



SUSE Linux Enterprise Micro 5.4

Security and Hardening Guide

Security and Hardening Guide

SUSE Linux Enterprise Micro 5.4

The security guide focuses on using SELinux, FIPS, and gives details about the remote attestation using the *Keylime* agent

Publication Date: December 18, 2025

<https://documentation.suse.com> 

Copyright © 2006–2025 SUSE LLC and contributors. All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or (at your option) version 1.3; with the Invariant Section being this copyright notice and license. A copy of the license version 1.2 is included in the section entitled “GNU Free Documentation License”.

For SUSE trademarks, see <https://www.suse.com/company/legal/> . All third-party trademarks are the property of their respective owners. Trademark symbols (®, [™] etc.) denote trademarks of SUSE and its affiliates. Asterisks (*) denote third-party trademarks.

All information found in this book has been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. Neither SUSE LLC, its affiliates, the authors nor the translators shall be held liable for possible errors or the consequences thereof.

Contents

Preface **vi**

- 1 Available documentation **vi**
- 2 Improving the documentation **vii**
- 3 Documentation conventions **viii**
- 4 Support **x**
 - Support statement for SUSE Linux Enterprise Micro **x** • Technology previews **xi**

1 SELinux 1

- 1.1 Getting SELinux **1**
- 1.2 SELinux modes **1**
- 1.3 SELinux policy overview **3**
 - Creating policies for containers **3**
- 1.4 SELinux security context **5**
- 1.5 Tools for managing SELinux **5**

2 Remote attestation using Keylime 8

- 2.1 About *Keylime* **9**
 - Keylime* agent **10** • *Keylime* registrar **10** • *Keylime* verifier **10**
- 2.2 Setting up the verifier and registrar **10**
- 2.3 Installing the *Keylime* agent **11**
- 2.4 Registering agents **13**
- 2.5 Secure payloads **14**
- 2.6 Enabling IMA tracking **14**

3 Authentication with PAM 16

- 3.1 What is PAM? 16
- 3.2 Structure of a PAM configuration file 17
- 3.3 The PAM configuration of sshd 19
- 3.4 Configuration of PAM modules 21
 - pam_env.conf 22 • limits.conf 22
- 3.5 Configuring PAM using pam-config 22
- 3.6 Manually configuring PAM 23
- 3.7 Configuring U2F keys for local login 24
 - Associating the U2F key with your account 24 • Updating the PAM configuration 25

4 Enabling compliance with FIPS 140-3 27

- 4.1 FIPS overview 27
- 4.2 When to enable FIPS mode 28
- 4.3 Installing FIPS 28
- 4.4 Running containers on SLE Micro 29
- 4.5 MD5 not supported in Samba/CIFS 29
- 4.6 More information 30

A GNU licenses 31

Preface

1 Available documentation

Online documentation

Our documentation is available online at <https://documentation.suse.com>. Browse or download the documentation in various formats.



Note: Latest updates

The latest updates are usually available in the English-language version of this documentation.

SUSE Knowledgebase

If you run into an issue, check out the Technical Information Documents (TIDs) that are available online at <https://www.suse.com/support/kb/>. Search the SUSE Knowledgebase for known solutions driven by customer need.

Release notes

For release notes, see <https://www.suse.com/releasesnotes/>.

In your system

For offline use, the release notes are also available under `/usr/share/doc/release-notes` on your system. The documentation for individual packages is available at `/usr/share/doc/packages`.

Many commands are also described in their *manual pages*. To view them, run `man`, followed by a specific command name. If the `man` command is not installed on your system, install it with `sudo zypper install man`.

2 Improving the documentation

Your feedback and contributions to this documentation are welcome. The following channels for giving feedback are available:

Service requests and support

For services and support options available for your product, see <https://www.suse.com/support/>.

To open a service request, you need a SUSE subscription registered at SUSE Customer Center. Go to <https://scc.suse.com/support/requests>, log in, and click *Create New*.

Bug reports

Report issues with the documentation at <https://bugzilla.suse.com/>.

To simplify this process, click the *Report an issue* icon next to a headline in the HTML version of this document. This preselects the right product and category in Bugzilla and adds a link to the current section. You can start typing your bug report right away.

A Bugzilla account is required.

Contributions

To contribute to this documentation, click the *Edit source document* icon next to a headline in the HTML version of this document. This will take you to the source code on GitHub, where you can open a pull request.

A GitHub account is required.



Note: *Edit source document* only available for English

The *Edit source document* icons are only available for the English version of each document. For all other languages, use the *Report an issue* icons instead.

For more information about the documentation environment used for this documentation, see the repository's README.

Mail

You can also report errors and send feedback concerning the documentation to doc-team@suse.com. Include the document title, the product version, and the publication date of the document. Additionally, include the relevant section number and title (or provide the URL) and provide a concise description of the problem.

3 Documentation conventions

The following notices and typographic conventions are used in this document:

- /etc/passwd: Directory names and file names
- PLACEHOLDER: Replace PLACEHOLDER with the actual value
- PATH: An environment variable
- ls, --help: Commands, options, and parameters
- user: The name of a user or group
- package_name: The name of a software package
- **Alt** , **Alt – F1** : A key to press or a key combination. Keys are shown in uppercase as on a keyboard.
- *File*, *File > Save As*: menu items, buttons
- **AMD/Intel** > This paragraph is only relevant for the AMD64/Intel 64 architectures. The arrows mark the beginning and the end of the text block. ◁
- **IBM Z, POWER** > This paragraph is only relevant for the architectures IBM Z and POWER. The arrows mark the beginning and the end of the text block. ◁
- *Chapter 1, “Example chapter”*: A cross-reference to another chapter in this guide.
- Commands that must be run with root privileges. You can also prefix these commands with the sudo command to run them as a non-privileged user:

```
# command  
> sudo command
```

- Commands that can be run by non-privileged users:

```
> command
```

- Commands can be split into two or multiple lines by a backslash character (\) at the end of a line. The backslash informs the shell that the command invocation will continue after the end of the line:

```
> echo a b \
```

```
c d
```

- A code block that shows both the command (preceded by a prompt) and the respective output returned by the shell:

```
> command  
output
```

- Notices



Warning: Warning notice

Vital information you must be aware of before proceeding. Warns you about security issues, potential loss of data, damage to hardware, or physical hazards.



