follows:

```
zypper in kernel-default-6.4.0-150700.53.22.1
```

To apply the kernel changes on AMD and ARM machines the following commands MUST be executed as well:

```
update-bootloader --refresh ;
grub2-mkconfig -o /boot/grub2/grub.cfg ;
test -e /boot/efi/EFI/sles/grub.cfg  && \
grub2-mkconfig -o /boot/efi/EFI/sles/grub.cfg ;
test -e /boot/efi/EFI/sle_rt/grub.cfg  && \
grub2-mkconfig -o /boot/efi/EFI/sle_rt/grub.cfg;
dracut -f --regenerate-all
```

### 2.3.1 x86 Configuration

On x86 CPUs, the AES-NI instruction set is available to speed up the AES operation. If software-only support shall be used instead, the following configuration MUST be applied.

Note, in the evaluated configuration, AES-NI MUST be deactivated. Thus, to comply with the evaluation result, you MUST apply the following configurations.

After the first boot, you MUST disable the AES-NI and Bluetooth Linux kernel modules to prevent them from being used. To do so, create the following file with the following content:

**/lib/modprobe.d/aesni_intel.conf**

```
install aesni_intel /bin/true
```

**/lib/modprobe.d/sha_ni.conf**

```
install sha1_ni /bin/true
install sha256_ni /bin/true
```

**/lib/modprobe.d/bluetooth.conf**

```
install bluetooth /bin/true
```

After updating the configuration, you MUST reboot the system.

### 2.3.2 ARM64 System Configuration

On ARM CPUs, the CE instruction set is available to speed up the AES and SHA operations. If software-only support shall be used instead, the following configuration MUST be applied.

Note, in the evaluated configuration, CE MUST be deactivated. Thus, to comply with the evaluation result, you MUST apply the following configurations.

After the first boot, you MUST disable the CE Linux kernel modules to prevent it from being used. To do so, create the following file with the following content:

**/lib/modprobe.d/aes_ce.conf**

```
install aes_ce_cipher /bin/true
install aes_ce_blk /bin/true
install sha1_ce /bin/true
install sha256_arm64 /bin/true
install sha512_arm64 /bin/true
install ghash_ce /bin/true
install crct10dif_ce /bin/true
```

After updating the configuration, you MUST reboot the system.

### 2.3.3 IBM Z System Configuration

For IBM Z Systems, the processor provides instructions implementing AES and SHA2 support called VMX. This support SHALL NOT be used and instead the software-only algorithms. Therefore the following configurations MUST be applied.

After the first boot, you MUST disable several Linux kernel modules to prevent them from being used. To do so, create the following files with the following contents:

**/lib/modprobe.d/aes_s390.conf**

```
install aes_s390 /bin/true
```

**/lib/modprobe.d/des_s390.conf**

```
install des_s390 /bin/true
```

**/lib/modprobe.d/ghash_s390.conf**

```
install ghash_s390 /bin/true
```

**/lib/modprobe.d/sha1_s390.conf**

```
install sha1_s390 /bin/true
```

**/lib/modprobe.d/sha256_s390.conf**

```
install sha256_s390 /bin/true
```

**/lib/modprobe.d/sha512_s390.conf**

```
install sha512_s390 /bin/true
```

After updating the configuration, you MUST reboot the system.

# Chapter 3

# System operation

To ensure that the systems remains in a secure state, special care MUST be taken during system operation.

## 3.1 System startup, shutdown and crash recovery

Use the *shutdown*(8), *halt*(8) or *reboot*(8) programs as needed to shut down or reboot the system.

When powered on (or on initial program load of the logical partition on a host system), the system will boot into the SLES operating system. If necessary (for example after a crash), a filesystem check will be performed automatically. In rare cases manual intervention is necessary, please refer to the *fsck*(8) and *debugfs*(8) documentation for details in this case.

In case a nonstandard boot process is needed (such as booting from floppy disk or CD-ROM to replace a defective hard drive), interaction with the boot loader and/or the host's management system can be used to modify the boot procedure for recovery. For example, the kernel command line argument for booting into the emergency mode can be helpful:

```
systemd.unit=emergency.target
```

Also, the rescue target provides a helpful environment:

```
systemd.unit=rescue.target
```

Please refer to the relevant documentation of the boot loader, as well as the SLES administrator guide, for more information.

## 3.2 Backup and restore

Whenever you make changes to security-critical files, you MAY need to be able to track the changes made and revert to previous versions, but this is not required for compliance with the evaluated configuration.

The *tar*(1) archiver is RECOMMENDED for backups of complete directory contents, please refer to section §6.6 "Data import / export" of this guide. Regular backups of the following files and directories (on removable media such as USB flash drive, or on a separate host) are RECOMMENDED:

```
/etc/
/var/spool/cron/
```

You MUST use the `--acl` option for *tar* if you intend to save or restore ACLs.

Depending on your site's audit requirements, also include the contents of */var/log/* in the backup plan. In that case, the automatic daily log file rotation needs to be disabled or synchronized with the backup mechanism, refer to sections §4.2 "System logging and accounting" and §4.3 "Configuring the audit subsystem" of this guide for more information.

You MUST protect the backup media from unauthorized access, because the copied data does not have the access control mechanisms of the original file system. Among other critical data, it contains the secret keys used by the *SSH* and *charon* servers, as well as the */etc/shadow* password database. Store the backup media at least as securely as the server itself.

## 3.3 Gaining administrative access

System administration tasks require superuser privileges. Directly logging on over the network as user 'root' is disabled. To gain superuser rights, you MUST first authenticate using an unprivileged user ID, and then use either the `su` or `sudo` command to switch identities. Note that you MUST NOT use the 'root' rights for anything other than those administrative tasks that require these privileges, all other tasks MUST be done using your normal (non-root) user ID.

User IDs that belong to administrative users are assigned to the *trusted* group. Note that SLES uses a group named *trusted* to provide administrator access to users unlike other Linux distributions which may use a group named *wheel*. That group MUST NOT be used for any other user ID. The `su` command can only be invoked by users belonging to the *trusted* group to prevent password attacks against the root user account.

### 3.3.1 Using su

The `su` command allows a permanent switch of the user ID for the current session.

You MUST use exactly the following *su*(1) command line to gain superuser access:

```
/usr/bin/su -
```

This ensures that the correct binary is executed irrespective of PATH settings or shell aliases, and that the root shell starts with a clean environment not contaminated with the starting user's settings. This is necessary because the *.profile* shell configuration and other similar files are writable for the unprivileged ID, which would allow an attacker to easily elevate privileges to root if able to subvert these settings.

Administrators MUST NOT add any directory to the root user's PATH that are writable for anyone other than 'root', and similarly MUST NOT use or execute any scripts, binaries or configuration files that are writable for anyone other than 'root', or where any containing directory is writable for a user other than 'root'.

### 3.3.2 Using sudo

The `sudo` command allows invoking of a command with a configured user ID, including the root user ID. The switch to the target user ID only remains for the duration of the execution time of the specified command.

The default configuration of `sudo` does not allow any unprivileged users to invoke privileged commands. Depending on your requirements, the following examples may be used as a guide to configure `sudo`. More information may be obtained from the *sudoers*(5) man page.

The following configuration allows all users associated with the *trusted* group to use all commands with privileges:

```
%trusted        ALL=(ALL)       ALL
```

The use of commands with the root identity, other system identities or system groups MUST be restricted to users of the *trusted* group.

## 3.4 Installation of additional software

Additional software packages MAY be installed as needed, provided that they do not conflict with the security requirements.

Any additional software added is not intended to be used with superuser privileges. The administrator MUST use only those programs that are part of the original evaluated configuration for administration tasks, except if the administrator has independently ensured that use of the additional software is not a security risk.

Administrators MAY add scripts to automate tasks as long as those only depend on and run programs that are part of the evaluated configuration.

The security requirements for additional software are:

- Kernel modules other than those provided as part of the evaluated configuration MUST NOT be installed or loaded. You MUST NOT load the *tux* kernel module (the in-kernel web server is not supported). You MUST NOT add support for non-ELF binary formats or foreign binary format emulation that circumvents system call auditing. You MUST NOT activate *knfsd* or export NFS file systems.

- Device special nodes MUST NOT be added to the system.

- SUID root, SGID root programs or programs with file system capabilities MUST NOT be added to the system. Programs which use the SUID or SGID bits to run with identities other than 'root' MAY be added if the numerical SUID and SGID values are not less than 500 as defined with the values *UID_MIN* and *GID_MIN* in the configuration file of */etc/login.defs*. This restriction is necessary to avoid conflict with system user and group IDs such as the "disk" group.

- The content, permissions, and ownership of all existing filesystem objects (including directories and device nodes) that are part of the evaluated configuration MUST NOT be modified. Files and directories MAY be added to existing directories provided that this does not violate any other requirement.

- Programs automatically launched with 'root' privileges MUST NOT be added to the system. Exception: processes that *immediately* and *permanently* switch to a non privileged identity on launch are permitted, for example by using `su USERID -c LAUNCH_COMMAND` in the startup file, or alternatively by using the *setgroups*(2), *setgid*(2) and *setuid*(2) system calls in a binary. (*seteuid*(2) etc. are insufficient -- if the administrator cannot identify when and how privileged are dropped, the application MUST NOT be installed.)

