Common Criteria EAL4+ Evaluated Configuration Guide for SUSE LINUX Enterprise Server 11 SP2

February 19, 2013; v1.1

Contents

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. 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 the one of the following hardware systems:

SLES with virtualization support

IBM based on x86 64bit Intel Xeon processors

- IBM System x: x3400 M2, x3400 M3, x3500 M2, x3500 M3, x3550 M2, x3550 M3, x3620 M3, x3630 M3, x3650 M2, x3650 M3

IBM based on AMD Opteron processors

- IBM System x: x3755 M3

SLES on IBM System z

- IBM System z based on z/Architecture processors version 10

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 protocol of SSHv2 is 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).

Please be also aware that when configuring KVM guest system, KVM allows physical network adapters to be shared among different guest systems. The sharing is based on IP addresses where KVM assigns a unique IP address to each guest. You should consider the configuration of shared network adapters akin to the use of physical switches in your network. If you want to achieve a stronger separation between the virtual machines, each guest domain should be assigned a dedicated physical network interface.

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`.

The administrator MUST NOT read the contents of the file holding the seed information in */var/lib/misc/random-seed*. This ensures that if an administrator is relieved from his duties and therefore the trust relationship is lifted, the administrator is not able to obtain or apply (partial) knowledge of the state of */dev/random* or */dev/urandom*.

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 11 SP2, on one of the supported hardware platforms as defined in section §1.3.1 "Hardware requirements" of this guide.

On the platforms that support virtualization (VM) or secure logical partitioning (LPAR), other operating systems MAY be installed and active at the same time as the evaluated configuration. This is if (and only if) the VM or LPAR configuration ensures that the other operating systems cannot access data belonging to the evaluated configuration or otherwise interfere with its operation. Setting up this type of configuration is considered to be part of the operating environment and is not addressed in this guide.

On the other platforms, 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 §3.9 "Using serial terminals" of this guide), but *not* modems, ISDN cards, or other remote access terminals.

USB keyboards and mice MAY be attached, as some of the supported hardware platforms would otherwise not have supported console input devices. If a USB keyboard or mouse is used, it MUST be connected before booting the operating system, and NOT added later to a running system. Other hot-pluggable hardware that depends on the dynamic loading of kernel modules MUST NOT be attached. Examples of such unsupported hardware are USB and IEEE1394/FireWire peripherals other than mice and keyboards.

Note, the IBM Crypto Express cards MUST NOT be installed or assigned to the virtual machine. 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 is 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 11 SP2** applicable for the chosen hardware system. Note that you MUST use the 64bit version marked as *x86_64* for the Intel based hardware.

- A static IP address if you are intending to attach the target system to a network; the evaluated configuration does not support DHCP. In addition, you will need to configure the netmask, gateway, and DNS server list manually.

- An Internet-connected system equipped with the *rpm* and *rpm2cpio* package management tools. This system does not need to be in the evaluated configuration, and no packages will be installed on it. It is used to download and verify the installation packages.

- 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 SLES ISO images. The methods include the storing of the ISO images on a DVD-R and booting from it, using TFTP and NFS, using HTTP-based distribution of images, etc. The possible installation methods are explained in the SLES 11 deployment guide provided at http://www.suse.com/documentation/sles11/.

  After obtaining the ISO images, you MUST perform the integrity verification of the downloaded files as outlined on the SUSE security web sites http://www.suse.com/security/download-verification.html.

- A method to make the AutoYaST installation XML file accessible to the YaST installer. A retrieval method for the AutoYaST XML file as outlined in the AutoYaST installation manual http://users.suse.com/~ug/autoyast_doc/ section 7.3 to allow the SLES installer YaST to retrieve the XML file.

- If additional packages are listed in section §2.2.2 "Obtaining of installation images", ensure that these packages are accessible via an HTTP or FTP server or via storage media that can be locally mounted, such as a CD-R or a USB stick.

2.2.2 Obtaining of installation images

You MUST download the distribution ISO images from the SUSE web site on a separate Internet-connected computer, and either burn DVD-Rs from them, or make the contents available as outlined in the SLES 11 deployment guide provided at http://www.suse.com/documentation/sles11/. The download location http://www.suse.com/products/server/ contains links to the platform-specific images.

You MUST use **SUSE Linux Enterprise Server 11 Service Pack 2**. 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 image files are correct. The checksums are shown on the SUSE Security Web Page at http://www.suse.com/security/download-verification.html. Perform the steps outlined at this web page, including the signature check of the hash sums.

You MUST download several additional packages not included in the ISO images to set up the evaluated configuration. These packages are available from the SUSE web site (the URL is spanned across two lines)

```
http://ftp.suse.com/pub/projects/security/CommonCriteria/
        OSPP-EAL4+-SLES11SP2/packages/
```

You MUST obtain the following packages:

- autoyast-eal4-config-sles11sp2
- libopenssl0_9_8-*.rpm and openssl-0.9.8j*.rpm for your architecture
- kernel-default-*.rpm for your architecture

The precise versions of the additional RPMs needed are listed in the AutoYaST XML file distributed with the *autoyast-eal4-config-sles11sp2* RPM package. If additional packages are needed, the AutoYaST XML file contains the listing of the packages in the variable *RPMS*.

2.2.3 Extraction of AutoYaST XML file

The file needed to initiate the automated installation of SLES is the AutoYaST XML file distributed with the *autoyast-eal4-config-sles11sp2* RPM package. This package contains one AutoYaST XML file per supported hardware platform.

Download the RPMs using a separate Internet-connected computer. Do NOT install the downloaded packages yet.

You MUST have the SUSE package signing key available to verify the integrity of the additional RPM packages. It is available at the root of first downloaded ISO image. A number of PGP are provided where the keys are named as *gpg-pubkey-xxxxxxxx-xxxxxxxx.asc* where "xxxxxxxx" are hexadecimal numbers.

To verify the integrity and authenticity of the *autoyast-eal4-config-sles11sp2* RPM package, mount the first downloaded ISO image on the download system. The following commands assume that the ISO image is mounted at */mnt*. Now run the following commands to verify the package integrity:

```
rpm --import /mnt/gpg-pubkey-307e3d54-4be01a65.asc
rpm -v --checksig autoyast-eal4-config-sles11sp2*.rpm
```

This MUST display the status for "Header V3 RSA/SHA256 Signature" and "V3 RSA/SHA256 Signature" as "OK". If it does not, you MUST NOT proceed with the installation using that file.

The web page http://support.novell.com/techcenter/sdb/en/2005/05/rdassen_rpm_lost_keys.html provides additional information about the verification of the integrity and authenticity of RPM packages.

As a next step, on the download system, unpack the contents of the *autoyast-eal4-config-sles11sp2* RPM into a temporary directory to obtain the AutoYaST XML file required for installation:

```
mkdir cc-install
cd cc-install
rpm2cpio ../autoyast-eal4-config-sles11sp2-*.rpm | cpio -id
```

This will create the following directory structure in the current working directory:

Guidance documents

The Evaluated Configuration Guide – i.e. this document – together with supporting documents are provided in

```
./usr/share/doc/autoyast-eal4-config-sles11sp2*/
```

AutoYaST XML files

The AutoYaST XML files are provided with

```
./usr/share/autoyast-cc/profile/*.xml
```

The RPM package distributes one XML file for each supported hardware platform.

The AutoYaST XML files are named as follows:

cc-x86_64.xml
Intel Xeon, and AMD Opteron systems

cc-s390x.xml
IBM System Z

Scripts and configuration files

The RPM package also contains the configuration files and scripts as separate files which are embedded into the AutoYaST XML files. These scripts and configuration files are provided for informational purposes only and are not needed for the installation procedure – only the AutoYaST XML file is required for the automated installation process. The scripts and configuration files are stored in the following directories:

```
./usr/share/autoyast-cc/profile/config/
./usr/share/autoyast-cc/profile/scripts/
```

2.2.4 Establish AutoYaST installation environment

The extracted AutoYaST XML file now must be delivered to the YaST installer which installs SLES onto the target.

A large variety of methods are available to start YaST together with the AutoYaST XML file and initiate the automated installation. A retrieval method for the AutoYaST XML file as outlined in the AutoYaST installation manual http://users.suse.com/~ug/autoyast_doc/ section 7.3 to allow the SLES installer YaST to retrieve the XML file.

The following incomplete listing show examples how the AutoYaST XML file is loaded with the YaST installer. You MAY choose *one* of the examples listed here or refer to the AutoYaST installation manual for more extensive methods to invoke YaST with the AutoYaST XML file.

All of the following commands must be supplied at the boot command prompt that appears when booting from the ISO image. The used file name of *cc-x86_64.xml* must be replaced with the AutoYaST XML file name appropriate for your hardware system as mentioned above.

- If the AutoYaST XML file is reachable at an HTTP web server, use the following command

  ```
  autoyast=http://[user:password@]<server>/<path>/cc-x86_64.xml
  ```

- If the AutoYaST XML file is accessible via an NFS server, use the following command

  ```
  autoyast=nfs://<server>/<path>/cc-x86_64.xml
  ```

- If the AutoYaST XML file is made available via an FTP server, use the following command

  ```
  autoyast=ftp://[user:password@]<server>/<path>/cc-x86_64.xml
  ```

- If the AutoYaST XML file is delivered on an USB stick, use the following command where YaST searches on all USB devices it can find

  ```
  autoyast=usb://<path>/cc-x86_64.xml
  ```

- If the AutoYaST XML file is provided on a generic storage device, use the following command where "<device>" points to the device file name without the "/dev/" path name (e.g. *sda1* instead of */dev/sda1*)

  ```
  autoyast=device://<device>/<path>/cc-x86_64.xml
  ```

  YaST can be instructed to search all accessible devices by not specifying the device name (note the three slashes)

  ```
  autoyast=device:///<path>/cc-x86_64.xml
  ```

After ensuring that the AutoYaST XML file can be loaded by YaST, the final step for preparation of the installation environment must be performed: establish a means to boot either from the ISO image or the network boot system delivered on the ISO image and make the contents of the ISO image available to YaST.

The easiest method is to burn the ISO image onto a DVD-R and boot off that DVD-R. It is also possible to boot via TFTP. The contents of the ISO image can be provided either with the aforementioned DVD-R, via HTTP, FTP, NFS or other means. The possible installation methods are explained in the SLES 11 deployment guide provided at http://www.suse.com/documentation/sles11/.

