



# SUSE Linux Enterprise Micro Security Guide

# SUSE Linux Enterprise Micro Security Guide

## WHAT?

This book describes the basics of SELinux and remote attestation using Keylime, and focuses on securing local access to a machine using PAM.

## WHY?

You would like to be introduced to security solutions available on SUSE Linux Enterprise Micro.

## EFFORT

The SELinux topic takes approx. 40 minutes to read. The PAM configuration takes approx. 20 minutes to read. The Keylime attestation takes approx. 15 minutes.

## GOAL

A better understanding of SELinux, PAM and Keylime.

Publication Date: 07 Aug 2025

<https://documentation.suse.com> 

# Contents

<b>1</b>	<b>SELinux 1</b>
1.1	About SELinux 1
1.2	Getting SELinux 1
1.3	SELinux modes 1
	Changing the SELinux mode 2 • Verifying that SELinux is functional 3
1.4	SELinux security context 4
1.5	SELinux policy overview 6
	Working with SELinux modules 6 • Creating policies for containers 6
1.6	SELinux Booleans 8
	Working with Booleans 8
1.7	Tools for managing SELinux 10
	Using the Z option 10 • The <b>chcon</b> command 10 • <b>getenforce</b> and <b>setenforce</b> commands 12 • The <b>fixfiles</b> script 12 • The <b>semanage</b> command 13 • The <b>sestatus</b> command 18
1.8	SELinux troubleshooting 19
	Switching on the logging service 19 • The /var/log/audit file 19 • Analyzing /var/log/audit/audit.log with <b>audit2allow</b> 21
<b>2</b>	<b>PAM on SLE Micro 24</b>
2.1	Introduction to PAM 24
2.2	Structure of PAM configuration 24
	An example of PAM configuration 27
2.3	Configuration of PAM modules 29
	pam_env.conf 30 • limits.conf 30
2.4	Configuring PAM using <b>pam-config</b> 30
2.5	Manually configuring PAM 31

- 2.6 Configuring SLE Micro to require U2F keys for local login 32  
Associating the U2F key with your account 32 • Updating the PAM configuration 33

## 3 Attestation using Keylime 35

- 3.1 Remote attestation using Keylime 35  
Terminology 35 • What is Keylime? 36 • Architecture 37 • Setting up the verifier, registrar and tenant 38 • Installing the agent 40 • Registering the agent 41 • Secure payloads 42 • Enabling IMA tracking 43
- 3.2 For more information 43

## A Legal Notice 45

## B GNU Free Documentation License 46

# 1 SELinux

## 1.1 About SELinux

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) to make access control 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.

## 1.2 Getting SELinux

SELinux is present on the pre-built images by default. However, in rare cases when 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 initrd.

## 1.3 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, we do not recommend using the disabled mode.

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. This is the default SELinux mode on SLE Micro, regardless of the deployment type.

For information about switching between SELinux modes, refer to [Section 1.3.1, “Changing the SELinux mode”](#).

## 1.3.1 Changing the SELinux mode

You can switch the SELinux mode temporarily or permanently.

### 1.3.1.1 Changing the SELinux mode temporarily

To set SELinux to the permissive or enforcing mode temporarily, use the command **setenforce**.

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.

Remember that you cannot disable SELinux using the **setenforce** command.

### 1.3.1.2 Changing the SELinux mode permanently

To perform changes to the SELinux mode that persists rebooting of the system, edit the /etc/selinux/config configuration file. In this file, you can also disable SELinux on your system. However, this action is not recommended. If SELinux is possibly causing issues to your system, switch to the permissive mode instead and debug your system.

In the file /etc/selinux/config, change the value of SELINUX to permissive, or enforced as follows:

```
SELINUX=permissive
```

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.

#### 1.3.1.3 Verifying the active SELinux mode

To verify the mode, run the following command:

```
# getenforce
```

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

#### 1.3.2 Verifying that SELinux is functional

If you are performing configuration changes, it may be useful to switch to permissive mode. During this time, users might label files incorrectly, and thus cause problems when switching back to enforcing mode.

To return the system back to its secured state, perform the following steps:

1. Reset the security context:

```
> sudo restorecon -R /
```

2. Switch to enforcing mode by setting `SELINUX=enforcing` in the `/etc/selinux/config`.
3. Reboot the system and log in again.

