

SELinux 查错

解释

包含 `enforcing` 模式 SELinux 的系统可能会导致拒绝访问错误，从而导致应用程序无法正常运行。您可以使用 `audit2allow` 或 `setroubleshoot` 以用户友好的方式分析拒绝访问错误消息。

原因

本文说明如何在不降低系统安全性的情况下解决 SELinux 导致的拒绝访问错误。

工作量

读完本文大约需要 30 分钟。

目标

您可以使用后文所述的工具之一来调试 SELinux 拒绝访问错误。

要求

- 一个正在运行的系统，其中已启用 SELinux。

出版日期：2025 年 12 月 11 日

目录

- 1 `/var/log/audit/audit.log` 文件 3
- 2 使用 `audit2allow` 分析 `/var/log/audit/audit.log` 6

- 3 使用 **setroubleshoot** 分析 AVC 消息 7
- 4 法律声明 13
- A GNU 自由文档许可证 13

1 /var/log/audit/audit.log 文件

默认情况下，如果 SELinux 是导致出错的原因，系统会将与此结果相关的日志消息发送到 /var/log/audit/audit.log 文件。



注意：空的 /var/log/audit/audit.log

如果您看到空的 /var/log/audit/audit.log，这通常表示 auditd 服务未运行。在这种情况下，请执行以下步骤：

1. 启动 auditd 服务：

```
> sudo systemctl start auditd
```

2. 使用以下命令在系统目标中启用该服务

```
> sudo systemctl enable auditd
```

/var/log/audit/audit.log 文件存储拒绝访问错误、服务事件等的消息。

在例 1：“/etc/audit/audit.log 中的示例行”中，可以看到 /var/log/audit/audit.log 内容的部分示例。

例 1：/etc/audit/audit.log 中的示例行

```
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_75PVMTO-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_75PVMTO-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_75PVMTO-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_75PVMTO-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_75PVMTO-
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_75PVMTO-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

```

日志中每条消息的外观如下：

```

type=AVC msg=audit(1348173901.081:299): avc: denied { getattr }
for pid=3426 comm="smartd" path="/var/lib/smartmontools/
smartd.WDC_WD2500BEKT_75PVMTO-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

```

消息的每一行可分解为若干部分。例如，最后一行包含以下部分：

type=AVC:

每个 SELinux 相关审计日志行都以类型标识（例如 type=AVC）开头。请注意，如果包含 type=SYSCALL 的消息后接在类型不同但 msg 值相同的消息后面，则它可能会提供有关事件的更多信息。

msg=audit(1348173901.309:300):

这是时戳，它是以纪元时间（自 1970 年 1 月 1 日开始经过的秒数）写入的。您可以针对纪元时间表示法中点号前面的部分使用 date -d，以确定该事件是何时发生的：

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

avc: denied { append }:

被拒绝的特定操作。在本例中，系统已拒绝将数据追加到文件。在浏览审计日志文件时，您可以看到其他系统操作，例如 write open、getattr 等。

for pid=1316:

发起操作的命令或进程的进程 ID

comm="rsyslogd":

与该 PID 关联的特定命令

name="smartmontools":

操作主体的名称

dev=sda6 ino=582296:

相关文件的块设备和 inode 编号

scontext=system_u:system_r:syslogd_t:

源环境，即操作发起者的环境

`tclass=file:`

主体的类标识

2 使用 **audit2allow** 分析 `/var/log/audit/audit.log`

您不必亲自解释 `/var/log/audit/audit.log` 中的事件，而可以使用

audit2allow 命令。

该命令可帮助您分析 `/var/log/audit/audit.log` 中晦涩难懂的日志消息。**audit2allow** 查错会话始终由三个不同的命令组成。首先，可以使用 **audit2allow -w -a** 以更易于理解的方式显示审计信息。默认情况下，**audit2allow -w -a** 会处理 `audit.log` 文件。如果您要分析 `audit.log` 文件中的特定消息，请将其复制到临时文件，并使用以下命令分析该文件：

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

例 2：分析审计消息

```
> 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
```

发生此问题的原因是：

缺少类型强制 (TE) 允许规则。

要生成可加载的模块以允许这种访问，请运行

```
> sudo audit2allow
```

要确定具体哪条规则拒绝了访问，您可以使用 **audit2allow -a** 显示已记录到 `audit.log` 文件中的所有事件中的强制规则，或使用 **audit2allow -i FILENAME** 显示应用于存储在特定文件中的消息的规则：

