



SUSE Linux Enterprise Server 15 SP3

安全和强化指南

安全和强化指南

SUSE Linux Enterprise Server 15 SP3

本指南解释了系统安全性的基本概念，并介绍了本产品随附的安全软件（例如 AppArmor、SELinux 或审计系统）的用法。本指南还可帮助系统管理员强化安装的系统。

出版日期：2025 年 6 月 18 日

<https://documentation.suse.com> 

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 (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。

目录

前言 xx

- 1 可用文档 xx
- 2 改进文档 xx
- 3 文档约定 xxi
- 4 支持 xxiii

SUSE Linux Enterprise Server 支持声明 xxiii • 技术预览 xxiv

1 安全性和机密性 1

- 1.1 概述 1
- 1.2 口令 2
- 1.3 备份 2
- 1.4 系统完整性 3
- 1.5 文件访问 4
- 1.6 网络 4
- 1.7 软件漏洞 5
- 1.8 恶意软件 6
- 1.9 重要安全提示 6
- 1.10 报告安全问题 7

2 通用准则 8

- 2.1 简介 8
- 2.2 评估保障级别 (EAL) 8

2.3 一般指导原则 9

2.4 更多信息 11

I 身份验证 12

3 通过 PAM 进行身份验证 13

3.1 PAM 是什么? 13

3.2 PAM 配置文件的结构 13

3.3 sshd 的 PAM 配置 16

3.4 PAM 模块的配置 19

pam_env.conf 19 • pam_mount.conf.xml 20 • limits.conf 20

3.5 使用 pam-config 配置 PAM 20

3.6 手动配置 PAM 21

3.7 更多信息 22

4 使用 NIS 23

4.1 配置 NIS 服务器 23

配置 NIS 主服务器 23 • 配置 NIS 从属服务器 28

4.2 配置 NIS 客户端 29

5 使用 YaST 设置身份验证客户端 31

5.1 使用 YaST 配置身份验证客户端 31

5.2 SSSD 31

检查状态 31 • 缓存 32

6 389 LDAP Directory Server 33

6.1 LDAP 目录树的结构 33

- 6.2 安装 389 Directory Server 36
 - 设置新的 389 Directory Server 实例 36 · 使用自定义配置文件创建 389 Directory Server 实例 37 · 基于模板创建 389 Directory Server 实例 39 · 停止和启动 389 Directory Server 40 · 配置用于本地管理的管理员身份凭证 41
- 6.3 防火墙配置 42
- 6.4 管理 LDAP 用户和组 42
- 6.5 设置 SSSD 45
- 6.6 管理模块 46
- 6.7 从 OpenLDAP 迁移到 389 Directory Server 47
 - 测试从 OpenLDAP 迁移 47 · 规划迁移 50
- 6.8 使用 TSL 的 CA 证书 51
- 6.9 更多信息 53
- 7 使用 Kerberos 进行网络身份验证 54**
 - 7.1 概念概述 54
 - 7.2 Kerberos 术语 54
 - 7.3 Kerberos 的工作原理 56
 - 首次联系 56 · 请求服务 56 · 相互身份验证 57 · 票据授予 — 联系所有服务器 57
 - 7.4 Kerberos 的用户视图 58
 - 7.5 安装和管理 Kerberos 59
 - Kerberos 网络拓扑 60 · 选择 Kerberos 领域 61 · 设置 KDC 硬件 61 · 配置时间同步 62 · 配置 KDC 63 · 配置 Kerberos 客户端 66 · 配置远程 Kerberos 管理 69 · 创建 Kerberos 服务主体 71 · 对 Kerberos 启用 PAM 支持 73 · 配置 SSH 进行 Kerberos 身份验证 73 · 使用 LDAP 和 Kerberos 74

- 7.6 Kerberos 和 NFS 77
 - 组成员资格 78 • 性能和可伸缩性 79 • 主 KDC、多个域和信任关系 80

- 7.7 更多信息 80

8 Active Directory 支持 81

- 8.1 集成 Linux 和 Active Directory 环境 81
- 8.2 有关 Linux Active Directory 支持的背景信息 82
 - 域加入 84 • 域登录和用户主目录 85 • 办公服务和策略支持 86
- 8.3 为 Active Directory 配置 Linux 客户端 87
 - 选择用于连接 Active Directory 的 YaST 模块 87 • 使用用户登录管理加入 Active Directory 88 • 使用 Windows 域成员资格加入 Active Directory 92 • 检查 Active Directory 连接状态 94
- 8.4 登录到 Active Directory 域 95
 - GDM 95 • 控制台登录 95
- 8.5 更改口令 96

9 设置 freeRADIUS 服务器 98

- 9.1 在 SUSE Linux Enterprise 上安装和测试 98

II 本地安全性 101

10 物理安全性 102

- 10.1 系统锁 102
- 10.2 锁定 BIOS 102
- 10.3 通过引导加载程序提供的安全性 103
- 10.4 淘汰包含敏感数据的 Linux 服务器 104
 - scrub: 磁盘重写实用程序 104

10.5	限制对可移动媒体的访问	106
11	使用 seccheck 进行自动安全检查	108
11.1	Seccheck 计时器	108
11.2	启用 seccheck 计时器	108
11.3	每日、每周和每月检查	109
11.4	自动注销	110
12	软件管理	112
12.1	去除不需要的软件包 (RPM)	112
12.2	修补 Linux 系统	114
	YaST 联机更新	114
	自动联机更新	114
	储存库镜像工具 — RMT	115
	SUSE Manager	116
13	文件管理	117
13.1	磁盘分区	117
13.2	检查文件权限和所有权	117
13.3	默认的 umask	118
13.4	SUID/SGID 文件	119
13.5	全局可写文件	120
13.6	孤立文件或无拥有者的文件	120
14	加密分区和文件	122
14.1	使用 YaST 设置加密文件系统	122
	在安装过程中创建加密分区	123
	在正在运行的系统上创建加密分区	124
	加密可移动媒体的内容	124
14.2	使用 GPG 加密文件	124

15 使用 cryptctl 对托管应用程序的储存区加密 126

- 15.1 设置 **cryptctl** 服务器 127
- 15.2 设置 **cryptctl** 客户端 129
- 15.3 使用服务器端命令检查分区解锁状态 132
- 15.4 手动解锁加密分区 132
- 15.5 维护停机过程 133
- 15.6 更多信息 133

16 用户管理 134

- 16.1 各种帐户检查 134
 - 未锁定的帐户 134 • 未使用的帐户 134
- 16.2 启用口令时效 135
- 16.3 实施更强的口令 137
- 16.4 使用 PAM 进行口令和登录管理 137
 - 口令强度 138 • 限制使用先前的口令 139 • 登录失败次数太多后锁定用户帐户 139
- 16.5 限制 root 登录 140
 - 限制本地文本控制台登录 141 • 限制图形会话登录 142 • 限制 SSH 登录 143
- 16.6 为交互式外壳会话设置无活动超时 143
- 16.7 防止意外拒绝服务 145
 - 限制系统资源示例 146
- 16.8 显示登录标题 148
- 16.9 连接统计实用程序 149