Important: Important notice

Important information you should be aware of before proceeding.



Note: Note notice

Additional information, for example about differences in software versions.



Tip: Tip notice

Helpful information, like a guideline or a piece of practical advice.

- Compact Notices



Additional information, for example about differences in software versions.



Helpful information, like a guideline or a piece of practical advice.

4 Support

Find the support statement for SUSE Linux Enterprise Micro and general information about technology previews below. For details about the product lifecycle, see <https://www.suse.com/lifecycle>.

If you are entitled to support, find details on how to collect information for a support ticket at <https://documentation.suse.com/sles-15/html/SLES-all/cha-adm-support.html>.

4.1 Support statement for SUSE Linux Enterprise Micro

To receive support, you need an appropriate subscription with SUSE. To view the specific support offers available to you, go to <https://www.suse.com/support/> and select your product.

The support levels are defined as follows:

L1

Problem determination, which means technical support designed to provide compatibility information, usage support, ongoing maintenance, information gathering and basic troubleshooting using available documentation.

L2

Problem isolation, which means technical support designed to analyze data, reproduce customer problems, isolate a problem area and provide a resolution for problems not resolved by Level 1 or prepare for Level 3.

L3

Problem resolution, which means technical support designed to resolve problems by engaging engineering to resolve product defects which have been identified by Level 2 Support.

For contracted customers and partners, SUSE Linux Enterprise Micro is delivered with L3 support for all packages, except for the following:

- Technology previews.
- Sound, graphics, fonts, and artwork.
- Packages that require an additional customer contract.
- Packages with names ending in `-devel` (containing header files and similar developer resources) will only be supported together with their main packages.


SUSE will only support the usage of original packages. That is, packages that are unchanged and not recompiled.

4.2 Technology previews

Technology previews are packages, stacks, or features delivered by SUSE to provide glimpses into upcoming innovations. Technology previews are included for your convenience to give you a chance to test new technologies within your environment. We would appreciate your feedback. If you test a technology preview, please contact your SUSE representative and let them know about your experience and use cases. Your input is helpful for future development.

Technology previews have the following limitations:

- Technology previews are still in development. Therefore, they may be functionally incomplete, unstable, or otherwise *not* suitable for production use.
- Technology previews are *not* supported.
- Technology previews may only be available for specific hardware architectures.
- Details and functionality of technology previews are subject to change. As a result, upgrading to subsequent releases of a technology preview may be impossible and require a fresh installation.
- SUSE may discover that a preview does not meet customer or market needs, or does not comply with enterprise standards. Technology previews can be removed from a product at any time. SUSE does not commit to providing a supported version of such technologies in the future.

For an overview of technology previews shipped with your product, see the release notes at <https://www.suse.com/releasenotes> .

1 SELinux

This chapter gives a brief overview of SELinux implementation on SLE Micro.

SELinux was developed as an additional Linux security solution that uses the security framework in the Linux kernel. The purpose was to allow for a more granular security policy that goes beyond the standard Discretionary Access Controls (DAC), the traditional file permissions of owner/group/world, and read/write/execute.

SELinux uses labels attached to objects (for example, files and network sockets) and uses them for access decisions.

The default action of SELinux is to deny any access. SELinux allows only actions that were specifically allowed in the SELinux policy. Another feature of SELinux that increases security is that SELinux allows strict confinement of processes up to the point where the processes cannot access files of other processes on the same system.

SELinux was designed to enhance existing security solutions, not to replace them. For example, discretionary access control (DAC) is still applied even if the system is using SELinux. If DAC denies access first, SELinux is then not used as the access was already blocked by another mechanism.

1.1 Getting SELinux

SELinux is installed by default when installing SLE Micro by YaST or is part of the pre-built images. The default mode is set to enforced on all deployment types, and the file system is labelled.

If in any case SELinux is not set up on your system, run the following command:

```
# transactional-update setup-selinux
```

Reboot your system after the command has finished. The command installs the SELinux policy if it is not installed, sets the enforcing SELinux mode and rebuilds initramfs.

1.2 SELinux modes

SELinux can run in one of three modes: disabled, permissive, or enforcing.

Using the disabled mode means that no rules from the SELinux policy are applied and your system is not protected. Therefore, the disabled mode is not recommended.

In the permissive mode, SELinux is active, the security policy is loaded, the file system is labeled and access denial entries are logged. However, the policy is not enforced and thus no access is actually denied.

In the enforced mode, the security policy is applied. Each access that is not explicitly allowed by the policy is denied.

You can switch between the enforcing and permissive modes by using the **setenforce** command. Alternatively, you can switch between all SELinux modes by editing the /etc/selinux/config configuration file. Changes performed by the **setenforce** command are valid only until the next reboot. For persistent changes of the SELinux mode, edit the /etc/selinux/config configuration file.

The **setenforce** command has the following syntax:

```
# setenforce MODE_ID
```

where MODE_ID is 0 for the permissive mode or 1 for the enforced mode.

To verify the mode, run the following command:

```
# getenforce
```

The command should return permissive or enforced, depending on the provided MODE_ID.

To change the SELinux mode permanently, in the file /etc/selinux/config, change the value of SELINUX to disabled, or permissive, or enforced as follows:

```
SELINUX=disabled
```

The changes in the file are applied after the next reboot.



Note: Relabeling your system after switching from the disabled mode

If you disable SELinux on your system and then enable it later, make sure that you relabel your system. When SELinux is disabled, and you perform changes to your file system, the changes are not reflected in the context anymore (for example, new files do not have any context). Therefore, you need to relabel your system by using the **restorecon** command, using the autorelabel boot parameter, or by creating a file that will trigger relabeling on the next boot. To create the file, run the following command:

```
# touch /etc/selinux/.autorelabel
```

After reboot, the file `/etc/selinux/.autorelabel` is replaced with another flag file: `/etc/selinux/.relabelled` to prevent relabeling on subsequent reboots.

1.3 SELinux policy overview

The policy is the key component in SELinux. Your SELinux policy defines rules that specify which objects can access which files, directories, ports, and processes on a system. To do this, a security context is defined for all of these.

An SELinux policy contains a huge number of rules. To make it more manageable, policies are often split into modules. This allows the administrator to switch protection on or off for different parts of the system.