  Automatic launch mechanisms are:

  - Targets and units as part of the *systemd* mechanism.
  - Scheduled jobs using `cron` (including entries in */etc/cron** files)
  - Applications started using the system DBUS which is configured via */etc/dbus-1/system.d/*.
  - Applications specified in */etc/sudoers* or with rules located in a file in the directory */etc/sudoers.d*. Note, that file may contain the keyword *ALL* as a placeholder for a command. In this case, the user allowed to execute all commands with that rule using the root user ID MUST ensure that additional applications are not executed using `sudo`. This requirement can only be met with operational procedures.
  - Applications spawned via `udev` where the rules are added to */lib/udev/rules.d*.

Examples of programs that usually do not conflict with these requirements and MAY be installed are compilers, interpreters, network services running with non-root rights, and similar programs. The requirements listed above MUST be verified in each specific case.

## 3.5 Scheduling processes using cron

The *cron*(8) program schedules programs for execution at regular intervals. Entries can be modified using the *crontab*(1) program - the file format is documented in the *crontab*(5) manual page.

You MUST follow the rules specified for installation of additional programs for all entries that will be executed by the 'root' user. Use non-root crontab entries in all cases where 'root' privileges are not absolutely necessary.

Errors in the non interactive jobs executed by `cron` are reported in the system log files in */var/log/*, and additionally via e-mail to the user who scheduled it.

Permission for users to schedule jobs with `cron` is controlled through the following *allow* and *deny* files:

```
/etc/cron.allow
/etc/cron.deny
```

The *allow* file has precedence if it exists, then only those users whose usernames are listed in it are permitted to use the service. If it does not exist, the *deny* file is used instead and all users who are *not* listed in that file can use the service. Note that the contents of these files are only relevant when the scheduling commands are executed, and changes have no effect on already scheduled commands.

In the SLES distribution, the *allow* files do not exist, and *deny* files are used to prevent system-internal IDs and/or guest users from using these services. By default, the evaluated configuration permits all non-system users to use *cron* and *at*.

It is RECOMMENDED to restrict the use of *cron* to human users and disallow system accounts from using these mechanisms. For example, the following commands add all system accounts other than root to the *deny* files:

```
awk -F: '{if ($3>0 && $3<1000) print $1}' /etc/passwd >/etc/cron.deny
chmod 600 /etc/cron.deny
```

Administrators MAY schedule jobs that will be run with the privileges of a specified user by editing the file */etc/crontab* with an appropriate username in the sixth field. Entries in */etc/crontab* are not restricted by the contents of the *allow* and *deny* files.

You MAY create */etc/cron.allow* to explicitly list users who are permitted to use these services. If you do create these files, they MUST be owned by the user 'root' and have file permissions 0600 (no access for group or others).

Note, the login ID is not retained for the following special case:

1. User A logs into the system.
2. User A uses su to change to user B.
3. User B now edits the cron or at job queue to add new jobs. This operation is appropriately audited with the proper login ID.
4. Now when the new jobs are executed as user B, the system does not provide the audit information that the jobs are created by user A.

## 3.6 Mounting filesystems

If any filesystems need to be mounted in addition to those set up at installation time, appropriate mount options MUST be used to ensure that mounting the filesystem does not introduce capabilities that could violate the security policy.

The special-purpose *proc*, *sysfs*, *devpts*, *securityfs*, *cgroups*, *binfmt_misc*, *devtmpfs*, *mqueue*, and *tmpfs* filesystems are part of the evaluated configuration. These are virtual filesystems with no underlying physical storage, and represent data structures in kernel memory. Access to contents in these special filesystems is protected by the normal discretionary access control policy and additional permission checks.

Note that changing ownership or permissions of virtual files and directories is generally NOT supported for the *proc* and *sysfs* filesystems (corresponding to directories */proc/* and */sys/*), and attempts to do so will be ignored or result in error messages.

A new file system can be integrated as part of the evaluated configuration, for example by installing an additional hard disk, under the following conditions:

- The device is protected against theft or manipulation in the same way as the server itself, for example by being installed inside the server.
- One or more new, empty, file systems with the file system formats listed in section §2.2.3 "Installation process" are created on it.
- The file systems of EXT4, BtrFS, or XFS are mounted using the `acl` option, for example with the following setting in the */etc/fstab* file:

  ```
  /dev/sdc1 /home2 ext3 acl 1 2
  ```

  Existing files and directories MAY then be moved onto the new file systems.
- If a device containing a file system is ever removed from the system, the device MUST be stored within the secure server facility, or alternatively MUST be destroyed in a way that the data on it is reliably erased.

Alternatively, media MAY be accessed without integrating them into the evaluated configuration, for example USB flash drives.

USB devices MUST be accessed using the `iso9660` filesystem type.

The following mount options MUST be used if the filesystems contain data that is not part of the evaluated configuration:

```
nodev,nosuid
```

Adding the *noexec* mount option to avoid accidental execution of files or scripts on additional mounted filesystems is RECOMMENDED.

Note that these settings do not completely protect against malicious code and data, you MUST also verify that the data originates from a trustworthy source and does not compromise the server's security. Specifically, be aware of the following issues:

- Even unprivileged programs and scripts can contain malicious code that uses the calling user's rights in unintended ways, such as corrupting the user's data, introducing trojan horses in the system, attacking other machines on the network, revealing confidential documents, or sending unsolicited commercial e-mail ("spam").
- Data on the additional filesystem MUST have appropriate access rights to prevent disclosure to or modification by unauthorized users. Be aware that imported data may have been created using user names and permissions that do not match your system's security policies.
- You MUST NOT write data on removable file systems such as floppy disks, since it cannot be adequately protected by the system's access control mechanisms after being removed from the system. Please refer to section §3.2 "Backup and restore" of this guide for more information regarding non-filesystem-based backup.

Each new file system MUST be mounted on an empty directory that is not used for any other purpose. It is RECOMMENDED using subdirectories of */mnt* for temporary disk and removeable storage media mounts.

For example:

```
# mount /dev/cdrom /media/cdrom -t iso9660 -o ro,nodev,nosuid,noexec
```

You MAY also add an equivalent configuration to */etc/fstab*, for example:

```
/dev/cdrom /media/cdrom iso9660 ro,noauto,nodev,nosuid,noexec 0 0
```

You MUST NOT include the *user* flag, ordinary users are not permitted to mount filesystems. This is also enforced by the deletion of the SUID bit on the *mount* command.

## 3.7 Encryption of partitions

SLES provides the dm-crypt mechanism for setting up partitions where all data stored on those partitions are encrypted on the fly. When data is read from those partitions, the data is decrypted without any intervention by any user.

As the block device of the partition is subject to the cryptographic operation, there is no restriction which filesystem is used together with the encrypted block device. If you selected a full disk encryption or configured encryption for different partitions during the initial installation time as outlined in section §2.2.3 "Installation process", you already store data on dm-crypt protected hard disks.

You MAY configure yet unused or newly added hard disks or partitions using dm-crypt before creating a filesystem on them. The setup of a dm-crypt protected partition is performed using the `cryptsetup` application. Please refer to the *cryptsetup*(8) man page for instructions on using dm-crypt.

Note, if the `cryptsetup` application is not available on the system it must be installed using the zypper command:

```
zypper in cryptsetup-2.7.0-150600.3.3.1
```

When using cryptsetup manually, you MUST use the LUKS extension and therefore the LUKS commands specified in *cryptsetup*(8). The cryptographic key encrypting all data is protected with a passphrase provided during creation time. Note: The selected passphrase must be strong enough to protect the encryption key.

The setup of a dm-crypt protected partition is performed with the *luksFormat* command to the `cryptsetup` application.

When you do not want to use the default cipher with *luksFormat* (see `cryptsetup --help` for the default), you MUST ensure that the following requirements are met when specifying the cipher:

- AES in GCM mode with 256 bits
- AES in CBC mode with 256 bits
- AES in XTS mode with 512 bits

After formatting, the *luksOpen* command has to be used to set up the encryption mechanism, i.e. to inform the kernel that any read and write operation is encrypted and decrypted on the fly. The device file created with the *luksOpen* command can now be used to create a file system which then can also be mounted.

For a regular operation, the *luksOpen* command has be used followed by a `mount` command with the device file created by *luksOpen*.

## 3.8 Secure erasure

To erase key material in files, such as TLS certificates / keys and SSH key pairs, the file containing the key material MUST be deleted and therefore the command *shred* MUST be used. However, please note that *shred* is not effective on all file system see for further information the man page of *shred*(1). Furthermore, for SSDs, the /usr/sbin/fstrim command MUST be invoked once the key file with the key material has been deleted to inform the underlying SSD to discard deleted blocks. One example is to use the

```
fstrim -a
```

command to trim all block devices including the block device that used to hold the sensitive data. For further information regarding *fstrim* please see the documentation in the *fstrim*(8) man page.

In addition, the disk MUST be overwritten multiple times with random numbers before removed. It is further RECOMMENDED to use encrypted partitions (see §3.7 "Encryption of partitions") as well as to physically destroy the disk upon disposal.

