Common Criteria Guide for Oracle Linux 7.3

Version 1.0

Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065
USA
Tel.: +1.650.506.7000

Trademarks

Oracle Linux and the Oracle logo are trademarks or registered trademarks of Oracle Corporation in the United States, other countries, or both.

atsec is a trademark of atsec information security GmbH.

Linux is a registered trademark of Linus Torvalds.

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

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

Legal Notice

Table of contents

Revision History

Version	Date	Author	Changes
1.0	2018-12-20	Stephan Müller	First public release

1 Introduction

1.1 Purpose of this Document

The Oracle Linux 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.

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.

The evaluated configuration is defined for the given software installed in accordance with the steps provided in this guide. Applying security patches released by the vendor SHOULD be considered even though that will cause a deviation from the evaluated configuration. Details about applying software updates are provided by the Oracle development team.

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 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 that is used to execute the Oracle Linux MUST be one of the following hardware systems:

- x86 64bit Intel Xeon processors:
  - Oracle Server X7-2

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 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 Oracle Linux, 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.

The evaluated configuration allows the use of DHCP acting benignly. It is RECOMMENDED to add static DNS names when using DHCP to allow users to access the server with a known identifier. If the administrator uses non-static IP addresses for servers, the administrator MUST ensure means to allow users to establish the authenticity of such servers. Such authenticity can be established using the SSH host keys. Section 3.15.4 provides details about the proper use of these SSH host keys to establish the authenticity of servers.

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 including the DHCP and potentially the DNS servers MUST be configured and managed under the same management control and operate under the same security policy constraints.

Be aware that information passed to another system leaves the control of the sending system, and the protection of this information against unauthorized access needs to be enforced by the receiving system. If an organization wants to implement a consistent security policy covering multiple

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

1.4.2 Requirements for Administrators

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

The system administrative personnel MUST NOT be careless, wilfully 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 Oracle Linux 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`.

It is part of your responsibility as a system administrator to verify that the requirements set forth for the users in section 1.5 are met, and to be available to users if they need your help in maintaining the security of their data.

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 chapter 5 of this guide. Appropriate training MUST be available to ensure this.

2 Installation

The evaluation covers a fresh installation of Oracle Linux 7.3, on one of the supported hardware platforms as defined in section 1.3.1 of this guide.

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

2.1 Supported Hardware

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

- Any storage devices and backup devices supported by the operating system (this includes hard disks, CD-ROM drives and tape drives).
- All Ethernet network adapters supported by the operating system. Other LAN or 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 3.8 of this guide), but *not* 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.

2.2 Kernels

The evaluated configuration covers the following types of kernels:

- UEK kernel
- “Red Hat Compatible” kernel

The user space packages are identical between both images. It is allowed to choose between the two kernels at installation time as well as during boot.

The following sections apply to systems with either kernel unless specifically noted.

2.3 Installation Process

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

All settings listed here are REQUIRED unless specifically declared otherwise.

2.3.1 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 of this guide for the list of supported hardware. The target system REQUIRES at least one local hard drive that will be erased and re-partitioned for use by the evaluated configuration.
- The availability of the Oracle Linux 7.3 ISO image dedicated to the Common Critera evaluation applicable for the chosen hardware system.
- A methods to make the contents of the ISO images containing Oracle Linux available to the target system, including ensuring the possibility to boot the boot image are as follows: 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 Oracle Linux 7 server guide.
- After obtaining the ISO images, you MUST perform the integrity verification of the downloaded files as outlined on the Oracle Linux web sites.

2.3.2 Obtaining of Installation Images

You MUST download the ISO images for the evaluated configuration from the Oracle Linux web site on a separate Internet-connected computer, and either burn DVD-Rs from them, or make the contents available as outlined in the Oracle Linux 7 server guide.

You MUST use the full DVD installation image. The boot image MUST NOT be used.

You MUST use Oracle Linux 7.3. Make sure that you are using the appropriate version for your platform, refer to section 1.3.1 of this guide for the list of supported hardware and the corresponding version needed.

You MUST verify that the integrity and authenticity of the image files are correct. The integrity is verified using the SHA-256 hash sums provided on the TLS protected Oracle Cloud download site when downloading Oracle Linux..

The following sections explain the steps of the installation process.

2.3.3 Standard Installation of Oracle Linux

The standard installation process of Oracle Linux is outlined in the Installation Guide.