17 Spectre/Meltdown Checker 150

17.1 使用 `spectre-meltdown-checker` 150

17.2 更多信息 152

18 使用 YaST 配置安全设置 153

18.1 安全概览 153

18.2 预定义安全配置 154

18.3 口令设置 155

18.4 引导设置 155

18.5 登录设置 155

18.6 用户添加 156

18.7 其他设置 156

19 Linux 中的访问控制列表 157

19.1 传统文件权限 157

setuid 位 158 • setgid 位 158 • 粘性位 158

19.2 ACL 的优势 159

19.3 定义 159

19.4 处理 ACL 160

ACL 项和文件模式权限位 161 • 具有 ACL 的目录 162 • 具有默认 ACL 的目录 164 • ACL 检查算法 167

19.5 应用程序中的 ACL 支持 167

19.6 更多信息 167

20 证书存储区 168

20.1 激活证书存储区 168

20.2 导入证书 168

21 使用 AIDE 进行入侵检测 170

21.1 为何要使用 AIDE? 170

21.2 设置 AIDE 数据库 170

21.3 本地 AIDE 检查 173

21.4 独立于系统的检查 174

21.5 更多信息 175

III 网络安全性 177

22 X Window 系统和 X 身份验证 178

23 SSH: 安全网络操作 179

23.1 **ssh** — 安全外壳 179

在远程主机上启动 X 应用程序 180 • 代理转发 180

23.2 **scp** — 安全复制 180

23.3 **sftp** — 安全文件传输 181

使用 **sftp** 181 • 设置文件上载权限 182

23.4 SSH 守护程序 (sshd) 183

维护 SSH 密钥 184 • 轮换主机密钥 184

23.5 SSH 身份验证机制 185

生成 SSH 密钥 186 • 复制 SSH 密钥 186 • 使用 **ssh-agent** 187

23.6 端口转发 188

23.7 在安装的系统上添加和去除公共密钥 188

23.8 更多信息 189

24 掩蔽和防火墙 191

24.1 使用 iptables 过滤包 191

24.2 关于掩蔽的基础知识 194

24.3 防火墙基础知识 195

24.4 firewalld 195

使用 NetworkManager 配置防火墙 197 • 在命令行上配置防火墙 197 • 访问监听动态端口的服务 202

24.5 从 SuSEfirewall2 迁移 205

24.6 更多信息 207

25 配置 VPN 服务器 208

25.1 概念概述 208

术语 208 • VPN 方案 208

25.2 设置简单测试方案 211

配置 VPN 服务器 212 • 配置 VPN 客户端 213 • 测试 VPN 示例方案 214

25.3 使用证书颁发机构设置 VPN 服务器 215

创建证书 215 • 配置 VPN 服务器 216 • 配置 VPN 客户端 218

25.4 使用 YaST 设置 VPN 服务器或客户端 219

25.5 更多信息 220

26 使用 XCA、X 证书和密钥管理器管理 PKI 221

26.1 安装 XCA 221

26.2 创建新 PKI 221

创建新的根 CA 222 • 创建已签名的主机证书 223 • 吊销证书 223

IV 通过 APPARMOR 限制特权 225

27 AppArmor 简介 226

27.1 AppArmor 组件 226

27.2 有关 AppArmor 配置文件构建的背景信息 226

28 入门 228

28.1 安装 AppArmor 228

28.2 启用和禁用 AppArmor 229

28.3 选择要构建配置文件的应用程序 230

28.4 构建和修改配置文件 230

28.5 更新您的配置文件 232

29 使程序免疫 233

29.1 AppArmor 框架简介 234

29.2 确定要使其免疫的程序 236

29.3 使 cron 作业免疫 236

29.4 使网络应用程序免疫 237

使 Web 应用程序免疫 238 • 使网络代理免疫 240

30 配置文件组件和语法 242

30.1 分解 AppArmor 配置文件 243

30.2 配置文件类型 245

标准配置文件 246 • 未关联的配置文件 246 • 本地配置文件 246 • 帽子 247 • 更改规则 247

30.3 Include 语句 248

抽象 250 • 程序块 250 • Tunables 250

- 30.4 功能项 (POSIX.1e) 250
- 30.5 网络访问控制 251
- 30.6 配置文件名称、标志、路径和通配 252
 - 配置文件标志 253 · 在配置文件中使用的变量 254 · 模式匹配 255 · 名称空间 256 · 配置文件命名和附件规范 256 · 别名规则 257
- 30.7 文件权限访问模式 257
 - 读取模式 (r) 258 · 写入模式 (w) 258 · 追加模式 (a) 258 · 文件锁定模式 (k) 258 · 链接模式 (l) 259 · 链接对 259 · 可选的允许规则和文件规则 259 · 拥有者条件规则 260 · 拒绝规则 261
- 30.8 挂载规则 262
- 30.9 Pivot Root 规则 263
- 30.10 PTrace 规则 264
- 30.11 信号规则 265
- 30.12 执行模式 266
 - 离散配置文件执行模式 (px) 266 · 离散本地配置文件执行模式 (cx) 266 · 未受限执行模式 (ux) 267 · 不安全的执行模式 267 · 继承执行模式 (ix) 267 · 允许可执行映射 (m) 267 · 命名配置文件转换 268 · 配置文件转换的回退模式 269 · 执行模式中的变量设置 269 · safe 和 unsafe 关键字 270
- 30.13 资源限制控制 271
- 30.14 审计规则 272
- 31 AppArmor 配置文件储存库 274**
- 32 使用 YaST 构建和管理配置文件 275**
- 32.1 手动添加配置文件 275

- 32.2 编辑配置文件 276
 - 添加项 278 • 编辑项 281 • 删除项 281
- 32.3 删除配置文件 281
- 32.4 管理 AppArmor 282
 - 更改 AppArmor 状态 283 • 更改单个配置文件的模式 283

33 从命令行构建配置文件 285

- 33.1 检查 AppArmor 状态 285
- 33.2 构建 AppArmor 配置文件 286
- 33.3 添加或创建 AppArmor 配置文件 287
- 33.4 编辑 AppArmor 配置文件 287
- 33.5 卸载未知的 AppArmor 配置文件 287
- 33.6 删除 AppArmor 配置文件 288
- 33.7 构建配置文件的两种方式 288
 - 独立式配置文件构建 289 • 系统性配置文件构建 289 • 构建配置文件的工具汇总 291
- 33.8 重要的文件名和目录 310

34 使用 ChangeHat 构建 Web 应用程序的配置文件 312

- 34.1 配置 Apache 以使用 mod_apparmor 313
 - 虚拟主机指令 314 • 位置和目录指令 314
- 34.2 管理 ChangeHat 感知型应用程序 315
 - 使用 AppArmor 的命令行工具 315 • 在 YaST 中向帽子添加帽子和项 321

35 使用 pam_apparmor 限制用户 323

36 管理已构建配置文件的应用程序 324

36.1 对安全事件拒绝做出反应 324

36.2 维护安全配置文件 324

备份安全配置文件 324 • 更改安全配置文件 325 • 将新软件引入您的环境 325

37 支持 326

37.1 联机更新 AppArmor 326

37.2 使用手册页 326

37.3 更多信息 328

37.4 查错 328

如何应对应用程序行为异常？ 328 • 我的配置文件似乎不再正常工作... 328 • 使用 Apache 解决问题 332 • 如何从使用的配置文件列表中排除特定的配置文件？ 332 • 我是否可以管理未安装在我系统上的应用程序的配置文件？ 332 • 如何找出和修复 AppArmor 语法错误 333

37.5 报告 AppArmor 的 Bug 334

38 AppArmor 术语表 335

V SELINUX 对比 337

39 配置 SELinux 338

39.1 为何要使用 SELinux？ 338

支持状态 339 • 了解 SELinux 组件 339

39.2 策略 341

39.3 安装 SELinux 软件包并修改 GRUB 2 342

39.4 SELinux 策略 344

- 39.5 配置 SELinux 346
- 39.6 管理 SELinux 348
 - 查看安全环境 348 • 选择 SELinux 模式 350 • 修改 SELinux 环境类型 351 • 应用文件环境 352 • 配置 SELinux 策略 354 • 使用 SELinux 模块 355
- 39.7 查错 356

VI LINUX 审计框架 360

40 了解 Linux 审计 361

- 40.1 Linux 审计组件简介 363
- 40.2 配置审计守护程序 365
- 40.3 使用 **auditctl** 控制审计系统 370
- 40.4 将参数传递到审计系统 373
- 40.5 了解审计日志和生成报告 376
 - 了解审计日志 376 • 生成自定义审计报告 381
- 40.6 使用 **ausearch** 查询审计守护程序日志 388
- 40.7 使用 **autrace** 分析进程 392
- 40.8 可视化审计数据 393
- 40.9 中继审计事件通知 395