## 3.9 Configuring password policy

To configure the minimum password length, the minimum number of special and numeric characters as well as the minimum number of uppercase and lowercase characters in passwords the following line can be added or updated in */etc/pam.d/common-password* by using *pam_cracklib*(8):

```
password requisite pam_cracklib.so minlen=X dcredit=X ocredit=X
ucredit=X lcredit=X
```

- *minlen* is the minimum acceptable size for the new password
- *dcredit* with N $<$ 0 is the minimum number of digits that must be met for a new password
- *ocredit* with N $<$ 0 is the minimum number of other characters that must be met for a new password
- *ucredit* with N $<$ 0 is the minimum number of upper case letters that must be met for a new password
- *lcredit* with N $<$ 0 is the minimum number of lower case letters that must be met for a new password

## 3.10 Network configuration

To configure the network interfaces modify the config files or use the tool *wicked*. For detailed information about *wicked* and manual configuration on how to enable or disable network interfaces and configure WiFi interfaces please see section 23.5 Configuring a Network Connection Manually of the SUSE Linux Enterprise Server 15 SP7 Administration Guide available at https://documentation.suse.com/sles/15-SP7/.

## 3.11 Managing user accounts

### 3.11.1 Creating users

Use the *useradd*(8) command to create new user accounts, then use the *passwd*(1) command to assign an initial password for the user. Alternatively, if the user is present when the account is created, permit them to choose their own password. Refer to the manual pages for *useradd*(8) and *passwd*(1) for more information.

If you assign an initial password for a new user, you MUST transfer this initial password in a secure way to the user, ensuring that no third party gets the information. For example, you can tell the password to a user personally known to you. If this is not possible, you MAY send the password in written form in a sealed letter. This applies also when you set a new password for a user in case the user has forgotten the password or it has expired. You MUST advise the user that he MUST change this initial password when he first logs into the system and select his own password in accordance with the rules defined in section §6.3 "Password policy" of this guide.

You MUST NOT use the `-p` option to *useradd*(8), specifying a password in that way would bypass the password quality checking mechanism.

The temporary password set by the administrator MUST be changed by the user as soon as possible. Use the *chage*(8) command with the `-d` option to set the last password change date to a value where the user will be reminded to change the password. The RECOMMENDED value is based on the settings in */etc/login.defs* and is equivalent to today's date plus PASS_WARN_AGE minus PASS_MAX_DAYS.

Example:

```
useradd -m -c "John Doe" jdoe
passwd jdoe
chage -d $(date +%F -d "53 days ago") jdoe
```

The `-m` option to *useradd*(8) creates a home directory for the user based on a copy of the contents of the */etc/skel/* directory. Note that you MAY modify some default configuration settings for users, such as the default *umask*(2) setting or time zone, by editing the corresponding global configuration files:

```
/etc/profile
/etc/bash.bashrc
/etc/csh.cshrc
```

### 3.11.2 Changing user passwords

If necessary, you MAY reset the user's password to a known value using `passwd` *USER*, and entering the new password. You cannot recover the previously used password, since the hash function used is not reversible.

### 3.11.3 SSH key-based authentication

The TOE allows the configuration of key-based authentication for SSH. Key-based authentication is configured on a per-user basis by managing the file *.ssh/authorized_keys* in the home directory of a user. For information on how to use that file, see sshd(8).

To generate keys that can be used for key-based authentication, the tool *ssh-keygen*(8) is provided and strongly RECOMMENDED as only the ssh-keygen utility provided with the TOE was subject to the security assessment. As the SSH daemon only accepts SSH protocol version 2, only the protocol 2 keys are supported with the SSH daemon. Therefore, you MUST only use the option `-t rsa` or `-t ecdsa` when generating a key with *ssh-keygen*.

The `ssh-keygen` utility allows you to specify the key size for RSA with the default of 2048 bits. You MUST change the default and use key sizes above 3072 bits. All supported key sizes for ECDSA are allowed.

The private key part MUST be stored in *~/.ssh/* inaccessible to other users. This file must be treated similarly to a password. It is strongly RECOMMENDED that this key is protect with a passphrase using `ssh-keygen`.

The following command line is an example that generates an ECDSA key:

```
ssh-keygen -t ecdsa -C "John Doe's key"
```

The command asks for a passphrase where a strong passphrase SHOULD be provided.

Please note that account locking does not prevent users to log onto the system with SSH key-based authentication.

### 3.11.4 Changing user properties

You MAY use the *usermod*(8) command to change a user's properties.

### 3.11.5 Locking and unlocking of user accounts

Users MAY be locked out (disabled) using `passwd -l USER`, and re-enabled using `passwd -u USER`. Note that this locking only prevents password-based authentication attempts. SSH key-based authentication is unaffected by using `passwd -l`. To prevent SSH key-based logins, the file *.ssh/authorized_keys* located in the home directory of the user MUST be removed.

The *pam_tally2.so* PAM module enforces automatic lockout after excessive failed authentication attempts. Use the program `pam_tally2` to view and reset the counter if necessary, as documented in the *pam_tally2*(8) man page.

Please note that the order is very important while adding configurations to the pam configuration files */etc/pam.d/login*, */etc/pam.d/su-l* and */etc/pam.d/su*. As the importance of the order is not described in depth in *pam_tally2*(8) an example is provided below where the access will be denied after 5 attempts:

File */etc/pam.d/login*:

```
#%PAM-1.0
auth       required   pam_env.so
auth       required   pam_tally2.so   onerr=fail deny=5
auth       requisite  pam_nologin.so
auth       include    common-auth
...
```

Files */etc/pam.d/su*:

```
auth       required   pam_tally2.so   onerr=fail deny=5
auth       include    common-auth
...
```

Files */etc/pam.d/su-l*:

```
auth    sufficient      pam_rootok.so
auth    required        pam_tally2.so   onerr=fail deny=5
auth    include         common-auth
...
```

It is important, that *pam_tally2.so* is stated as second entry after *pam_env.so* in */etc/pam.d/login* before the inclusion of *common-auth*.

Note that the *pam_tally2* mechanism does not *prevent* password guessing attacks, it only prevents *use* of the account after such an attack has been detected. Therefore, you MUST assign a new password for the user before reactivating an account. For example:

```
# view the current counter value
pam_tally2 --user jdoe

# set new password, and reset the counter
passwd jdoe
pam_tally2 --user jdoe --reset
```

The *chage*(1) utility MAY be used to view and modify the expiry settings for user accounts. Unprivileged users are able to view but not modify their own expiry settings.

### 3.11.6 Removing users

The `userdel`(8) utility removes the user account from the system, but does not remove files outside the home directory (and the mail spool file), or kill processes belonging to this user. Use `kill` (or reboot the system) and `find` to do so manually if necessary, for example:

```
# Which user to delete?
U=jdoe

# Lock user account, but don't remove it yet
passwd -l $U

# Kill all user processes, repeat if needed (or reboot)
kill -9 `ps -la --User $U --user $U |awk '{print $4}'`

# Recursively remove all files and directories belonging to user
# (Careful - this may delete files belonging to others if they
# are stored in a directory owned by this user.)
# Use the applicable file system type for your system.
find / -depth \( ! -fstype ext3 -prune -false \) \
        -o -user $U -exec rm -rf {} \;

# Remove cron and at jobs
crontab -u $U -r
find /var/spool/atjobs -user $U -exec rm {} \;

# Now delete the account
userdel $U
```

Please note that similar concerns apply when a group is removed. The administrator MUST ensure that the files associated with the group are reassigned to other groups or deleted. In addition, the administrator MUST handle the processes currently executing with the deleted group.

In addition, the administrator should consider that the user ID may be used in ACLs where these ACLs should be checked for their validity.

If you need to create additional groups or modify existing groups, use the *groupadd*(8), *groupmod*(8) and *groupdel*(8) commands.

Group passwords are NOT supported in the evaluated configuration, and have been disabled by removing the SUID bit from the *newgrp*(8) program. You MUST NOT re-enable this feature and MUST NOT use *passwd*(1) with the `-g` switch or the *gpasswd*(1) command to set group passwords.

### 3.11.7 Defining administrative accounts

Administrative users MUST be member of the *trusted* group. Specify the `-G trusted` option for the *useradd*(8) command when creating administrative users.

You MAY also use the *usermod*(8) command to change group membership. For example, if you want to add the user 'jdoe' to the *trusted* group, you could use the following:

```
# List the groups the user is currently a member of:
groups jdoe

# Add the additional group
usermod -G $(groups jdoe | sed 's/.*: //; s/ /,/g'),trusted jdoe
```

## 3.12 Using serial terminals

You MAY attach serial terminals to the system for use by system administrators.

Serial terminals are activated by systemd automatically when the kernel command line option `console` is used to enable a serial console. For example:

```
console=ttyS0,115200n8
```

## 3.13 SSH Agent Forwarding

The usage of the ssh agent forwarding on the system is not allowed and MUST therefore be disabled. The file */etc/ssh/ssh_config* MUST be modified as follows:

```
Host *
        ForwardAgent no
```

## 3.14 Managing data objects

### 3.14.1 Revoking access

As with most operating systems, access rights are checked only once, when the object is first accessed by the process. If the initial permission check was successful, read and/or write operations are permitted indefinitely without further checking, even if the access rights to the object are changed or revoked.

If this delayed revocation is not acceptable to you and you need to definitely ensure that no user processes are accessing an object after you have changed the access rights to that object, you MUST reboot the system. This ensures that no processes have open descriptors which could permit continued access.