To ensure consistency with the requirements of the Common Criteria evaluated configuration, the following restrictions must be applied to the options offered during the initial installation. All other options offered during the installation process can be freely set as desired.

When configuring an administrative user account, the Oracle Linux installer requires a password to be set. Although the password's quality is checked and the user is informed about passwords with insufficient quality, the installer allows such challenging passwords to be configured. Therefore, you MUST ensure that a password with sufficient quality is set. Guidance for passwords with sufficient strength is given in section 5.3.

- When partitioning the hard disk, you MUST ensure that the previous content of the entire hard disk is deleted. This is ensured by allowing the installer to erase all previous data and reformat the hard disk. You MUST NOT reuse previously existing partitions. It is

RECOMMENDED that the directory `/var/log` is stored on a separate partition to avoid log data filling up system partitions.

- When partitioning the system, you MUST NOT use the encrypted LVM option (i.e. select “Encrypt my data” option in the installer) or guarantee that sufficient entropy is present in the system. The reason is that when selecting this option, the master volume key used to encrypt the data for the lifetime of the partition is created at this point. However, insufficient entropy is yet present which may generate a master volume key with insufficient cryptographic strength. Note, encrypted partitions using `cryptsetup` can be created during runtime of the system as sufficient entropy is available at that time.
- The Oracle Linux installer allows the configuration of the system update approach. You MUST select "no automated update" to prevent the system from automatically pulling and installing updates.

After the installation is completed, the system MUST be rebooted.

2.3.4 Achieving the Evaluated Configuration

This section outlines the configuration steps to be performed to achieve the evaluated configuration.

Before performing the configuration steps outlined in this section, it is assumed that the installation process outlined in section 2.3.3 has been completed and Oracle Linux has been installed successfully.

The following subsections outline different configuration aspects that apply to different components of Oracle Linux.

2.3.4.1 Add Update Packages

After performing the installation of the operating system, a set of software components must be updated. To perform these updates, the RPM packages with the exact versions must be downloaded from the Oracle Linux web site into a separate directory.

Before installing these packages, the additional OL7 package signing key must be installed. That signing key can be obtained as outlined in the Oracle Linux GPG key website. This web page outlines how the key can be obtained, its authenticity is verified and how it is installed.

This is followed by executing the command `rpm -Uhv <download-directory>/*`. After completing the installation, the system must be rebooted.

The updates are intended to provide a system which is compliant with the FIPS 140-2 validation.

The following packages must be obtained and subsequently installed if the respective package is already installed with an earlier version on the system. Update packages are available at the Linux 7 (x86_64) Security Validation webpage. All other packages are derived from the operating system base repository:

- curl 7.29.0-35.0.1.el7.x86_64
- libcurl 7.29.0-35.0.1.el7.x86_64
- kernel 3.10.0-862.3.3.0.1.el7.x86_64
- kernel-headers 3.10.0-862.3.3.0.1.el7.x86_64
- kernel-tools 3.10.0-862.3.3.0.1.el7.x86_64

- kernel-tools-libs 3.10.0-862.3.3.0.1.el7.x86_64
- kernel-uek 4.1.12-124.16.4.el7uek.x86_64
- kernel-uek-firmware 4.1.12-124.16.4.el7uek.x86_64
- microcode_ctl 2:2.1-22.5.0.3.el7_4.x86_64
- fipscheck 1.4.1-5.el7.x86_64
- fipscheck-lib 1.4.1-5.el7.x86_64
- gmp 6.0.0-12.el7.x86_64
- gnutls 3.3.24-1.0.3.el7.x86_64
- gnutls-c++ 3.3.24-1.0.3.el7.x86_64
- nettle 2.7.1-8.el7.x86_64
- libgcrypt 1.5.3-13.el7_3.1.el7.x86_64nss 3.21.0-17.0.3.el7.x86_64
- nss-sysinit 3.21.0-17.0.3.el7.x86_64
- nss-tools 3.21.0-17.0.3.el7.x86_64
- nss-util 3.21.0-17.0.3.el7.x86_64
- nss-softokn 3.16.2.3-14.4.0.1.el7.x86_64
- nss-softokn-freebl 3.16.2.3-14.4.0.1.el7.x86_64
- openssl 1.0.1e-60.0.1.el7_3.1.x86_64
- openssl-libs 1.0.1e-60.0.1.el7_3.1.x86_64
- openssh 6.6.1p1-35.el7_3.x86_64
- openssh-clients 6.6.1p1-35.el7_3.x86_64
- openssh-server 6.6.1p1-35.el7_3.x86_64
- pam_ssh_agent_auth-0.9.3-9.35.el7_3.x86_64
- stunnel-4.56-6.el7.x86_64