2.2.5 Package selection

The AutoYaST XML file contains the list of RPM packages that form the TOE. The package listing can be found inside the XML tag of "<software>". The list of packages is partitioned into different package sets. Each package set is described an marked as either "Mandatory", "Default", "Optional", or "Prohibited" where these verdicts are documented in a comment with the package listing. You MAY modify the package listing according to these verdicts.

2.2.6 IBM System Z installation

The installation on IBM System Z may involve some considerations on how to access the installation system. Usually, the IBM System Z machines are headless systems where the z/VM console is accessed remotely.

To boot the IBM System Z, SLES allows numerous kernel command line parameters to be supplied. These parameters are documented in the general SLES guidance for the s390x architecture. The following example listing may be used as a guidance to set up the boot parameters:

```
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
hostip=1.2.3.4 netmask=255.255.255.0 gateway=1.2.3.1
nameserver=1.2.3.2 install=http://1.2.3.3/SLES11SP2
InstNetDev=osa Layer2=0 autoyast=http://1.2.3.3/autoyast.xml
OSAInterface=qdio OSAMedium=eth PortNo=1 ReadChannel=0.0.F404
WriteChannel=0.0.F405 DataChannel=0.0.F406
```

The parameters given in the first line should not be changed. The parameters in the second and third line specify the network configuration of the current system. In addition, the remote archive hosting the SLES RPMs is provided. The parameters of *InstNetDev*, *Layer2* from the fourth line as well as the fifth and sixth line should be known to the z/VM administrator as they specify the network device as well as the disk device channels to be used by the installation system. In case parameters are not provided but needed by the installation image, YaST will ask for the information before the installation proceeds.

When booting the installation image, the administrator is asked how YaST is supposed to be accessed. Various options are possible. 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.

After booting and selecting the access method for installation, the boot image waits for the administrator to log in and start "yast". This additional step is different from the installation with VNC, which may run through unattended.

The following methods must be used to start up YaST:

Initial startup

After booting the installation system for the first time, you only need to invoke `yast` on the command line to start up YaST. YaST will automatically invoke the auto-installation process based on the AutoYaST XML file provided the boot parameter outlined in section §2.2.4 "Establish AutoYaST installation environment" are present.

The auto-installation process displays a warning about missing disks during the initial startup of YaST. Please ignore this warning.

Post-installation startup

The installation process described in section §2.2.7 "Installation process" explains that after the initial installation of the base system, a reboot is performed. This first reboot shall launch the post-installation phase of the auto-installation process. On IBM System Z, you again must start up YaST manually after logging in. However, this time YaST has to be started with the following command:

```
/usr/lib/YaST2/startup/YaST2.ssh
```

YaST Firstboot

When the YaST post-installation phase is completed and YaST displays that the *Firstboot* sequence is to be started, invoke the firstboot sequence with:

```
firstboot
```

Please note that the installation process requires a reboot after the firstboot phase is completed. On IBM System Z, this reboot MUST be performed manually.

After starting YaST with the listed commands, the installation process follows the steps outlined in section §2.2.7 "Installation process".

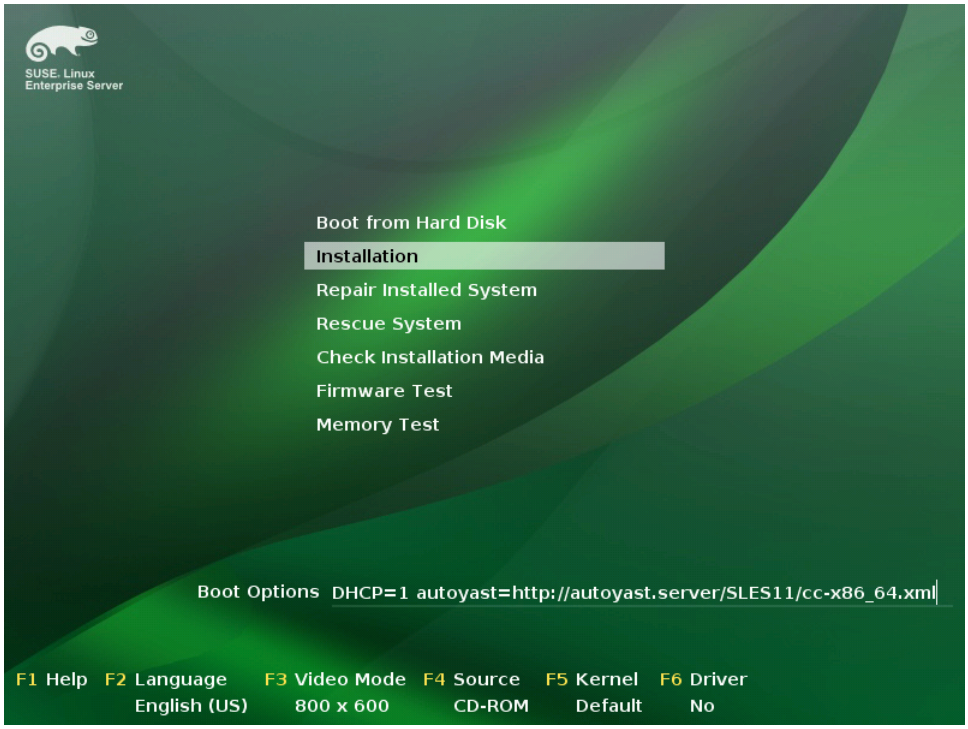
2.2.7 Installation process

After establishing the installation environment, YaST together with the AutoYaST XML file MUST be booted to initiate the automated 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, including the use of an NCURSES YaST interface depending on your hardware configuration.

The following illustration shows an example of how to boot with the AutoYaST XML file by booting directly from the SLES DVD. If a different boot mechanism is used, this illustration may not be applicable.

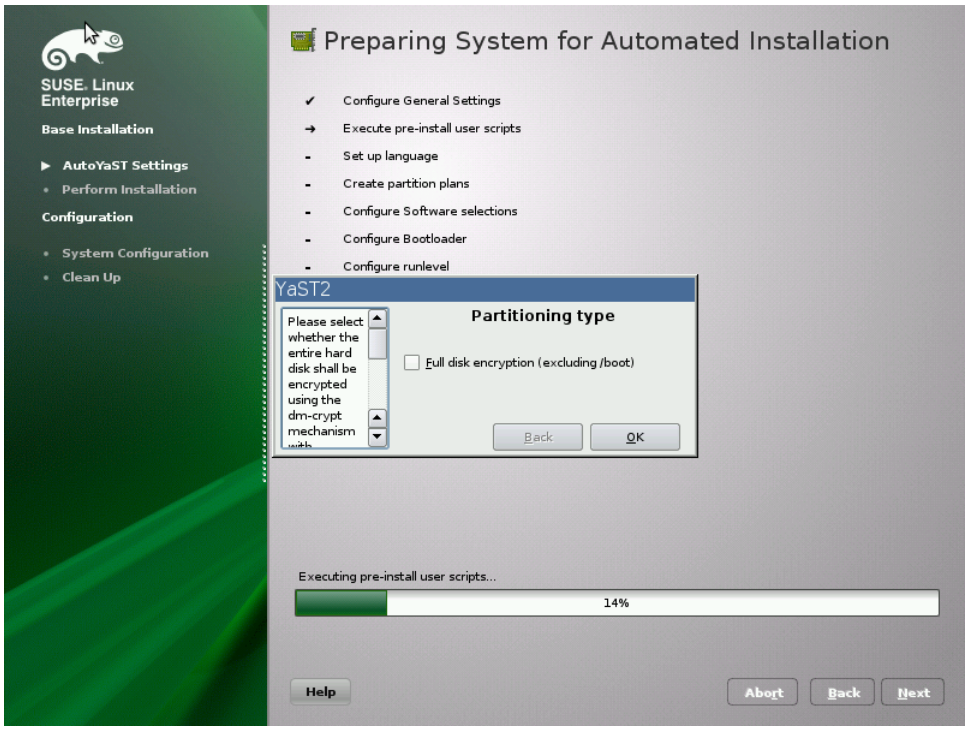


When the installer asks for the installation language, you MUST select "English" as this is the only language supported for the evaluated configuration.

In case an IBM System Z is installed, YaST now asks for the DASD device(s) to be used for the installation system. Please select the proper DASD device channel and activate it by using the "Perform Action" button. If needed, you may also request a low-level format of the DASD device.

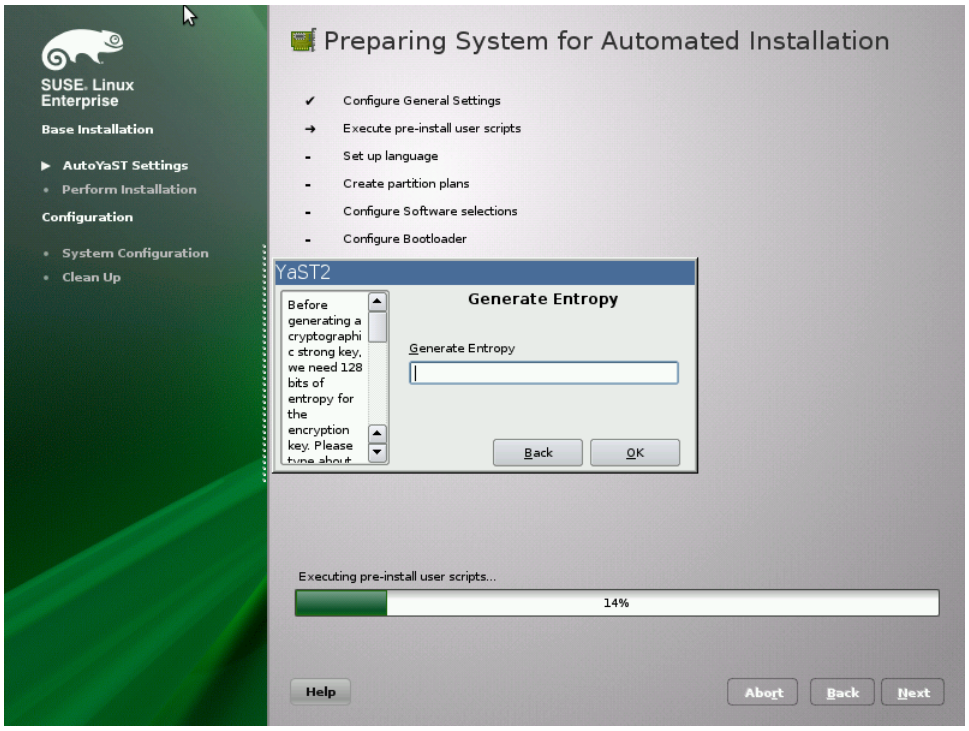
The first question the automated installation process asks is whether the the target system shall be installed on a fully encrypted disk. The encryption mechanism used is *dm-crypt* and all partitions including the SWAP partition except */boot* are located inside the *dm-crypt* container. The administrator MAY create *dm-crypt* containers in a running system. However, the full-disk encryption scheme can only be set up at this point.

