SUSE

使用 Keylime 保护 SLE Micro

解释

Keylime 是基于 TPM 的远程引导证明和运行时完整性测量服务。

原因

本文介绍如何在 SLE Micro 上配置和运行 Keylime。

工作量

读完本文大约需要 25 分钟。

目标

您将详细了解 Keylime - 它的工作原理，应如何配置它以及它是如何运行的。

要求

- 正在运行的 SLE Micro 实例

出版日期：2025 年 12 月 11 日

目录

1 使用 Keylime 进行远程证明

随着保护设备免遭未经授权的更改的需求日益增长，**远程证明 (RA)** 这一安全机制的应用范围也不断扩大。借助 RA，主机（客户端）可以对其引导链状态以及远程主机（验证者）上运行的软件进行身份验证。RA 通常会与公钥加密（使用 TPM2）结合使用，这样发送的信息便只能被请求证明的服务读取，同时又可验证数据的有效性。

SLE Micro 上的远程证明通过 **Keylime** 实现。

1.1 术语

远程证明技术使用了以下术语：

认证密钥 (AK)

一种数据签名密钥，可证明数据来自真实的 TPM 且未被篡改。

用于测量的核心信任根

计算自己的哈希以及引导过程中下一步的哈希，是测量链的发起端。

背书密钥 (EK)

在生产 TPM 时永久嵌入其中的加密密钥。TPM 中存储的该密钥的公共部分和证书用于识别真实 TPM。

身份管理体系结构 (IMA)

提供相应方法来检测恶意更改文件行为的内核完整性子系统。

测量引导

引导序列中的每个组件在委派下一个组件执行之前用于计算下一个组件哈希的方法。该哈希会扩展 TPM 的一个或多个 PCR。系统会创建相应事件，以记录有关进行测量的位置以及测量的内容等信息。这类事件会随扩展的 PCR 值一起收集在事件日志中，并且可能会与代表健康状况良好的系统的预期值进行比较。

平台配置寄存器 (PCR)

TPM 中用于存储引导层哈希等数据的内存位置。只能通过不可逆的操作 `extend` 来更新 PCR。在 TPM 上可以使用 **`quote`** 命令来获得当前 PCR 值的签名列表，在证明过程中，第三方可以对此引用进行验证。

安全引导

引导过程中的每一步都会在启动下一步前检查加密签名，以确定该步骤是否可执行。

可信平台模块 (TPM)

系统中作为硬件存在或在固件中实施以充当信任根的一个独立安全加密处理器。TPM 提供了 PCR 来存储各个引导层的哈希。典型的 TPM 会提供多项功能，例如随机数字生成器、计数器或本地时钟。此外，它还存储了 24 个 PCR，并按每个支持的加密哈希函数（SHA1、SHA256、SHA384 或 SHA512）的内存库分组。

注意

默认情况下，TPM 处于禁用状态。因此不会进行测量引导。要启用远程证明，请在 EFI/BIOS 菜单中启用 TPM。

安全有效负载

一种用于向健康状况良好的代理传递加密数据的机制。有效负载用于提供密钥、口令、证书、配置或进而可被代理使用的脚本。

1.2 Keylime 是什么?

Keylime 是一款远程证明解决方案，可用于将 TPM 作为测量的信任根来监控远程节点的健康状况。利用 Keylime，您可以执行多种任务，例如：

- 验证测量引导期间扩展的 PCR。
- 创建分析并建立事件日志的断言。
- 建立远程系统中任意 PCR 的值的断言。
- 监控打开的文件或所执行文件的有效性。
- 通过**安全有效负载**向经过验证的节点传递加密数据。
- 执行在计算机未通过所认证的测量结果时触发的自定义脚本。

1.3 体系结构

Keylime 由代理、验证者、注册者和命令行工具 (tenant) 组成。代理位于需要接受认证的系统上。验证者和注册者位于执行代理注册和证明的远程系统上。请注意，SLE Micro 上仅可使用代理角色。有关每个组件的详细信息，请参见以下几节。

1.3.1 Keylime 代理

代理是在需要接受认证的系统上运行的服务。代理会向验证者发送事件日志、IMA 哈希以及关于测量引导的信息，并使用本地 TPM 作为数据有效性的证明者。

新代理启动时需要先在注册者中注册自己。为此，代理需要使用 TLS 证书来建立连接。TLS 证书由注册者生成，但用户需要手动将其安装到代理中。注册之后，代理会将其认证密钥和背书密钥的公共部分发送给注册者。注册者会在称为“身份凭证激活”的过程中向代理返回一个质询，该质询会验证代理的 TPM。当代理在注册者中注册后即可登记进行认证。