When compiling the policy for your system, you will have a choice to either work with a modular policy, or a monolithic policy, where one huge policy is used to protect everything on your system. It is strongly recommended to use a modular policy and not a monolithic policy. Modular policies are much easier to manage.

SLE Micro is shipped with the `targeted` SELinux policy.

1.3.1 Creating policies for containers

SLE Micro is delivered with a policy that by default does not allow containers to access files outside the container data. On the other hand, all network access is allowed. Typically, containers are created with bind mounts and should be able to access other directories like `/home` or `/var`. You may want a possibility to allow access to these directories or, on the contrary, restrict some ports to the container even if SELinux is used on your system. In this case, you need to create new policy rules that enable or disable the access. SLE Micro provides the `Udica` tool for this purpose. The following procedure describes how to create a custom policy for your containers:

1. Make sure that SELinux is in the enforcing mode. For details, refer to [Section 1.2, “SELinux modes”](#).
2. Start a container using the following parameters:

```
# podman run -v /home:/home:rw -v /var:/var:rw -p 21:21 -it sle15 bash
```

The container runs with the default policy that does not allow access to the mount points but does not restrict other ports.

3. You can exit the container.

4. Obtain the container ID:

```
# podman ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e59f9d0f86f2	registry.opensuse.org/devel/bci/tumbleweed/containerfile/opensuse/bci/ruby:latest	/bin/bash	8 minutes ago	Up 8 seconds ago	0.0.0.0:21->21/tcp	zen_ramanujan

5. Create a JSON file that Udica will use to create a custom policy for the container:

```
# podman inspect e59f9d0f86f2 > OUTPUT_JSON_FILE
```

For example, substitute `OUTPUT_JSON_FILE` with `container.json`

6. Run Udica to generate a policy according to the container parameters:

```
# udica -j OUTPUT_JSON_FILE CUSTOM_CONTAINER_POLICY
```

For example:

```
# udica -j container.json custom_policy
```

7. According to the provided instructions, load the policy modules by running:

```
# semodule -i custom_policy.cil /usr/share/udica/templates/{base_container.cil,net_container.cil,home_container.cil}
```

8. Run a container with the new policy module by using the `--security-opt` option as follows:

```
# podman run --security-opt label=type:custom_policy.process -v /home:/home:rw -v /var:/var:rw -p 21:21 -it sle15 bash
```

1.4 SELinux security context

The security context is a set of information assigned to a file or a process. It consists of SELinux user, role, type, level and category. This information is used to make access control decisions.

SELINUX CONTEXT FIELDS

SELinux user

is an identity defined in the policy that is authorized for a specific set of roles and for a specific *level* range. Each Linux user is mapped to an SELinux user. SELinux does not use the list of user accounts maintained by Linux in `/etc/passwd`, but uses its own database and mapping. By convention, the identity name is suffixed with `_u`, for example: `user_u`.

role

defines a set of permissions that a user can be granted. A role defines which *types* a user assigned to this role can access. By convention, the role name is suffixed with `_r`, for example: `system_r`.

type

conveys information on how particular files and processes can interact. A process consists of files with a concrete SELinux type, and it cannot access files outside of this type. By convention, the type name is suffixed with `_t`, for example: `var_t`.

level

is an optional attribute that specifies the range of levels of clearance in the multilevel security.

category

is an optional attribute that allows you to add categories to processes, files, and users. A user can then access files that have the same category.

1.5 Tools for managing SELinux

SLE Micro provides you with tools to manage SELinux on your system. If, in any case, the below described tools are not installed on your system, install the tools by running:

```
# transactional-update pkg install policycoreutils-python-utils
```

After successful installation, reboot the system.

SELINUX COMMANDS

chcon

changes the security context of provided files to the context provided to the command

getenforce

displays the current SELinux mode

fixfiles

enables you to check for issues with a mismatched security context and then fix them

ls -Z PATH

shows security context of all files/directories in the specified PATH, for example:

```
# ls -Z /
system_u:object_r:bin_t:s0 bin
system_u:object_r:boot_t:s0 boot
system_u:object_r:device_t:s0 dev
system_u:object_r:etc_t:s0 etc
system_u:object_r:home_root_t:s0 home
system_u:object_r:lib_t:s0 lib
system_u:object_r:lib_t:s0 lib64
system_u:object_r:mnt_t:s0 mnt
system_u:object_r:usr_t:s0 opt
system_u:object_r:proc_t:s0 proc
system_u:object_r:default_t:s0 root
system_u:object_r:var_run_t:s0 run
system_u:object_r:bin_t:s0 sbin
system_u:object_r:var_t:s0 srv
system_u:object_r:sysfs_t:s0 sys
system_u:object_r:tmp_t:s0 tmp
system_u:object_r:usr_t:s0 usr
system_u:object_r:var_t:s0 var
```

restorecon

restores a file context to the default value (as stored in the SELinux policy)

semanage

enables you to adjust context and configure certain elements of SELinux policy. The command provides several subcommands. For details, use:

```
# semanage --help
```

setenforce

enables you to temporarily set a SELinux mode to permissive or enforcing

sestatus

displays the current status of SELinux, for example:

```
# sestatus

SELinux status:                enabled
SELinuxfs mount:              /sys/fs/selinux
SELinux root directory:       /etc/selinux
Loaded policy name:            targeted
Current mode:                  enforcing
Mode from config file:         enforcing
Policy MLS status:             enabled
Policy deny_unknown status:    allowed
Memory protection checking:    requested (insecure)
Max kernel policy version:     31
```



Note: The Z option available to other commands

You can also use the Z option with other commands, for example: cp, ps, and id.

2 Remote attestation using Keylime

This chapter focuses on the remote attestation of SLE Micro using the *Keylime* agent. With the growing demand on securing devices against unauthorized changes, the use of the security mechanism called *remote attestation (RA)* has been experiencing significant growth. Using RA, a host (client) can authenticate its boot chain status and running software on a remote host (verifier). *Remote attestation* is usually combined with public-key encryption (by using TPM2), thus the sent information can only be read by the services that requested the attestation, and the validity of the data can be verified.

Throughout this document, the following terms are used:

TERMINOLOGY

Attestation key (AK)

is a data signing key that proves that the data comes from a real TPM and has not been tampered with.

Core root of trust for measurement

is a starting point of the boot process that cannot be altered and calculates the first hash of the layer above.

Endorsement key (EK)

is an encryption key that is permanently embedded in the TPM when it is manufactured. The public part of the key is used to recognize a genuine TPM.

Integrity management architecture (IMA)

the kernel integrity subsystem that provides a means of detecting malicious changes to files.

Measured boot

in this method, each component in the booting sequence calculates a hash of the next one before delegating the execution of the next component. The hash extends one or several PCRs of the TPM. An event is created with the information about where the measurement took place and what was measured. Such events are collected in an event log, and, along with the extended PCR values, the events can be compared with the expected values representing a healthy system.

Platform Configuration Register (PCR)

is a memory location in TPM that, for example, stores hashes of booting layers. PCR can be updated only by using the non-reversible operation—extend. A signed list of current PCR values can be obtained by the quote command on TPM, and this quote can be verified by a third party during the attestation process.

Secure boot

each step of the booting process checks a cryptographic signature on the executable of the next step before launching it.

Trusted Platform Module (TPM)

is a self-contained security cryptographic processor present in the system as hardware or implemented in the firmware that serves as a root of trust. TPM provides a PCR for storing the hashes of booting layers. A typical TPM provides several functions, like a random number generator, counters, or a local clock. It also stores 24 PCRs, grouped by banks per each supported cryptographic hash function (SHA1, SHA256, SHA384, or SHA512).

Usually, by default, TPM usage is disabled. Therefore, the measured boot does not take place. To enable the remote attestation, enable TPM in the EFI/BIOS menu.

2.1 About Keylime

Keylime is a remote attestation solution that enables you to monitor the health of remote nodes using a TPM as a root of trust for measurement. With *Keylime*, you can perform several tasks like validation of the PCRs extended during the measured boot, analysis and assertions of the event log, assertion of the value of any PCR in the remote system, and also monitoring the validity of open or executed files.

Keylime also provides a framework for delivering encrypted data to verified nodes. This data is the so-called payload. The framework can also execute custom scripts that are triggered when a machine fails the attested measurements.

Keylime also provides a framework for delivering encrypted data to verified nodes. This data is the so-called payload. The framework can also execute custom scripts that are triggered when a machine fails the attested measurements.

Keylime consists of an agent, a verifier, a registrar, and a command-line tool (tenant). Agents are on those systems that need to be attested. The verifier and registrar are on remote systems that perform the registration and attestation of agents. Keep in mind that only the agent role is available on SLE Micro. For details about each component, refer to the following sections.

2.1.1 *Keylime* agent

The agent is a service that runs on the system that needs to be attested. The agent sends the event log, IMA hashes, and information about the measured boot to the verifier, using the local TPM as a certifier of the data validity.

When a new agent is started, the agent needs to register itself in the registrar first. To do so, the agent needs a TLS certificate to establish the connection. The TLS certificate is generated by the registrar, but it needs to be installed manually to the agent. After the registration, the agent sends its attestation key and the public part of the endorsement key to the registrar. The registrar responds to the agent with a challenge in a process called credential activation, which will validate the TPM of the agent. Once the agent has been registered, it is ready to be enrolled for attestation.

2.1.2 *Keylime* registrar

The registrar is used to register agents that should be attested. The registrar collects the agent's attestation key, the public part of the endorsement key and the endorsement key certification, and verifies that the agent attestation key belongs to the endorsement key.

2.1.3 *Keylime* verifier

The verifier performs the actual attestation of agents and continuously pulls the required attestation data from agents (amongst others, the PCR values, IMA logs, and UEFI event logs).

2.2 Setting up the verifier and registrar

Before you start installing and registering agents, prepare the verifier and the registrar on remote hosts as described in the following procedure.

1. Check the content of configuration files in `/usr/etc/keylime/*.conf`. The defaults are usually sufficient without any changes, but if you need to adjust the configuration, add the changes to either `/etc/keylime/verifier.conf.d/` or `/etc/keylime/registrar.conf.d/`. Set the ownership to `keylime:tss` and change the permissions as follows (for the verifier node):

```
# chown -R keylime:tss /etc/keylime/verifier.conf.d/

# chmod -R 600 /etc/keylime/verifier.conf.d/
```

2. Start and enable the verifier service:

```
# systemctl enable --now keylime_verifier.service
```

Starting the service generates certificates that are further required by the registrar.

3. If the previous command completes successfully, you can start and enable the registrar service:

```
# systemctl enable --now keylime_registrar.service
```

2.3 Installing the *Keylime* agent

The *Keylime* agent is not present on SLE Micro by default, you need to install it manually. To install the agent, proceed as follows:

1. Install the `rust-keylime` package as follows:

```
# transactional-update pkg in rust-keylime
```

2. Reboot your system.

3. Adjust the default agent's configuration.

- a. Create a directory to store a new configuration file for your changes in `/etc/keylime/agent.conf.d/`. The default configuration is stored in `/usr/etc/keylime/agent.conf`, but we do not recommend editing this file, because it might get overwritten with next system updates.

```
# mkdir -p /etc/keylime/agent.conf.d
```

b. Create a new file `/etc/keylime/agent.conf.d/agent.conf`:

```
# cat << EOF > /etc/keylime/agent.conf.d/agent.conf
[agent]

uuid = "d111ec46-34d8-41af-ad56-d560bc97b2e8"
registrar_ip = "<REMOTE_IP>"
revocation_notification_ip = "<REMOTE_IP>"
EOF
```

where:

- `uuid` is generated each time the agent is run. However, you can define a specific value by this option.
- `<REMOTE_IP>` is an IP address of the registrar.
- `<REMOTE_IP>` is an IP address of the verifier.

c. Change the owner of the `/etc/keylime/` directory to `keylime:tss`:

```
# chown -R keylime:tss /etc/keylime
```

d. Change permissions on the `/etc/keylime/` directory:

```
# chmod -R 600 /etc/keylime
```

4. Copy the certificates generated by the CA to the agent node. On the agent node, run:

a. Prepare a directory for the certificate:

```
# mkdir -p /var/lib/keylime/cv_ca
```

b. Copy the certificate to the agent:

```
# scp CERT_SERVER_ADDRESS:/var/lib/keylime/cv_ca/cacert.crt /var/lib/keylime/
cv_ca
```

c. Change the owner of the certificate to `keylime`:

```
# chown -R keylime:tss /var/lib/keylime/cv_ca
```

5. Start and enable the `keylime_agent.service`:

```
# systemctl enable --now keylime_agent.service
```

2.4 Registering agents



Note: SLE Micro does not provide the tenant, registrar or verifier

SLE Micro provides only the *Keylime* agent capability, therefore, the tasks performed in this chapter cannot be performed from your SLE Micro.

You can register a new agent either by using the CLI tenant or by editing the configuration of the verifier. Using the tenant on the verifier host, run the following:

```
# keylime_tenant -v 127.0.0.1 \  
    -t AGENT \  
    -u UUID \  
    --cert default \  
    -c add  
    [--include PATH_TO_ZIP_FILE]
```

Where:

- AGENT is an IP address of the agent to be registered.
- UUID is the agent's UUID.
- the file passed by the include option is used to deliver secret payload data to the agent. For details, refer to [Section 2.5, "Secure payloads"](#).

You can list registered agents by using the **reglist** command on the verifier host as follows:

```
# keylime_tenant -v 127.0.0.1 \  
    --cert default \  
    -c reglist
```

To remove a registered agent, specify the agent using the -t and -u options and the -c delete command as follows:

```
# keylime_tenant -v 127.0.0.1 \  
    -t AGENT -u UUID -c delete
```

```
-t AGENT \  
-u UUID \  
-c delete
```

2.5 Secure payloads

A secure payload enables you to deliver encrypted data to healthy agents. Typically, these payloads are used to provide keys, passwords, certificates, configurations, or scripts that are further used by the agent.

The secure payload is delivered to the agent in a zip file that must contain a shell script—autorun.sh. The script will be executed only if the agent has been properly registered and verified. To deliver the zip file, use the `--include` option of the **keylime_tenant** command.

The script autorun.sh contains steps that will enable the use of passwords, certificates and so on. For example, the script can create a directory structure and copy SSH keys there:

```
#!/bin/bash  
  
mkdir -p /root/.ssh/  
cp id_rsa* /root/.ssh/  
chmod 600 /root/.ssh/id_rsa*  
cp /root/.ssh/id_rsa.pub /root/.ssh/authorized_keys
```

In this case, do not forget to include the SSH keys in the zip file.

2.6 Enabling IMA tracking

When using IMA, the kernel calculates a hash of accessed files. The hash is then used to extend the PCR 10 in the TPM and also log a list of accessed files. The verifier can request a signed quote to the agent for PCR 10 to get the logs of all accessed files including the file hashes. Verifiers then compare the accessed files with a local allowlist of approved files. If any of the hashes are not recognized, the system is considered unsafe, and a revocation event is triggered.

For a high-level overview of IMA/EVM, refer to [IMA/EVM introduction \(https://en.opensuse.org/SDB:Ima_evm#Introduction\)](https://en.opensuse.org/SDB:Ima_evm#Introduction).

Before *Keylime* can collect information, IMA/EVM needs to be enabled. To enable the process, boot a kernel of the agent with the parameters: `ima_appraise=log` and `ima_policy=tcb`. To use the boot parameters on boot, proceed as follows:

1. Update the `GRUB_CMDLINE_LINUX_DEFAULT` option with the parameters in `/etc/default/grub`:

```
GRUB_CMDLINE_LINUX_DEFAULT="ima_appraise=log ima_policy=tcb"
```

2. Regenerate `grub.cfg` by running:

```
# transactional-update grub.cfg
```

3. Reboot your system.

The procedure above uses the default kernel IMA policy, but we recommend creating a new policy to avoid monitoring too many files and therefore creating long logs. For details, refer to the *Keylime* documentation (https://keylime-docs.readthedocs.io/en/latest/user_guide/runtime_ima.html) [↗](#).

To indicate the expected hashes, use the `--allowlist` option of the `keylime_tenant` command when registering the agent. To view the excluded or ignored files, use the `--exclude` option of the `keylime_tenant` command:

```
# keylime_tenant --allowlist  
-v 127.0.0.1 \  
-u UUID
```

3 Authentication with PAM

Linux uses PAM (pluggable authentication modules) in the authentication process as a layer that mediates between user and application. PAM modules are available on a system-wide basis, so they can be requested by any application. This chapter describes how the modular authentication mechanism works and how it is configured.

3.1 What is PAM?

System administrators and programmers often want to restrict access to certain parts of the system or to limit the use of certain functions of an application. Without PAM, applications must be adapted every time a new authentication mechanism, such as LDAP, Samba, or Kerberos, is introduced. However, this process is time-consuming and error-prone. One way to avoid these drawbacks is to separate applications from the authentication mechanism and delegate authentication to centrally managed modules. Whenever a newly required authentication scheme is needed, it is sufficient to adapt or write a suitable *PAM module* for use by the program in question. The PAM concept consists of:

- *PAM modules*, which are a set of shared libraries for a specific authentication mechanism.
- A *module stack* with of one or more PAM modules.
- A PAM-aware *service* which needs authentication by using a module stack or PAM modules. Usually a service is a familiar name of the corresponding application, like login or su. The service name other is a reserved word for default rules.
- *Module arguments*, with which the execution of a single PAM module can be influenced.
- A mechanism evaluating each *result* of a single PAM module execution. A positive value executes the next PAM module. The way a negative value is dealt with depends on the configuration: “no influence, proceed” up to “terminate immediately” and anything in between are valid options.

3.2 Structure of a PAM configuration file

PAM on SLE Micro comes with a set of configuration files stored in `/etc/pam.d`. Every service (or program) that relies on the PAM mechanism has its own configuration file in this directory. For example, the service for `sshd` can be found in the `/etc/pam.d/sshd` file.

The files under `/etc/pam.d/` define the PAM modules used for authentication. Each file consists of lines, which define a service, and each line consists of a maximum of four components:

```
TYPE  CONTROL
MODULE_PATH  MODULE_ARGS
```

The components have the following meaning:

TYPE

Declares the type of the service. PAM modules are processed as stacks. Different types of modules have different purposes. For example, one module checks the password, another verifies the location from which the system is accessed, and yet another reads user-specific settings. PAM knows about four different types of modules:

auth

Check the user's authenticity, traditionally by querying a password. However, this can also be achieved with a chip card or through biometrics (for example, fingerprints or iris scan).

account

Modules of this type check if the user has general permission to use the requested service. As an example, such a check should be performed to ensure that no one can log in with the user name of an expired account.

password

The purpose of this type of module is to enable the change of an authentication token. Usually this is a password.

session

Modules of this type are responsible for managing and configuring user sessions. They are started before and after authentication to log login attempts and configure the user's specific environment .

CONTROL

Indicates the behavior of a PAM module. Each module can have the following control flags:

required

A module with this flag must be successfully processed before the authentication may proceed. After the failure of a module with the required flag, all other modules with the same flag are processed before the user receives a message about the failure of the authentication attempt.

requisite

Modules having this flag must also be processed successfully, in much the same way as a module with the required flag. However, in case of failure a module with this flag gives immediate feedback to the user and no further modules are processed. In case of success, other modules are subsequently processed, like any modules with the required flag. The requisite flag can be used as a basic filter checking for the existence of certain conditions that are essential for a correct authentication.

sufficient

After a module with this flag has been successfully processed, the requesting application receives an immediate message about the success and no further modules are processed, provided there was no preceding failure of a module with the required flag. The failure of a module with the sufficient flag has no direct consequences, in the sense that any subsequent modules are processed in their respective order.

optional

The failure or success of a module with this flag does not have any direct consequences. This can be useful for modules that are only intended to display a message (for example, to tell the user that mail has arrived) without taking any further action.

include

If this flag is given, the file specified as argument is inserted at this place.

MODULE_PATH

Contains a full file name of a PAM module. It does not need to be specified explicitly, as long as the module is located in the default directory /lib/security (for all 64-bit platforms supported by SUSE® Linux Enterprise Micro, the directory is /lib64/security).

MODULE_ARGS

Contains a space-separated list of options to influence the behavior of a PAM module, such as debug (enables debugging) or nullok (allows the use of empty passwords).

In addition, there are global configuration files for PAM modules under `/etc/security`, which define the exact behavior of these modules (examples include `pam_env.conf` and `time.conf`). Every application that uses a PAM module actually calls a set of PAM functions, which then process the information in the various configuration files and return the result to the requesting application.

To simplify the creation and maintenance of PAM modules, common default configuration files for the types `auth`, `account`, `password`, and `session` modules have been introduced. These are retrieved from every application's PAM configuration. Updates to the global PAM configuration modules in `common-*` are thus propagated across all PAM configuration files without requiring the administrator to update every single PAM configuration file.

The global PAM configuration files are maintained using the **pam-config** tool. This tool automatically adds new modules to the configuration, changes the configuration of existing ones or deletes modules (or options) from the configurations. Manual intervention in maintaining PAM configurations is minimized or no longer required.

3.3 The PAM configuration of sshd

Consider the PAM configuration of `sshd` as an example:

EXAMPLE 3.1: PAM CONFIGURATION FOR SSHD (`/etc/pam.d/sshd`)

```
#%PAM-1.0 ❶
auth    requisite    pam_nologin.so          ❷
auth    include      common-auth          ❸
account requisite    pam_nologin.so          ❷
account include      common-account       ❸
password include     common-password      ❸
session required     pam_loginuid.so       ❹
session include      common-session       ❸
session optional     pam_lastlog.so       ❺ silent noupdate showfailed
```

- ❶ Declares the version of this configuration file for PAM 1.0. This is merely a convention, but could be used in the future to check the version.
- ❷ Checks, if `/etc/nologin` exists. If it does, no user other than `root` may log in.
- ❸ Refers to the configuration files of four module types: `common-auth`, `common-account`, `common-password`, and `common-session`. These four files hold the default configuration for each module type.
- ❹ Sets the login UID process attribute for the process that was authenticated.

- 5 Displays information about the last login of a user.

By including the configuration files instead of adding each module separately to the respective PAM configuration, you automatically get an updated PAM configuration when an administrator changes the defaults. Formerly, you needed to adjust all configuration files manually for all applications when changes to PAM occurred or a new application was installed. Now the PAM configuration is made with central configuration files and all changes are automatically inherited by the PAM configuration of each service.

The first include file (`common-auth`) calls three modules of the `auth` type: `pam_env.so`, `pam_gnome_keyring.so` and `pam_unix.so`. See [Example 3.2, “Default configuration for the auth section \(common-auth\)”](#).

EXAMPLE 3.2: DEFAULT CONFIGURATION FOR THE `auth` SECTION (`common-auth`)

```
auth    required    pam_env.so                ①
auth    optional    pam_gnome_keyring.so       ②
auth    required    pam_unix.so try_first_pass ③
```

- ① `pam_env.so` loads `/etc/security/pam_env.conf` to set the environment variables as specified in this file. It can be used to set the `DISPLAY` variable to the correct value, because the `pam_env` module knows about the location from which the login is taking place.
- ② `pam_gnome_keyring.so` checks the user's login and password against the GNOME key ring
- ③ `pam_unix` checks the user's login and password against `/etc/passwd` and `/etc/shadow`.

The whole stack of `auth` modules is processed before `sshd` gets any feedback about whether the login has succeeded. All modules of the stack having the `required` control flag must be processed successfully before `sshd` receives a message about the positive result. If one of the modules is not successful, the entire module stack is still processed and only then is `sshd` notified about the negative result.

When all modules of the `auth` type have been successfully processed, another include statement is processed, in this case, that in [Example 3.3, “Default configuration for the account section \(common-account\)”](#). `common-account` contains only one module, `pam_unix`. If `pam_unix` returns the result that the user exists, `sshd` receives a message announcing this success and the next stack of modules (password) is processed, shown in [Example 3.4, “Default configuration for the password section \(common-password\)”](#).

EXAMPLE 3.3: DEFAULT CONFIGURATION FOR THE `account` SECTION (`common-account`)

```
account required    pam_unix.so try_first_pass
```

EXAMPLE 3.4: DEFAULT CONFIGURATION FOR THE `password` SECTION (`common-password`)