Please note that full disk encryption is not supported on IBM System Z in the evaluated configuration. The following questions around full disk encryption are therefore not available on IBM System Z.

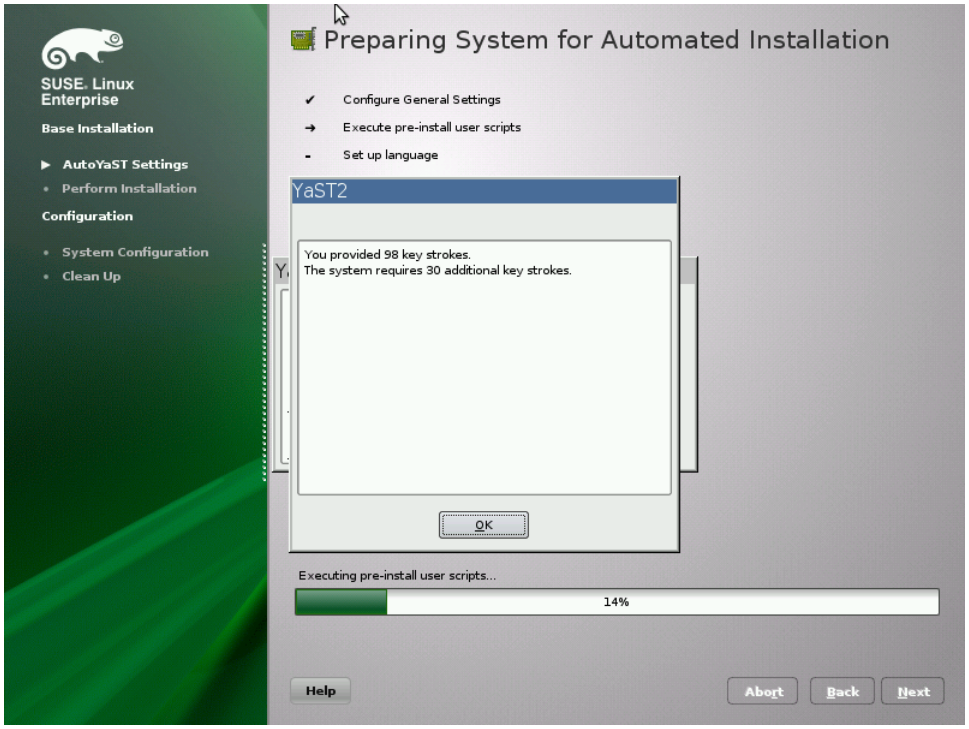


If the full-disk encryption configuration is selected, the installer asks for operations to increase the entropy in the system for generating a strong encryption key and for a passphrase to protect the encryption key.

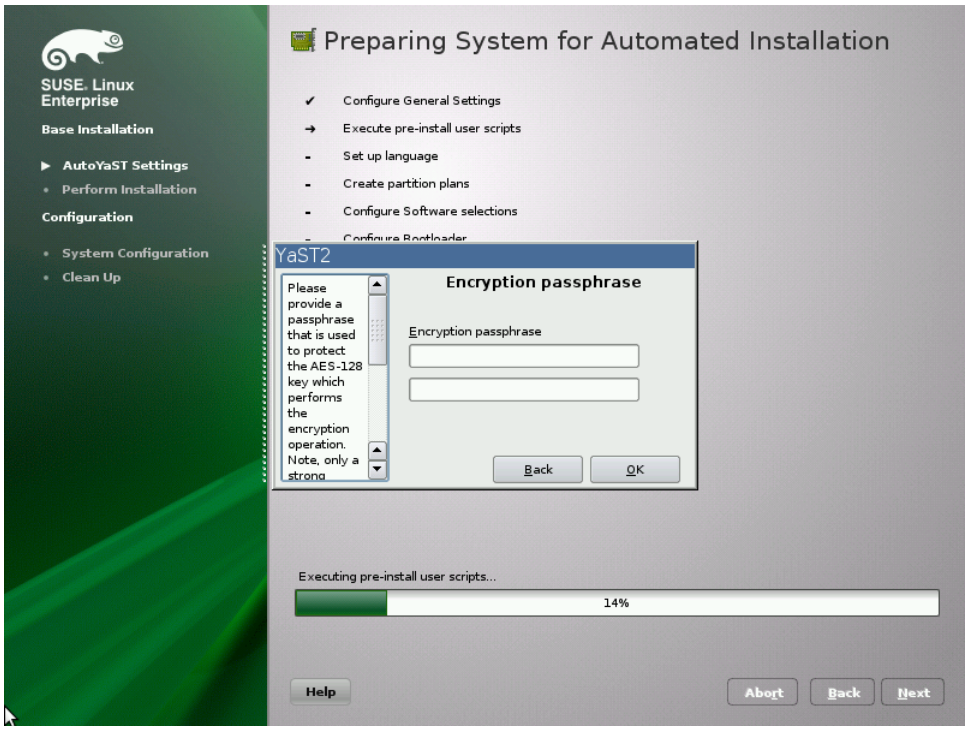
When YaST asks for topping off the entropy available in the system, you are requested to hit a number of keys. WARNING: you MUST NOT hit one key and hold it down as this does not add sufficient entropy. Also, if you perform the installation by accessing the system remotely (e.g. via serial console or network), you MUST access the physical console of the system to generate 128 bit of entropy (e.g. 128 key strokes, some mouse movements, etc.). The following figure illustrates the question:



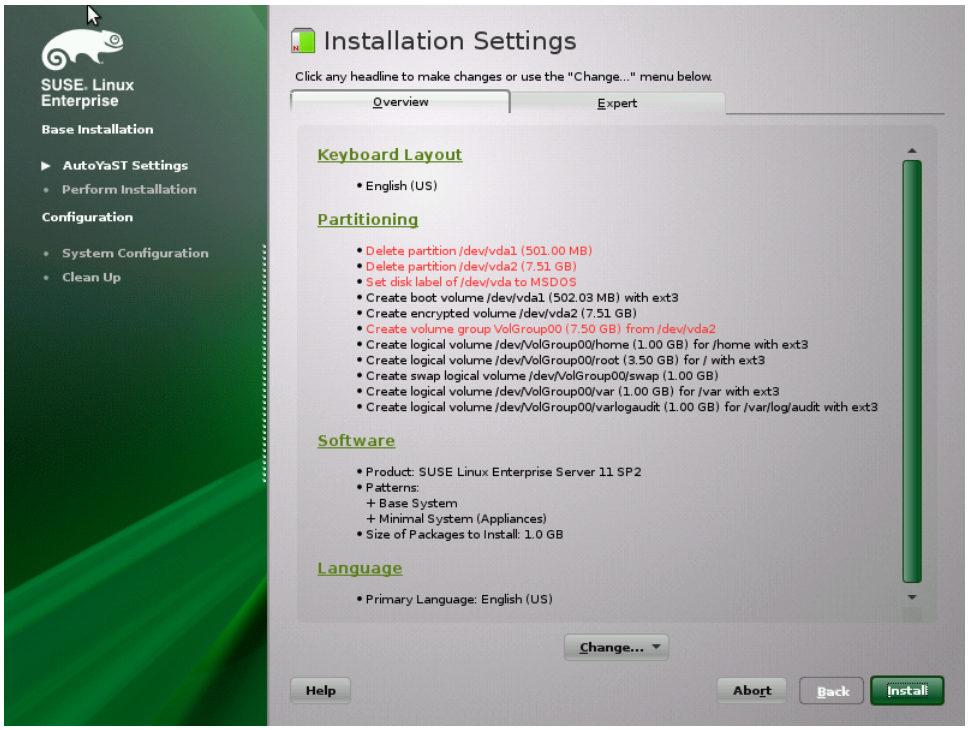
YaST will show you an information display when the number of key strokes is not yet sufficient as shown in this graphic.



After obtaining sufficient entropy, the passphrase is asked for. This passphrase is asked for during every reboot. You MUST choose a well-suited passphrase to protect the symmetric key used to encrypt data on disk.



As a last step before the installation commences, the YaST installer provides an overview of the installation actions.



You MAY modify the following settings at this point:

- Keyboard layout to suit your layout.

- 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*. In addition you MAY change the used filesystems to either EXT3, 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.

  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.

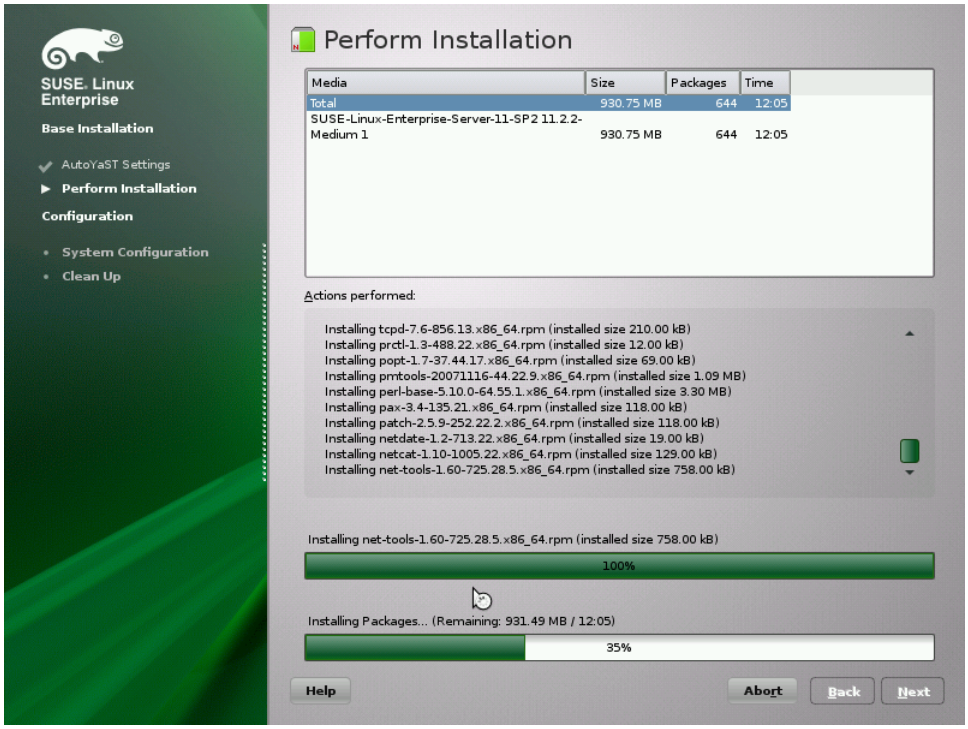
- You MAY modify the boot loader options such as setting a password. However, you MUST NOT modify the boot loader type, the boot loader location and the boot loader section of "Linux - CC evaluated configuration" which must be the default selection.