### 3.14.2 SYSV shared memory and IPC objects

The system supports SYSV-compatible shared memory, IPC objects, and message queues. If programs fail to release resources they have used (for example, due to a crash), the administrator MAY use the *ipcs*(8) utility to list information about them, and *ipcrm*(8) to force deletion of unneeded objects. Note that these resources are also released when the system is rebooted.

For additional information, please refer to the *ipc*(2) manual page.

### 3.14.3 Posix Message Queues

POSIX message queues are supported as an alternative to SYSV message queues. Users and administrators MAY use the system calls and corresponding library functions documented in the *mq_overview*(7) man page, such as *mq_open*(2) and *mq_unlink*(2).

The message queue filesystem (type *mqueue*) MAY be mounted in case filesystem-based access to POSIX message queues is requested.

## 3.15 Configuring object access rights

Administrators MAY use the *chown*(1), *chgrp*(1), and *chmod*(1) tools to configure DAC access rights. You MUST NOT grant additional access to objects that are part of the evaluated configuration.

Please refer to the respective man pages for more information about these tools.

## 3.16 Setting the system time and date

You MUST verify periodically that the system clock is sufficiently accurate, otherwise log and audit files will contain misleading information. When starting the system, the time and date are copied from the computer's hardware clock to the kernel's software clock, and written back to the hardware clock on system shutdown.

All internal dates and times used by the kernel, such as file modification stamps, use universal time (UTC), and do not depend on the current time zone settings. Userspace utilities usually adjust these values to the currently active time zone for display. Note that text log files will contain ASCII time and date representations in local time, often without explicitly specifying the time zone.

The *date*(1) command displays the current time and date, and can be used by administrators to set the software clock, using the argument *mmddHHMMyyyy* to specify the numeric month, day, hour, minute and year respectively. For example, the following command sets the clock to May 1st 2004, 1pm in the local time zone:

```
date 050113002004
```

The *hwclock*(8) can query and modify the hardware clock on supported platforms, but is not available in virtual environments such as LPAR. The typical use is to copy the current value of the software clock to the hardware clock. Note that the hardware clock MAY be running in either local time or universal time, as indicated by the *UTC* setting in the */etc/sysconfig/clock* file. The following command sets the hardware clock to the current time using UTC:

```
hwclock -u -w
```

Use the command *tzselect*(8) to change the default time zone for the entire system. Note that users MAY individually configure a different time zone by setting the *TZ* environment variable appropriately in their shell profile, such as the *$HOME/.bashrc* file.

## 3.17 Configuring time synchronization with NTP

To configure the time servers *chrony(1)* reads its configuration from */etc/chrony.conf*. Therefore, */etc/chrony.conf* needs to be updated. To do so, specific server names or IP addresses like

```
0.suse.pool.ntp.org
1.suse.pool.ntp.org
```

or a pool name like

```
pool pool.ntp.org
```

can be specified. Pool name resolves to several IP addresses. See *chrony.conf(5)* for more details.

Afterwards, *chrony* needs to be executed using

```
systemctl start chronyd.service
```

Until the system time is stable it may take some time. To additionally, start *chrony* at boot time

```
systemctl enable chronyd.service
```

should be used.

Note, since only one instance of *chronyd* can be running at any given time, do not enable or start *yast-timesync.service*.

Further note, *chronyc*(1) can be used to see status reports about the operation of *chronyd*. For details see the man page of *chronyc*(1).

## 3.18 Firewall configuration

You MAY enable, reconfigure, or disable the builtin network firewall as required. SLES allows the following types of firewall configuration to control traffic:

- The packet filtering of IP, TCP, UDP, ICMP protocols is implemented with `firewalld` which can be used via the command line utility *firewall-cmd* or the graphical user interface *firewall-config*. *firewalld* can be used to set up very small and lean packet filter rules as well as complex and sophisiticated filtering rules as needed by the administrator. See section 23.4 of https://documentation.suse.com/sles/15-SP7/html/SLES-all/book-security.html "Security and Hardening Guide" as well as *firewall-cmd*(1) for details on the use of the application.
- SLES allows the configuration of virtual machines using the KVM functionality and connects the guest software with external networks using the Linux kernel's bridging functionality. As the bridging functionality is enforced at Ethernet layer, the Linux kernel does not engage its TCP/IP stack for packets traveling through the bridge to received by either some KVM guest software or remote systems. SLES provides a packet filter mechanism for filtering packets at the Ethernet layer using the `ebtables` functionality. The man page *ebtables*(8) discusses the concept as well as the use of the packet filter.

### 3.18.1 firewalld auditing of packet filter operations

firewalld contains the *set-log-denied* statement which is documented in *firewall-cmd*(1).

This allows the logging of denied packets. When this option is set the Linux kernel will generate an audit entry for each denied (rejected or dropped) packet.

See *firewall-cmd*(1) for more information.

## 3.19 Screen saver configuration

The `screen` application is used to provide a locking mechanism of the current terminal for every user.

In the default configuration, `screen` is not enabled. You MAY enable screen by executing the script:

```
/usr/share/doc/packages/certification-sles-eal4/
        screen-script-screensaver
```

This script modifies */etc/profile* to allow screen to be started at logon time. The following discussion applies only when the mentioned script was executed. Note that irrespectively of enabling screen with the mentioned script, the scrollback buffer in the terminal is disabled using a kernel command line. The kernel command line option for the scrollback buffer is discussed in the following paragraphs.

The `screen` locking is invoked by the following means:

- The locking is executed automatically after a period of inactivity on the terminal defined by a timeout in either */etc/screenrc* or *~/.screenrc* using the *idle X lockscreen* configuration value where X is an integer value specifying the idle time in seconds before the screen is locked.
- Every user can lock his screen by executing the `C-a C-x` screen key binding combination.

You MAY change the timeout value for locking the session in */etc/screenrc* with the value for *lockscreen*.

Please note that users can modify the timeout by providing their own *~/.screenrc*. You can disable the support for per-user configuration files by invoking screen with the option of *-c /dev/null*.

**WARNING**: If a user accesses the system remotely and the screen saver functionality is triggered, the TOE ensures that the session is locked. However, it is possible that the remote terminal implements a scroll-back buffer that is not under the control of the TOE. Therefore, it is possible that the remote terminal has the session locked but a user can scroll back and list the history of actions. If the user shall not be able to use the scroll back buffer of the remote terminal, that terminal must be configured accordingly as this buffer is not under the control of the TOE. The local scroll back buffer is disabled with the kernel command line entries of:

```
no-scroll fbcon=scrollback:0
```

To invoke `screen` automatically upon log in, you MAY enter the following lines to either */etc/bash_profile* for system-wide enforcement or *~/.bash_profile* for a per-user enforcement. Note that a user can change *~/.bash_profile*.

```
exec screen
```

## 3.20 Update configuration

As a prerequisite to receive continuous updates you need to register the system with the SUSE Customer Center (SCC) as described in the section Installation process "Installation process".

To install updates YaST is used. This is inline with the evaluated configuration.

### 3.20.1 Manual update

To open the online update dialog either start YaST and select Software › Online Update or alternatively, start it from the command line with *yast2 online_update*.

The application will now show available patches sorted by security relevance: security, recommended, and optional. You can now select an entry in the Summary section to view a short Patch Description at the bottom left corner of the dialog. The upper right section lists the packages included in the selected patch.

By default, all new patches (except optional ones) that are currently available for your system are already marked for installation. They will be applied automatically once Accept or Apply is clicked. If one or multiple patches require a system reboot, this will be shwon before the patch installation starts. After the installation is complete, click Finish to leave the YaST Online Update. Your system is now up to date.

### 3.20.2 Automatic update

To open the automatic online update dialog either start YaST and select Software › Online Update › Configuration › Online Update or alternatively, start it from the command line with *yast2 online_update_configuration*.

Now the update interval can be set to Daily, Weekly, or Monthly.

Sometimes patches may require the attention of the administrator. Before these patches are installed, the user is informed about the consequences and is asked to confirm the installation of the patch. Such patches are called "Interactive Patches". When installing patches automatically, it is assumed that you have accepted the installation of interactive patches. If you prefer to review these patches before they get installed, check Skip Interactive Patches. In this case, interactive patches will be skipped during automated patching. Make sure to periodically run a manual online update, to check whether interactive patches are waiting to be installed.

To automatically accept any license agreements, activate Agree with Licenses. Activate Include Recommended Packages to automatically install all packages recommended by updated packages. To disable the use of delta RPMs (for performance reasons), un-check Use Delta RPMs.

To filter the patches by category (such as security or recommended), check Filter by Category and add the appropriate patch categories from the list. Only patches of the selected categories will be installed. It is a good practice to enable only automatic Security updates, and to manually review all others. Patching is usually reliable, but you may wish to test non-security patches, and roll them back if you encounter any problems.

As last step confirm your configuration by clicking OK.

Note: The automatic online update does not automatically restart the system afterward. If there are package updates that require a system reboot, this needs to be done manually.

Note: To disable automatic updates, uncheck Automatic Online Update and confirm by clicking OK.