```
password requisite pam_cracklib.so
password requisite pam_cracklib.so
password required pam_unix.so use_authtok nullok shadow try_first_pass
```

Again, the PAM configuration of `sshd` involves only an include statement referring to the default configuration for `password` modules located in `common-password`. These modules must successfully be completed (control flags `requisite` and `required`) whenever the application requests the change of an authentication token.

Changing a password or another authentication token requires a security check. This is achieved with the `pam_cracklib` module. The `pam_unix` module used afterward carries over any old and new passwords from `pam_cracklib`, so the user does not need to authenticate again after changing the password. This procedure makes it impossible to circumvent the checks carried out by `pam_cracklib`. Whenever the account or the auth type are configured to complain about expired passwords, the `password` modules should also be used.

EXAMPLE 3.5: DEFAULT CONFIGURATION FOR THE `session` SECTION (`common-session`)

```
session required pam_selinux.so close
session optional pam_systemd.so
session required pam_limits.so
session required pam_unix.so try_first_pass
session optional pam_umask.so
session required pam_selinux.so open
session optional pam_env.so
```

As the final step, the modules of the `session` type (bundled in the `common-session` file) are called to configure the session according to the settings for the user in question. The `pam_limits` module loads the file `/etc/security/limits.conf`, which may define limits on the use of certain system resources. The `pam_unix` module is processed again. The `pam_umask` module can be used to set the file mode creation mask. Since this module carries the `optional` flag, a failure of this module would not affect the successful completion of the entire session module stack. The `session` modules are called a second time when the user logs out.

3.4 Configuration of PAM modules

Some PAM modules are configurable. The configuration files are located in `/etc/security`. This section briefly describes the configuration files relevant to the `sshd` example—`pam_env.conf` and `limits.conf`.

3.4.1 pam_env.conf

`pam_env.conf` can be used to define a standardized environment for users that is set whenever the `pam_env` module is called. With it, preset environment variables using the following syntax:

```
VARIABLE [DEFAULT=VALUE] [OVERRIDE=VALUE]
```

VARIABLE

Name of the environment variable to set.

[DEFAULT=<value>]

Default VALUE the administrator wants to set.

[OVERRIDE=<value>]

Values that may be queried and set by `pam_env`, overriding the default value.

A typical example of how `pam_env` can be used is the adaptation of the `DISPLAY` variable, which is changed whenever a remote login takes place. This is shown in [Example 3.6, “pam_env.conf”](#).

EXAMPLE 3.6: `PAM_ENV.CONF`

```
REMOTEHOST  DEFAULT=localhost          OVERRIDE=@{PAM_RHOST}  
DISPLAY     DEFAULT=${REMOTEHOST}:0.0  OVERRIDE=${DISPLAY}
```

The first line sets the value of the `REMOTEHOST` variable to `localhost`, which is used whenever `pam_env` cannot determine any other value. The `DISPLAY` variable in turn contains the value of `REMOTEHOST`. Find more information in the comments in `/etc/security/pam_env.conf`.

3.4.2 limits.conf

System limits can be set on a user or group basis in `limits.conf`, which is read by the `pam_limits` module. The file allows you to set hard limits, which may not be exceeded, and soft limits, which may be exceeded temporarily. For more information about the syntax and the options, see the comments in `/etc/security/limits.conf`.

3.5 Configuring PAM using pam-config

The `pam-config` tool helps you configure the global PAM configuration files (`/etc/pam.d/common-*`) and several selected application configurations. For a list of supported modules, use the `pam-config --list-modules` command. Use the `pam-config` command to maintain your

PAM configuration files. Add new modules to your PAM configurations, delete other modules or modify options to these modules. When changing global PAM configuration files, no manual tweaking of the PAM setup for individual applications is required.

A simple use case for **pam-config** involves the following:

1. **Auto-generate a fresh unix-style PAM configuration.** Let **pam-config** create the simplest possible setup which you can extend later on. The **pam-config --create** command creates a simple Unix authentication configuration. Pre-existing configuration files not maintained by **pam-config** are overwritten, but backup copies are kept as ***.pam-config-back-up**.
2. **Add a new authentication method.** Adding a new authentication method (for example, SSSD) to your stack of PAM modules comes down to a simple **pam-config --add --sss** command. SSSD is added wherever appropriate across all **common-*-pc** PAM configuration files.
3. **Add debugging for test purposes.** To make sure the new authentication procedure works as planned, turn on debugging for all PAM-related operations. The **pam-config --add --sss-debug** command turns on debugging for SSSD-related PAM operations.
4. **Query your setup.** Before you finally apply your new PAM setup, check if it contains all the options you wanted to add. The **pam-config --query --MODULE** command lists both the type and the options for the queried PAM module.
5. **Remove the debug options.** Finally, remove the debug option from your setup when you are entirely satisfied with its performance. The **pam-config --delete --sss-debug** command turns off debugging for the **pam_ssh.so** module. In case you had debugging options added for other modules, use similar commands to turn these off.

For more information on the **pam-config** command and the options available, refer to the manual page of **pam-config(8)**.

3.6 Manually configuring PAM

If you prefer to manually create or maintain your PAM configuration files, make sure to disable **pam-config** for these files.

When you create your PAM configuration files from scratch using the `pam-config --create` command, it creates symbolic links from the `common-*` to the `common-*-pc` files. `pam-config` only modifies the `common-*-pc` configuration files. Removing these symbolic links effectively disables `pam-config`, because `pam-config` only operates on the `common-*-pc` files and these files are not put into effect without the symbolic links.



Warning: Include `pam_systemd.so` in configuration

If you are creating your own PAM configuration, make sure to include `pam_systemd.so` configured as `session optional`. Not including the `pam_systemd.so` can cause problems with `systemd` task limits. For details, refer to the man page of `pam_systemd.so`.

3.7 Configuring U2F keys for local login

To provide more security during the local login, you can configure two-factor authentication using the `pam-u2f` framework and the U2F feature on Yubikeys and Security Keys.

To set up U2F on your system, you need to associate your key with your account. After that, configure your system to use the key. The procedure is described in the following sections.

3.7.1 Associating the U2F key with your account

To associate your U2F key with your account proceed as follows:

1. Log in to your machine.
2. Insert your U2F key.
3. Create a directory for the U2F key configuration:

```
> sudo mkdir -p ~/.config/Yubico
```

4. Run the `pamu2fcfg` that outputs configuration lines:

```
> sudo pamu2fcfg > ~/.config/Yubico/u2f_keys
```

5. When your device begins flashing, touch the metal contact to confirm the association.

We recommend using a backup U2F device, which you can set up by running the following commands:

1. Run:

```
> sudo pamu2fcfg -n >> ~/.config/Yubico/u2f_keys
```

2. When your device begins flashing, touch the metal contact to confirm the association.

You can move the output file from the default location to a directory that requires the `sudo` permission to modify the file to increase security, for example, to the `/etc` directory. To do so, follow the steps:

1. Create a directory in `/etc`:

```
> sudo mkdir /etc/Yubico
```

2. Move the created file:

```
> sudo mv ~/.config/Yubico/u2f_keys /etc/Yubico/u2f_keys
```



Note: Placing the `u2f_keys` to a non-default location

If you move the output file to a different directory than is the default (`$HOME/.config/Yubico/u2f_keys`), you need to add the path to the `/etc/pam.d/login` file as described in [Section 3.7.2, “Updating the PAM configuration”](#).

3.7.2 Updating the PAM configuration

After you have created the U2F keys configuration, you need to adjust the PAM configuration on your system.

1. Open the file `/etc/pam.d/login`.
2. Add the line `auth required pam_u2f.so` to the file as follows:

```
#%PAM-1.0
auth      include      common-auth
auth      required     pam_u2f.so
account   include      common-account
password  include      common-password
```

```
session optional pam_keyinit.so revoke
session include common-session
#session optional pam_xauth.so
```

3. If you placed the `u2f_keys` file to a different location than `$HOME/.config/Yubi-co/u2f_keys`, you need to use the `authfile` option in the `/etc/pam.d/login` PAM file as follows:

```
#!/PAM-1.0
auth requisite pam_nologin.so
auth include common-auth
auth required pam_u2f.so authfile=<PATH_TO_u2f_keys>
...
```

where `<PATH_TO_u2f_keys>` is the absolute path to the `u2f_keys` file.

4 Enabling compliance with FIPS 140-3

FIPS 140-3 is a security accreditation program for validating cryptographic modules produced by private companies. The Federal Information Processing Standards (FIPS) Publication 140 is a series of computer security standards developed by the National Institute of Standards and Technology (NIST) to ensure the quality of cryptographic modules.


If your organization does any work for the United States federal government, it is likely that your cryptography applications (such as openssl, GnuTLS, and OpenJDK) will be required to be in compliance with Federal Information Processing Standards (FIPS) 140-3. If your organization is not required by compliance rules to run SUSE Linux Enterprise in FIPS mode, it is most likely best to not do it. This chapter provides guidance on enabling FIPS mode, and links to resources with detailed information.



Important: SUSE Linux Enterprise Micro5.4 and FIPS 140-3

The relevant binaries are currently undergoing FIPS 140-3 certification. Until the certification has been achieved, full FIPS 140-3 compliance cannot be guaranteed.

4.1 FIPS overview

Every vendor that develops and maintains cryptographic applications and wants them to be tested for FIPS compliance must submit them to the Cryptographic Module Validation Program (CMVP) (see <https://csrc.nist.gov/projects/cryptographic-module-validation-program> .

The latest FIPS 140-3 standard was approved in March 2019 and replaces 140-2.

4.2 When to enable FIPS mode



Warning: FIPS requires expertise

Administering FIPS is complex and requires significant expertise. Implementing it correctly, testing and troubleshooting all require a high degree of knowledge.

Only run your SLE Micro in FIPS mode when it is required to meet compliance rules. Otherwise, we do not recommend running your systems in FIPS mode.

Below are some reasons to *not* use FIPS mode (if not required explicitly):

- FIPS is restrictive. It enforces the use of specific validated cryptographic algorithms and specific certified binaries that implement these validated algorithms. You must use only the certified binaries.
- Upgrades may break functionality.
- The approval process is very long, so certified binaries are always several releases behind the newest release.
- Certified binaries, such as `ssh`, `sshd` and `sftp-server`, run their own self-checks at start-up and run only when these checks succeed. This creates a small performance degradation.
- Administering FIPS is complex and requires significant expertise.

4.3 Installing FIPS

To install the FIPS pattern on a running system, proceed as follows:

1. Install the `patterns-microos-fips` pattern:

```
# transactional-update pkg install -t pattern microos-fips
```

2. Reboot your system.

3. Add the kernel command line parameter `fips=1` to the boot loader configuration. To do so, edit the file `/etc/default/grub` as follows:

```
GRUB_CMDLINE_LINUX_DEFAULT="... fips=1..."
```

4. After logging in to the system, run

```
# transactional-update grub.cfg
```

5. Reboot your system.

Alternatively, you can install the pattern during the manual installation under *Software* as described in Book “Deployment Guide”, Chapter 12 “Installation steps”, Section 12.9 “Installation Settings”. Then adjust the boot loader configuration as described in the procedure above.



Important: Undergoing FIPS 140-3 certification

The relevant binaries are currently undergoing FIPS 140-3 certification. Until the certification has been achieved, full FIPS 140-3 compliance cannot be guaranteed



Note: Installing and enabling FIPS on a running system

If you install and enable the FIPS mode on a running system, you might need to make adjustments, such as regenerating keys and auditing your setup to ensure it is set up correctly.

4.4 Running containers on SLE Micro

If you run SLE Micro in the FIPS mode and you use only the SLE 15 SP4 BCI-based containers, then such a setup can serve as a FIPS-compliant platform. If you intend to run a third party container on SLE Micro, check the container's FIPS compatibility before deploying it.

4.5 MD5 not supported in Samba/CIFS

According to the FIPS standards, MD5 is not a secure hashing algorithm, and it must not be used for authentication. If you run a FIPS-compliant network environment, and you have clients or servers that run in FIPS-compliant mode, you must use a Kerberos service for authenticating Samba/CIFS users. This is necessary as all other Samba authentication modes include MD5.

4.6 More information

For more information, refer to:

- Man 8 [fips-mode-setup](#)
- Man 8 [fips-finish-install](#)
- Man 7 [crypto-policies](#)
- Man 8 [update-crypto-policies](#)

A GNU licenses

This appendix contains the GNU Free Documentation License version 1.2.

GNU Free Documentation License

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary

formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <https://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

```
Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.