- Time zone applicable to the system.

- You MAY enable kdump.

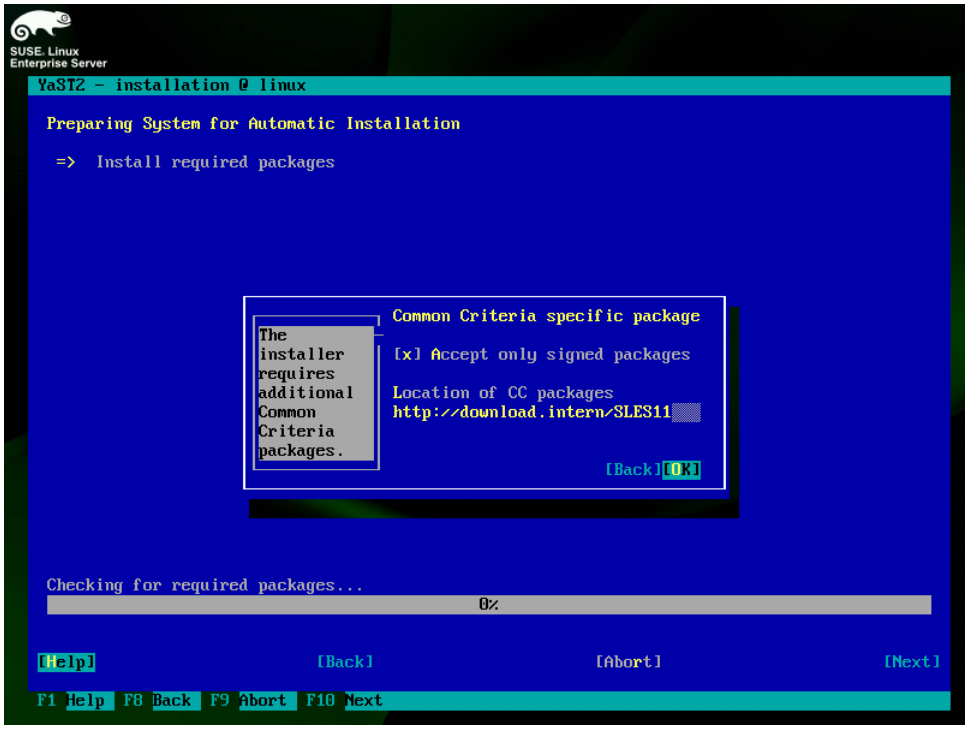
You MUST NOT modify any other settings.

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.

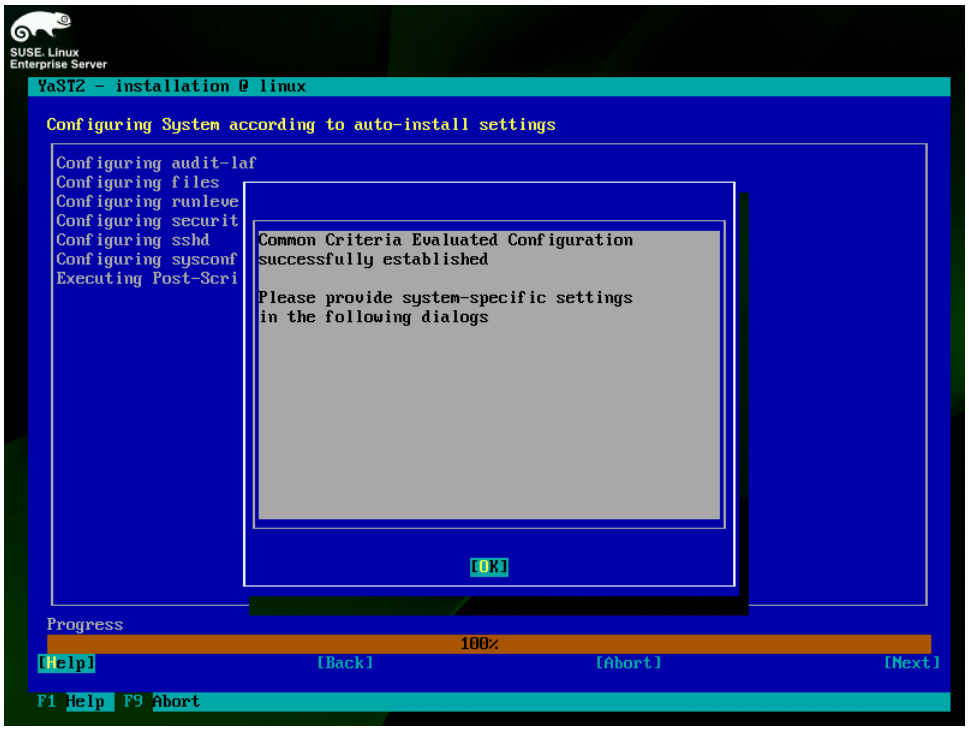


After the installation process completes, the system reboots into the post-installation phase. The first step of the post-installation phase is the verification whether additional update packages are needed. If additional packages are needed a dialog is displayed asking for the location of these packages and whether they must be signed. Note, for the evaluated configuration, the check box to verify the signature MUST be selected. YaST displays an error if the signatures cannot be verified and re-displays the dialog. The location of the additional packages can only be specified as either "http", "ftp", or "file". You MAY change into a second terminal by pressing ALT+F2 and perform the necessary operations to make the packages available via HTTP, FTP or on the local file system. For example, you MAY mount a CD, USB stick or an NFS server that holds the packages.

If no updated packages are needed, this dialog is skipped automatically.



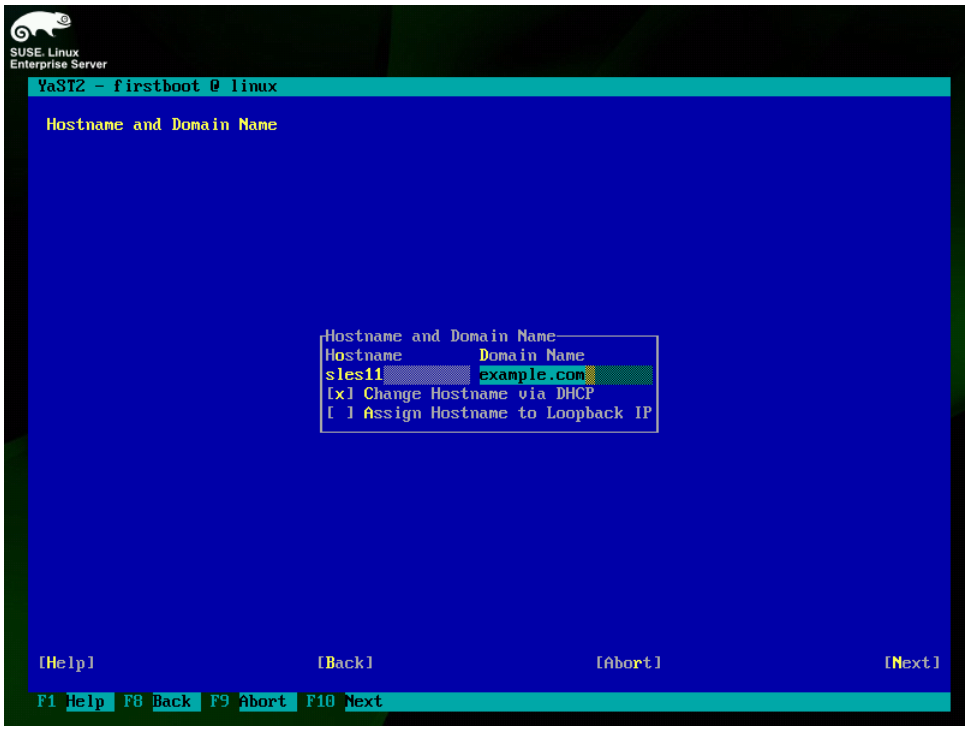
As a next step, the system is automatically configured. Intervention by the administrator is not needed. When concluding the automated configuration phase, the result of the configuration is displayed.