4. Run the **`sestatus -v`** command. It should give you an output similar to the following one:

```
> sudo sestatus -v
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:     33

Process contexts:
Current context:                unconfined_u:unconfined_r:unconfined_t:s0-
s0:c0.c1023
Init context:                   system_u:system_r:init_t:s0
/usr/sbin/sshd                  system_u:system_r:sshd_t:s0-s0:c0.c1023

File contexts:
Controlling terminal:          unconfined_u:object_r:user_tty_device_t:s0
/etc/passwd                    system_u:object_r:passwd_file_t:s0
/etc/shadow                    system_u:object_r:shadow_t:s0
/bin/bash                      system_u:object_r:shell_exec_t:s0 \
-> system_u:object_r:shell_exec_t:s0
/bin/login                     system_u:object_r:login_exec_t:s0
/bin/sh                        system_u:object_r:bin_t:s0 \
-> system_u:object_r:shell_exec_t:s0
/sbin/agetty                   system_u:object_r:bin_t:s0 \
-> system_u:object_r:getty_exec_t:s0
/sbin/init                     system_u:object_r:bin_t:s0 -> \
system_u:object_r:init_exec_t:s0
/usr/sbin/sshd                 system_u:object_r:sshd_exec_t:s0
```

5. If the system is not working properly, check the log files in `/var/log/audit/audit.log`. For more details, refer to [SELinux troubleshooting \(https://documentation.suse.com/sle-micro/6.0/html/Micro-setroubleshoot/setroubleshoot.html\)](https://documentation.suse.com/sle-micro/6.0/html/Micro-setroubleshoot/setroubleshoot.html).

## 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 user**

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 only one SELinux user. However, one SELinux user can have several roles.

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`.

When a new Linux account is created and the SELinux user is not assigned to the account, the default SELinux user is used. Usually, the default value is `unconfined_u`. For a procedure on how to change the default value, refer to [Section 1.7.5.2, “The `semanage login` command”](#).

**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**

The 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**

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

**category**

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.

Here is an example of an SELinux context:

```
allow user_t bin_t:file {read execute gettattr};
```

This example rule states that the user who has the context type `user_t` (this user is called the source object) is allowed to access objects of the class *file* with the context type `bin_t` (the target), using the permissions `read`, `execute` and `gettattr`.

## 1.5 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. We strongly recommend using a modular policy and not a monolithic policy. Modular policies are much easier to manage.

SLE Micro is shipped with the targeted SELinux policy.

### 1.5.1 Working with SELinux modules

As an administrator, you can switch modules on or off. This can be useful if you want to disable only a part of the SELinux policy and you do not want to run a specific service without SELinux protection.

To view all SELinux policy modules in use, run the command:

```
semodule -l
```

After you get the name of the module you want to switch off, run the command:

```
> sudo semodule -d MODULENAME
```

To switch on the policy module, run the command:

```
> sudo semodule -e MODULENAME
```

### 1.5.2 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 1.3.1, “Changing the SELinux mode”](#).

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
```

## 1.6 SELinux Booleans

SELinux Booleans support a flexible policy management approach. For example, Booleans enable you to disable a particular policy on one server, while keeping the same policy active on another one. In other words, a Boolean can be understood as a switch for a policy rule. Instead of changing a particular policy, you can switch it off. In the policy code, Booleans are called a *tunable*. Because Booleans are included in the policy, they are available as soon as a policy is loaded.

The changes to the Booleans value may be persistent or temporary, lasting until the end of the session.

SELinux offers tools that enable you to list and view details or change the state of Booleans. See the following sections for details.

### 1.6.1 Working with Booleans

#### 1.6.1.1 Listing Booleans

You can use the `getsebool` or `semanage` command to list currently defined Booleans. To list all currently defined Booleans, along with their state, run the following command:

```
# getsebool -a

abrt_anon_write --> off
abrt_handle_event --> off
abrt_upload_watch_anon_write --> on
...
```

To get more details about particular Booleans, you can use the `semanage` command as follows:

```
# semanage boolean -l
```

SELinux boolean	State	Default	Description
abrt_anon_write	(off , off)		Allow abrt to anon write
abrt_handle_event	(off , off)		Allow abrt to handle event
abrt_upload_watch_anon_write	(on , on)		Allow abrt to upload watch anon write

To get the status of an individual Boolean, you can use the following command:

```
# getsebool BOOLEAN_NAME
```

Alternatively, you can just use the **grep** command on the **semanage boolean** output:

```
# semanage boolean -l | grep BOOLEAN_NAME
```

### 1.6.1.2 Toggling Booleans

The commands **setsebool** and **semanage** can be used to toggle the value of Booleans. You can change the Boolean status persistently or just temporarily until the session ends. To change a Boolean value temporarily, run the following command:

```
# setsebool BOOLEAN_NAME BOOLEAN_VALUE
```

where **BOOLEAN\_VALUE** is either **on** or **off**.

To change a Boolean value persistently, run one of the following two commands:

```
# setsebool -P BOOLEAN_NAME BOOLEAN_VALUE
```

Alternatively, using the **semanage** command:

```
# semanage boolean -m --BOOLEAN_VALUE BOOLEAN_NAME
```

where **BOOLEAN\_VALUE** is either **on** or **off**.

A single Boolean can enable or disable several policy rules. To see which policy rules are enabled or disabled by specific Booleans, use the **sedispol** tool to analyze the policy file:

```
# sedispol /etc/selinux/targeted/policy/policy.32
```

As the policy rules are usually huge, we recommend setting an output file by selecting the **f** and specifying a file name. After specifying the file name, press **6**. Then you can inspect the file.

## 1.7 Tools for managing SELinux

SLE Micro provides you with tools to manage SELinux on your system. If 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.

### 1.7.1 Using the Z option

Where SELinux is installed and configured, you can use the `-Z` to regular commands like `ls`, `id` or `ps`. Using this option, you can display the security context of files or processes. For example, with the `ls` command:

```
> ls -Z /etc/shadow

system_u:object_r:shadow_t:s0 /etc/shadow
```

### 1.7.2 The `chcon` command

The command name `chcon` stands for change context. The command can change the full security context of a file to the value provided on the CLI, or it can change parts of the context. Alternatively, you can provide a file that serves as a reference.

To change the full security context of a file, the command syntax looks as follows:

```
# chcon SECURITY_CONTEXT FILENAME
```

where:

- `SECURITY_CONTEXT` is in the format: `SELinux_USER:ROLE:TYPE:LEVEL:CATEGORY`. For example, the context could be: `system_u:object_r:httpd_config_t:s0`.
- `FILENAME` is a path to the file whose context should be changed.

To set a security context according to a provided file that serves as a reference, run `chcon` as follows:

```
# chcon --reference=REFERENCE_FILE FILENAME
```

where:

- REFERENCE\_FILE is a path to a file that should be used as a reference.
- FILENAME is a path to the file whose context should be changed.

Alternatively, you can change only one part of the security context. The general syntax of the **chcon** command is as follows:

```
# chcon CONTEXT_OPTION CONTEXT_PART FILENAME
```

The options and arguments have the following meaning:

- depending on the context part, CONTEXT\_OPTION can be any of the following:

-u resp --user

denotes that an SELinux user context will be changed on the provided file:

```
# chcon -u system_u logind.conf
```

-r resp --role

only the role part will be changed in the context of the provided file:

```
# chcon -r object_r logind.conf
```

-t resp --type

only the type part will be changed in the context of the provided file:

```
# chcon -t etc_t logind.conf
```

-l resp --range

only the range part of the security context will be changed:

```
# chcon -l s0 logind.conf
```

- CONTEXT\_PART is the particular value of the security context to be set.
- FILENAME is a path to the file whose context will be changed.



## Note: Using **chcon** on symbolic links

By default, when you change the security context on a symbolic link, the context of the link target is changed and the symbolic link context is **not** changed. To force **chcon** to change the context of the symbolic link and not the link target, use the `--no-dereference` option as shown below:

```
# chcon --no-dereference -u system_u -t etc_t network.conf
```

You can change the context of all files in a directory by using the recursive option:

```
# chcon --recursive system_u:object_r:httpd_config_t:s0 conf.d
```

### 1.7.3 **getenforce** and **setenforce** commands

The **getenforce** command returns the current SELinux mode: Enforcing, Permissive or Disabled.

```
# getenforce
```

```
Permissive
```

The **setenforce** command temporarily changes the SELinux mode to enforcing or permissive. You cannot use this command to disable SELinux. Remember that the change persists only until the next reboot. To change the state permanently, follow the description in [Section 1.3.1, “Changing the SELinux mode”](#).

```
# setenforce MODE_ID
```

where MODE\_ID is 0 for the permissive mode or 1 for the enforced mode.

### 1.7.4 The **fixfiles** script

The script enables you to perform the following tasks with the security context:

- check if the context is correct
- change any incorrect file context labels
- relabel your system if you added a new policy



The script syntax is as follows:

```
# fixfiles [OPTIONS] ARGUMENT
```

where:

- OPTIONS can be the following:

-l LOGFILE

saves the output to the provided file

-o OUTPUT\_FILE

saves to the provided output file the names of all files whose file context differs from the default

-F

forces a reset of context

- ARGUMENT can be one of the following:

check

shows previous and current file context for an incorrect label without performing any changes

relabel

relabels incorrect file contexts according to the currently loaded policy

restore

restores incorrect file contexts to the default values

verify

lists all files with incorrect file context labels without performing any changes

## 1.7.5 The **semanage** command

The **semanage** command can be used to configure parts of the policy without the need to re-compile the policy from sources. The command enables you to perform the following tasks:

- manage Booleans by using the boolean argument. For details about Booleans, refer to [Section 1.6.1, "Working with Booleans"](#).
- adjust the context of files by using the fcontext argument

- manage user mappings using the login argument
- manage SELinux users using the user argument
- manage SELinux policy modules using the module argument

The general command syntax looks as follows:

```
# semanage ARGUMENT OPTIONS [OBJECT_NAME]
```

where:

- ARGUMENT is one of the following: **login**, **user**, **fcontext**, **boolean**, **module**.
- OPTIONS depends on the provided ARGUMENT. Common options are described in *Common options*.
- OBJECT\_NAME, depending on the provided ARGUMENT, can be a login name, module name, file name or SELinux user.

#### COMMON OPTIONS

-a, --add

adds a provided object

-h, --help

prints the command help

--extract

displays commands that were used to change the system (Booleans, file context, and so on)

-l, --list

lists all objects

-m, --modify

modifies the provided object

-n, --noheading

modifies the output of the listing operation by omitting headings

-s, --seuser

specifies the SELinux user

Other options are specific to particular semanage commands and are described in corresponding sections.

### 1.7.5.1 The **semanage fcontext** command

Using the **semanage fcontext** command, you can perform the following tasks:

- query file context definitions
- add contexts on files
- add your own rules

Changes performed to the file context using the **semanage fcontext** command do not require modifications or recompilation of the policy.

On top of the common options described in *Common options*, the **semanage fcontext** command takes the following options:

#### **-e, --equal**

The option enables you to use the context of the provided path context to label files in a different directory (the provided target path). For example, you want to assign the same context as `/home` has to an alternative home directory `/export/home`. If you use this option, you need to provide the source path and the target path:

```
# semanage fcontext -a -e /home /export/home
```

#### **-f, --ftype**

To specify a file type. Use one of the following values:

- **a** - all files, which is also the default value
- **b** - a block device
- **c** - a character device
- **d** - a directory
- **f** - regular files
- **l** - a symbolic link
- **p** - a named pipe
- **s** - a socket

### 1.7.5.2 The **semanage login** command

The **semanage login** enables you to perform the following tasks:

- Mapping of Linux users on a particular SELinux user. For example, to map the Linux user *tux* on *sysadm\_u*, run the command:

```
# semanage login -a -s sysadm_u tux
```

- Mapping of a group of Linux users on a particular SELinux user. For example, to map users of the *writers* group on *user\_u*, run the command:

```
# semanage login -a -s user_u %writers
```

The group is then listed in the output of **semanage login -l**, prefixed with the % character. Keep in mind that the user group should be primary because mapping SELinux users on supplementary groups may result in incompatible mappings.

```
# semanage login -m -s staff_u %writers
```

- Mapping of Linux users on a particular SELinux MLS/MCS security range.
- Modifying of the already created mapping. For this purpose, just replace the *-a* option with *-m* in the previous commands.
- Setting the default SELinux user for new Linux users. The usual default SELinux user is *unconfined\_u*. To change the value to *staff\_u*, run the command:

```
# semanage login -m -s staff_u __default__
```

### 1.7.5.3 The **semanage boolean** command

The **semanage boolean** command is used to control Booleans in the SELinux policy.

The command synopsis looks as follows:

```
semanage boolean [-h] [-n] [ --extract |  
--deleteall | --list [-C] | --modify ( --on | --off | -1 | -0 ) boolean ]
```

On top of the common options, you can use the following ones specific to the **semanage boolean** command:

**--list -C**

To display a list of local modifications to Booleans.

**-m --on | -1**

To switch the provided Boolean on.

**-m --off | -0**

To switch the provided Boolean off.

**-D, --deleteall**

To delete all local modifications to Booleans.

The most common usage of the command is to switch on or off a particular Boolean. For example, to switch on the `authlogin_yubikey` Boolean, run:

```
# semanage boolean -m on authlogin_yubikey
```

#### 1.7.5.4 The **semanage user** command

The **semanage user** command controls the mapping between the SELinux user and the roles and MLS/MCS levels.

On top of the common options described in *Common options*, the **semanage use** command takes the following options:

**-R [ROLES], --roles [ROLES]**

A list of SELinux roles. You can enclose multiple roles within double quotes and separate them by spaces, or you can use the `-R` several times.

Using this command, you can perform the following tasks:

- Listing the mapping of SELinux users on roles by running:

```
# semanage user -l
```

- Changing the roles assigned to the `user_u` SELinux user:

```
# semanage user -m -R "system_r unconfined_r user_r"
```

- Assigning to `admin_u` the role `staff_r` and a category `s0`:

```
# semanage user -a -R "staff_r -r s0 admin_u"
```

- Creating a new SELinux user, for example, `admin_u` with the `staff_r` role. You also need to define the labeling prefix for this user by using the `-P`:

```
# semanage user -a -R "staff_r" -P admin admin_u
```

### 1.7.5.5 The `semanage module` command

The `semanage module` command can install, remove, disable or enable SELinux policy modules. On top of the common options described in *Common options*, the `semanage fcontext` command takes the following options:

`-d, --disable`

To disable the provided SELinux policy module:

```
# semanage module --disable MODULE_NAME
```

`-e, --enable`

To enable the provided SELinux policy module:

```
# semanage module --enable MODULE_NAME
```

## 1.7.6 The `sestatus` command

The `sestatus` gets the status of a system where SELinux is running.

The generic syntax of the command looks as follows:

```
sestatus [OPTION]
```

When run without any options and arguments, the command outputs the following information:

```
# 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:    33
```

The command can take the following options:

**-b**

Displays the status of Booleans on the system.

**-v**

Displays the security context of files and processes listed in the /etc/sestatus.conf file.

## 1.8 SELinux troubleshooting

### 1.8.1 Switching on the logging service

By default, if SELinux is the reason something is not working, a log message to this effect is sent to the /var/log/audit/audit.log file. For the message to be sent, the auditd service must be running. If you see an empty /var/log/audit.log, start the auditd service using

```
> sudo systemctl start auditd
```

and enable it in the targets of your system, using

```
> sudo systemctl enable auditd
```

### 1.8.2 The /var/log/audit file

The /var/log/audit file stores messages of access denials, service events and so on.

In *Example 1.1: "Example lines from /etc/audit/audit.log"*, you can see a partial example of the contents of /var/log/audit/audit.log.

EXAMPLE 1.1: **EXAMPLE LINES FROM /etc/audit/audit.log**

```
type=DAEMON_START msg=audit(1348173810.874:6248): auditd start, ver=1.7.7 format=raw
kernel=3.0.13-0.27-default auid=0 pid=4235 subj=system_u:system_r:auditd_t res=success
```

```

type=AVC msg=audit(1348173901.081:292): avc: denied { write } for
  pid=3426 comm="smartd" name="smartmontools" dev=sda6 ino=581743
  scontext=system_u:system_r:fsdaemon_t tcontext=system_u:object_r:var_lib_t tclass=dir
type=AVC msg=audit(1348173901.081:293): avc: denied { remove_name } for pid=3426
  comm="smartd" name="smartd.WDC_WD2500BEKT_75PVMT0-WD_WXC1A21E0454.ata.state~" dev=sda6
  ino=582390 scontext=system_u:system_r:fsdaemon_t tcontext=system_u:object_r:var_lib_t
  tclass=dir
type=AVC msg=audit(1348173901.081:294): avc: denied { unlink } for pid=3426
  comm="smartd" name="smartd.WDC_WD2500BEKT_75PVMT0-WD_WXC1A21E0454.ata.state~" dev=sda6
  ino=582390 scontext=system_u:system_r:fsdaemon_t tcontext=system_u:object_r:var_lib_t
  tclass=file
type=AVC msg=audit(1348173901.081:295): avc: denied { rename } for pid=3426
  comm="smartd" name="smartd.WDC_WD2500BEKT_75PVMT0-WD_WXC1A21E0454.ata.state" dev=sda6
  ino=582373 scontext=system_u:system_r:fsdaemon_t tcontext=system_u:object_r:var_lib_t
  tclass=file
type=AVC msg=audit(1348173901.081:296): avc: denied { add_name } for pid=3426
  comm="smartd" name="smartd.WDC_WD2500BEKT_75PVMT0-WD_WXC1A21E0454.ata.state~"
  scontext=system_u:system_r:fsdaemon_t tcontext=system_u:object_r:var_lib_t tclass=dir
type=AVC msg=audit(1348173901.081:297): avc: denied { create } for pid=3426
  comm="smartd" name="smartd.WDC_WD2500BEKT_75PVMT0-WD_WXC1A21E0454.ata.state"
  scontext=system_u:system_r:fsdaemon_t tcontext=system_u:object_r:var_lib_t tclass=file
type=AVC msg=audit(1348173901.081:298): avc: denied { write open } for pid=3426
  comm="smartd" name="smartd.WDC_WD2500BEKT_75PVMT0-WD_WXC1A21E0454.ata.state" dev=sda6
  ino=582390 scontext=system_u:system_r:fsdaemon_t tcontext=system_u:object_r:var_lib_t
  tclass=file
type=AVC msg=audit(1348173901.081:299): avc: denied { getattr } for pid=3426
  comm="smartd" path="/var/lib/smartmontools/smartd.WDC_WD2500BEKT_75PVMT0-
WD_WXC1A21E0454.ata.state" dev=sda6 ino=582390 scontext=system_u:system_r:fsdaemon_t
  tcontext=system_u:object_r:var_lib_t tclass=file
type=AVC msg=audit(1348173901.309:300): avc: denied { append } for pid=1316

```

A single message looks as follows:

```

type=AVC msg=audit(1348173901.081:299): avc: denied { getattr } for
  pid=3426 comm="smartd" path="/var/lib/smartmontools/smartd.WDC_WD2500BEKT_75PVMT0-
WD_WXC1A21E0454.ata.state" dev=sda6 ino=582390 scontext=system_u:system_r:fsdaemon_t
  tcontext=system_u:object_r:var_lib_t tclass=file

```

Every line of the message can be broken down into sections. For example, the sections in the last line are:

type=AVC:

Every SELinux-related audit log line starts with the type identification, for example, type=AVC. Note that a message with the type=SYSCALL that follows one with a different type and has the same value of msg may provide further information regarding the event.



msg=audit(1348173901.309:300):

This is the time stamp, which is written in epoch time, the number of seconds that have passed since Jan 1, 1970. You can use **date -d** on the part up to the dot in the epoch time notation to find out when the event happened:

```
> date -d @1348173901
Thu Sep 20 16:45:01 EDT 2012
```

avc: denied { append }:

The specific action that was denied. In this case, the system has denied the appending of data to a file. While browsing through the audit log file, you can see other system actions, such as write open, getattr and more.

for pid=1316:

the process ID of the command or process that initiated the action

comm="rsyslogd":

the specific command that was associated with that PID

name="smartmontools":

the name of the subject of the action

dev=sda6 ino=582296:

the block device and inode number of the file that was involved

scontext=system\_u:system\_r:syslogd\_t:

the source context, which is the context of the initiator of the action

tclass=file:

a class identification of the subject

### 1.8.3 Analyzing /var/log/audit/audit.log with **audit2allow**

Instead of interpreting the events in /var/log/audit/audit.log yourself, you can use the **audit2allow** command. The command helps analyze the cryptic log messages in /var/log/audit/audit.log. An **audit2allow** troubleshooting session always consists of three different commands. First, you would use **audit2allow -w -a** to present the audit information in a more

readable way. The `audit2allow -w -a` by default works on the `audit.log` file. If you want to analyze a specific message in the `audit.log` file, copy it to a temporary file and analyze the file with:

```
> sudo audit2allow -w -i FILENAME
```

#### EXAMPLE 1.2: ANALYZING AUDIT MESSAGES

```
> sudo audit2allow -w -i testfile
type=AVC msg=audit(1348173901.309:300): avc: denied { append } for pid=1316
comm="rsyslogd" name="acpid" dev=sda6 ino=582296
scontext=system_u:system_r:syslogd_t tcontext=system_u:object_r:apmd_log_t tclass=file
```

This was caused by:

A missing type enforcement (TE) allow rule.

To generate a loadable module to allow this access, run

```
> sudo audit2allow
```

To find out which specific rule has denied access, you can use `audit2allow -a` to show the enforcing rules from all events that were logged into the `audit.log` file, or `audit2allow -i FILENAME` to show it for messages that you have stored in a specific file:

#### EXAMPLE 1.3: VIEWING WHICH LINES DENY ACCESS

```
> sudo audit2allow -i testfile
#===== syslogd_t =====
allow syslogd_t apmd_log_t:file append;
```

To create an SELinux module with the name `mymodule` that you can load to allow the access that was previously denied, run

```
> sudo audit2allow -a -R -M mymodule
```

If you want to do this for all events that have been logged into the `audit.log` file, use the `-a -M` command arguments. To do it only for specific messages that are in a specific file, use `-i -M` as in the example below:

#### EXAMPLE 1.4: CREATING A POLICY MODULE ALLOWING AN ACTION PREVIOUSLY DENIED

```
> sudo audit2allow -i testfile -M example
***** IMPORTANT *****
To make this policy package active, execute:
```