The following packages must be installed to provide the functionality mandated by the evaluation – the dependencies required by these packages must be resolved and installed as well:

- cryptsetup 1.7.4-4.el7.x86_64
- scrub 2.5.2-7.el7.x86_64
- audispd-plugins 2.6.5-3.el7
- screen 4.1.0-0.25.el7.x86_64

During installation, the system may require the installation of some depending package. The administrator MUST load the respective packages and install them with the versions given in the dependency listing.

2.3.4.2 Configure SSH Server

The SSH server configuration is limited to use only approved ciphers. To ensure that the server process limits the available ciphers to the list of approved ciphers, the following configuration options must be set in `/etc/ssh/sshd_config`. Other configuration options can be chosen as required by the administrator:

- `Protocol 2`
- `PermitRootLogin no`
- `PermitEmptyPasswords no`
- `ChallengeResponseAuthentication no`
- `KerberosAuthentication no`
- `GSSAPIAuthentication no`
- `UsePAM yes`
- `PubkeyAuthentication yes`
- `PasswordAuthentication yes`
- `HostbasedAuthentication no`
- `IgnoreRhosts yes`
- `Ciphers aes256-ctr,aes128-ctr,aes256-gcm@openssh.com,aes128-gcm@openssh.com,aes256-cbc,aes128-cbc`
- `MACs hmac-sha2-512,hmac-sha2-256,hmac-sha1,hmac-sha1-96`
- `KexAlgorithms ecdh-sha2-nistp521,ecdh-sha2-nistp384,ecdh-sha2-nistp256,diffie-hellman-group14-sha1`

The configuration implies that in order to access the root account, the administrator first has to log on with his own regular user account and then switch to the root user using `su` or `sudo`.

It is permissible to change the order of the ciphers in the options `Ciphers, MACs, KexAlgorithms, HostKeyAlgorithms,` and `PubkeyAcceptedKeyTypes` to ensure the priority required by the operational environment is met.

2.3.4.3 Configure SSH Client

The SSH client is allowed to only use only approved ciphers. This can be configured with the configuration file of `/etc/ssh/ssh_config`. Other configuration options can be chosen as required by the administrator.

- `Ciphers aes256-ctr,aes128-ctr,aes256-gcm@openssh.com,aes128-gcm@openssh.com,aes256-cbc,aes128-cbc`
- `MACs hmac-sha2-512,hmac-sha2-256,hmac-sha1,hmac-sha1-96`
- `KexAlgorithms ecdh-sha2-nistp521,ecdh-sha2-nistp384,ecdh-sha2-nistp256,diffie-hellman-group14-sha1`
- `HostKeyAlgorithms ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-rsa`

- `PubkeyAcceptedKeyTypes ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-rsa`

It is permissible to change the order of the ciphers in the options `Ciphers, MACs, KexAlgorithms, HostKeyAlgorithms,` and `PubkeyAcceptedKeyTypes` to ensure the priority required by the operational environment is met.

Note: A user may override the system-wide selection given above by either using command line options or create a user-specific configuration file for the SSH client. In this case, however, the user's choice is limited to his session.

2.3.4.4 Configure PAM

For the authentication mechanism PAM, the account lockout after a given number of failed login attempts must be configured. The PAM module pam_tally2.so is used for this purpose. It must be added to the /etc/pam.d/system-auth and /etc/pam.d/password-auth based on the following example:

```
auth     required      pam_env.so
auth     required      pam_tally2.so file=/var/log/tallylog deny=3
auth     sufficient    pam_unix.so nullok try_first_pass
auth     requisite     pam_succeed_if.so uid >= 1000 quiet_success
auth     required      pam_deny.so

account  required      pam_unix.so
account  required      pam_tally2.so
account  sufficient    pam_localuser.so
account  sufficient    pam_succeed_if.so uid < 1000 quiet
account  required      pam_permit.so
```

2.3.4.5 Enable FIPS Mode

The evaluated configuration requires to execute the operating system in a FIPS 140-2 compliant mode. Therefore, the following instructions must be performed:

To bring the Module into FIPS Approved mode, perform the following:

1. Install the dracut-fips package:
   `# yum install dracut-fips`
2. Recreate the INITRAMFS image:
   `# dracut -f`

After regenerating the initramfs, the following string must be appended to the kernel command line by changing the setting in the boot loader:

```
fips=1
```

If /boot or /boot/efi resides on a separate partition, the kernel parameter `boot=<partition of /boot or /boot/efi>` must be supplied. The partition can be identified with the command:

```
"df /boot"
```

or

```
"df /boot/efi"
```

For example:

```
$ df /boot
Filesystem 1K-blocks Used   Available Use% Mounted on
/dev/sda1  233191    30454  190296    14%  /boot
```

The partition of /boot is located on /dev/sda1 in this example. Therefore, the following string needs to be appended to the kernel command line:

```
"boot=/dev/sda1"
```

Reboot to apply these settings.

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

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

Also, the rescue target provides a helpful environment:

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

Please refer to the relevant documentation of the boot loader, as well as the Oracle Linux 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 5.7 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 `-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 section 4.2 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 IKE 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 package for more information. Alternatively, you can 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 the `su` or `sudo` commands 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 `wheel` group. That group MUST NOT be used for any other user ID.

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 writeable 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 writeable for anyone other than root, and similarly MUST NOT use or execute any scripts, binaries or configuration files that are writeable for anyone other than root, or where any containing directory is writeable 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 `wheel` group to use all commands with privileges:

```
%wheel        ALL=(ALL:ALL)        ALL
```

The use of commands with the root identity, other system identities or system groups MUST be restricted to users of the `wheel` 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 add support for non-ELF binary formats or foreign binary format emulation that circumvents system call auditing. You MUST NOT activate `knfsd` or export NFS file systems.
- Device special nodes MUST NOT be added to the system.
- SUID root, SGID root programs or programs with file system capabilities MUST NOT be added to the system. Programs which use the SUID or SGID bits to run with identities other than 'root' MAY be added if the numerical SUID and SGID values are not less than 1000 as defined with the values `UID_MIN` and `GID_MIN` in the configuration file of `/etc/login.defs`. This restriction is necessary to avoid conflict with system user and group IDs such as the `disk` group.
- The content, permissions, and ownership of all existing filesystem objects (including directories and device nodes) that are part of the evaluated configuration MUST NOT be modified. Files and directories MAY be added to existing directories provided that this does not violate any other requirement.
- Programs automatically launched with 'root' privileges MUST NOT be added to the system. Exception: processes that immediately and permanently switch to a non privileged identity on launch are permitted, for example by using `su USERID -c LAUNCH_COMMAND` in the startup file, or alternatively by using the `setgroups(2)`, `setgid(2)` and `setuid(2)` system calls in a binary. (`seteuid(2)` etc. are insufficient -- if the administrator cannot identify when and how privileged are dropped, the application MUST NOT be installed.)

  Automatic launch mechanisms are:

  - Targets and units as part of the `systemd` mechanism.
  - Scheduled jobs using `cron` (including entries in `/etc/cron*` files)
  - Applications started using the system DBUS which is configured via `/etc/dbus-1/system.d/`.
  - Applications specified in `/etc/sudoers` or with rules located in a file in the directory `/etc/sudoers.d`. Note, that file may contain the keyword `ALL` as a placeholder for a command. In this case, the user allowed to execute all commands with that rule using

the root user ID MUST ensure that additional applications are not executed using `sudo`. This requirement can only be met with operational procedures.

- Applications spawned via `udev` where the rules are added to `/lib/udev/rules.d`.

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

3.5 Scheduling Processes Using cron

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

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

Errors in the non interactive jobs executed by `cron` are reported in the system log files in `/var/log/`, and additionally via e-mail to the user who scheduled it. Permission for users to schedule jobs with `cron` is controlled through the following `allow` and `deny` files:

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

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

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 user name 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.

When creating additional file systems, one of the following file systems are allowed to be used:

- Ext3
- Ext4
- XFS

Other file systems may be supported by the operating system, but are not allowed to be used for supplemental partitions. The exception is the use of VFAT for `/boot/efi` in case a UEFI boot environment is used.

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.
- 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 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 removable storage media mounts.