41 设置 Linux 审计框架 398

- 41.1 确定要审计的组件 398
- 41.2 配置审计守护程序 399
- 41.3 对系统调用启用审计 401
- 41.4 设置审计规则 401

41.5 配置审计报告 403

41.6 配置日志可视化 407

42 审计规则集简介 410

42.1 添加基本审计配置参数 410

42.2 添加审计日志文件和配置文件监测项 411

42.3 监视文件系统对象 412

42.4 监视安全配置文件和数据库 413

42.5 监视其他系统调用 416

42.6 过滤系统调用参数 416

42.7 使用键管理审计事件记录 419

43 有用资源 421

A 实现 PCI DSS 合规 423

A1 什么是 PCI DSS? 423

A2 本文档的重点：与操作系统相关的方面 424

A3 要求详细说明 425

要求 1：安装并维护防火墙配置以保护持卡人数据 425 • 要求 2：不要使用供应商为系统口令和其他安全参数提供的默认值 427 • 要求 3：保护存储的持卡人数据 430 • 要求 4：对在开放的公共网络上传输的持卡人数据进行加密 431 • 要求 5：保护所有系统免遭恶意软件的攻击，并定期更新防病毒软件或程序 432 • 要求 6：开发并维护安全系统和应用程序 432 • 要求 7：限制企业仅可访问其需要知道的持卡人数据 434 • 要求 8：识别并验证对系统组件的访问 436 • 要求 9：限制对持卡人数据的物理访问 437 • 要求 10：跟踪并监控对网络资源和持卡人数据的所有访问 437 • 要求 11：定期测试安全系统和流程 438 • 要求 12：维护用于处理所有个人信息安全性的策略 438

B GNU licenses 439

前言

1 可用文档

联机文档

<https://documentation.suse.com/#sles> 上提供了此产品的相关联机文档。您可浏览或下载各种格式的文档。

<https://documentation.suse.com/> 上提供了其他产品的相关联机文档。



注意：最新更新

最新的文档更新通常会在文档的英文版中提供。

发行说明

有关发行说明，请参见 <https://www.suse.com/releasesnotes/>。

在您的系统中

要以脱机方式使用，请参见安装的系统中 `/usr/share/doc` 下的文档。许多命令的**手册页**中也对相应命令进行了详细说明。要查看手册页，请运行 `man` 后跟特定的命令名。如果系统上未安装 `man` 命令，请使用 `sudo zypper install man` 加以安装。

2 改进文档

欢迎您提供针对本文档的反馈及改进建议！我们提供了多种反馈渠道：

服务请求和支持

有关产品可用的服务和支持选项，请参见 <https://www.suse.com/support/>。

要创建服务请求，需在 SUSE Customer Center 中获取一个订阅。请转到 <https://scc.suse.com/support/requests> 并登录，然后单击新建。

Bug 报告

在 <https://bugzilla.suse.com/> 中报告文档问题。要简化此过程，可以使用本文档 HTML 版本中的标题旁边的报告文档 Bug 链接。这样，就会在 Bugzilla 中预先选择正确的产品和类别，并添加当前章节的链接。然后，您便可以立即开始键入 Bug 报告。需要一个 Bugzilla 帐户。

贡献

要帮助改进本文档，请使用本文档 HTML 版本中的标题旁边的编辑源代码链接。这些链接会将您转到 GitHub 上的源代码，在其中可以创建拉取请求。需要一个 GitHub 帐户。

有关本文档使用的文档环境的详细信息，请参见[储存库的 README \(https://github.com/SUSE/doc-sle/blob/master/README.adoc\)](https://github.com/SUSE/doc-sle/blob/master/README.adoc)。

邮件

另外，您也可以将有关本文档中的错误以及相关反馈发送至：doc-team@suse.com。请确保反馈中含有文档标题、产品版本和文档发布日期。请引用相关的章节号和标题（或者包含 URL），并提供问题的简要说明。

3 文档约定

本文档中使用了以下通知和排版约定：

- /etc/passwd：目录名称和文件名
- PLACEHOLDER：PLACEHOLDER 将会替换为实际的值
- PATH：环境变量 PATH
- ls、--help：命令、选项和参数
- 用户：用户和组
- 软件包名称：软件包名称
- **Alt**、**Alt - F1**：按键或组合键；这些键以大写形式显示，如在键盘上一样
- 文件、文件 > 另存为：菜单项，按钮

- **AMD/Intel** 本段内容仅与 AMD64/Intel 64 体系结构相关。箭头标记文本块的开始位置和结束位置。 ◁
- **IBM Z, POWER** 本段内容仅与 IBM Z 和 POWER 体系结构相关。箭头标记文本块的开始位置和结束位置。 ◁

- **跳舞的企鹅**（**企鹅**一章，↑其他手册）：此内容参见自其他手册中的一章。
- 必须使用 `root` 特权运行的命令。您往往还可以在这些命令前加上 `sudo` 命令，以非特权用户身份来运行它们。

```
root # command
tux > sudo command
```

- 可以由非特权用户运行的命令。

```
tux > command
```

- 注意



警告：警报通知

在继续操作之前，您必须了解的不可或缺的信息。向您指出有关安全问题、潜在数据丢失、硬件损害或物理危害的警告。



重要：重要通知

在继续操作之前，您必须了解的重要信息。



注意：注意通知

额外信息，例如有关软件版本差异的信息。



提示：提示通知

有用信息，例如指导方针或实用性建议。

4 支持

下面提供了 SUSE Linux Enterprise Server 的支持声明和有关技术预览的一般信息。有关产品生命周期的细节，请参见《升级指南》，第 2 章“生命周期和支持”。

如果您有权获享支持，可在《管理指南》，第 39 章“收集系统信息以供支持所用”中查找有关如何收集支持票据所需信息的细节。

4.1 SUSE Linux Enterprise Server 支持声明

要获得支持，您需要一个适当的 SUSE 订阅。要查看为您提供的具体支持服务，请转到 <https://www.suse.com/support/> 并选择您的产品。

支持级别的定义如下：

L1

问题判定，该技术支持级别旨在提供兼容性信息、使用支持、持续维护、信息收集，以及使用可用文档进行基本查错。

L2

问题隔离，该技术支持级别旨在分析数据、重现客户问题、隔离问题领域，并针对级别 1 不能解决的问题提供解决方法，或作为级别 3 的准备级别。

L3

问题解决，该技术支持级别旨在借助工程方法解决级别 2 支持所确定的产品缺陷。

对于签约的客户与合作伙伴，SUSE Linux Enterprise Server 将为除以下软件包外的其他所有软件包提供 L3 支持：

- 技术预览。
- 声音、图形、字体和作品。
- 需要额外客户合同的软件包。
- **工作站扩展**模块随附的某些软件包仅享受 L2 支持。
- 名称以 `-devel` 结尾的软件包（包含头文件和类似的开发人员资源）只能同其主软件包一起接受支持。

SUSE 仅支持使用原始软件包，即，未发生更改且未重新编译的软件包。

4.2 技术预览

技术预览是 SUSE 提供的旨在让用户大致体验未来创新的各种软件包、堆栈或功能。随附这些预览只是为了提供方便，让您有机会在自己的环境中测试新的技术。非常希望您能提供反馈！如果您测试了技术预览，请联系 SUSE 代表，将您的体验和用例告知他们。您的反馈对于我们的未来开发非常有帮助。