```
semodule -i example.pp
```

As indicated by the **audit2allow** command, you can now run this module by using the **semodule -i** command, followed by the name of the module that **audit2allow** has created for you (example.pp in the above example).

## 2 PAM on SLE Micro

### 2.1 Introduction to 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 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.

### 2.2 Structure of PAM configuration

PAM on SLE Micro comes with a so-called directory based configuration. The set of configuration files is 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.



## Note: File-based configuration (`/etc/pam.conf`) not used on SLE Micro

The configuration of each service can be also stored in `/etc/pam.conf`. However, for maintenance and usability reasons, this configuration scheme is not used in SUSE Linux Enterprise Micro.

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, if 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 calls a set of PAM functions, which then process the information in the 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.

### 2.2.1 An example of PAM configuration

To demonstrate a real use case example of PAM configuration, the configuration of `sshd` has been used in this section:

EXAMPLE 2.1: 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      silent noupdate showfailed ❺
```

- 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.

The first include file (`common-auth`) calls modules of the `auth` type: `pam_env.so`, `pam_gnome_keyring.so` and `pam_unix.so`. See [Example 2.2, “Default configuration for the auth section \(common-auth\)”](#). Keep in mind that the modules may differ according to your installation.

EXAMPLE 2.2: 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 2.3, “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 2.4, “Default configuration for the password section \(common-password\)”](#).

EXAMPLE 2.3: DEFAULT CONFIGURATION FOR THE `account` SECTION (`common-account`)

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

EXAMPLE 2.4: 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 2.5: 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.

## 2.3 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`.

### 2.3.1 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 2.6, “pam\\_env.conf”](#).

#### EXAMPLE 2.6: 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`.

### 2.3.2 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`.

## 2.4 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-backup**.
2. **Add a new authentication method.** Adding a new authentication method (for example, LDAP) to your stack of PAM modules comes down to a simple **pam-config --add --ldap** command. LDAP 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 --ldap-debug** turns on debugging for LDAP-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 the performance of it. The **pam-config --delete --ldap-debug** command turns off debugging for LDAP authentication. 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)**.

## 2.5 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`.

## 2.6 Configuring SLE Micro to require U2F keys for local login

To provide more security during the local login to SLE Micro, 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 SLE Micro system, you need to associate your key with your account on SLE Micro. After that, configure your system to use the key. The procedure is described in the following sections.

### 2.6.1 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` command 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, move it 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 2.6.2, “Updating the PAM configuration”](#).

## 2.6.2 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
```

<b>auth</b>	<b>required</b>	<b>pam_u2f.so</b>
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.

## 3 Attestation using Keylime

### 3.1 Remote attestation using Keylime

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). RA is often combined with public-key encryption (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. Remote attestation on SLE Micro is implemented by *Keylime*.

#### 3.1.1 Terminology

Remote attestation technology uses the following terms:

**Attestation key (AK)**

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**

Calculates its own hash and the hash of the next step in the boot process, initiating the chain of measurements.

**Endorsement key (EK)**

An encryption key that is permanently embedded in the TPM when it is manufactured. The public part of the key and the certification stored in the TPM are used to recognize a genuine TPM.

**Integrity management architecture (IMA)**

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

**Measured boot**

A method with which 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)

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)

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).



#### Note

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.

#### Secure payload

A mechanism to deliver encrypted data to healthy agents. Payloads are used to provide keys, passwords, certificates, configurations or scripts that are further used by the agent.

### 3.1.2 What is 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 multiple tasks, for example:

- Validate of the PCRs extended during the measured boot.
- Create analysis and make assertions of the event log.



- Make assertion of the value of any PCR in the remote system.
- Monitor the validity of open or executed files.
- Deliver encrypted data to verified nodes via *secure payloads*.
- Execute custom scripts that are triggered when a machine fails the attested measurements.

### 3.1.3 Architecture

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.

#### 3.1.3.1 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, it 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 validates the TPM of the agent. Once the agent has been registered, it is ready to be enrolled for attestation.

#### 3.1.3.2 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.

#### 3.1.3.3 Keylime verifier

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

### 3.1.4 Setting up the verifier, registrar and tenant



#### Note

The container described in this article delivers control plane services *verifier* and *registrar* and a *tenant* command-line tool (CLI) that are part of the Keylime project.

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

1. Identify the Keylime workload image.

```
# podman search keylime
[...]  
registry.opensuse.org/devel/microos/containers/containerfile/opensuse/keylime-  
control-plane
```

2. Pull the image from the registry.