For example:

```
# mount /dev/cdrom /mnt -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 Managing User Accounts

3.7.1 Creating Users

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

If you assign an initial password for a new user, you MUST transfer this initial password in a secure way to the user, ensuring that no third party gets the information. For example, you can tell the password to a user personally known to you. If this is not possible, you MAY send the password in written form in a sealed letter. This applies also when you set a new password for a user in case the user has forgotten the password or it has expired. You MUST advise the user that he MUST change this initial password when he first logs into the system and select his own password in accordance with the rules defined in section 5.3 of this guide.

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:

```
adduser --gecos "John Doe" jdoe
passwd jdoe
chage -d $(date +%F -d "53 days ago") jdoe
```

The `useradd` command 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
```

3.7.2 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.

3.7.3 SSH key-based Authentication

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

To generate keys that can be used for key-based authentication, the tool `ssh-keygen(8)` is provided. As the SSH daemon only accepts SSH protocol version 2, only the protocol 2 keys are supported with the SSH daemon.

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 key sizes larger than 2048 bits. All supported key sizes for ECDSA except NIST P-521 are allowed.

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

3.7.4 Changing User Properties

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

3.7.5 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.

3.7.6 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 --no-headers --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.

3.7.7 Defining Administrative Accounts

Administrative users MUST be member of the `wheel` group.

3.8 Using Serial Terminals

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

Serial terminals are activated by `systemd` which monitors the serial line. `systemd` 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.

3.9 Managing Data Objects

3.9.1 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.

3.9.2 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 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.

The `hwclock(8)` can query and modify the hardware clock on supported platforms. 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 UTC. It uses UTC by default. See hwclock(8) for further details. The following command sets the hardware clock to the current time using UTC:

```
hwclock -u -w
```

3.13 Firewall Configuration

You MAY enable, reconfigure, or disable the built-in network firewall as required.

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 sophisticated filtering rules can be configured as well to suit the need of the administrator. An elaborate introduction is given in the Oracle Linux Security Guide. In addition, `iptables(8)` discusses use of the application.

3.14 Screen Saver Configuration

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

The `screen` command may be started at logon time. The following discussion applies only when the user did not disable screen. Note that irrespective of enabling screen with the mentioned script, the scroll-back buffer in the terminal is disabled using a kernel command line. The kernel command line option for the scroll-back 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 time-out in either `/etc/screenrc` or `~/.screenrc` using the `idle X lockscreen` configuration value where X is an integer value specifying the idle time in seconds before the screen is locked.
- Every user can lock his screen by executing the `CTRL-a CTRL-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`.

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 `~/.profile` for a per-user enforcement. Note that a user can change `~/.profile`.

```
exec screen
```

3.15 Cryptographic Support

3.15.1 TLS Key Handling

The TOE supports the generation of the RSA key pair used by the client. The evaluated configuration also allows the use of an externally-generated certificate. A widely accepted Certification Authority might be used to generate and/or sign such a certificate (allowing a wide community trusting this CA to validate the certificate). In a closed community it might also be sufficient to have one server within the community to act as a CA. The OpenSSL library provides the functions to set up such a CA, but those functions are not subject of this Security Target. To perform administrative actions regarding the key and certificate handling, following commands are provided:

- `openssl genrsa` documented in the `genrsa(1)` man page is to be used for generating an RSA private key. The generated key SHOULD be encrypted using either AES128, AES192 or AES256 offered with the mentioned command. You MUST NOT use the `-3` command line option to use the RSA public exponent of 3. Only the public exponent of 65537 is allowed to be used. In addition, you MUST NOT use an OpenSSL engine with the `-engine` command line option. The RSA modulus size MUST be 2048 bits or larger.
- With `openssl ecparam` explained in the `ecparam(1)` man page, an ECDSA private key can be generated. You MUST use one of the following curves for generating the key: `prime256v1` (NIST P-256), `secp384r1` (NIST P-384), or `secp521r1` (NIST P-521). Again, OpenSSL is not allowed be used with an engine which implies that the `-engine` command line option MUST NOT be used.
- The command `openssl req` can be used to derive a certificate signing request from the private key generated with the aforementioned command. Please see the man page `req(1)` for details. In case a new key shall be generated, the constraints given for the `genrsa` command above MUST be met.
- Using `openssl ca` as documented in the `ca(1)` man page can be used to sign a certificate signing request.

Using the mentioned commands, a self-signed CA certificate can be generated which MUST be subject to the same cryptographic constraints outlined above.

3.15.2 TLS Configuration

When establishing TLS communication channels, you MUST configure one or more cipher suites from the following list. It is not permissible to use other cipher suites:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA

- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA256

In addition, you MUST configure the TLS communication channel to use the TLS v1.2 protocol.