1.3.2 Keylime 注册者

注册者用于注册应接受认证的代理。注册者会收集代理的认证密钥，背书密钥的公共部分以及背书密钥证书，并会验证代理认证密钥是否属于背书密钥。

1.3.3 Keylime 验证者

验证者会对代理执行实际的证明，并会持续从代理那里提取所需证明数据（包括 PCR 值、IMA 日志和 UEFI 事件日志等）。

2 使用 Podman 运行 Keylime 工作负载

Keylime 是一种远程证明解决方案，可用于监控远程节点的健康状况。**验证者**和**注册者**是远程系统上的 Keylime 的重要组成部分，用于执行 Keylime 代理的注册和证明。

注意

本文中介绍的容器提供了作为 Keylime 项目一部分的控制平面服务：**验证者**、**注册者**和 **tenant** 命令行工具 (CLI)。

在开始安装和注册代理之前，请按照下面所述的过程在远程主机上准备验证者和注册者。

1. 找到 Keylime 工作负载映像。

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

2. 从注册表中提取映像。

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

3. 创建 `keylime-control-plane` 卷以持久存放数据库以及证明过程中所需的证书。

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

4. 启动容器及相关服务。

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

系统即会创建 `keylime-control-plane` 容器。该容器中包含已配置且正在运行的注册者和验证者服务。容器会使用默认值在内部向主机公开端口 8881、8890 和 8891。请验证防火墙配置，以允许访问这些端口并允许容器之间进行通讯，因为 tenant CLI 需要如此。

提示

如果您需要停止 Keylime 服务，请运行以下命令：

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

2.1 监控 Keylime 服务

要获取主机上运行的容器的状态，请运行以下命令：

```
# podman ps
```

要查看 Keylime 服务的日志，请运行以下命令：

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

2.2 执行 tenant CLI

tenant CLI 工具包含在容器中，如果主机防火墙未影响 Keylime 服务公开的端口，您可以使用相同的映像执行该工具，例如：

```
# 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
```

2.3 提取 Keylime 证书

Keylime 容器首次执行时，其服务会创建多个代理所需的证书。您需要从容器中提取证书，并将其复制到代理的 `/var/lib/keylime/cv_ca/` 目录。

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

提示

有关安装代理的详细信息，请参见第 3 节 “安装 Keylime 代理”。

3 安装 Keylime 代理

Keylime 是一种远程证明解决方案，可用于监控远程节点的健康状况。Keylime 代理是在需要证明的系统上运行的服务，会将事件日志、IMA 哈希和有关测量引导的信息发送给验证者。

SLE Micro 默认未安装 Keylime 代理，您需要手动安装。要安装代理，请执行以下操作：

1. 按如下所示安装 `rust-keylime` 软件包：

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

   然后重引导系统。

2. 调整代理的默认配置。

   a. 创建一个目录，在 `/etc/keylime/agent.conf.d/` 中存储用于保存您的更改的新配置文件。默认配置文件存储在 `/usr/etc/keylime/agent.conf` 中，但我们不建议编辑此文件，因为将来进行系统更新时会覆盖该文件。

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

   b. 创建新文件 `/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 每次代理运行时都会生成唯一标识符，不过，您可以使用此选项定义一个特定的值。

❷ 注册者的 IP 地址。

❸ 验证者的 IP 地址。

c. 将 `/etc/keylime/` 目录的拥有者更改为 `keylime:tss`：

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

d. 更改对 `/etc/keylime/` 目录的权限：

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

3. 将 CA 生成的证书复制到代理节点。在代理节点上执行以下操作：

a. 准备用于存储证书的目录：

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

b. 将证书复制到代理上：

```
# scpCERT_SERVER_ADDRESS:/var/lib/keylime/cv_ca/cacert.crt /var/lib/
keylime/cv_ca
```

c. 将证书的拥有者更改为 `keylime:tss`：

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

4. 启动并启用 `keylime_agent.service`：

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

4 注册 Keylime 代理

Keylime 是一种远程证明解决方案，可用于监控远程节点的健康状况。Keylime 代理是在需要证明的系统上运行的服务，会将事件日志、IMA 哈希和有关测量引导的信息发送给验证者。