## 3.21 Cryptographic Support

### 3.21.1 OpenSSL on x86 Architecture

Per default, OpenSSL uses the AES-NI instruction set if the underlying x86 CPU provides it. If the administrator wants to revert to the software implementation of AES instead of AES-NI as the software implementation can be reviewed, the following environment variable MUST be set for the applications that use OpenSSL:

```
OPENSSL_ia32cap="~0x200000200000000"
```

Note, in the evaluated configuration, AES-NI MUST be deactivated. Thus, to comply with the evaluation result, you MUST apply the environment variable for OpenSSH and the *openssl* application.

This environment variable ensures that AES-NI as well as the PCLMULQDQ support in the underlying x86 CPU are disabled.

### 3.21.2 OpenSSL on IBM Z Architecture

Per default, OpenSSL uses the VMX instruction set if the underlying IBM Z System CPU provides it. If the administrator wants to revert to the software implementation of AES instead of VMX as the software implementation can be reviewed, the following environment variable MUST be set for the applications that use OpenSSL:

```
OPENSSL_s390xcap="kimd:0:0;klmd:0:0;km:0:0;kmc:0:0;kmac:0:0;kmctr:0:0;
kmo:0:0;kmf:0:0;prno:0:0;kma:0:0;pcc:0:0;kdsa:0:0"
```

Note, in the evaluated configuration, VMX MUST be deactivated. Thus, to comply with the evaluation result, you MUST apply the environment variable for OpenSSH and the *openssl* application.

After updating the configuration, you MUST reboot the system.

### 3.21.3 OpenSSL on ARM Architecture

Per default, OpenSSL uses the CE instruction set if the underlying ARM CPU provides it. If the administrator wants to revert to the software implementation of AES instead of CE as the software implementation can be reviewed, the following script MUST be created to define the environment variable for the applications that use OpenSSL:

*disable_CE.sh*

```
#!/bin/bash

armcap=0

lscpu | grep "Flags" | grep -q "neon" && armcap=$((armcap | (1 << 0)))
lscpu | grep "Flags" | grep -q "aes" && armcap=$((armcap | (1 << 2)))
lscpu | grep "Flags" | grep -q "sha1" && armcap=$((armcap | (1 << 3)))
lscpu | grep "Flags" | grep -q "sha2" && armcap=$((armcap | (1 << 4)))
lscpu | grep "Flags" | grep -q "pmull" && armcap=$((armcap | (1 << 5)))
lscpu | grep "Flags" | grep -q "sha512" && armcap=$((armcap | (1 << 6)))
lscpu | grep "Flags" | grep -q "cpuid" && armcap=$((armcap | (1 << 7)))
lscpu | grep "Flags" | grep -q "rng" && armcap=$((armcap | (1 << 8)))
lscpu | grep "Flags" | grep -q "sm3" && armcap=$((armcap | (1 << 9)))
lscpu | grep "Flags" | grep -q "sm4" && armcap=$((armcap | (1 << 10)))
if lscpu | grep "Flags" | grep -q "sha3"; then
        armcap=$((armcap | (1 << 11)))
        cat /proc/cpuinfo | grep "CPU part" | grep -q "0xd40" &&
        armcap=$((armcap | (1 << 12)))
        cat /proc/cpuinfo | grep "CPU part" | grep -q "0xd49" &&
        armcap=$((armcap | (1 << 12)))
        cat /proc/cpuinfo | grep "CPU part" | grep -q "0xd4f" &&
        armcap=$((armcap | (1 << 12)))
        cat /proc/cpuinfo | grep "CPU part" | grep -q "0xd40" &&
        armcap=$((armcap | (1 << 16)))
        cat /proc/cpuinfo | grep "CPU part" | grep -q "0xd4f" &&
        armcap=$((armcap | (1 << 16)))
fi
lscpu | grep "Flags" | grep -q "sve" && armcap=$((armcap | (1 << 13)))
lscpu | grep "Flags" | grep -q "sve2" && armcap=$((armcap | (1 << 14)))

# Disable CE instruction set features, i.e. bits 2 (AES), 0 (NEON).
# 16 (UNROLL12_EOR3), 12 (UNROLL8_EOR3),
# 5 (PMULL), 3 (SHA1), 4 (SHA256), 6 (SHA512), 15 (SHA3_AND_WORTH_USING)
# and 11 (SHA3)
disable_mask=$((0x58 | 0x11025 | 0x8801))
armcap=$((armcap & ~disable_mask))

export OPENSSL_armcap=$armcap
```

Afterwards the script MUST be executed.

### 3.21.4 SSH Client Configuration

The evaluated configuration requires that keys generated for OpenSSH applications including the `sshd` daemon, the `ssh` client application and `ssh-keygen` must be generated using a random number generator that is seeded with at least 256 bits of entropy.

OpenSSH uses the OpenSSL deterministic random number generator for generating keys. This deterministic random number generator is seeded by reading the seed from the `Jitter RNG`.

For the OpenSSH client protection, you MUST add the following option to the file */etc/ssh/ssh_config*

```
UseRoaming no
```

For additional parameters you MUST specify see §3.21.5 "SSH Server Configuration" below.

### 3.21.5 SSH Server Configuration

The evaluated configuration requires that the file */etc/ssh/sshd_config* MUST be modified as follows - irrespective whether the configuration file contains a comment that it is compliant to the CC configuration:

- The `HostbasedAcceptedKeyTypes`, `HostbasedAcceptedAlgorithms` and `HostKeyAlgorithms` values MUST be limited to the algorithms
  - rsa-sha2-256
  - rsa-sha2-512
  - ecdsa-sha2-nistp384
  - ecdsa-sha2-nistp521
- The values of `Ciphers` MUST be limited to
  - aes256-gcm@openssh.com
- The values of `KexAlgorithms` MUST be limited to
  - ecdh-sha2-nistp384
  - ecdh-sha2-nistp521
  - diffie-hellman-group16-sha512
  - diffie-hellman-group18-sha512
- The values of `MACs` MUST be limited to
  - hmac-sha2-256
  - hmac-sha2-512
- The values of `PubkeyAcceptedAlgorithms` MUST be limited to
  - ecdh-sha2-nistp384
  - ecdh-sha2-nistp521
- The `RekeyLimit` option MUST be set to `1G 1h`.
- Only `PubkeyAuthentication` and `PasswordAuthentication` are allowed to be enabled.

### 3.21.6 Cryptographic key handling

The cryptographic network protocols of SSH and TLS provide a secure channel to protect sensitive data. During the establishment of the secure channel, the protocols use certificates and private keys to support mutual authentication using X.509v3 certificates. The entire trust of the secure communication also rests on the appropriate authentication to establish the identity and authenticity of the remote peer. The X.509v3 certificate validation includes CRL or OCSP checks for received certificates, verification of the certificate chain terminated by a trusted CA certificate with the corresponding basicConstrains extension, the CA flag set to true and the Certificate Signing key as well as the verification of the extendedKeyUsage field. The verification is done for all TLS and SSH communications using mutual authentication.

Therefore, you MUST ensure that the generation and use of the certificates, certificate revocation lists (CRLs) used for certificate validation, and private and public keys used for these network protocols meet the corresponding standards and providing sufficient security strength through the use of appropriate key lengths and message digest algorithms.

You also MUST verify the integrity and authenticity of digital certificates and key material before importing them into the TOE, and verifying that certificates are signed using strong hash algorithms.

Please note, ephemeral cryptographic key material kept in volatile memory are completely cleared only after a full powercycle.

The following cryptographic mechanisms MUST be used for SSH:

**Encryption algorithms**

AES256-GCM@openssh.com

**Public key algorithms**

RSA-SHA2-256, RSA-SHA2-512, ECDSA-SHA2-NISTP384, or ECDSA-SHA2-NISTP521

**MAC algorithms**

HMAC-SHA2-256, HMAC-SHA2-512, AES256-GCM@openssh.com

**Key exchange**

Diffie-Hellman-group16-SHA512, Diffie-Hellman-group18-SHA512, ECDH-SHA2-NISTP384 or ECDH-SHA2-NISTP521

The TLS client support is implemented in OpenSSL. The authentication of the TLS server certificate is performed using FQDN or the IP address. Further wild cards for the server identifier resolution are supported. Certificate pinning, however, is not supported. The following cipher suites MUST be used:

- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

The allowed Supported Groups Extension in the Client Hello for TLS are:

- secp384r1
- secp521r1

In addition, when setting up disk encryption configurations, the passphrase protects the master volume key used to encrypt the data to be stored on disk. To ensure appropriate protection of that master volume key, you MUST use a sufficiently strong passphrase. For disk encryption you MUST use the allowed ciphers and allowed block chaining modes as specified in §3.7 "Encryption of partitions".

Furthermore, usernames and passwords used for authentication as well as private keys and certificates are stored in files protected by the operating system independent of the disk encryption. Username files are write protected and password files are read and write protected, except for the root user. Per-user SSH keys are read and writable only for the owning user.