例 3：查看哪些行拒绝了访问

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

要创建您可以加载以允许先前被拒绝的访问的 SELinux 模块 `mymodule`，请运行

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

如果您要对已记录到 `audit.log` 文件中的所有事件执行此操作，请使用 `-a -M` 命令参数。要仅针对特定文件中的特定消息执行此操作，请按下面的示例所示使用 `-i -M`：

例 4：创建允许先前被拒绝的操作的策略模块

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

semodule -i example.pp
```

如 `audit2allow` 命令所示，您现在可按以下方式运行此模块：使用 `semodule -i` 命令，后接 `audit2allow` 为您创建的模块的名称（在上例中为 `example.pp`）。

3 使用 `setroubleshoot` 分析 AVC 消息

要以用户友好的方式分析 AVC 拒绝访问错误消息，可以使用 `setroubleshoot` 工具。

3.1 `setroubleshoot` 概览

3.1.1 什么是 `setroubleshoot`？

`setroubleshoot` 是可以从内核收集 SELinux 审计事件并对其进行分析的工具。如果发生此类事件，`setroubleshoot` 会通知管理员。

3.1.2 setroubleshoot 组件

SELinux 查错过程涉及以下组件，默认情况下，所有这些组件均已安装在 SLE Micro 上。

- setroubleshoot-server 提供以下工具：
 - setroubleshootd - 用于处理传入请求和插件定义的主守护程序。该守护程序是按需激活的，不需要通过 systemd 服务来运行。它只能由特权用户和专门的 setroubleshoot 用户来管理。
 - /var/lib/setroubleshoot/setroubleshoot_database.xml 文件中的警报数据库
 - sealert - 用于分析 /var/log/audit.log 的命令行用户界面
 - sedispatch - 审计调度程序，用于扫描 SELinux AVC 消息并将其转换为 DBus 消息，然后将其传递给守护程序。
- setroubleshoot-plugins - 插件用于 AVC 消息分析，并提供有关如何解决问题的建议。

3.1.3 setroubleshoot 的工作原理

setroubleshoot 包含守护程序和分析插件。当插件检测到问题时，会向守护程序报告此问题，然后守护程序将检查这是否为已知问题。如果不是，则将新问题连同建议的解决方案一起添加到数据库中。

3.1.4 setroubleshoot 的优点

setroubleshoot 提供以下功能来帮助您解决 SELinux 安全系统上的问题：

- 发生 AVC 拒绝访问错误时向管理员发送警报。
- 自动分析 AVC 拒绝访问错误。
- 建议可能的修复方法，例如调整系统配置或安装更新等。
- 浏览以前的警报。

3.2 配置 setroubleshoot

尽管 `setroubleshoot` 的配置不需要调整，但在某些特殊用例中，您可能需要更改默认值。以下章节提供了一般用例。

`setroubleshoot` 的配置文件是 `/etc/setroubleshoot`。通常，除了设置电子邮件通知之外，您无需修改其他配置。但是，如果您需要更改配置，可以编辑该文件，或使用 `setroubleshootd` 命令配置特定项目。命令语法如下：

```
# setroubleshootd -c  
SECTION.OPTION=VALUE
```

例如，要设置 `from_address` 选项，请运行如下命令：

```
# setroubleshootd -c  
email.from_address="example@mail.com"
```

3.2.1 配置 setroubleshoot 日志记录级别

默认日志记录级别（`sealert_log` 和 `setroubleshootd_log` 的值）设置为 `warning`。但是，您可以将该值设置为下列其中一项：

critical

仅记录导致系统无法正常运行的严重错误。

error

报告可能影响系统的严重错误。

warning

指出发生了意外情况，或者不久的将来可能会出现的问题。但系统可按预期运行。

info

记录确认系统正常运行的信息。

debug

记录用于调试的详细信息。

3.2.2 配置 setroubleshoot 以发送电子邮件通知

如果系统中发生 AVC 拒绝访问错误，setroubleshoot 可向您发送电子邮件通知。

要获取这些通知，请执行以下步骤：

1. 打开 /etc/setroubleshoot/setroubleshoot.conf。

2. 在该文件中，根据需要调整以下配置项目：

smtp_host

如果 SMTP 服务器未在本地主机上运行，请填写服务器地址。

smtp_port

默认值为 25。通常不需要对此值进行任何调整。

from_address

添加发件人地址。

subject

配置所有邮件的通用主题。

recipients_filepath

指定通知收件人列表的位置。

use_sendmail

如果您使用 SendMail，请设置为 true。

3. 在 recipients_filepath 选项定义的路径（默认为 /var/lib/setroubleshoot/email_alerts-recipients）中创建邮件收件人文件。

每个电子邮件地址必须独占一行。注释以 # 符号表示。

3.2.3 配置 setroubleshoot 数据库

可以更改 setroubleshootd 数据库中的记录数量、数据库位置或文件名前缀。

database_dir

指定数据库 XML 文件所在目录的绝对路径。

filename

配置数据库文件名的自定义前缀。然后，文件名如下所

示：FILENAME_PREFIX_database.xml。

max_alerts

定义数据库中的记录数上限。指定 0 表示记录数没有限制。

max_alert_age

超过设置限制的警报将从数据库中删除。可使用以下单位：**年、月、天、小时、分钟和秒**，甚至可以使用复数形式；并且可以使用多个单位，例如 3 weeks 2 days（相当于 23 天）。如果留空，则表示没有限制。

3.2.4 配置 setroubleshoot 以从远程服务器收集信息

可以配置 setroubleshoot 以从远程服务器收集 SELinux 审计数据。为此请配置地址列表。

[listen_for_client] address_list

在服务器端。

[client_connect_to] address_list

在客户端。

列表中的地址采用以下格式：