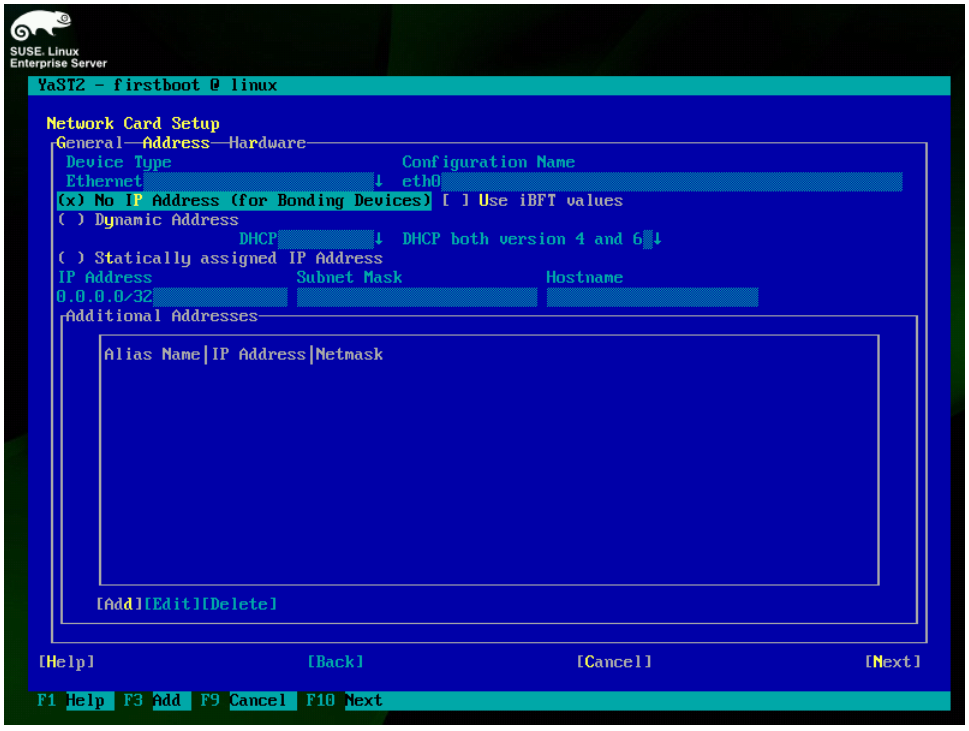
You MUST ensure that the configuration result shows a successful configuration. If at least one step in the configuration fails, the result shows an error. You MUST NOT proceed if an error is displayed. You MAY analyze the the automated configuration operation by reviewing the configuration log at */var/log/YaST/cc-scripts.log*. Search for *non-recoverable ERROR* to identify the areas of failures. Once the issue is identified and resolved, you MAY re-perform the entire installation and configuration process. You MUST NOT proceed at the current configuration state as the system is in an unknown state at this time.

The last configuration steps require administrator intervention and cover configuration options specific to the system. YaST first asks for the acceptance of the end user license agreement. You MUST accept the license or you cannot proceed with the configuration and the use of the system.

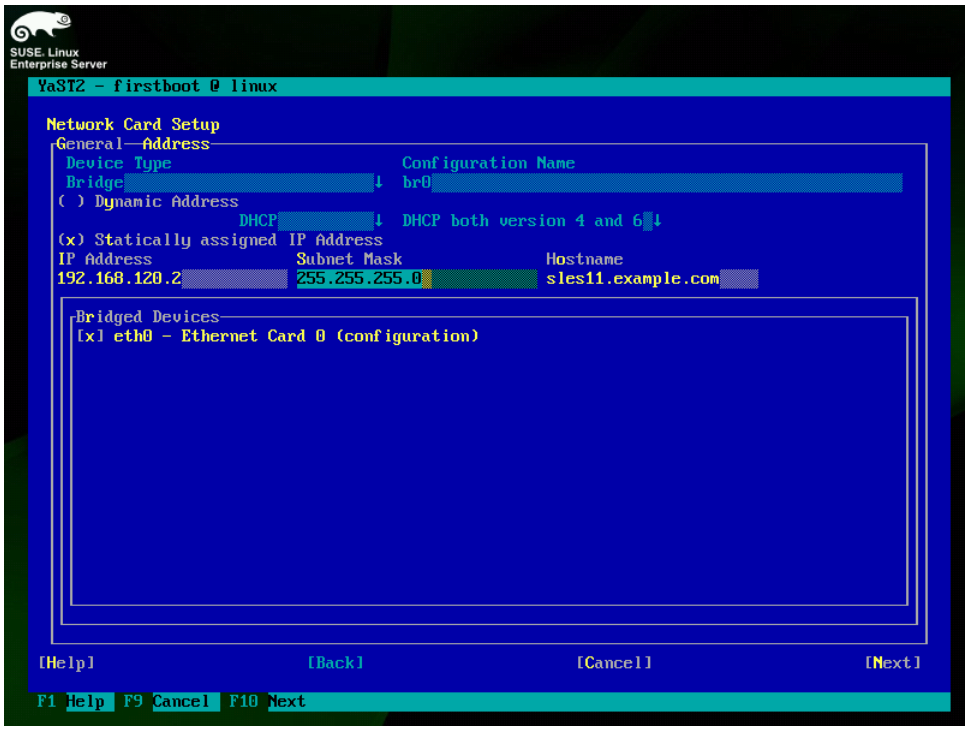
Following the license acceptance, YaST ask for the host name of the system.



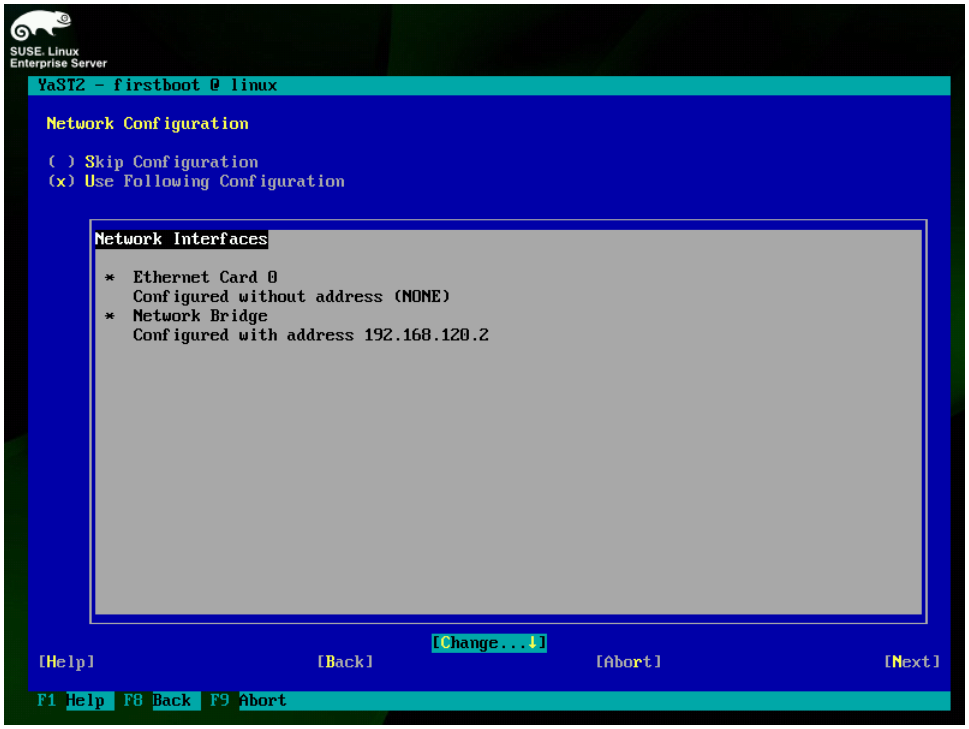
In a next step, the network configuration can be provided. You MUST use a static IP address for the system. If the installation includes KVM (only available on x86_64), you MAY configure the bridge or bridges allowed for the virtual machines. In this configuration, you edit the network device that shall be bound to the bridge and select "No IP Address (for Bonding Devices)". The following illustrations show the example of configuring a bridge for KVM.



In addition, the bridge now is configured with a static IP address and marked to be bound to the network interface that is left without an IP address.

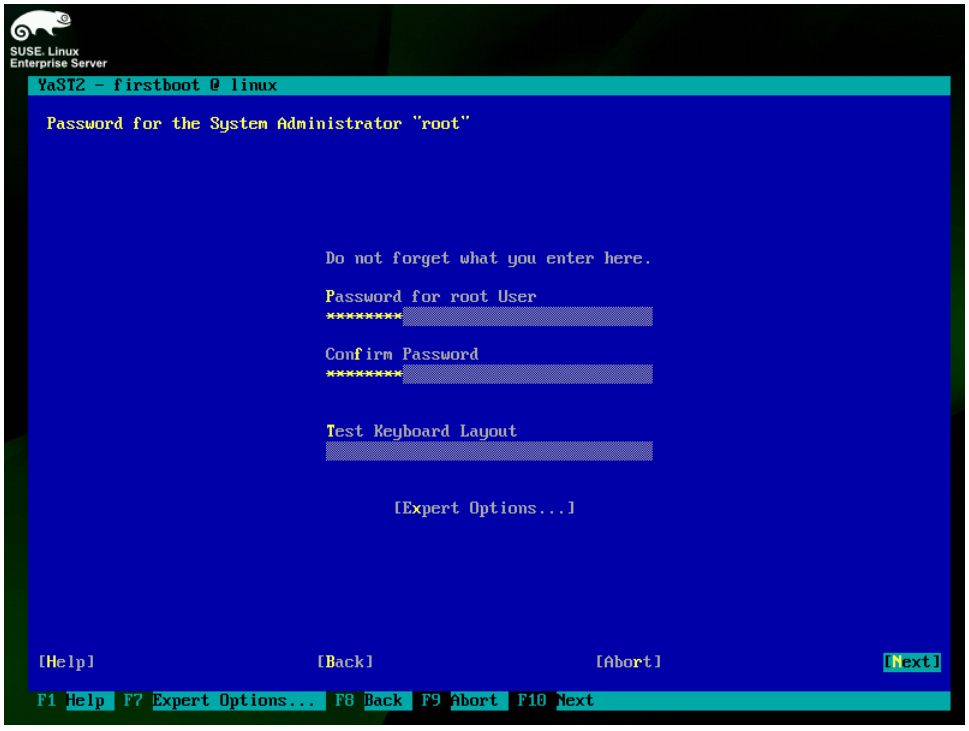


The summary of the network configuration with a bridge device for virtual machines may look similar to the following illustration.