### 3.21.7 Cryptographic key generation and establishment

Asymmetric cryptographic keys MUST be generated using one of the following cryptographic key generation algorithms (defined through the involved certificates):

- RSA with key sizes of 3072-bit, and 4096 bit, using RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017
- Elliptic Curve Cryptography (ECC) using the NIST curves NIST NIST P-384, or NIST P-521
- Finite-Field Cryptography (FFC) using approved Safe-Prime Groups as specified in NIST Special Publication 800-56A Revision 3

The cryptographic key establishment MUST be implemented using one of the following cryptographic key establishment methods:

- Elliptic Curve Cryptography (ECC) with ECC CDH
- Finite-Field Cryptography (FFC) with FFC DH

## 3.22 Certificate signing limitation

In the evaluated configuration it MUST be ensured that the CA certificate only signs a certificate whose public key uses one of the following

- NIST P-384
- NIST P-521
- RSA with modulus 3072 or larger

# Chapter 4

# Monitoring, Logging & Audit

## 4.1 Reviewing the system configuration

It is RECOMMENDED that you review the system's configuration at regular intervals to verify if it still agrees with the evaluated configuration. This primarily concerns those processes that may run with 'root' privileges.

The permissions of the device files */dev/** MUST NOT be modified.

In particular, review settings in the following files and directories to ensure that the contents and permissions have not been modified:

```
/etc/apparmor/*
/etc/audit/*
/etc/cron.allow
/etc/cron.d/*
/etc/cron.deny
/etc/cron.daily/*
/etc/cron.hourly/*
/etc/cron.monthly/*
/etc/cron.weekly/*
/etc/crontab
/etc/group
/etc/gshadow
/etc/hosts
/etc/systemd/*
/etc/ld.so.conf
/etc/libvirt/*
/etc/localtime
/etc/login.defs
/etc/modprobe.conf
/etc/pam.d/*
/etc/passwd
/etc/securetty
/etc/security/opasswd
/etc/security/*
/etc/shadow
/etc/ssh/ssh_config
/etc/ssh/sshd_config
/etc/sysconfig/*
```

```
/var/log/audit.d/*
/var/log/faillog
/var/log/lastlog
/var/spool/cron/*
```

Use the command `lastlog` to detect unusual patterns of logins.

Also verify the output of the following commands (run as 'root') to analyze that no unknown applications or commands are listed as they may indicate a breach of the security of the system:

```
# list of commands executed as root by cron
crontab -l

# list files with the SUID/SGID bit set
find / \( -perm -4000 -o -perm -2000 \) -ls

# list of world writable files, directories or block device files
find / \( -type f -o -type d -o -type b \) -perm -0002 -ls

# list of files in system directories which are not owned by root
find /bin /boot /etc /lib /sbin /usr \
        ! -type l \( ! -uid 0 -o -perm +022 \)
```

## 4.2 System logging and accounting

System log messages are stored in the */var/log/* directory tree in plain text format, most are logged through the *syslogd*(8) and *klogd*(8) programs, which MAY be configured via the */etc/syslog.conf* file.

The *logrotate*(8) utility, launched from */etc/cron.daily/logrotate*, starts a fresh log file every week or when they reach a maximum size and automatically removes or archives old log files. You MAY change the configuration files */etc/logrotate.conf* and */etc/logrotate.d/** as required.

In addition to the *syslog* messages, various other log files and status files are generated in */var/log* by other programs:

```
File           Source
-------------+------------------------------------------------------------
YaST2          Directory for YaST2 log files
audit.d        Directory for LAF logs
boot.msg       Messages from system startup
lastlog        Last successful log in  (see lastlog(8))
libvirt        Log maintained by libvirtd
localmessages  Written by syslog
mail           Written by syslog, contains messages from the MTA (postfix)
messages       Written by syslog, contains messages from su and ssh
news/          syslog news entries (not used in the evaluated configuration)
warn           Written by syslog
wtmp           Written by the PAM susbystem, see who(1)
```

Please see *syslog*(3), *syslog.conf*(5) and *syslogd*(8) man pages for details on syslog configuration.

The *ps*(1) command can be used to monitor the currently running processes. Using `ps faux` will show all currently running processes and threads.

## 4.3 Configuring the audit subsystem

The audit subsystem implements a central monitoring solution to keep track of security relevant events, such as changes and change attempts to security critical files.

This is accomplished through two separate mechanisms. All system calls are intercepted, and the kernel writes the parameters and return value to the audit log for those calls that are marked as security relevant in the filter configuration. In addition, some trusted programs contain audit-specific code to write audit trails of the actions they are requested to perform.

Please refer to the *auditd*(8), *auditd.conf*(5), and *auditctl*(8) man pages for more information.

For information regarding remote auditing please refer to *audisp-remote*(8) and *audisp-remote.conf*(5).

### 4.3.1 Intended usage of the audit subsystem

The Operating System Protection Profile (OSPP) specifies the auditing capabilities that a compliant system must support. The evaluated configuration described here is based on these requirements.

**WARNING:** Some of the protection profile requirements may conflict with your specific requirements for the system. For example, an OSPP-compliant system MUST disable logins if the audit subsystem is not working. Please ensure that you are aware of the consequences if you enable auditing.

OSPP is designed for a multiuser system, with multiple unique users who maintain both shared and private resources. The auditing features are intended to support this mode of operation with a reliable trail of security-relevant operations. It is less useful for a pure application server with no interactive users.

Please be aware that the auditing subsystem will, when activated, cause some slowdown for applications on the server. The impact depends on what the application is doing and how the audit subsystem is configured. As a rule of thumb, applications that open a large number of separate files are most affected, and CPU-bound programs should not be measurably affected. You will need to balance the performance requirements against your security needs when deciding if and how you want to use auditing.

### 4.3.2 Selecting the events to be audited

You MAY make changes to the set of system calls and events that are to be audited. OSPP requires that the system has the *capability* to audit security relevant events, but it is up to you to choose how you want to use these capabilities. It is acceptable to turn off system call auditing completely even in an evaluated configuration, for example on a pure application server with no interactive users on the system.

The audit package provides several suggested audit configuration files, for example the */usr/share/doc/packages/audit/capp.rules* file for systems covering the basic system functionality. It contains a suggested setup for a typical multiuser system, all access to security relevant files is audited, along with other security relevant events such as system reconfiguration. You MAY copy one of the sample rules files to */etc/audit/audit.rules* and modify the configuration according to your local requirements, including the option of using an empty audit rules file to disable auditing if not required.

The man page *auditctl*(8) provides a discussion of the audit rules.

### 4.3.3 Reading and searching the audit records

Use the *ausearch*(8) tool to retrieve information from the audit logs. The information available for retrieval depends on the active filter configuration. If you modify the filter configuration, it is RECOMMENDED keeping a datestamped copy of the applicable configuration with the log files for future reference.

For example:

```
# search for events with a specific login UID
ausearch -ul jdoe

# search for events by process ID
ausearch -p 4690
```

Please refer to the *ausearch*(8) man page for more details.

For some system calls on some platforms, the system call arguments in the audit record can be slightly different than you may expect from the program source code due to modifications to the arguments in the C library or in kernel wrapper functions. For example, the *mq_open*(3) glibc library function strips the leading '/' character from the path argument before passing it to the *mq_open*(2) system call, leading to a one character difference in the audit record data. Similarly, some system calls such as *semctl*(2), *getxattr*(2), and *mknodat*(2) can have additional internal flags automatically added to the flag argument. These minor modifications do not change the security relevant information in the audit record.

Of course, you can use other tools such as plain *grep*(1) or scripting languages such as *awk*(1), *python*(1) or *perl*(1) to further analyze the text audit log file or output generated by the low-level *ausearch* tool.

### 4.3.4 Starting and stopping the audit subsystem

If the audit daemon is terminated, no audit events are saved until it is restarted. To avoid lost audit records when you have modified the filter configuration, you MUST use the command `service auditd reload` to re-load the filters.

You MUST NOT use the *KILL* signal (-9) to stop the audit daemon, doing so would prevent it from cleanly shutting down.

It is RECOMMENDED that you add the kernel parameter `audit=1` to your boot loader configuration file to ensure that all processes, including those launched before the *auditd* service, are properly attached to the audit subsystem. Please refer to the documentation of your boot loader for more details on how to modify the kernel command line.

### 4.3.5 Storage of audit records

The default audit configuration stores audit records in the */var/log/audit/audit.log* file. This is configured in the */etc/audit/auditd.conf* file. You MAY change the *auditd.conf* file to suit your local requirements.

It is RECOMMENDED that you configure the audit daemon settings appropriately for your local requirements, for example by changing the log file retention policy to never delete old audit logs with the following setting in the */etc/audit/auditd.conf* file:

```
max_log_file_action = KEEP_LOGS
```

The most important settings concern handling situations where the audit system is at risk of losing audit information, such as due to lack of disk space or other error conditions. You MAY choose actions appropriate for your environment, such as switching to single user mode (action `single`) or shutting down the system (action `halt`) to prevent auditable actions when the audit records cannot be stored.

**Warning:** Switching to single user mode does not automatically kill all user processes when using the system default procedure. You MAY kill processes of users by using `killall -u`. Please note that system services SHOULD NOT be terminated.

Halting the system is RECOMMENDED and most certain way to ensure all user processes are stopped. The following settings are RECOMMENDED in the */etc/audit/auditd.conf* file if a fail-secure audit system is required:

```
admin_space_left_action = SINGLE
disk_full_action = HALT
disk_error_action = HALT
```

It is RECOMMENDED that you configure appropriate disk space thresholds and notification methods to receive an advance warning when the space for audit records is running low.

It is RECOMMENDED that you use a dedicated partition for the */var/log/audit/* directory to ensure that *auditd* has full control over the disk space usage with no other processes interfering.