Please note that only the cipher implementations part of the OpenSSL libraries were subject to cryptographic validation. None of the OpenSSL engines were validated.

3.15.3 Cryptographic Key Handling for OpenSSH

The cryptographic network protocol of SSH provides a secure channel to protect sensitive data. During the establishment of the secure channel, the protocol uses certificates and private keys to support the mutual authentication. The entire trust of the secure channel also rests on the appropriate authentication to establish the identity and authenticity of the remote peer.

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

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

In addition, when setting up disk encryption configurations, the passphrase protects the master volume key used to encrypt the data to be stored on disk. To ensure appropriate protection of that master volume key, you MUST use a sufficiently strong passphrase.

All certificates of the certificate chain MUST use the following cryptographic mechanisms as applicable:

- RSA key length: RSA with 2048 bit or larger
- ECDSA curves: ECDSA NIST P-256, NIST P-384, or NIST P-521
- Hash algorithm: SHA-224, SHA-256, SHA-384, or SHA-512; SHA-1 is deprecated

You MUST ensure that the exponent of an RSA certificate is at least 2^16 + 1 i.e. 65537 or larger.

3.15.4 SSH Host Key Usage

In addition to user authentication, SSH performs an authentication of the server. This mechanism provides a guarantee to the user that the intended server is reached and no man-in-the-middle attack is performed.

The authentication of the SSH server by the SSH client application is implemented by using SSH host keys. Oracle Linux generates such SSH host keys automatically during the first boot of the operating system.

Before the authentication of the server can be verified, the SSH host keys must be transferred from the server to the client. Commonly this is done automatically upon first access of the SSH server by the SSH client. During the first contact, the SSH host keys are exchanged and the SSH client displays the fingerprint of the retrieved SSH host keys to the user and requests the user to confirm the fingerprint. After the confirmation of the fingerprint, the SSH host keys are stored in the file `.ssh/known_hosts` by the SSH client. Every subsequent connection attempt verifies the SSH host keys with the respective entries in `.ssh/known_hosts`. If the SSH host keys differ for a given SSH server compared to previous connections, the SSH client displays a warning and rejects any communication attempts.

To ensure that the SSH host keys are transmitted to the SSH client in a way to guarantee a proper authentication of the SSH server, the administrator MUST either provide the SSH host key fingerprints to the users and require that the users verify the fingerprint. Alternatively, the administrator MUST distribute the `known_hosts` file to the users with the known good SSH host keys of the server.

3.15.5 Cryptographic Key Destruction

The keys stored in files as outlined in the preceding sections can be securely erased using the `scrub(1)` command. You MUST NOT delete a file holding cryptographic keys by any means other than `scrub`.

Please note that when using SSDs, the wear levelling mechanism prevents software to overwrite the exact physical location where the keys are stored. Thus, even when using `scrub`, the key data may still reside on the physical data store albeit it cannot be retrieved by the operating system any more. Yet, forensic tools may recover that data. Thus, an SSD must be physically destroyed at the end of life to guarantee that no cryptographic keys remain.

The system uses many more keys than outlined in the preceding sections. Those keys are always ephemeral and maintained in RAM. These keys will be securely erased by the system without user intervention.

3.16 Trusted Updates

Oracle provides regular updates to the Oracle Linux operating system. After initial installation, the update mechanism is fully configured to obtain updates.

The administrator can start the update process with the following command: `yum update`. This command fetches the current list of available updates for the current installation base. If updates are available for one or more of the currently installed packages, the update command will list them and

asks the administrator whether to install them. In case the update fails for any reason, `yum` will list the reason and prevent the update operation.

When installing new software components for Oracle Linux using the `yum install` command, the command ensures that the most current version of the software component is installed.

The automated installation of updates can be achieved by using yum-cron which is installed with the following command:

```
yum --enablerepo=ol7_optional_latest install yum-cron
```

The configuration file for yum-cron allows the definition of how the automated update is to be performed. When deinstalling the package, the automated update is deactivated. Also, the service can be started and stopped using the `systemctl` start/stop command.

4 Monitoring, Logging & Audit

4.1 Reviewing the System Configuration

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

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

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

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

```
/var/log/audit/*
/var/log/lastlog
/var/log/tallylog
```

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.

4.2 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.

4.2.1 Intended Usage of the Audit Subsystem

As part of the evaluation, various auditing capabilities are assessed and ensured that they are supported. The evaluated configuration described here is based on these requirements.