您可以使用 CLI tenant 或者编辑验证者的配置来注册新代理。在验证者主机上使用 tenant 运行以下命令：

```
# keylime_tenant -v 127.0.0.1 \
  -tAGENT \❶
  -u UUID \❷
  --cert default \
  -c add
  [--include PATH_TO_ZIP_FILE]❸
```

❶ `AGENT` 是要注册的代理的 IP 地址。

❷ `UUID` 是代理的唯一标识符。

❸ `include` 选项传递的文件用于向代理传送机密的有效负载数据。有关细节，请参见第 5 节 “Keylime 安全有效负载”。

您可以在验证者主机上使用 **`reglist`** 命令列出已注册的代理，如下所示：

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

要去除注册的代理，请使用 `-t` 和 `-u` 选项指定代理并运行 `-c delete` 命令，如下所示：

```
# keylime_tenant -v 127.0.0.1 \
  -tAGENT \
 -u UUID \
  -c delete
```

5 Keylime 安全有效负载

Keylime 是一种远程证明解决方案，可用于监控远程节点的健康状况。

5.1 什么是安全有效负载?

使用 Keylime 安全有效负载，您可以向健康状况良好的代理传递加密数据。有效负载用于提供密钥、口令、证书、配置或 Keylime 代理在后面的阶段中使用的脚本。

5.2 安全有效负载的工作原理

安全有效负载通过 `zip` 文件传递给代理，该文件必须包含名为 `autorun.sh` 的外壳脚本。仅当代理正确注册并验证后，该脚本才会执行。要传递 `zip` 文件，需使用 **`keylime_tenant`** 命令的 `--include` 选项。

例如，下面的 `autorun.sh` 脚本会创建一个目录结构并将 SSH 密钥复制到其中。相关的 `zip` 存档必须包含这些 SSH 密钥。

```
> 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
```

6 为 Keylime 启用 IMA 跟踪

Keylime 是一种远程证明解决方案，可用于监控远程节点的健康状况。**完整性管理体系结构** (IMA) 是一个内核完整性子系统，提供了检测恶意更改文件的行为的方法。

使用 IMA 时，内核会计算被访问文件的哈希。然后，该哈希会用于扩展 TPM 中的 PCR 10，还会记录被访问文件的列表。验证者可以向代理请求 PCR 10 的签名引用，以获取所有被访问文件的日志（包括文件哈希）。然后，验证者会将被访问文件与本地的获批文件许可列表进行比较。如果无法识别其中的任何哈希，系统即被视为不安全，并触发撤消事件。

必须启用 IMA/EVM，Keylime 才能收集信息。要启用该进程，请使用 `ima_appraise=log` 和 `ima_policy=tcb` 参数引导代理的内核：

1. 在 `/etc/default/grub` 中使用这两个参数更新 `GRUB_CMDLINE_LINUX_DEFAULT` 选项：

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

2. 运行以下命令重新生成 `grub.cfg`：

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

3. 重引导系统。

上面的过程使用了默认的内核 IMA 策略。为了避免监控的文件太多，因而产生过长的日志，您可以创建新的自定义策略。有关详细信息，请参见 Keylime 文档 (https://keylime-docs.readthedocs.io/en/latest/user_guide/runtime_ima.html)。

要指明预期的哈希，请在注册代理时使用 **`keylime_tenant`** 命令的 `--allowlist`选项。要查看排除或忽略的文件，请使用 **`keylime_tenant`** 命令的 `--exclude` 选项：

```
# keylime_tenant --allowlist
    -v 127.0.0.1 \
    -uUUID
```

7 更多信息

- Keylime 主页：https://keylime.dev。
- 最新的 Keylime 文档：https://keylime.readthedocs.io/en/latest/。

- 有关 IMA/EVM 的简要概述，请访问 https://en.opensuse.org/SDB:Ima_evm#Introduction。
- 有关创建新内核 IMA 策略的详细信息，请访问 https://keylime-docs.readthedocs.io/en/latest/user_guide/runtime_ima.html。

8 法律声明

A GNU 自由文档许可证

0. 导言

此许可证的目的是赋予手册、教科书或其他功能性的和有用的文档以“自由”：即保证每个人都有复制和再分发此类文档的有效自由，无论是否进行修改，也无论将其用于商业或非商业用途。其次，此许可证为作者和出版者保留了因工作获得声誉但不视为对他人所做修改负责的方式。