但是，技术预览存在以下限制：

- 技术预览仍处于开发阶段。因此，它们的功能可能不完备、不稳定，或者在其他方面**不适合**用于生产。
- 技术预览**不受支持**。
- 技术预览可能仅适用于特定的硬件体系结构。
- 技术预览的细节和功能可能随时会发生变化。因此，可能无法升级到技术预览的后续版本，而只能进行全新安装。
- 我们随时可能会放弃技术预览。例如，如果 SUSE 发现某个预览不符合客户或市场的需求，或者不能证明它符合企业标准，则可能会放弃该预览。SUSE 不承诺未来将提供此类技术的受支持版本。

有关产品随附的技术预览的概述，请参见 <https://www.suse.com/releasesnotes/>  上的发行说明。

1 安全性和机密性

本章介绍计算机安全的基本概念，其中会介绍威胁和基本缓解方法。本章还提供了其他包含更多信息的章节、指南和网站的参考内容。

1.1 概述

Linux 的一个主要特征是它能够同时处理多个用户（多用户），并允许这些用户在同一台计算机上同时执行任务（多任务）。对于用户而言，处理本地储存的数据与处理网络中储存的数据没有任何差别。

由于存在多用户功能，不同用户的数据必须分开储存，以确保安全性和隐私性。Linux 的另一个重要特征是，即使数据媒体（例如硬盘）丢失或受损，它也能保持数据的可用性。

本章侧重于机密和隐私方面，不过综合性的安全概念还包括定期更新、可正常工作且经过测试的备份。如果没有备份，在发生数据篡改或者硬件故障后，数据恢复就会变得非常困难。

使用**深层防御**方法实现安全性：我们认为，没有任何一种威胁缓解措施可以完全保护系统和数据，但多层防御能够大大提高攻击的难度。深层防御策略可由以下部分构成：

- 将口令进行哈希处理（例如，使用 PBKDF2、bcrypt 或 scrypt）并将口令加盐
- 加密数据（例如，使用 AES）
- 日志记录、监视和入侵检测
- 防火墙
- 防病毒扫描程序
- 明确规定的成文紧急程序
- 备份
- 物理安全性
- 审计、安全扫描和入侵测试

SUSE Linux Enterprise Server 中包含用于解决上面所列要求的软件。下列章节提供了保护系统的起点措施。

《安全和强化指南》中提供了有关强化系统的更多细节。

1.2 口令

Linux 系统上只会储存口令的哈希。哈希是可以方便加密数据的单向算法。同时，哈希算法使得攻击者很难根据哈希计算出原始机密。

哈希储存在普通用户不能读取的 `/etc/shadow` 文件中。由于性能强大的计算机能够恢复口令，因此不应向普通用户显示哈希加密的口令。

美国国家标准技术研究院 (NIST) 发布了有关口令的指导原则（可在 <https://pages.nist.gov/800-63-3/sp800-63b.html#sec5> 上找到）

有关如何设置口令策略的细节，请参见第 18.3 节“口令设置”。有关 Linux 上的身份验证的一般信息，请参见第 I 部分“身份验证”。

1.3 备份

如果您的系统被入侵，可以使用备份来恢复先前的系统状态。发生 bug 或意外时，也可以使用备份将当前系统与旧版本进行比较。对于生产系统，请务必进行某种类型的非现场备份以应对灾难等情况（例如磁带/可刻录媒体的非现场储存，或非现场发起的储存）。

出于法律原因，一些公司和组织必须谨慎考虑备份过多信息以及保留时间过长的情况。如果您的环境设有销毁旧纸质文件的相关政策，您可能还需要将此政策扩展至 Linux 备份磁带。

有关服务器的物理安全的规则也适用于备份。此外，还建议您对备份数据进行加密。可对单个备份存档执行此操作，也可对整个备份文件系统执行此操作（如适用）。如果备份媒体丢失（例如在运输途中），加密可保护数据免受未经授权的访问。如果备份系统本身遭到入侵，则适用相同的规则。在某种程度上，加密还可确保备份的完整性。但请注意，相应人员需要能够在紧急情况下解密备份。此外，还应考虑加密密钥本身被入侵而需要替换的情况。

如果已知某个系统被入侵或疑似被入侵，请务必确定备份的完整性状态。如果在很长一段时间内都未检测到系统入侵，则有可能备份已包含被操控的配置文件或恶意程序。请保留足够长的备份历史以便检查可能的不合理差异。

即使不存在任何已知安全违规，也应定期检查各备份中重要配置文件之间的差异，这有助于发现安全问题（甚至可能的意外错误配置）。此方法最适合内容不会频繁发生更改的文件和环境。

1.4 系统完整性

如果能够以物理方式访问某台计算机，当已获授权的人员引导该计算机时，他们可以操控固件和引导进程来获取访问权限。您的第一项措施应该就是以物理方式锁住服务器机房，尽管并非所有计算机都能锁在不允许进入的机房中。

另外，请记得以安全的方式处置旧设备。保护引导加载程序并限制可移动媒体也有助于确保物理安全性。有关更多信息，请参见第 10 章“物理安全性”。

考虑采取以下附加措施：

- 配置您的系统，使其无法从可移动设备引导。
- 使用 UEFI 口令、安全引导和 GRUB2 口令保护引导进程。
- Linux 系统由引导加载程序启动，该程序通常允许向引导的内核传递其他选项。可以通过为引导加载程序额外设置一个口令，来防止其他人在引导期间使用此类参数。这对于系统安全至关重要。不仅内核本身以 root 权限运行，而且内核还是在系统启动时授予 root 权限的第一个权威对象。
有关在引导加载程序中设置口令的详细信息，请参见《管理指南》，第 14 章“引导加载程序 GRUB 2”，第 14.2.6 节“设置引导口令”。
- 启用硬盘加密。有关更多信息，请参见第 14 章“加密分区和文件”。
- 使用 **cryptctl** 加密托管的储存区。有关更多信息，请参见第 15 章“使用 cryptctl 对托管应用程序的储存区加密”。
- 使用 AIDE 检测系统配置中发生的任何更改。有关更多信息，请参见第 21 章“使用 AIDE 进行入侵检测”。

1.5 文件访问

由于 Linux 采用**一切设置都在文件中指定**的方法，文件权限对于控制对大多数资源的访问权限至关重要。这意味着，您可以使用文件权限来定义对普通文件、目录和硬件设备的访问权限。默认情况下，大多数硬件设备只能由 `root` 访问。但是，某些设备（例如串行端口）可供普通用户访问。

一般来说，执行某项任务时应始终尽量使用限制性最强的特权。例如，以 `root` 权限读写电子邮件是完全没有必要的。如果邮件程序存在 bug，攻击者可能会利用此 bug 在攻击时使用该程序所具有的权限发起攻击。如若遵守上述规则，则可以尽量减少可能的损失。

有关细节，请参见第 19.1 节“传统文件权限”和第 19.2 节“ACL 的优势”。

AppArmor 和 SELinux 允许您为应用程序和用户设置约束。有关细节，请参见第 IV 部分“通过 AppArmor 限制特权”和第 V 部分“SELinux 对比”。

如果存在能够从所安装操作系统的外部访问硬盘的可能性（例如，通过引导在线系统或拆除硬件），请将数据加密。SUSE Linux Enterprise Server 允许您加密包含数据和操作系统的分区。有关详细信息，请参见第 14 章“加密分区和文件”。

1.6 网络

保护网络服务是个至关重要的任务。应当力求保护尽可能多的 **OSI 模型层**。

在传输层或应用层上，应使用最新的加密算法对所有通讯进行身份验证和加密。使用虚拟专用网 (VPN) 作为物理网络上的附加安全层。

SUSE Linux Enterprise Server 提供了许多选项用于保护网络：

- 使用 `openssl` 可以创建 X509 证书。这些证书可用于对许多服务进行加密和身份验证。您可以设置自己的**证书颁发机构 (CA)**，并在网络中将其用作信任源。有关细节，请参见 `man openssl`。
- 通常至少会向公共因特网公开网络的某些部分。使用防火墙规则关闭端口并卸载（最起码要禁用）不需要的服务，从而减小受攻击面。有关详细信息，请参见第 24 章“掩蔽和防火墙”。