Please refer to the *auditd.conf*(5) man page for more information about the storage and handling of audit records.

### 4.3.6 Remote auditing

In the evaluated configuration remote auditing is allowed using *audisp-remote*(8) and *audisp-remote.conf*(5) as well as *ssh*(1) for the secure communication between the remote logging server and the client. For details please refer to the respective man pages. Nevertheless, an example configuration is provided below.

Please note, the example below MAY be adapted based on your needs, however, the communication to the remote server MUST use SSH with the allowed cipher suites as configured in §3.21.6"Cryptographic key handling" and key-based authentication as described in §6.4 "SSH key-based authentication".

1. On the receiving logging server side, set the receiving port and IP address to localhost and a not used port, therefore auditd is not exposed to the network and SSH handles the forwarding to the localhost and port. An example configuration is shown below for */etc/audit/auditd.conf*

   ```
   tcp_listen_port = <server_port_for_traffic>
   tcp_listen_address = 127.0.0.1
   ```

2. Configure on the TOE */etc/audisp/audisp-remote.conf* as follows

   ```
   transport = tcp
   remote_server = 127.0.0.1
   port = <local_port_for_traffic>
   # use a local queue to buffer logs temporarily in case the ssh
   # tunnel is interrupted
   queue_file = <Path to the buffer file for unsent logs>
   queue_depth = <Maximum number of records in the queue>
   queue_max_file_size = <Maximum size of the buffer file>
   queue_max_age = <Maximum time logs are kept in the queue>
   ```

3. In the file */etc/auditd/plugins.d/au-remote.conf* set

   ```
   active = yes
   ```

4. Create an SSH tunnel between client and server

   ```
   ssh -L 127.0.0.1:<local_port_for_traffic>:127.0.0.1:<server_port_for_traffic>
           <username>@<remote_server>
   ```

5. At last step restart auditd on both systems

   ```
   systemctl restart auditd
   ```

In addition setup a systemd service for the SSH tunnel to ensure that the tunnel restarts if it crashes or the connection drops. Otherwise logs exceeding the `queue_depth` number are dropped. Please ensure that the audit policies adhere to your company policies.

You MAY need to adjust the firewall settings to allow the sending and receiving of logs on the corresponding machine.

### 4.3.7 Reliability of audit data

You MAY choose an appropriate balance between availability of the system and secure failure mode in case of audit system malfunctions based on your local requirements.

You MAY configure the system to cease all processing immediately in case of critical errors in the audit system. When such an error is detected, the system will then immediately enter "panic" mode and will need to be manually rebooted. To use this mode, add the following line to the */etc/audit/audit.rules* file:

```
-f 2
```

Please refer to the *auditctl*(8) man page for more information about the failure handling modes.

You MAY edit the */etc/libaudit.conf* file to configure the desired action for applications that cannot communicate with the audit system. Please refer to the *get_auditfail_action*(3) man page for more information.

*auditd* writes audit records using the normal Linux filesystem buffering, which means that information can be lost in a crash because it has not been written to the physical disk yet. Configuration options control how *auditd* handles disk writes and allow the administrator to choose an appropriate balance between performance and reliability.

Any applications that read the records while the system is running will always get the most current data out of the buffer cache, even if it has not yet been committed to disk, so the buffering settings do not affect normal operation.

The default setting is `flush = DATA`, ensuring that record data is written to disk, but metadata such as the last file time might be inconsistent.

The highest performance mode is `flush = none`, but be aware that this can cause loss of audit records in the event of a system crash.

If you want to ensure that auditd always forces a disk write for each record, you MAY set the `flush = SYNC` option in */etc/audit/auditd.conf*, but be aware that this will result in significantly reduced performance and high strain on the disk.

A compromise between crash reliability and performance is to ensure a disk sync after writing a specific number of records to provide an upper limit for the number of records lost in a crash. For this, use a combination of `flush = INCREMENTAL` and a numeric setting for the `freq` parameter, for example:

```
flush = INCREMENTAL
freq = 100
```

The audit record files are *not* protected against a malicious administrator, and are not intended for an environment where the administrators are not trustworthy.

## 4.4 System configuration variables in */etc/sysconfig*

The system uses various files in */etc/sysconfig* to configure the system. Most files in this directory tree contain variable definitions in the form of shell variables that are either read by the rc scripts at system boot time or are evaluated by the `SuSEconfig` command and used as input to re-write other configuration files on the system.

In the evaluated configuration, no changes are permitted that would require running the `SuSEconfig` command to re-write other configuration files. You MAY run `SuSEconfig`, but it will have no effect on the evaluated configuration.

# Chapter 5

# Application Developers

When creating application running on SLES the application developers can use the included `gcc` compiler and linker. When invoking `gcc`, the best practices for secure development should be followed by developers:

- Include the enabling of stack smashing protections through the following compiler flags:

```
-fstack-protector-strong --param=ssp-buffer-size=4
```

- Include the enabling of ASLR through the following compiler and linker flags:

```
-fpie -Wl,-pie
```

# Chapter 6

# Security guidelines for users

## 6.1 Online Documentation

The system provides a large amount of online documentation, usually in text format. Use the `man` program to read entries in the online manual, for example:

```
man ls
man man
```

to read information about the `ls` and `man` commands respectively. You can search for keywords in the online manual with the *apropos*(1) utility, for example:

```
apropos password
```

When this guide refers to manual pages, it uses the syntax ENTRY(SECTION), for example *ls*(1). Usually you do not need to provide the section number, but if there are several entries in different sections, you can use the optional `-S` switch and pick a specific one.

Some programs provide additional information GNU 'texinfo' format, use the `info` program to read it, for example:

```
info diff
```

Additional information, sorted by software package, can be found in the */usr/share/doc/*/* directories. Use the *less*(1) pager to read it, for example:

```
less /usr/share/doc/packages/bash/FAQ
```

Many programs also support a `--help`, `-?` or `-h` switch you can use to get a usage summary of supported command-line parameters.

Note that this Configuration Guide has precedence over other documents in case of conflicting recommendations.

## 6.2 Authentication

You MUST authenticate (prove your identity) before being permitted to use the system. When the administrator created your user account, he or she will have assigned a user name and default password, and provided that information for you along with instructions how to access the system.

Please note, next to the password authentication also key-based authentication as described in §6.4 "SSH key-based authentication" is allowed. However, these are the only two allowed authentication methods in the evaluated configuration and one of these MUST be used.

Logging in to the system will usually be done using the Secure Shell (SSH) protocol, alternatively a serial terminal may be available. Use the `ssh` command to connect to the system unless instructed otherwise by the administrator, for example:

```
ssh jdoe@172.16.0.1
```

The *ssh*(1) manual page provides more information on available options. If you need to transfer files between systems, use the *scp*(1) or *sftp*(1) tools.

If this is the first time you are connecting to the target system, you will be prompted if you want to accept the host key. If the administrator has provided a key fingerprint for comparison, verify that they match, otherwise type `yes` to continue. You MUST immediately change your initially assigned password with the *passwd*(1) utility.

You MUST NOT under any circumstances attempt to log in from an insecure device, such as a public terminal or a computer belonging to a friend. Even if the *person* owning the computer is trustworthy, the *computer* may not be due to having been infected with malicious code. Always remember that the device you are typing your password into has the ability to save and re-use your authentication information, so you are in effect giving the computer you are using the right to do any and all actions in your name. Insecure handling of authentication information is the leading cause for exploits of otherwise secure systems, and SSH can only protect the information during transit, and offers no protection at all against an insecure endpoint.

When you log out from the system and leave the device you have used for access (such as a terminal or a workstation with terminal emulation), you MUST ensure that you have not left information on the screen or within an internal buffer that should not be accessible to another user. You should be aware that some terminals also store information not displayed on the terminal (such as passwords, or the contents of a scrollback buffer). Nevertheless this information may be extractable by the next user unless the terminal buffer has been cleared. Safe options include completely shutting down the client software used for access, powering down a hardware terminal, or clearing the scrollback buffer by switching among virtual terminals in addition to clearing the visible screen area.

If you ever forget your password, contact your administrator who will be able to assign a new password.

You MAY use the *chsh*(1) and *chfn*(1) programs to update your login shell and personal information if necessary. Not all settings can be changed this way, contact your administrator if you need to change settings that require additional privileges.

## 6.3 Password policy

All users, including the administrators, MUST ensure that their authentication passwords are strong (hard to guess) and handled with appropriate security precautions. The password policy described here is designed to satisfy the requirements of the evaluated configuration. If your organization already has a password policy defined, your administrator MAY refer you to that policy if it is equivalently strong.

You MUST change the initial password set by the administrator when you first log into the system. You MUST select your own password in accordance with the rules defined here. You MUST also change the password if the administrator has set a new password, for example if you have forgotten your password and requested the administrator to reset the password.

Use the *passwd*(1) program to change passwords. It will first prompt you for your old password to confirm your identity, then for the new password. You will be prompted to enter the new password twice, to catch mistyped passwords.

The *passwd*(1) program will automatically perform some checks on your new password to help ensure that it is not easily guessable, but you MUST nevertheless follow the requirements in this chapter.

Note that the administrators MUST also ensure that their own passwords comply with this password policy, even in cases where the automatic checking is not being done, such as when first installing the system.