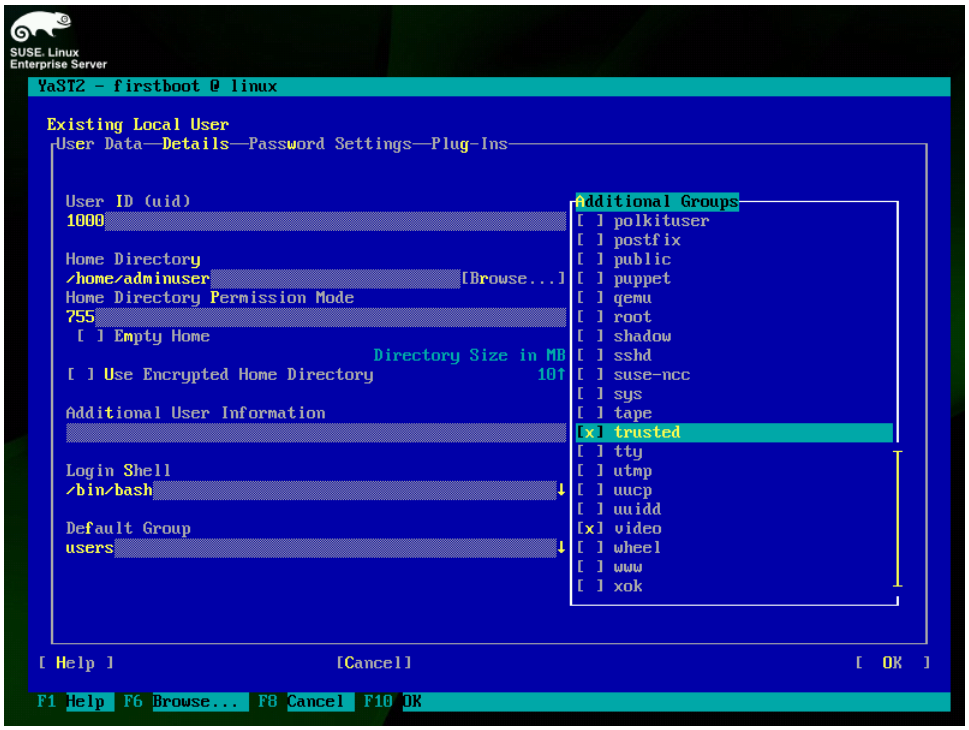
YaST now asks for the time zone applicable for the system. You MAY select the appropriate time zone. In addition, you MAY configure an NTP server at this point by selecting "Change" in the box "Date and Time".

In a next step, you MUST provide a password for the root user. This password is subject to the password quality constraints required by the Security Target. You MAY change the "Password Encryption" scheme by selecting "Expert Options".

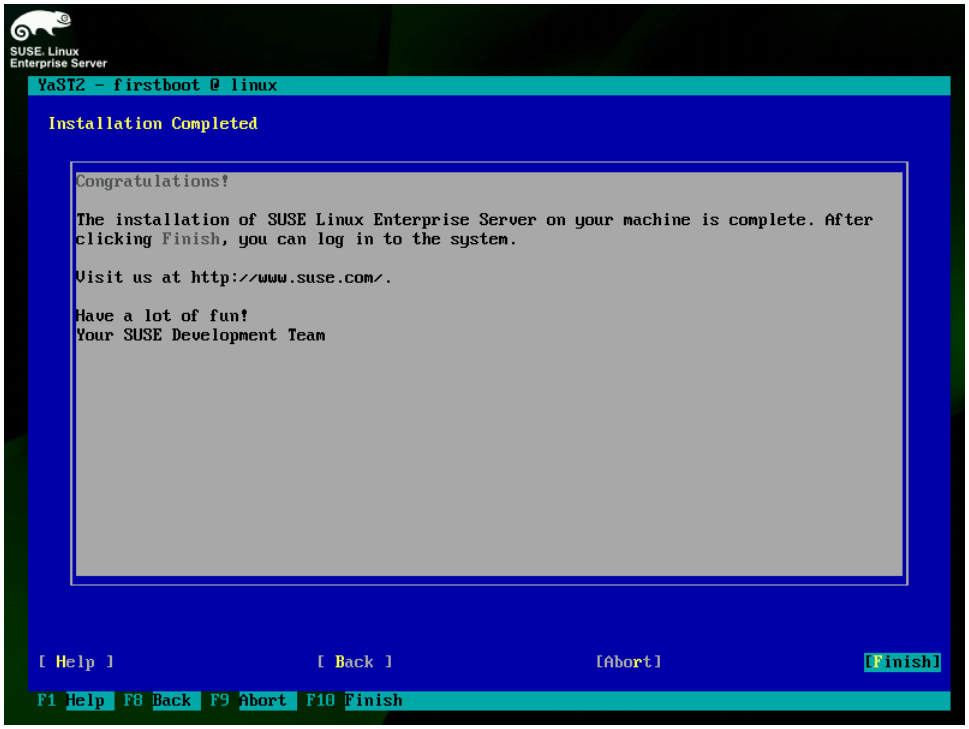


Finally, you MUST specify at least one user account for an administrative user. Administrators can only log on as root on the local console. If administration shall be performed via SSH, administrators first must log on with their regular

user ID and then use either `su` or `sudo` to acquire root privileges. The difference between an administrative user and a regular unprivileged user is that administrative users are assigned to the *trusted* group. See section §3.3 "Gaining administrative access" for more details.



The system is now fully configured as displayed by the summary.



After clicking "Finish", the system reboots to load the evaluated configuration.

Note, IBM System Z does not perform the reboot automatically. Therefore, you MUST reboot the system after finishing.

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, on systems using the *grub* boot loader you can add the following command to the boot prompt to launch a shell directly from the kernel, bypassing the normal init/login mechanism:

```
init=/bin/sh
```

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 *star*(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 tapes or CD-R, or on a separate host) are RECOMMENDED:

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

You MUST use the `-H=exustar -acl` option for *star* 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 §5.2 "System logging and accounting" and §5.3 "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 *stunnel* servers, as well as the */etc/shadow* password database. Store the backup media at least as securely as the server itself.

A RECOMMENDED method to track changes is to use a version control system. RCS is easy to set up because it does not require setting up a central repository for the changes, and you can use shell scripting to automate the change tracking. RCS is not included in the evaluated configuration, see *rcsintro*(1) in the rcs RPM package for more information. Alternatively, you can create manually create backup copies of the files and/or copy them to other servers using *scp*(1).

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:

```
/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 or SGID root programs 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:

  - Entries in */etc/inittab*
  - Executable files or links in */etc/init.d/* and its subdirectories
  - Entries in */etc/xinetd.conf*
  - 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 §2.2.7 "Installation process" are created on it.
- The file systems 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 CD-ROMs or DVDs.

CD/DVD 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 §2.2.7 "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.

When using cryptsetup manually, you MUST use the LUKS extension and therefore the LUKS commands specified in *cryptsetup*(8).

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

`cryptsetup` obtain the key material by reading data out of */dev/urandom* XORed with 13 bytes of */dev/random*. This implementation ensures that `cryptsetup` generates the key used for the cryptographic operations out of a random number generator seeded by */dev/random* to ensure cryptographically strong keys. Note that the key generation process may stall while reading from */dev/random* in case insufficient entropy is present.

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:

Allowed ciphers:

- AES (128 bits, 192 bits, 256 bits)
- Serpent (128 bits, 192 bits, 256 bits)
- Twofish (128 bits, 192 bits, 256 bits)

Allowed block chaining modes:

- CBC

Allowed IV-handling mechanisms:

- CBC: ESSIV

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 Managing user accounts

3.8.1 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. Alteratively, 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
```

3.8.2 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.

3.8.3 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 DSA or RSA keys that can be used for key-based authentication, the tool *ssh-keygen*(8) is provided. As the SSH daemon only accepts SSH protocol version 2, only the protocol 2 keys are supported with the SSH daemon. Therefore, you SHOULD only use the option `-t rsa` or `-t dsa` when generating a key with *ssh-keygen*.

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

3.8.4 Changing user properties

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

3.8.5 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. 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.

3.8.6 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.

3.8.7 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
```

3.9 Using serial terminals

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

Serial terminals are activated by adding an entry in the file */etc/inittab* for each serial terminal that causes *init*(8) to launch an *agetty*(8) process to monitor the serial line. *agetty* runs *login*(1) to handle user authentication and set up the user's session.

If you use serial terminals and require the fail-safe audit mode, you MUST ensure that the file */etc/pam.d/login* is configured to use the *require_auditd* option for the *pam_loginuid.so* module in the `session` stack.

For example, adding the following line to */etc/inittab* activates a VT102-compatible serial terminal on serial port /dev/ttyS1, communicating at 19200 bits/s:

```
S1:3:respawn:/sbin/agetty 19200 ttyS1 vt102
```

The first field MUST be an unique identifier for the entry (typically the last characters of the device name). Please refer to the *agetty*(8) and *inittab*(5) man pages for further information about the format of entries.

You MUST reinitialize the *init* daemon after any changes to */etc/inittab* by running the following command:

```
init q
```

3.10 Managing data objects

3.10.1 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.

3.10.2 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.

3.10.3 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.

3.11 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.

3.12 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 z/VM or 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.

3.13 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 the `iptables` command which uses kernel support for controlling traffic. Iptables can be used to set up very small and lean packet filter rules. On the other hand, very complex and sophisiticated filtering rules can be configured as well to suit the need of the administrator. An elaborate introduction is given in the SUSE Linux Enterprise Server Security Guide section 15.1 accessible at http://www.suse.com/documentation/sles11/. In addition, *iptables*(8) discusses 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 travelling 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.

3.13.1 iptables auditing of packet filter operations

Iptables is extended by an *AUDIT* target that conceptually works similar to the *LOG* target documented in *iptables*(8).

This target allows the logging of matching packets with the audit facility. When this option is set for a rule, the Linux kernel will generate an audit entry for each matching packet which holds details about the packet depending on the protocol of the packet.

See *iptables*(8) for more information.

3.13.2 ebtables auditing of packet filter operations

Similarly to iptables, ebtables is extended by an *AUDIT* watcher. The implementation of the *AUDIT* watcher for ebtables is identical to the *AUDIT* target of the iptables mechanism.

See *ebtables*(8) for more information.

3.14 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/autoyast-cc/profile/scripts/screen` out of the *autoyast-eal4-config-sles11sp2*.rpm* RPM. 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 *lockscreen* configuration value.
- 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*.

The screen locking and unlocking is provided by the `/usr/bin/vlock` which MUST be invoked with screen as otherwise the `screen` cannot be unlocked. `screen` is informed about the lock program using the *LOCKPRG* environment variable. This environment variable MUST be set to `/usr/bin/vlock` for every invocation of screen.

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 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*.