```
[{FAMILY}]ADDRESS[:PORT_NUMBER]
```

其中 {FAMILY} 为 {inet} 或 {unix}%{path}s。如果地址族为 inet，则您可以选择性地指定端口号，否则端口号将设置为 default_port 配置选项指定的默认值。默认值 {unix}%{path}s hostname 表示侦听本地 Unix 域套接字。

3.3 运行 /var/log/audit/audit.log 分析

要让 setroubleshoot 工具分析审计日志文件，请运行以下命令：

```
> sudo sealert -a /var/log/audit/audit.log
```

在以下示例输出中，有两个端口值已指派到 SSHD 服务：

```
100% done
found 1 alerts in /var/log/audit/audit.log
-----

SELinux is preventing sshd from name_bind access on the tcp_socket port 2222.

**** Plugin bind_ports (92.2 confidence) suggests ****

If you want to allow sshd to bind to network port 2222
Then you need to modify the port type.
Do
# semanage port -a -t PORT_TYPE -p tcp 2222 ❶
    where PORT_TYPE is one of the following: ssh_port_t, vnc_port_t,
    xserver_port_t.

**** Plugin catchall_boolean (7.83 confidence) suggests ****

If you want to allow nis to enabled
Then you must tell SELinux about this by enabling the 'nis_enabled' boolean.

Do
setsebool -P nis_enabled 1

**** Plugin catchall (1.41 confidence) suggests ****

If you believe that sshd should be allowed name_bind access on the port 2222
tcp_socket by default.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do
allow this access for now by executing:
# ausearch -c 'sshd' --raw | audit2allow -M my-sshd
# semodule -X 300 -i my-sshd.pp

Additional Information:
...
```

First Seen	2024-02-07 14:26:27 UTC
Last Seen	2024-02-08 03:30:12 UTC
Local ID	b5cbdd75-3f8d-425d-af75-f6cbf1540ffd

Raw Audit Messages

```
type=AVC msg=audit(1707363012.797:25): avc: denied { name_bind } for
pid=841 comm="sshd" src=2222 scontext=system_u:system_r:sshd_t:s0-s0:c0.c1023
tcontext=system_u:object_r:unreserved_port_t:s0 tclass=tcp_socket permissive=0
```

Hash: sshd,sshd_t,unreserved_port_t,tcp_socket,name_bind

- ❶ bind_ports 插件为此处出现的问题提供了最合适解决方案。

4 法律声明

版权所有 © 2006–2025 SUSE LLC 和贡献者。保留所有权利。

根据 GNU 自由文档许可证 (GNU Free Documentation License) 版本 1.2 或（根据您的选择）版本 1.3 中的条款，在此授予您复制、分发和/或修改本文档的权限；本版权声明和许可证附带不可变部分。许可版本 1.2 的副本包含在题为“GNU Free Documentation License”的部分。

有关 SUSE 商标，请参见 <https://www.suse.com/company/legal/> 。所有其他第三方商标分别为相应所有者的财产。商标符号（®、™ 等）代表 SUSE 及其关联公司的商标。星号 (*) 代表第三方商标。

本指南力求涵盖所有细节，但这不能确保本指南准确无误。SUSE LLC 及其关联公司、作者和译者对于可能出现的错误或由此造成的后果皆不承担责任。

A GNU 自由文档许可证

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. 允许任何人复制和分发此许可证文档的逐字副本，但禁止对其进行更改。

0. 引言

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 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。

您也可以在这样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。