- 使用 OpenVPN 保护通过不安全的物理网络建立的通讯通道。有关详细信息，请参见第 25 章 “配置 VPN 服务器”。
- 对网络服务使用强身份验证。有关详细信息，请参见第 I 部分 “身份验证”。

1.7 软件漏洞

软件漏洞是软件中存在的问题，攻击者可以利用此类问题来获取未经授权的访问权限或滥用系统。如果漏洞影响到了远程服务（例如 HTTP 服务），则会造成特别严重的问题。计算机系统非常复杂，因此它们总是存在某些漏洞。

当此类问题变成已知问题时，通常软件开发人员必须在软件中予以修复。然后，系统管理员必须及时在受影响的系统上以安全的方式安装推出的更新。

漏洞通常在中心数据库（例如，由美国政府维护的**国家漏洞数据库**）中公告。您可以订阅这些信息源，及时了解最新发现的漏洞。在某些情况下，可以在软件更新推出之前对 bug 造成的问题加以缓解。漏洞会分配到一个**公共漏洞和暴露 (CVE)** 编号和一个**公共漏洞评分系统 (CVSS)** 分数。该分数有助于识别漏洞的严重性。

SUSE 会提供安全建议源。可通过 <https://www.suse.com/en-us/support/update/> 获得。<https://www.suse.com/support/security/> 上还按 CVE 编号列出了安全更新。



注意：向后移植和版本号

SUSE 采用在较旧稳定软件版本中应用重要源代码修复的做法（**向后移植**）。因此，即使 SUSE Linux Enterprise Server 中某个软件版本号低于上游项目中的最新版本号，SUSE Linux Enterprise Server 中的软件版本也已包含最新的漏洞修复。

有关更多信息，请参见《升级指南》，第 6 章 “源代码向后移植”。

一般情况下，管理员应该为系统中的严重漏洞做好应对准备。这包括尽最大努力强化所有计算机。另外，我们建议制定好预定义的程序，以快速安装用于解决严重漏洞的更新。

为了减轻可能的攻击所造成的损害，请使用限制性文件权限。请参见第 19.1 节 “传统文件权限”。

其他有用链接：

- <http://lists.opensuse.org/opensuse-security-announce/> , 包含 openSUSE 安全公告的邮件列表
- <https://nvd.nist.gov/> , 国家漏洞数据库
- <https://cve.mitre.org/> , MITRE 的 CVE 数据库
- https://www.bsi.bund.de/SiteGlobals/Forms/Suche/BSI/Sicherheitswarnungen/Sicherheitswarnungen_Formular.html , 德国联邦信息安全局漏洞信息源
- <https://www.first.org/cvss/> , 有关公共漏洞评分系统的信息

1.8 恶意软件

恶意软件是旨在扰乱计算机正常运行或窃取数据的软件，包括病毒、蠕虫、勒索软件或 Rootkit。恶意软件有时会利用软件漏洞来攻击计算机。不过，它们往往是用户意外执行的，尤其是从未知来源安装第三方软件时。SUSE Linux Enterprise Server 在其下载储存库中提供了详细的程序（软件包）列表。这可以减少下载第三方软件的需要。SUSE 提供的所有软件包都已签名。下载后，SUSE Linux Enterprise Server 的软件包管理器会检查软件包的签名，以校验其完整性。

`rpm --checksig RPM_FILE` 命令可显示软件包的校验和及签名是否正确。可以在 SUSE Linux Enterprise Server 的第一张 DVD 以及全球大多数密钥服务器上找到签名密钥。

您可以使用 ClamAV 防病毒软件来检测系统上的恶意软件。ClamAV 可以集成到多个服务中，例如邮件服务器和 HTTP 代理。这样就可以在用户启动恶意软件之前将其过滤掉。

限制性用户特权可以减少意外执行代码的风险。

1.9 重要安全提示

以下提示简要概括了上述章节的内容：

- 及时了解最新的安全问题。尽快获取并安装安全公告中建议的已更新软件包。
- 尽可能避免使用 `root` 特权。设置限制性文件权限。
- 仅使用加密的协议进行网络通讯。

- 禁用您绝对不需要的所有网络服务。
- 展开定期安全审计。例如，扫描网络中的开放端口。
- 使用 AIDE（高级入侵检测环境）监视系统上文件的完整性。
- 安装任何第三方软件时都要小心谨慎。
- 定期检查所有备份。
- 检查日志文件（例如，使用 logwatch）。
- 将防火墙配置为阻止所有未显式列入白名单的端口。
- 采用冗余的安全措施设计。
- 在可能的情况下使用加密（例如，针对移动计算机的硬盘）。

1.10 报告安全问题

如果您发现了安全相关的问题，请先检查是否有可用的更新软件包。如果没有可用的更新，请向 security@suse.de 发送电子邮件。请提供问题的详细说明以及相关的软件包版本号。我们建议使用 GPG 加密电子邮件。

<https://www.suse.com/support/security/contact/>  上提供了最新版本的 SUSE GPG 密钥。

2 通用准则

通用准则是指用于评估和衡量 IT 产品安全价值的最知名且使用最广泛的方法。该方法旨在保持独立性，以独立的实验室开展评估，然后由认证机构予以认证。安全功能要求 (SFR) 汇总在所谓的保护配置文件 (PP) 中。如果安全目标 (ST) 的定义与评估保障级别 (EAL) 相若，便可以比较不同产品的安全功能。（安全目标的定义通常会参考 PP — 如果存在满足产品用途的 PP。）

2.1 简介

要清楚定义 IT 产品中的安全并非易事。安全应被视为一个永无止境的过程，而不是一个可满足或不可满足的静态条件。通用准则证书（EAL7 以下级别）并未对系统的错误倾向做出清楚说明，但却为产品添加了一项无法仅通过技术存在来描述的重要价值：即，某人已独立检查了系统设计，所用方法与其声明相对应，并已在生产和维护产品时进行明确关注。

该证书为产品及其安全功能以及公司设计、构建和工程处理产品的流程指定了成熟程度，并将整个产品生命周期内对其进行维护。因此，通用准则旨在做到面面俱到，将与 IT 产品安全相关的一切因素全部考虑在内。

2.2 评估保障级别 (EAL)

评估保障级别表示产品满足所述声明的可信度。级别范围从 1 到 7：

- EAL1：功能测试
- EAL2：结构测试
- EAL3：系统测试及检查
- EAL4：系统设计、测试和审核
- EAL5：半形式化设计和测试
- EAL6：半形式化验证的设计和测试
- EAL7：形式化验证的设计和测试

EAL1 只为产品满足安全要求提供基本保障，EAL2 到 EAL4 为中等保障级别。EAL5 到 EAL7 描述中到高以及高级别保障。如果某个产品从设计之初便未以较高级别保障为目标，则 EAL4 将是产品预期可拥有的最高保障级别。

2.3 一般指导原则

本指南中的大多数建议基于以下准则。在您定义自己的安全流程或确定本文中未明确涉及的配置时，请考虑这些准则。

尽可能使用数据加密

请注意，加密法固然有用，但只适用于它能胜任的特定目的。使用加密法并非获得更高系统安全性的通用途径；它也可能会给系统带来额外的风险。请就是否使用加密法做出明智决策，并有责任为您的决策提供理由。虚假的安全感比本身的弱点可能更有害。

SUSE Linux Enterprise Server 支持对以下项目进行加密：

- 用于远程登录 ([openssh](#)、[man ssh\(1\)](#)) 的网络连接 ([openssl](#) 命令、[stunnel](#))
- 文件 ([gpg](#))
- 位于块层的整个文件系统 ([dm-crypt](#)、[cryptsetup](#))
- VPN ([ipsec](#)、[openvpn](#))