本许可证是一种“非盈利版权”，这意味着从该文档衍生的作品也必须是以同一方式自由的。它补充了 GNU 通用公共许可证（为自由软件设计的非盈利版权许可证）。

我们设计此许可证旨在将其用于免费软件的手册，因为免费软件需要自由文档：免费程序所附手册应具有与软件本身同样的自由。但是此许可证不限于软件手册；它可以用于任何文本作品，无论主题如何或它是否作为印刷书籍出版。建议本许可证主要用于目的是指导或参考的作品。

1. 适用性和定义

此许可证适用的对象：由版权所有者在其中明确声明可按照此许可证条款以任何媒体分发的任何手册和其他作品。此类声明授予在此处所述的条款和条件下使用该作品的全球无限期无版权许可证。下述“文档”指任何此类手册或作品。任何公众成员都是一个被许可人，以下称为“您”。如果您以需要版权法许可的任何方式复制、修改或分发该作品，则表示您接受该许可证。

该文档的“修改版本”表示包含该文档或其一部分（或者逐字复制或者有修改和/或翻译为另一语言）的任何作品。

“次要章节”是该文档的命名附录或扉页章节，专门讲述该文档的出版者或作者与该文档整个主题（或相关问题）的关系，不包含与整个主题相关的内容。（因此，如果该文档是数学课本的一部分，则辅助部分可能不说明任何数学问题。）这种关系可以是与主题或相关问题的历史联系，或与它们相关的法律、商业、哲学、伦理或政治地位。

在该文档基于此许可证项发布的声明中，“固定章节”是将其标题指定为固定章节标题的一些辅助章节。如果一个章节不适用上述辅助章节的定义，则不允许将其指定为固定章节。该文档可能不包含固定章节。如果该文档不标识任何固定章节，则表示没有固定章节。

在该文档基于此许可证项发布的声明中，“封页文本”是作为封面文本或封底文本列出的简短文本段落。封面文本最多 5 个单词，封底文本最多 25 个单词。

文档的“透明”副本是一个机器可读的副本，使用公众可以得到其规范的格式表达，这样的副本适合于使用通用文本编辑器、（对于像素构成的图像）通用绘图程序、（对于绘制的图形）广泛使用的绘画编辑器直接修改文档，也适用于输入到文本格式处理程序或自动翻译成各种适于适用于输入到文本格式处理程序的格式。一个用其他透明文件格式表示的副本，如果该格式的标记（或缺少标记）已经构成了对读者的后续修改的障碍，那么就是不透明的。表示实质性数量的文本的图像格式都是不透明的。不“透明”的副本称为“不透明”。

适于作为透明副本的格式的示例有：没有标记的纯 ASCII 文本、Texinfo 输入格式、LaTeX 输入格式、使用公共可用 DTD 的 SGML 或 XML，符合标准的简单 HTML、可以人为修改的 PostScript 或 PDF。透明图像格式的示例有 PNG、XCF 和 JPG。不透明的格式包括：仅可以

被私有版权的字处理软件使用的私有版权格式、所用的 DTD 和/或处理工具不是广泛可用的的 SGML 或 XML、机器生成的 HTML、一些字处理器生成的只用于输出目的的 PostScript 或 PDF。

对于印刷书籍，“扉页”就是扉页本身以及随后的一些用于补充的页，显然本许可资料需要出现在扉页上。对于那些没有扉页的作品形式，“扉页”代表接近作品最突出标题的、在文本正文之前的文本。

“命名为 XYZ”的章节表示文档的一个特定的子单元，其标题就是 XYZ 或在括号中包含 XYZ 且后跟 XYZ 的其他语言翻译文本。（这里 XYZ 代表下面提及的特定章节名称，比如“致谢”、“题献”、“签名”或“历史”。）要在修改文档时对这类章节“保留标题”就是依据此定义保持这样一个“命名为 XYZ”的章节。

文档可能在文档遵照此许可证的声明后面包含免责声明。这些免责声明应作为参考信息包含在此许可证中，但是只能将其视作免责声明：这些免责声明暗指的任何其他含义均无效，且对此许可证的含义不产生任何影响。

2. 逐字复制

您可以用任何媒体复制并分发文档，无论是出于商业还是非商业目的，只要保证此许可证、版权声明和声称此许可证应用于文档的声明都完整地、无任何附加条件地存在于所有副本中。不能使用任何技术手段阻碍或控制您制作或发布的副本的阅读或再次复制。不过您可以在副本交易中得到报酬。如果发布足够多的副本，则您必须遵循下面第三节中的条件。