- Your password MUST be a minimum of 8 characters in length. More than 12 characters MAY be used (it is RECOMMENDED to use more than 12, best is to use passphrases), and all characters are significant.
- You MUST combine characters from different character classes to construct a sufficiently strong password, using at least one character from each class. The character classes are defined as follows:

  ```
  Lowercase letters: abcdefghijklmnopqrstuvwxyz
  Uppercase letters: ABCDEFGHIJKLMNOPQRSTUVWXYZ
  Digits:            0123456789
  Punctuation:       !"#$%&'()*+,-./:;<=>?[\]^_`{|}~
  ```

  Note that non-7-bit ASCII characters MAY be used for passwords.
- You MUST NOT base the password on a dictionary word, your real name, login name, or other personal details (such as dates, names of relatives or pets), or names of real people or fictional characters.
- Instead of a password, you MAY use a passphrase consisting of multiple unrelated words (at least three) joined with random punctuation characters. Such a passphrase MUST have a length of at least 16 characters. (This corresponds to automatically generated pass phrases constructed by choosing 3 words from a 4096 word dictionary and adding two punctuation characters from a set of 8, equivalent to 42 bits of entropy.)
- You MUST NOT use a simple alphabetic string, palindrome or combinations of adjacent keyboard keys.
- When you choose a new password, it MUST NOT be a simple variation or permutation of a previously used one.
- You MUST NOT write the password on paper or store it on electronic devices in unprotected form. Storage in a secure location (such as an envelope in a safety deposit box, or encrypted storage on an electronic device) MAY be acceptable, contact your administrator first to ensure that the protection is strong enough to make password recovery infeasible for the types of attackers the system is intended to protect against.
- The password is for you and you only. A password is like a toothbrush - you do not want to share it with anybody, even your best friend. You MUST NOT disclose your password to anybody else, or permit anybody else to use the system using your identity.

  Note that administrators will never ask you for your password, since they do not need it even if they are required to modify settings affecting your user account.
- You MUST NOT use the same password for access to any systems under external administration, including Internet sites. You MAY however use the same password for accounts on multiple machines within one administrative unit, as long as they are all of an equivalent security level and under the control of the same administrators.
- You MUST inform the administrator and select a new password if you have reason to believe that your password was accidentally disclosed to a third party.
- If the system notifies you that your password will expire soon or has expired, choose a new one as instructed. Contact your administrator in case of difficulty.

A RECOMMENDED method of generating passwords that fits these criteria while still being easy to memorize is to base it on letters of words in a sentence (NOT a famous quotation), including capitalization and punctuation and one or two variations. Example:

```
"Ask not for whom the bell tolls."
=> An4wtbt.

"Password 'P'9tw;citd' too weak; contained in this document"
=> P'9tw;citd
```

## 6.4 SSH key-based authentication

You MAY use the SSH key-based authentication documented in *sshd*(8) section "AUTHORIZED_KEYS FILE FORMAT". Before the SSH key-based authentication can be used, you must generate a key pair.

As only the *ssh-keygen*(1) utility provided with the TOE was subject to the security assessment, including the proper key generation support, it is strongly RECOMMENDED that you use this tool from the TOE.

You MUST generate key pairs for SSHv2 using the `-t rsa` or `-t ecdsa` command line switch.

The `ssh-keygen` utility allows you to specify the key size for RSA with the default of 2048 bits. You MUST change the default and use key sizes above 3072 bits. All supported key sizes for ECDSA are allowed.

You MUST keep the private key part stored in *~/.ssh/* inaccessible to any other user. This file must be treated similarly to a password. It is strongly RECOMMENDED that you protect that key with a passphrase using `ssh-keygen`.

The following command line is an example that generates an ECDSA key:

```
ssh-keygen -t ecdsa -C "John Doe's key"
```

The command asks you for a passphrase where you SHOULD provide a strong passphrase.

After the generation of the key pair, you MAY copy the applicable file out of the files *~/.ssh/*.pub* to your server system and append it to the file *~/.ssh/authorized_keys*. Create that file if it does not exist and ensure that its permission prevents others from accessing this file. More information can be found in *sshd*(8) section "AUTHORIZED_KEYS FILE FORMAT".

In case you fail to meet the above mentioned requirements, your account protection may be weakened. This can be considered similar to choosing a weak password or fail to keep the password confidential.

To erase the private key generated for the SSH key-based authentication see §3.8 "Secure erasure"".

Please note that using the key-based authentication is not subject to the account locking mechanism enforced for passwords.

## 6.5 Access control for files and directories

Linux is a multiuser operating system. You can control which other users will be able to read or modify your files by setting the Unix permission bits and user/group IDs, or (if more precise control is needed) by using POSIX-style access control lists (ACLs).

Note that the administrators ('root') are able to override these permissions and access all files on the system. Use of encryption is RECOMMENDED for additional protection of sensitive data. This may be achieved using gpg, see *GPG*(1) using

```
gpg -c <file>
```

and providing a secure passphrase.

### 6.5.1 Discretionary Access Control

You can control which other users will be able to read or modify your files by setting the Unix permission bits and user/group IDs, or (if more precise control is needed) by using POSIX-style access control lists (ACLs). This is referred to as discretionary access control (DAC).

The 'umask' setting controls the permissions of newly created files and directories and specifies the access bits that will be *removed* from new objects. Ensure that the setting is appropriate, and never grant write access to others by default. The umask MUST include at least the 002 bit (no write access for others), and the RECOMMENDED setting is 027 (read-only and execute access for the group, no access at all for others). The default configuration is even more strict as it sets 077 (accessible to the owner only).

Do not set up world-writable areas in the filesystem - if you want to share files in a controlled manner with a fixed group of other users (such as a project group), please contact your administrator and request the creation of a user group for that purpose.

Always remember that **you** are responsible for the security of the data you create and use. Choose permissions that match the protection goals appropriate for the content, and that correspond to your organization's security policy. Access to confidential data MUST be on a need-to-know basis, do not make data world-readable unless the information is intended to be public.

Whenever you start a program or script, it will execute with your access rights. This implies that a malicious program would be able to read and modify all files that you have access to. Never execute any code that you have received from untrustworthy sources, and do not run commands that you do not understand. Be aware that manipulations to the environment a program is run in can also cause security flaws, such as leaking sensitive information. Do not use the shell variables LD_LIBRARY_PATH or LD_PRELOAD that modify the shared library configuration used by dynamically linked programs.

Programs can be configured to run with the access rights of the program file's owner and/or group instead of the rights of the calling user. This is the SUID/SGID mechanism, which utilities such as *passwd*(1) use to be able to access security-critical files. You could also create your own SUID/SGID programs via *chmod*(1), but DO NOT do that unless you fully understand the security implications - you would be giving away *your* access privileges to whoever launches the SUID program. Please refer to the "Secure Programming HOWTO" in the unlikely case that you need to create such a program, there you will find explanations of the many aspects that must be considered, such as the risk of unintended shell escapes, buffer overflows, resource exhaustion attacks and many other factors. Note that SUID root programs MUST NOT be added to the evaluated configuration, the only permitted use of the SUID bit is for setting non-root user IDs.

Please refer to the *chmod*(1), *umask*(2), *chown*(1), *chgrp*(1), *acl*(5), *getfacl*(1), and *setfacl*(1) manual pages for information, or any of the many available books covering Linux security (cf. Appendix 'Literature'), or ask your system administrator for advice.

## 6.6 Data import / export

The system comes with various tools to archive data (*tar*, *star*, *cpio*). If ACLs are used, then only *star* MUST be used to handle the files and directories as the other commands do not support ACLs. The options *-H=exustar -acl* must be used with *star*.

Please see the *star*(1) man page for more information.

## 6.7 Screen saver

The system is provided with the possibility to lock your terminal. To unlock the terminal, you MUST provide your password.

The locking is established using the `screen` application. Depending on the system configuration, `screen` MAY already be started during login. If the `screen` application is not started, you may start it manually.

The `screen` application allows the following two types of screen locking:

- Automated locking of the screen after a period of inactivity on the terminal defined by a timeout in either */etc/screenrc* or *~/.screenrc* using the *lockscreen* configuration value.
- Manual locking by executing the `C-a C-x` screen key binding combination.

You MAY change the timeout value for locking the session in *~/.screenrc* with the value for *lockscreen*. Note that the administrator MAY disable the ability to use the *~/.screenrc* configuration file.

**WARNING**: If a user accesses the system remotely and the screen saver functionality kicks in, the TOE ensures that the session is locked. However, it is possible that the remote terminal implements a scroll-back buffer that is not under the control of the TOE. Therefore, it is possible that the remote terminal has the session locked but a user can scroll back and list the history of actions. If the user shall not be able to use the scroll back buffer of the remote terminal, that terminal must be configured accordingly as this buffer is not under the control of the TOE. The local scroll back buffer is disabled.

```
no-scroll fbcon=scrollback:0
```

If `screen` is not invoked automatically during startup, you MAY enter the following line to *~/.bash_profile*.

```
exec screen
```

# Chapter 7

# Appendix

## 7.1 Online Documentation

If there are conflicting recommendations in this guide and in one of the sources listed here, the Configuration Guide has precedence concerning the evaluated configuration.

SUSE Linux Enterprise Server Guidance, https://documentation.suse.com/sles/15-SP7/