```
# podman pull \\\n  registry.opensuse.org/devel/microos/containers/containerfile/opensuse/keylime-  
control-plane:latest
```

3. Create the keylime-control-plane volume to persist the database and certificates required during the attestation process.

```
# podman container runlabel install \\\n  registry.opensuse.org/devel/microos/containers/containerfile/opensuse/keylime-  
control-plane:latest
```

4. Start the container and related services.

```
# podman container runlabel run \\\n  registry.opensuse.org/devel/microos/containers/containerfile/opensuse/keylime-  
control-plane:latest
```

The keylime-control-plane container is created. It includes configured and running registrar and verifier services. Internally, the container exposes ports 8881, 8890 and 8891 to the host using the default values. Validate the firewall configuration to allow access to the ports and to allow communication between containers, because the tenant CLI requires it.



## Tip

If you need to stop Keylime services, run the following command:

```
# podman kill keylime-control-plane-container
```

### 3.1.4.1 Monitoring Keylime services

To get the status of running containers on the host, run the following command:

```
# podman ps
```

To view the logs of Keylime services, run the following command:

```
# podman logs keylime-control-plane-container
```

### 3.1.4.2 Executing the tenant CLI

The tenant CLI tool is included in the container, and if the host firewall does not interfere with the ports exposed by Keylime services, you can execute it using the same image, for example:

```
# podman run --rm \
-v keylime-control-plane-volume:/var/lib/keylime/ \
keylime-control-plane:latest \
keylime_tenant -v 10.88.0.1 -r 10.88.0.1 --cert default -c reglist
```

### 3.1.4.3 Extracting the Keylime certificate

The first time that the Keylime container is executed, its services create a certificate required by several agents. You need to extract the certificate from the container and copy it to the agent's /var/lib/keylime/cv\_ca/ directory.

```
# podman cp \
keylime-control-plane-container:/var/lib/keylime/cv_ca/cacert.crt
.
# scp cacert.crt
AGENT_HOST:/var/lib/keylime/cv_ca/
```



## Tip

Find more details about installing the agent in [Section 3.1.5, "Installing the agent"](#).

### 3.1.5 Installing the agent

The Keylime agent is not present on SLE Micro by default, and 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
```

Then reboot the system.

2. 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 may be overwritten in upcoming 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" ❶
registrar_ip = "<REMOTE_IP>" ❷
revocation_notification_ip = "<REMOTE_IP>" ❸
EOF
```

- ❶ The unique identifier is generated each time the agent is run. However, you can define a specific value by this option.
- ❷ IP address of the registrar.
- ❸ 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
```

3. Copy the certificates generated by the CA to the agent node. On the agent node, do the following:

- 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:tss`:

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

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

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

### 3.1.6 Registering the agent

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 \①  
-u UUID \②  
--cert default \  
-c add  
[--include PATH_TO_ZIP_FILE] ③
```

① AGENT is an IP address of the agent to be registered.

② UUID is the agent's unique identifier.

- ③ The file passed by the include option is used to deliver secret payload data to the agent. For details, refer to [Section 3.1.7, “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
```

## 3.1.7 Secure payloads

### 3.1.7.1 What is a secure payload?

A Keylime secure payload enables you to deliver encrypted data to healthy agents. Payloads are used to provide keys, passwords, certificates, configurations or scripts that are used by the Keylime agent at a later stage.

### 3.1.7.2 How does a secure payload work?

A secure payload is delivered to the agent in a zip file that must contain a shell script named autorun.sh. The script is 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.

For example, the following autorun.sh script creates a directory structure and copies SSH keys there. The related zip archive must include these SSH keys.

```
> cat autorun.sh  
#!/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
```

### 3.1.8 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.

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

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. To avoid monitoring too many files and therefore creating long logs, create a new custom policy. Find more details in the [Keylime documentation \(https://keylime-docs.readthedocs.io/en/latest/user\\_guide/runtime\\_ima.html\)](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
```

## 3.2 For more information

- Keylime home page is at <https://keylime.dev>.
- Latest Keylime documentation is at <https://keylime.readthedocs.io/en/latest/>.


- For a high-level overview of IMA/EVM, refer to [https://en.opensuse.org/SDB:Ima\\_evm#Introduction](https://en.opensuse.org/SDB:Ima_evm#Introduction).
- Find more details about creating a new kernel IMA policy in [https://keylime-docs.readthedocs.io/en/latest/user\\_guide/runtime\\_ima.html](https://keylime-docs.readthedocs.io/en/latest/user_guide/runtime_ima.html).



## A Legal Notice

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 other 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.

## B 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 automati-

cally 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.