您也可以在如上的条件下出租副本和向公众放映副本。

3. 大量复制

如果您出版的文档印刷版副本（或是有印制封页的其他媒体副本）多于 100 份，而文档的许可证声明中要求有封页文本，则您必须将它清晰地置于封页之上，封面文本在封面上，封底文本在封底上。封面和封底上还必须标明您是这些副本的出版者。封面必须同等显著地完整展现标题的所有文字。您可以在封页上加入其他资料。改动仅限于封页的复制，只要保持文档的标题不变并满足这些条件，可以在其他方面被视为逐字复制。

如果需要加上的文本对于封面或封底过多，无法明显地表示，您应该在封页上列出前面的（在合理的前提下尽量多），把其他的放在邻近的页面上。

如果您出版或分发了超过 100 份文档的不透明副本，则必须在每个不透明副本中包含一份计算机可读的透明副本，或是在每个不透明副本中给出一个计算机网络地址，通过这个地址，网络公共用户可以使用标准网络协议下载文档的无任何附加资料的完整透明副本。如果您选择后者，则必须在开始大量分发非透明副本的时候采用相当谨慎的步骤，保证透明副本在其所给出的位置在（直接或通过代理和零售商）分发最后一次该版本的非透明副本的时间之后一年之内始终是有效的。

在重新大量发布副本之前，请您（但不是必须）与文档的作者联系，以便他们可以有机会向您提供文档的更新版本。

4. 修改

在上述第 2、3 节的条件下，您可以复制和分发文档的修改版本，前提是严格按照此许可证发布修改后的文档，将修改版本用作文档，从而允许任何拥有此修改版副本的人执行分发或修改。另外，在修改版中，您需要做到如下几点：

A. 用于与文档以及以前各个版本（如果有，应该列在文档的“历史”章节中）显著不同的扉页（和封页，如果有）。如果那个版本的原始发行者允许的话，您可以使用和以前版本相同的标题。

B. 与作者一样，在扉页上列出承担修改版本中的修改的作者责任的一个或多个人或实体和至少五个文档的原作者（如果原作者不足五个就全部列出），除非他们免除了您的这个责任。

C. 与原来的发行者一样，在扉页上列出修改版的发行者的姓名。

D. 保持该文档的全部版权声明不变。

E. 在与其他版权声明邻近的位置加入恰当的针对您的修改的版权声明。

F. 在紧接着版权声明的位置加入许可声明，按照下面附录中给出的形式，以本许可证给公众授于是用修订版本的权利。

G. 保持原文档的许可声明中的全部不可变章节、封面文字和封底文字的声明不变。

H. 包含一份未作任何修改的本协议的副本。

I. 保持命名为“历史”的章节不变，保持它的标题不变，并在其中加入一项，至少声明扉页上的已修改版本的标题、年份、新作者和出版者。如果文档中没有命名为“历史”章节，则请新建它，并加入一项以声明原文档扉页上所列的标题、年份、作者与出版者，再在其后加入如上所说的描述修改版本的项。

J. 如果文档中有用于公共用户访问的文档透明副本的网址，则保持网址不变，并同样提供它所基于的以前文档版本的网址。这些网址可以放在“历史”章节。您可以不给出那些在原文档发行之前已经发行至少四年的版本给出的网址，或者该版本的发行者授权不列出网址。

K. 对于任何命名为“致谢”或“题献”的章节，保持其标题不变，并保持其全部内容以及对每位贡献者致谢和/或题献的语气不变。

L. 保持文档的所有固定章节不变，不改变它们的标题和内容。章节的编号或相当的内容不被认为是章节标题的一部分。

M. 删除命名为“签名”的章节。这样的章节不可以被包含在修改后的版本中。

N. 不要把任何现有章节重命名为“签名”或与任何不可变章节相冲突的标题。

O. 保持任何免责声明不变。

如果修改版本加入了新的符合次要章节定义的引言或附录章节，并且不含有从原文档中复制的内容，您可以选择将其标记为固定。如果需要这样做，则将它们的标题加入修改版本许可声明的不可变章节列表之中。这些标题必须和其他章节的标题相区分。

您可以加入一个命名为“签名”的章节，只要它只包含对您的修改版本由不同的各方给出的签名，例如书评或是声明文本已经被一个组织认定为一个标准的权威定义。

您可以加入一个最多 5 个字的段落作为封面文本和一个最多 25 个字的段落作为封底文本，将它们加入修改版本的封页文本列表末端。一个实体只可以添加（或编排）一段封面和一段封底文本。如果原文档已经为该封页（封面或封底）包含了封页文本，由您或您所代表的实体先前加入或排列的文本，不能再新加入一个，但您可以在原来的发行者的明确许可下替换掉原来的那个。