最小软件包安装

将您系统中的已安装软件包限制为最小数量会很有用。无法执行未安装的二进制文件。在系统安装期间，您可以限制所要安装的软件包集。例如，您可以取消选择所有软件包，而只选择您要使用的软件包。例如，在 YaST 中选择 [apache2-mod_perl](#) 软件包将自动选择安装 Apache 软件包运行所需的所有软件包。常常会人为减少依赖项，以便更灵活地处理系统的依赖关系树。您可以选择最小系统，并以其为基础通过（叶）软件包选择构建依赖关系树。

服务隔离 — 在独立的系统上运行不同的服务

可能的情况下，服务器应专用于提供一项服务或一个应用程序。如果攻击者能够成功利用一项服务中的软件缺陷（假设该缺陷允许访问其他服务），这将限制可能被入侵的其他服务数。

为系统上提供的服务使用 AppArmor 是一种有效的限制方法。有关详细信息，请参见第 IV 部分“通过 AppArmor 限制特权”和 [apparmor](#) 手册页。

SUSE Linux Enterprise Server 支持使用虚拟化技术。虽然虚拟化一般用于服务器整合目的，但它对于服务隔离也很有用。不过，虚拟化技术**无法**比拟通过在不同物理计算机上运行服务所提供的隔离强度，也无法代替后者。请注意，超级管理程序隔离虚拟机的能力并不高于或强于 Linux 内核隔离进程及其地址空间的能力。

系统指纹和备份

执行定期备份及设置系统指纹十分重要，在应对系统成功攻击的情况下尤其如此。请将其作为安全惯例不可或缺的一部分，以校验备份是否正常工作。

快速直接的可访问备份可增加您对系统完整性的信心。但备份机制/解决方案拥有足够的版本控制支持亦十分重要，以便您能够跟踪系统中的更改。示例：软件包的安装时间 (`rpm -q --queryformat='%{INSTALLTIME} %{NAME}\n'` 软件包名称) 必须与备份日志文件中的已更改文件对应。

SUSE Linux Enterprise Server 15 SP3 上存在多个工具，可用于检测未知但却成功的攻击。无需花费太多精力对其进行配置。

特别建议您使用文件和目录完整性检查器 [AIDE](#)（高级入侵检测环境）。当它运行以进行初始化时，会创建系统中列于其配置文件中的所有文件的哈希数据库。这样就可以在稍后校验所有已分类文件的完整性。



警告：后门

如果您使用 AIDE，请将哈希数据库复制到潜在攻击者无法访问的位置。否则，攻击者可能会在植入后门之后修改完整性数据库，导致完整性措施目标无法达成。

攻击者也可能已在内核中植入后门。除了很难检测之外，基于内核的后门还可有效去除系统入侵的所有痕迹，致使系统更改几乎变得无法察觉。因此，需要通过救援系统（或已手动装入目标系统文件系统的任何其他独立系统）来执行完整性检查。

请注意，安全更新的应用程序会令完整性数据库失效。`rpm -qlv packagename` 可列出软件包中包含的文件。RPM 子系统非常强大，所含数据由其自行维护。可使用 `--queryformat` 命令行选项对其进行访问。利用 RPM 的某些精细功能，可以更方便地管理包含已更改文件的完整性数据库的差异更新。

2.4 更多信息

通用准则评估可检查已评估设置中产品的特定配置。有关如何安装和配置用作通用准则评估中基线的参考系统，请参见通用准则评估文档的“管理员指南”部分。

不过，将已评估配置理解为**已强化**配置是不正确的。在安装后去除 `setuid` 位和管理过程的规定有助于实现合理的特定配置。但这对于强化声明是不够的。

- 有关 SUSE Linux Enterprise Server 安全认证和功能的详细信息，请参见 <https://www.suse.com/support/security/certifications/>。
- <https://www.suse.com/support/security/> 上提供有 SUSE 安全资源的列表。
- 除与通用准则工作相关的文档外，另请参见以下手册页：

`pam(8)`、`pam(5)`

`apparmor(7)` 和参考的手册页

`rsyslogd(8)`、`syslog(8)`、`syslogd(8)`

`fstab(5)`、`mount(8)`、`losetup(8)`、`cryptsetup(8)`

`haveged(8)`、`random(4)`

`ssh(1)`、`sshd(8)`、`ssh_config(5)`、`sshd_config(5)`、`ssh-agent(1)`、`ssh-add(1)`、`ssh-keygen(1)`

`cron(1)`、`crontab(5)`、`at(1)`、`atd(8)`

`systemctl(1)`、`daemon(7)`、`systemd.unit(5)`、`systemd.special(5)`、`kernel-command-line(7)`、`bootup(7)`、`systemd.directives`

| 身份验证

- 3 通过 PAM 进行身份验证 **13**
- 4 使用 NIS **23**
- 5 使用 YaST 设置身份验证客户端 **31**
- 6 389 LDAP Directory Server **33**
- 7 使用 Kerberos 进行网络身份验证 **54**
- 8 Active Directory 支持 **81**
- 9 设置 freeRADIUS 服务器 **98**

3 通过 PAM 进行身份验证

Linux 在身份验证进程中使用 PAM（可插拔身份验证模块）作为用户和应用程序之间的中间层。PAM 模块在整个系统范围内可用，因此任何应用程序都可以请求 PAM 模块。本章介绍模块化身份验证机制的工作原理和配置方法。

3.1 PAM 是什么？

系统管理员和编程人员经常要将访问限制在系统的某些部分或限制对应用程序某些功能的使用。没有 PAM，每次引入新的身份验证机制（例如 LDAP、Samba 或 Kerberos）时都必须对应用程序进行调整，而此过程非常耗时且容易出错。避免这些缺点的一种方法是将应用程序从身份验证机制中分开并将身份验证委托给集中管理的模块。每当需要使用新的必要身份验证模式时，只要调整或编写合适的 **PAM 模块** 供相关程序使用即可。

PAM 的概念包括：

- **PAM 模块**，用于特定身份验证机制的一组共享库。
- **模块堆栈**，其中包含一个或多个 PAM 模块。
- PAM 感知**服务**，需要使用模块堆栈或 PAM 模块进行身份验证。通常，服务是用户所熟悉的相应应用程序名称，例如 login 或 su。服务名称 other 是默认规则的保留字。
- **模块参数**，可用于影响单个 PAM 模块的执行。
- 用于评估执行单个 PAM 模块所产生的每种**结果**的机制。如果为正值，则执行下一个 PAM 模块。对负值的处理方式取决于配置：“无影响，继续”到“立即终止”之间的所有选项都有效。

3.2 PAM 配置文件的结构

可通过两种方式配置 PAM：

基于文件的配置 (/etc/pam.conf)

每个服务的配置储存在 /etc/pam.conf 中。不过，出于维护和可用性原因，SUSE Linux Enterprise Server 中未使用此配置模式。

基于目录的配置 (/etc/pam.d/)

依赖于 PAM 机制的每个服务（或程序）在 /etc/pam.d/ 目录中都有各自的配置文件。例如，可以在 /etc/pam.d/sshd 文件中找到 sshd 的服务。

/etc/pam.d/ 下的文件定义用于身份验证的 PAM 模块。每个文件都包含用于定义某个服务的行，而每行最多包含四个组成部分：

```
TYPE    CONTROL
MODULE_PATH  MODULE_ARGS
```