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

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

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

4.2.2 Selecting the Events to be Audited

You MAY make changes to the set of system calls and events that are to be audited. Oracle Linux 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/audit/rules` file for systems covering the basic system functionality. It contains a suggested setup for a typical multi-user system, all access to security relevant files is audited, along with other security relevant events such as system reconfiguration. You MAY copy one of the sample rules files to `/etc/audit/audit.rules` and modify the configuration according to your local requirements, including the option of using an empty audit rules file to disable auditing if not required.

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

4.2.3 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.

The following search options are relevant for operations specified in the Security Target:

- Start of audit operation: `ausearch -m SERVICE_START`
- End of audit operation: `ausearch -m SERVICE_END`
- User authentication: `ausearch -m USER_AUTH`
- Start of a user session: `ausearch -m USER_START`
- End of a user session: `ausearch -m USER_END`
- Use of privileged, special rights events:

    - Use of root user ID: `ausearch -ui “root”`
    - Use of the effective user ID of zero: `ausearch --uid-effective 0`
    - Watch use SUID applications: after adding a watch to all SUID applications, the command `ausearch -x` can be used to log all invocations of the SUID application.
- Privilege or role escalation: `ausearch -x “sudo”`

The audit log information contains the following information:

- Date and time is marked with “`time->`”
- Type of event is referenced with “`type`”
- Subject identity is specified by “`uid`” containing the numeric user ID of the respective user
- Outcome of the event identified with the field “`success`”
- User identity is given with the “`auid`” field

4.2.4 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 `systemctl reload auditd` to re-load the filters.

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

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

4.2.5 Storage of Audit Records

The default audit configuration stores audit records in the `/var/log/audit/` directory. 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 or shutting down the system to prevent auditable actions when the audit records cannot be stored.

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

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

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

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

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

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

4.2.6 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 Security Guidelines for Users

5.1 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)`.

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/bash/README
```

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.

5.2 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.

You MAY use SSH user keys for authentication as outlined in ssh(1). When using SSH key-based authentication, the SSH server first attempts to authenticate the user with the provided key. If this authentication attempt is unsuccessful, the SSH server falls back to password-based authentication and ask the user for the password.

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.

5.3 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:

```
"My first pet was a dog named Furry."
=> Mfpwadn4y.
"Password 'P'9tw;citd' too weak; contained in this document"
=> P'9tw;citd
```

5.4 SSH key-based Authentication

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

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 a larger bit size. All supported key sizes for ECDSA are allowed.

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

The following command line is an example that generates a ECDSA key with NIST curve P-256 key size:

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

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

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

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

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

5.5 SSH Host Keys

With the use of SSH, the SSH server is always authenticated by the SSH client application. This prevents man-in-the-middle attacks.

This technique rests on SSH host keys which are exchanged during the first connection with the SSH server. During that connection attempt, you are asked to verify and confirm the fingerprint of the SSH host key. You MUST confirm the fingerprint with a known good fingerprint provided by the administrator.

The SSH client application will display a warning sign in case the SSH host key of the server has changed indicating a man-in-the-middle attack. You MUST NOT proceed with the communication attempt in this case and contact the administrator instead.

5.6 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.

5.6.1 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 B<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.

5.7 Data Import / Export

The system comes with various tools to archive data (`tar`, `cpio`). If ACLs are used, then only `tar` MUST be used to handle the files and directories as the other commands do not support ACLs.

The options `--acls --xattrs` must be used with `tar` to back up the ACLs and the user extended attributes.

Please see the `tar(1)` man page for more information.

5.8 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.

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 `~/.profile`.

```
exec screen
```

Appendix A 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.

A list of all available online resources is given at the Oracle Linux 7 Documentation website.

Appendix B Abbreviations

Abbreviation	Description
AH	Authentication Header
CC	Common Criteria
DAC	Discretionary Access Control
EAL	Evaluation Assurance Level
ESP	Encapsulating Security Payload
IKE	Internet Key Exchange
IPSEC	IP Security Protocol
MAC	Mandatory Access Control
OSPP	Operating System Protection Profile
PP	Protection Profile
SAR	Security Assurance Requirement
SFP	Security Function Policy
SFR	Security Functional Requirement
SSH	Secure Shell
ST	Security Target
TOE	Target of Evaluation
TLS	Transport Layer Security
TSF	TOE security function
TSFI	TSF Interface
TSP	TOE security policy