作者和发行者不能通过本许可证授权公众使用他们的名字推荐或暗示认可任何一个修改版本。

5. 组合文档

遵照第 4 节所说的修改版本的规定，您以将文档和其他文档合并并以本许可证发布，只要您在合并结果中包含原文档的所有不可变章节，对它们不加以任何改动，并在合并结果的许可声明中将它们全部列为不可变章节，而且维持原作者的免责声明不变。

合并作品仅需要包含一份此许可证，多个相同的固定章节可以由一个副本取代。如果有多个名称相同但内容不同的固定章节，通过在章节名称后面的括号中加上原作者或出版者的姓名（如果已知）来加以区别，或者使用唯一编号加以区别。并对合并作品许可声明中的固定章节列表中的章节标题做相同的调整。

在合并过程中，必须合并不同原始文档中任何命名为“历史”的章节，从而形成新的命名为“历史”的章节；类似地，还要合并命名为“致谢”和“题献”的章节。必须删除所有命名为“签名”的章节。

6. 文档的合集

您可以制作一个文档和其他文档的合集，在本许可证下发布，并在合集中将不同文档中的多个本许可证的副本以一个单独的副本来代替，只要您在文档的其他方面遵循本许可证的逐字复制的条款即可。

您可以从一个这样的合集中提取一个单独的文档，并将它在本许可证下单独发布，只要您想这个提取出的文档中加入一份本许可证的副本，并在文档的其他方面遵循本许可证的逐字复制的原则。

7. 独立作品的聚合体

将文档或其派生品以及其他独立和无关文档或作品编撰在一个储存卷中或分发媒体上，这称为文档的“聚合体”，前提是编撰成品的著作权对其使用者的法律权限的限制未超出各个独立作品的许可范围。当基于此许可证发布的文档包含在一个聚合体中时，此许可证不适用于聚合体中的本非该文档派生作品的其他作品。

如果第 3 节中的封页文本要求适用于这些文档的副本，则若文档在聚合体中所占的比重小于全作品的一半，文档的封页文本可以放置在聚合体内包含文档部分的封页上，或是电子文档中的等效部分。否则，它必须位于整个聚合体的印刷的封页上。

8. 翻译

翻译被视为一种修改，因此您可以根据第 4 节的条款分发文档的翻译。将固定章节替换为翻译内容需要经得其版权所有者的特别许可，但除了这些固定章节的原始版本之外，您还可以包含一部分或所有固定章节的翻译。您可以包含一个此许可证以及所有许可证声明和免责声明的翻译版本，前提是同时包含它们的原始英文版本。当翻译版本和英文版发生冲突的时候，原始版本有效。

如果在文档中有命名为“致谢”、“题献”或“历史”的章节，保持标题（第 1 节）的要求（第 4 节）恰恰需要更换实际的标题。

9. 终止

除非此许可证中有明确规定，否则您不能对该文档进行复制、修改、分授许可或分发。在此许可证规定外对该文档所进行的任何复制、修改、分授许可或分发都是无效的，并且将自动终止您在此许可证下所拥有的权利。但是，对于在此许可证的规定下从您这里获得副本或权利的各方，只要其完全遵守此许可证的规定，其许可证将不会被终止。

10. 本许可的未来修订版本

自由软件基金会有时会发布 GNU 自由文档许可证的新的修订版版本。这些新版本的主旨和精神与当前版本是一致的，但在解决新问题的具体细节方面可能有所不同。请参见 https://www.gnu.org/copyleft/。

许可证的每个版本都有一个不同的版本号。如果文档指定了适用于它的此许可证“或任何后续版本”的特定带编号版本，则您可以选择遵从指定版本或自由软件基金会发布的任何随后版本（非草稿）的条款和条件。如果文档没有指定此许可证的版本号，您可以选择自由软件基金会发布的任何许可证版本（非草稿）。

附录：如何针对您的文档使用此许可证

```
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”.
```

如果您有固定章节、封面文本和封底文本，请将“with...Texts”部分替换为：

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

如果有不可变章节而没有封页文本，或这三种内容（不可变章节、封面文本、封底文本）的任何其他组合，请合并这两个备选项以适应您的情况。

如果您的文档包含不一般的程序代码示例，建议同时选择自由软件许可证（如 GNU 通用公共许可证）发布这些示例，以允许它们可以用于自由软件。