组成部分的含义如下：

TYPE

声明服务的类型。PAM 模块是成批处理的。不同类型的模块具有不同的用途。例如，一个模块检查口令，一个模块校验访问系统的位置，还有一个模块读取用户特定的设置。PAM 可以识别四种不同类型的模块：

auth

检查用户的真实性，传统方法是通过查询口令进行检查，但也可以通过芯片卡或生物特征（例如指纹或虹膜扫描）来实现此目的。

account

这种类型的模块会检查用户是否具有使用所请求服务的一般权限。例如，应执行这种检查以确保任何人都不能使用失效帐户的用户名登录。

password

这种类型的模块的用途是启用身份验证令牌的更改。这通常是一个口令。

session

这种类型的模块负责管理和配置用户会话。这些模块在身份验证前后启动，以记录登录尝试并配置用户的特定环境（邮件帐户、主目录、系统限制等）。

CONTROL

指示 PAM 模块的行为。每个模块都可以具有以下控制标志：

required

在进行身份验证之前，必须先成功处理带有此标志的模块。在处理带有 required 标志的模块失败后，将继续处理带有相同标志的所有其他模块，之后用户才会收到有关身份验证尝试失败的消息。

requisite

也必须成功处理带有此标志的模块，处理方式在很大程度上与带有 required 标志的模块类似。但是，如果某个带有此标志的模块失败，将立即向用户提供反馈并且不再继续处理其他模块。如果成功，则接着处理其他模块，就像带有 required 标志的任何模块一样。requisite 标志可用作基本过滤器，检查进行正确身份验证所必需的某些条件是否存在。

sufficient

在成功处理带有此标志的模块后，请求方应用程序会立即收到处理成功的消息并且不再处理其他模块，但前提是之前所有带有 required 标志的模块均未失败。带有 sufficient 标志的模块失败没有任何直接后果，所有随后的模块都将按其各自的顺序进行处理。

optional

带有此标志的模块成功或失败不会产生任何直接后果。此标志可用于只用来显示消息（例如，通知用户收到了邮件）而不采取任何进一步操作的模块。

include

如果给出此标志，则在此处插入指定为参数的文件。

MODULE_PATH

包含 PAM 模块的完整文件名。只要模块位于默认目录 /lib/security（对于 SUSE® Linux Enterprise Server 支持的所有 64 位平台，默认目录均为 /lib64/security）中，就无需显式指定此文件名。

MODULE_ARGS

包含用于影响 PAM 模块行为的选项的空格分隔列表，例如 debug（启用调试）或 nullok（允许使用空口令）。

另外，`/etc/security` 下提供了用于 PAM 模块的全局配置文件，它们定义这些模块的确切行为（其中包括 `pam_env.conf` 和 `time.conf`）。使用 PAM 模块的每个应用程序实际上会调用一组 PAM 函数，这些函数随后将处理不同配置文件中的信息，并将结果返回给请求方应用程序。

为了简化 PAM 模块的创建和维护，现已引入 `auth`、`account`、`password` 和 `session` 模块类型的通用默认配置文件。这些配置取自每个应用程序的 PAM 配置。因此，对 `common-*` 中全局 PAM 配置模块进行的更新将在所有 PAM 配置文件中传播，而无需管理员更新每个 PAM 配置文件。

您可以使用 **pam-config** 工具维护全局 PAM 配置文件。此工具可自动将新模块添加到配置、更改现有模块的配置，或者从配置中删除模块（或选项）。它最大限度地减少甚至完全消除了维护 PAM 配置时所需的人工干预。



注意：64 位和 32 位混合安装

使用 64 位操作系统时，还可以包含 32 位应用程序的运行时环境。在这种情况下，请确保同时安装 32 位版本的 PAM 模块。

3.3 sshd 的 PAM 配置

以 `sshd` 的 PAM 配置为例：

例 3.1：SSH 的 PAM 配置 (`/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 为 PAM 1.0 声明此配置文件的版本。这只是一项惯例，但将来可以使用它来检查版本。

- ② 检查 `/etc/nologin` 是否存在。如果不存在，则除 `root` 以外的任何用户都无法登录。
- ③ 参考四种模块类型的配置文件：`common-auth`、`common-account`、`common-password` 和 `common-session`。这 4 个文件包含每种模块类型的默认配置。
- ④ 设置已经过身份验证的进程的登录 UID 进程属性。
- ⑤ 显示有关用户上次登录的信息。

通过包含配置文件而不是将每个模块单独添加到相应的 PAM 配置，您可以在管理员更改默认设置后自动获取更新的 PAM 配置。以前，在 PAM 发生更改或安装新应用程序后，您需要手动调整所有应用程序的所有配置文件。而现在 PAM 配置是通过中央配置文件进行的，每个服务的 PAM 配置都将自动继承所有的更改。

第一个 `include` 文件 (`common-auth`) 会调用三个 `auth` 类型的模

块：`pam_env.so`、`pam_gnome_keyring.so` 和 `pam_unix.so`。请参见例 3.2 “`auth` 部分的默认配置 (`common-auth`)”。

例 3.2：`auth` 部分的默认配置 (`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` 会装载 `/etc/security/pam_env.conf`，以根据此文件中指定的配置来设置环境变量。它可用于将 `DISPLAY` 变量设置为正确的值，因为 `pam_env` 模块知道登录发生的位置。
- ② `pam_gnome_keyring.so` 根据 GNOME 密钥环检查用户的登录名和口令
- ③ `pam_unix` 根据 `/etc/passwd` 和 `/etc/shadow` 检查用户的登录名和口令。

整个 `auth` 模块堆栈处理完后，`sshd` 才会获得有关登录是否成功的反馈。堆栈中带有 `required` 控制标志的所有模块都必须成功处理，`sshd` 才能收到有关正面结果的消息。如果其中的某个模块不成功，则仍将继续处理整批模块，在此之后 `sshd` 才能得到处理失败的通知。

成功处理所有 `auth` 类型的模块后，将处理另一条 `include` 语句，在本例中为例 3.3 “`account` 部分的默认配置 (`common-account`)” 中的语句。`common-account` 仅包含一个模块：`pam_unix`。如果 `pam_unix` 返回的结果证明用户存在，则 `sshd` 会收到一条处理成功的消息，然后处理下一批模块 (`password`)，如例 3.4 “`password` 部分的默认配置 (`common-password`)” 中所示。

例 3.3：account 部分的默认配置 (common-account)

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

例 3.4：password 部分的默认配置 (common-password)

```
password requisite pam_cracklib.so
password optional pam_gnome_keyring.so use_authok
password required pam_unix.so use_authok nullok shadow try_first_pass
```

同样，`sshd` 的 PAM 配置仅涉及一条 `include` 语句，该语句引用了 `password` 模块的默认配置（位于 `common-password` 中）。每当应用程序请求获取身份验证令牌的更改信息时，都必须成功完成这些模块（控制标志为 `requisite` 和 `required`）。

更改口令或另一个身份验证令牌需要进行安全检查。可以使用 `pam_cracklib` 模块实现此目的。随后使用的 `pam_unix` 模块带有来自 `pam_cracklib` 的任何旧口令和新口令，因此用户在更改口令后无需再次进行身份验证。此过程可确保不能绕过 `pam_cracklib` 所执行的检查。每当配置了 `account` 或 `auth` 类型来指出口令失效时，还应使用 `password` 模块。

例 3.5：session 部分的默认配置 (common-session)

```
session required pam_limits.so
session required pam_unix.so try_first_pass
session optional pam_umask.so
session optional pam_systemd.so
session optional pam_gnome_keyring.so auto_start only_if=gdm,gdm-
password,lxdm,lightdm
session optional pam_env.so
```

最后，调用 `session` 类型的模块（捆绑在 `common-session` 文件中）以根据相关用户的设置来配置会话。`pam_limits` 模块装载文件 `/etc/security/limits.conf`，该文件定义对某些系统资源使用的限制。系统会再次处理 `pam_unix` 模块。`pam_umask` 模块可用于设置文件模式创建掩码。由于此模块带有 `optional` 标志，因此此模块的失败将不会影响整个会话模块堆栈的成功完成。当用户注销时，将再次调用 `session` 模块。

3.4 PAM 模块的配置

某些 PAM 模块是可配置的。配置文件位于 `/etc/security` 中。本节简要介绍与 `sshd` 示例相关的配置文件 — `pam_env.conf` 和 `limits.conf`。