```
LOCKPRG="/usr/bin/vlock"
export LOCKPRG
exec screen
```

3.15 OpenSSH 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 48 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 */dev/random*. The seeding process is described in the configuration file */etc/sysconfig/sshd*.

Using */dev/random* in the evaluated configuration deviates from the default behavior of the OpenSSH applications which per default employs */dev/urandom*. The difference is that */dev/random* blocks until sufficient entropy is available to satisfy the request for entropy. This implies that the OpenSSH applications block when they seed or re-seed until the requested amount of entropy is delivered.

The blocking may cause irritation by a user as the application seemingly stalls without any notification. The administrator is advised to inform the user base about this blocking behavior.

The blocking behavior is controlled with the environment variable *SSH_USE_STRONG_RNG* which must be set according to the following rules:

- For the `sshd`, the environment variable must be set in */etc/sysconfig/sshd*. After setting, the `sshd` daemon must be restarted using the start script.
- For applications invoked by users, the file */etc/profile/cc-configuration.*sh* contains the setting of this environment variable for Bourne shell and derivatives as well as C-shells and its derivatives.

In addition to the use of the proper random number generator, the AES-NI support provided with OpenSSL must be deconfigured for the OpenSSH applications. The AES-NI support was not subject to this current evaluation and can therefore not be used to encrypt the communication channel. The AES-NI support is disabled for OpenSSL by setting the *OPENSSL_DISABLE_AESNI* environment variable. The evaluated configuration already sets this environment variable in */etc/sysconfig/sshd* for the `sshd` daemon. Also, the environment variable is set in */etc/profile.d/cc-configuration.*sh*.

Chapter 4

Kernel-based virtual machine (KVM) management

The TOE provides virtual machine environments which allow other operating systems to execute concurrently with the host system. Each virtual machine is represented as a process in the host system and is subject to the standard Linux constraints for processes. The virtualization and simulation logic that supports the operation of a virtual machine executes within the same process of the respective virtual machine. When operating virtual machines, the host kernel acts as the hypervisor for the guest systems.

Treating each virtual machine as a normal process by the host system allows the concurrent execution of standard applications at the same time. Administrative tools are implemented as standard Linux applications. In addition, the computing environment provided by the host system to users logging into that system does not differ from a standard system even while virtual machines execute. Therefore, standard Linux applications and daemons may operate concurrently with virtual machines. To access the host system, the TOE provides console access via OpenSSH as well as access via the virtual machine management daemon provided by `libvirtd` via OpenSSH. In addition, the virtual machine consoles can be accessed using VNC using an already established OpenSSH channel. The VNC server is implemented by the `QEMU` virtualization and simulation logic. The use of OpenSSH for accessing the `libvirtd` management daemon as well as VNC is encapsulated in different client applications which are not part of the TOE, like `virsh` and `virt-manager`.

When using SLES as a host system for virtual machines, different UIDs and GIDs are used to separate different users, services, or virtual machines provided by the system. In such a case, it is assumed that these processes are responsible for the safeguarding of their data. Note: the evaluated configuration permits the use of the host system for regular operation by normal users at the same time when virtual machines execute. It depends on the administrator-defined policies whether such a dual use is allowed.

Hosting virtual machines with guest operating systems increase the availability requirements on the host system. Without guest systems, a failing system renders the services offered by that system offline. However, a failing system with a number of guest systems terminates the services offered by the host and all hosted guest systems. The impact of such a discontinuity is increased by a several orders of magnitude. Therefore, you MUST plan the deployment of a host system for other operating systems carefully to prevent service discontinuities affecting the overall goals of your IT infrastructure.

The following sections discuss the configuration of virtual machines and their resources.

4.1 Hardware configuration

The hardware covered by this evaluation provides all support needed for KVM. However, the evaluation is limited to the execution of virtual machines to the *x86_64* architecture, i.e. Intel Xeon and AMD Opteron processors. The KVM

support offered for other systems, such as PowerPC or IBM System Z MUST NOT be used as they were not subject to evaluation.

You MUST ensure that the virtualization hardware mechanisms are enabled. It MAY be possible that they are disabled using BIOS settings. Please check your hardware manuals or hardware vendor to ensure that the following hardware support is enabled:

Intel x86_64 bit based systems

- The processor's Virtualization Technology extensions (VT-x) support must be enabled. If that support is enabled, the following command returns with a listing of processor capabilities:

```
cat /proc/cpuinfo | grep vmx
```

- The processor's Enhanced Page Tables (EPT) support must be enabled. If that support is enabled, the following command returns with a listing of processor capabilities:

```
cat /proc/cpuinfo | grep ept
```

AMD Opteron based systems

- The AMD-V processor's support for virtual machines must be enabled. If that support is enabled, the following command returns with a listing of processor capabilities:

```
cat /proc/cpuinfo | grep svm
```

- The processor's Nested Page Tables (NPT) support must be enabled. If that support is enabled, the following command returns with a listing of processor capabilities:

```
cat /proc/cpuinfo | grep npt
```

4.2 Kernel-based Virtual Machine (KVM) configuration

The evaluated configuration allows the administration of virtual machines. The management daemon of `libvirtd` as well as a virtual machine's console are accessed via SSH.

The `libvirtd` virtual machine management daemon allows the specification of virtual machines, and the assignment of resources to these virtual machines. Once a virtual machine is configured, this daemon also allows the starting and stopping of a virtual machine.

Another aspect of `libvirtd` is the configuration of the host system and its resources which are made available to virtual machines, including the configuration of networks and storage areas.

The functionality mentioned for `libvirtd` allows users to define and start virtual machines. However, access to the virtual machine's console is established separately. The virtual machines are instantiated as a Linux process using the `QEMU` virtualization support. `QEMU` emulates various devices for a virtual machine, including the console devices with keyboard, mouse and display. These simulated devices are translated into a VNC server that the owner of the virtual machine can connect to.

Access to the `libvirtd` management daemon and the VNC server instantiated for a virtual machine by `QEMU` is established via SSH. Virtual machine management client applications that are distributed by the host can be used to access both, the `libvirtd` management daemon and the VNC server. For example, the client applications of `virsh` or `virt-manager` allow the access of both components via SSH. Note that both client applications are not covered by the evaluation.

Users managing virtual machines and who are allowed to access the VNC servers MUST have a local account on the TOE and MUST be members of the *libvirt* group. These users do not need any other elevated privileges like root access. Before these users are able to access `libvirtd` and the VNC servers they must authenticate via SSH using their accounts. To access the `libvirtd` virtual machine management daemon remotely via SSH, use the following command:

```
virsh connect qemu+ssh://USER@HOST/system
```

Replace the strings USER with the user ID you want to authenticate with and HOST with the hostname or IP address of the system executing the `libvirtd` management daemon.

To access `libvirtd` locally on the TOE system, you MAY use the following command:

```
virsh connect qemu:///system
```

In this case, the `virsh` command directly accesses the Unix domain sockets provided by `libvirtd` in */var/run/libvirt/*.

To manage virtual machines and resource assignments, please see the *virsh*(1) man page. This command allows the administration of virtual machines, their resources, and maintaining the life cycle of virtual machines.

If virtual machine administrators accessing `libvirtd` from remote shall be restricted to accessing `libvirtd` only without the ability to invoke any command on the host, you MAY add the following options to the file *~/.ssh/authorized_keys* where all option shall be on one line preceding the applicable public key – see *sshd*(8) for more information about options to be provided in the *authorized_keys* file.

```
no-agent-forwarding,no-pty,no-port-forwarding,no-user-rc,no-X11-forwarding,
no-agent-forwarding,command="/usr/local/bin/libvirt-access.sh"
```

In addition to the configuration above, you must generate the file */usr/local/bin/libvirt-access.sh* with the permissions of 755 with the following contents:

```
#!/bin/bash
if (echo $SSH_ORIGINAL_COMMAND | grep -q "127")
then
  sh -c nc -q 2>&1 | grep "requires an argument" >/dev/null
  if [ $? -eq 0 ]
  then
    CMD="nc -q 0 127.0.0.1 5900"
  else
    CMD="nc 127.0.0.1 5900"
  fi
  eval "$CMD";
else
  sh -c nc -q 2>&1 | grep "requires an argument" >/dev/null
  if [ $? -eq 0 ]
  then
    CMD="nc -q 0 -U /var/run/libvirt/libvirt-sock"
  else
    CMD="nc -U /var/run/libvirt/libvirt-sock"
  fi
  eval "$CMD"
fi
```

These options ensure that only the tunnel to the `libvirtd` Unix Domain sockets can be opened without causing any other operation on the host.

Please note: this configuration prevents a virtual machine administrator from accessing the local console. However, a virtual machine administrator still is able to access any device if he desires to do so. For example, a virtual machine administrator may assign the entire hard disk holding the host system to a virtual machine. Now, the virtual machine administrator is able to read the entire contents of the host disk space using the virtual machine functionality, bypassing any DAC or other restrictions enforced by the host system.

4.2.1 Virtual machines started outside libvirtd

In the TOE version, only virtual machines instantiated with the `libvirtd` virtual machine management daemon are covered with the separation mechanisms that were subject to evaluation. Virtual machines MAY be started outside of `libvirtd` by accessing */dev/kvm* directly, e.g. by spawning *QEMU* via command line. These virtual machines have the same privileges on the host system as the user that started them.

Users may invoke the `QEMU` application with similar command line options as the `libvirtd` daemon. In this case, the virtualization environment starts and the guest operating system may become active. However, this startup does not provide any assurance evaluated with the CC evaluation. In particular, the following support for virtual machines is missing rendering the execution of such virtual machines less effective:

- The hardware virtualization support MAY not used as the Linux kernel interface allowing user space to utilize the hardware support MAY not be accessible to normal users. The device file */dev/kvm* provides access to the hardware support.
- The separation mechanism between multiple instances of virtual machines as discussed in §4.3.2 "AppArmor sVirt support" is not set up and enforced.

To ensure that all support mechanisms for the proper operation and separation of virtual machines are employed, virtual machines MUST be configured and managed using the `libvirtd` virtual machine management daemon.

4.2.2 System versus session instances of libvirtd

The `libvirtd` virtual machine management daemon has the capability to provide unprivileged users (i.e. users not part of the *libvirt* group) to manage virtual machines. In this scenario, a `libvirtd` instance executes with the user ID of the user. This operational mode is called *session* mode.

Contrary, the privileged instance of `libvirtd` started at boot time is known as *system* instance.

The operational mode that the connecting user wants to access is supplied with the URI when connecting to `libvirtd`. The URI listed in §4.2 "Kernel-based Virtual Machine (KVM) configuration" contains the string *system* and therefore connects to the privileged instance of `libvirtd`.

The virtual machines started from the *session* instance of the `libvirtd` virtual machine management daemon lack the same capabilities as virtual machines that are started completely outside the `libvirtd` daemon. Therefore, the capabilities listed in §4.2.1 "Virtual machines started outside libvirtd" are missing for virtual machines started from the *session* instance of `libvirtd` as well.

This guidance document and all described virtual machine capabilities apply only to virtual machines controlled by the *system* instance of `libvirtd`.

4.3 Separation of virtual machines

The `libvirtd` virtual machine management daemon ensures that the execution environment as well as the resources allocated to a specific virtual machine are separated from other virtual machine instances as well as the host system. Such a transparent setup aids the administrator in keeping a secure configuration for the host system as well as the virtual machines.

4.3.1 Unprivileged user and group IDs

The host system protects its operation against the virtual machines by executing them with a fully unprivileged user and group ID, the ID *qemu*. This user ID has equivalent rights to normal unprivileged users and can therefore not

harm the host system even if the guest operating system would be able to exploit any vulnerabilities in the hardware emulation code provided with the `QEMU` application. The `libvirtd` virtual machine management daemon ensures that the processes implementing the virtual machine environment are started with the *qemu* user and group ID. The specification of these IDs is given in */etc/libvirt/qemu.conf*.

The configuration option option that enables the startup of a virtual machine with the configured user ID and the modification of the ownership of the file system resources is found in */etc/libvirt/qemu.conf* by setting the configuration value *dynamic_ownership* to 1.

4.3.2 AppArmor sVirt support

In addition to the protection of the host system from the virtual machines, the host system uses another mechanism to ensure full separation between virtual machines. As all virtual machines execute with the same user ID and group ID it is possible in theory that these virtual machine processes may interfere with each other. The AppArmor policy template enforced with `libvirtd` implements a strict separation between virtual machines and its resources.

With the sVirt mechanism implemented with the `libvirtd` virtual machine management daemon, an AppArmor label is automatically generated. These sVirt-generated unique AppArmor labels are assigned to each virtual machine and its resources. In addtion, each AppArmor label is assigned with a separate AppArmor label that is derived from the file */etc/apparmor.d/libvirt/TEMPLATE* which is extended by another AppArmor policy file that specifies allow rules to file system objects granted to the respective virtual machine. The instantiated files are stored in */etc/apparmor.d/libvirt/<UUID*>* where UUID is the unique label that corresponds with the UUID of the virtual machine. The AppArmor policies prevent any access request by a virtual machine to a resource that is not specified in the two policy files applicable to the virtual machine. In the evaluated configuration, the unique AppArmor labels for each virtual machine is derived automatically by the `libvirtd` virtual machine management daemon during startup of the virtual machine.

The virtual machine process is labeled with its AppArmor label during invocation of the virtual machine by `libvirtd`.

More information about the dynamic labeling can be found at *http://libvirt.org/drvqemu.html#securitysvirt*.

4.3.3 Sharing of resources

The sVirt functionality discussed in section §4.3.2 "AppArmor sVirt support" ensures that the virtual machines are isolated from each other. This isolation covers the resources used by the virtual machines, such as USB or PCI devices assigned to virtual machines, disk backend files, etc. The AppArmor profiles only allow access to the configured devices referenced by file system objects – such as device files or regular files – and deny access to any other devices.

The virtual machine administrator MAY assign the same device file or disk backend file to multiple virtual machines. Such sharing of resources is desirable, for example, if one ISO image shall be assigned to multiple virtual machines as an IDE CD-ROM backend. For example, if multiple virtual machines executing SLES11 shall be installed, one SLES11 installation ISO image can be assigned to all these virtual machines at the same time. Other examples include high-availability configurations where a common shared disk device must be configured.

sVirt allows such an assignment of one resource to multiple virtual machines as a requested breach of the isolation policy. Virtual machine administrators MUST anticipate that the virtual machine isolation does not apply to these shared resources.

If one virtual machine shall be fully isolated from other virtual machines, the virtual machine administrator MUST ensure that all device and disk backend resources assigned to this fully isolated virtual machine are not shared with other virtual machines.

Please note, disk backend files MAY be assigned to virtual machines in a read-only mode. In this case, the AppArmor profile only allows read access and denies write access to the disk backend file. Such read-only mode is typically used for ISO images that are configured as CD-ROM devices for virtual machines.

4.4 Networking considerations

Virtual machines MAY be configured with one or more network interfaces. The `libvirtd` virtual machine management daemon allows the configuration of the networking capabilities for virtual machines.

Two aspects MUST be configured to establish the network connectivity for virtual machines:

1. The `libvirtd` MUST be configured with one or more networks. Each network configuration results in an automated setup of a bridge network interface on the host system. That bridge interface is assigned with a physical network interface on the host. To achieve separation between these networks, each network MUST be assigned to an individual physical network interface.
2. Each virtual machine network configuration must be assigned to belong to one network defined for `libvirtd`. With this link, the virtual machine becomes part of the bridge defined for the network interface.

A network bridge provided with the host system can be considered to resemble a network switch. Each member of the bridge is able to directly communicate with each other member. Therefore, if virtual machines shall be configured that must not communicate with each other, separate networks must be specified with `libvirtd`. These networks must be assigned with different physical network interfaces on the host. The network interfaces of the virtual machines that shall be isolated must be assigned different networks.

When `libvirtd` instantiates a virtual machine, it sets up a TAP device connected to the bridge the virtual machine is configured to have access to. The file descriptor of that TAP device is transferred to the `QEMU` process by the `libvirtd` virtual machine management daemon as the unprivileged `QEMU` is not allowed to open the TAP device. With this file descriptor, the virtual machine process is now able to implement all network protocols starting with OSI-layer 2.

As `libvirtd` provides virtual machine processes with a TAP device to the host system's bridge interface, a virtual machine is able to generate full Ethernet communication including sniffing the network or flooding the network with data. This implies that although the host system fully isolates the virtual machines from the host system as well as from each other, virtual machines have the same network capabilities to the assigned network as bare-metal systems connected to the same network! Therefore, guest software executing within the virtual machines must be as trustworthy as individual physical machines on the same network. KVM does not and is not designed to stop rogue guest software from misusing access to the networks they are allowed to access.

4.4.1 Packet filtering

The host system provides packet filtering functionality for the bridges implemented with *ebtables*(8). Please see the man pages for details about setting up packet filters on bridges.

More information about ebtables is presented in section §3.13 "Firewall configuration".

4.5 Device assignment

Together with the support of the *libvirtd* management daemon, KVM provides the functionality of assigning PCI as well as USB devices to virtual machines for their exclusive use. A device can only be assigned to one virtual machine using this mechanism.

The following sections discuss the device assignment mechanisms separately for PCI and USB devices.

4.5.1 PCI device assignment

PCI device assignment MUST NOT be configured as it is considered to be unsafe in secure environments.

4.5.2 USB device assignment

Unlike the PCI device assignment, USB assignment does not require special considerations. If one USB device is assigned to a virtual machine, access is mediated by *QEMU* via the USB device file. The access permissions on the USB device file are restricted by *libvirtd* modifying the device file ownership and its AppArmor access permission such that only the virtual machine which is given access to the device is able to access the device file.

WARNING: Neither the *libvirtd* management daemon, nor the KVM functionality implement any constraints in which USB devices can be assigned to a virtual machine. A virtual machine administrator may assign any device listed in the `lsusb` output to a virtual machine. This includes devices that are needed by the host system. Virtual machine administrators have the ability to disrupt the operation of the host. The evaluated configuration does not place any constraints on which USB devices are assigned to virtual machines. This gives the virtual machine administrator full flexibility over USB device assignment configuration. However, the administrator MUST be very careful about which USB devices are assigned.

4.6 Disk space management

The evaluated configuration allows virtual machine administrators to configure various storage backends that are passed to the virtual machines as disk devices. All configurations offered by `libvirtd` are allowed.

Chapter 5

Monitoring, Logging & Audit

5.1 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.d/*       # excluding /etc/apparmor.d/libvirt/<UUID>*
/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/init.d/*
/etc/init/*
/etc/inittab
/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 \( ! -uid 0 -o -perm +022 \)
```

5.2 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 LAuS 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.

5.3 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.

5.3.1 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.

5.3.2 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.

5.3.3 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.

5.3.4 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.

5.3.5 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. Depending on your local policy, you MAY need to shut down KVM guests. As these KVM guests act like normal processes on the Linux host system, the same commands to terminate these processes may be used.

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.

5.3.6 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.

5.4 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 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.

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 end point.

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.
- Combine characters from different character classes to construct a sufficiently strong password, using either 12 total characters containing 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 either an RSA or DSA key pair using the *ssh-keygen*(1) utility.

As only the `ssh-keygen` 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 either RSA or DSA key pairs for SSHv2 using the `-t rsa` or `-t dsa` command line switch.

The `ssh-keygen` utility allows you to specify the key size for RSA with the default of 2048 bits. If you select a different key size, you MUST use either 1024 bits or 3072 bits.

You MUST keep the private key part stored in *~/.ssh/id_dsa* or *~/.ssh/id_rsa* 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 a DSA key with 1024 bit key size:

```
ssh-keygen -t dsa -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 file *~/.ssh/id_dsa.pub* or *~/.ssh/id_rsa.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.

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.

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.

The screen locking and unlocking is provided by the `/usr/bin/vlock` which MUST be invoked with screen as otherwise the `screen` cannot be unlocked. `screen` is informed about the lock program using the *LOCKPRG* environment variable. This environment variable MUST be set to `/usr/bin/vlock` for every invocation of screen.

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*.

```
LOCKPRG="/usr/bin/vlock"
export LOCKPRG
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, http://www.suse.com/documentation/sles11/

David A. Wheeler, "Secure Programming for Linux and Unix HOWTO", *http://tldp.org/HOWTO/Secure-Programs-HOWTO/*

Kevin Fenzi, Dave Wreski, "Linux Security HOWTO", *http://www.linuxsecurity.com/docs/LDP/Security-HOWTO/*

7.2 Literature

Ellen Siever, Stephen Spainhour, Stephen Figgins, & Jessica P. Hekman, "Linux in a Nutshell, 6rd Edition", O'Reilly 2009, ISBN 978-0596154486

W. Richard Stevens, "Advanced Programming in the UNIX(R) Environment", Addison-Wesley 1992, ISBN 0201563177