3.4.1 `pam_env.conf`

`pam_env.conf` 可用于定义每次调用 `pam_env` 模块时为用户设置的标准化环境。它允许您使用以下语法预设环境变量：

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

VARIABLE

要设置的环境变量的名称。

[DEFAULT=<value>]

管理员要设置的默认值。

[OVERRIDE=<value>]

可能由 `pam_env` 查询并设置的值，覆盖默认值。

有关 `pam_env` 如何使用的典型示例就是 `DISPLAY` 变量的调整，在发生远程登录是该变量会改变。例 3.6 “`pam_env.conf`” 中显示了这一点。

例 3.6：PAM_ENV.CONF

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

第一行将 `REMOTEHOST` 变量的值设置为 `localhost`，当 `pam_env` 不能确定任何其他值时就会使用该值。`DISPLAY` 变量又包含 `REMOTEHOST` 的值。`/etc/security/pam_env.conf` 的注释中提供了更多信息。

3.4.2 pam_mount.conf.xml

`pam_mount` 用于在登录期间装入用户主目录，以及在注销期间从中心文件服务器用来存放所有用户主目录的环境中卸载这些主目录。使用此方法就无需装入一个完整的 `/home` 目录（通过该目录可访问所有用户主目录），而是仅装入即将登录的用户的主目录。

安装 `pam_mount` 后，`/etc/security` 下会有一个 `pam_mount.conf.xml` 模板。手册页 **man 5 pam_mount.conf** 中提供了各个元素的说明。

可以使用 YaST 完成此功能的基本配置。选择网络设置 > Windows 域成员资格 > 专家设置以添加文件服务器。请参见《储存管理指南》，第 20 章 “Samba”，第 20.5 节 “配置客户端”。



注意：LUKS2 支持

`cryptsetup` 2.0 中已添加了 LUKS2 支持；从 SUSE Linux Enterprise Server 12 SP3 开始，SUSE Linux Enterprise Server 在 `pam_mount` 中包含了 LUKS2 支持。

3.4.3 limits.conf

可以在 `pam_limits` 模块将会读取的 `limits.conf` 中基于用户或组设置系统限制。该文件可让您设置硬限制（即不能超出的限制）和软限制（即可以暂时超出的限制）。有关语法和选项的详细信息，请参见 `/etc/security/limits.conf` 中的注释。

3.5 使用 pam-config 配置 PAM

`pam-config` 工具可帮助您配置全局 PAM 配置文件 (`/etc/pam.d/common-*`) 和多个选定的应用程序配置。有关受支持模块的列表，请使用 `pam-config --list-modules` 命令。使用 `pam-config` 命令可以维护 PAM 配置文件。可将新模块添加到 PAM 配置，删除其他模块，或修改这些模块的选项。更改全局 PAM 配置文件时，无需手动调整单个应用程序的 PAM 设置。

`pam-config` 的简单用例包括：

1. **自动生成全新的 Unix 样式 PAM 配置。** 让 `pam-config` 创建最简单的可行设置，供您以后扩展。`pam-config --create` 命令会创建简单的 Unix 身份验证配置。`pam-config` 不负责维护的现有配置文件将被重写，但会以 `*.pam-config-backup` 的形式保留备份副本。
2. **添加新的身份验证方法。** 可通过简单的 `pam-config --add --ldap` 命令将新的身份验证方法（例如 LDAP）添加到 PAM 模块堆栈。在适用的情况下，LDAP 将添加到所有 `common-* -pc` PAM 配置文件中。
3. **添加调试以进行测试。** 为确保新的身份验证过程按预期工作，请对所有 PAM 相关操作开启调试。`pam-config --add --ldap-debug` 会对 LDAP 相关的 PAM 操作开启调试。在 `systemd` 日记中查找调试输出（请参见《管理指南》，第 17 章 “`journalctl`：查询 `systemd` 日志”）。
4. **查询您的设置。** 在最终应用您的新 PAM 设置之前，请检查该设置是否包含您要添加的所有选项。`pam-config --query --MODULE` 命令会列出所要查询的 PAM 模块的类型和选项。
5. **去除调试选项。** 最后，当您对设置性能完全满意时，请从设置中去除调试选项。`pam-config --delete --ldap-debug` 命令会对 LDAP 身份验证关闭调试。如果您为其他模块添加了调试选项，请使用类似的命令关闭这些选项。

有关 `pam-config` 命令和可用选项的详细信息，请参见 `pam-config(8)` 手册页。

3.6 手动配置 PAM

如果您偏向于手动创建或维护 PAM 配置文件，请确保对这些文件禁用 `pam-config`。

当您使用 `pam-config --create` 命令从头开始创建 PAM 配置文件时，此命令会创建从 `common-*` 到 `common-* -pc` 文件的符号链接。`pam-config` 仅会修改 `common-* -pc` 配置文件。去除这些符号链接会有效禁用 `pam-config`，因为 `pam-config` 仅对 `common-* -pc` 文件运行，而在没有符号链接的情况下，这些文件不起作用。



警告：在配置中包含 `pam_systemd.so`

如果您要创建自己的 PAM 配置，请务必包含配置为 `session optional` 的 `pam_systemd.so`。不包含 `pam_systemd.so` 可能会导致 `systemd` 任务限制出现问题。有关细节，请参见 `pam_systemd.so` 的手册页。

3.7 更多信息

安装 `pam-doc` 软件包后，可在 `/usr/share/doc/packages/pam` 目录中找到以下附加文档：

README 文件

在此目录的顶层，有一个 `modules` 子目录提供了可用 PAM 模块的 README 文件。

Linux-PAM 系统管理员指南

此文档包含系统管理员应该了解的有关 PAM 的所有内容。它讨论了一系列主题，从配置文件的语法到 PAM 的安全特性。

Linux-PAM 模块编写人员手册

此文档从开发人员的角度对多个主题进行了总结，提供了有关如何编写符合标准的 PAM 模块的信息。

Linux-PAM 应用程序开发人员指南

此文档包含要使用 PAM 库的应用程序开发人员所需了解的所有内容。

PAM 手册页

整个 PAM 及其各个模块都随附了手册页，其中全面概述了所有组件的功能。

4 使用 NIS

当网络中的多个 Unix 系统都要访问公共资源时，所有用户和组身份对于该网络中的所有计算机而言是否相同就显得极其重要。网络应该对用户透明：不管用户实际正在使用哪台计算机，其环境都不应该有变化。可以通过 NIS 和 NFS 服务完成此操作。NFS 通过网络分发文件系统，详细信息请参见《储存管理指南》，第 19 章“通过 NFS 共享文件系统”。

NIS（网络信息服务）可以说是一种数据库式服务，用于跨网络访问 `/etc/passwd`、`/etc/shadow` 和 `/etc/group` 的内容。NIS 也可用于其他目的（如提供 `/etc/hosts` 或 `/etc/services` 之类文件的内容），但这里不作介绍。人们常把 NIS 称作 **YP**，也就是网络中的“电话黄页”。

4.1 配置 NIS 服务器

要想通过网络分发 NIS 信息，请安装一台服务器（**主服务器**）管理所有客户端，或者安装多台 NIS 从属服务器用于向主服务器请求此信息，然后将信息中继到各自的客户端。

- 要只为网络配置一台 NIS 服务器，请参见第 4.1.1 节“配置 NIS 主服务器”继续操作。
- 如果您的 NIS 主服务器需将其数据导出到从属服务器，请按第 4.1.1 节“配置 NIS 主服务器”中所述设置主服务器，按第 4.1.2 节“配置 NIS 从属服务器”中所述在子网中设置从属服务器。

4.1.1 配置 NIS 主服务器

要使用 YaST 管理 NIS 服务器功能，请以 root 身份运行 `zypper in yast2-nis-server` 命令来安装 `yast2-nis-server` 软件包。要为网络配置 NIS 主服务器，请按如下所示继续：

1. 启动 YaST > 网络服务 > NIS 服务器。
2. 如果网络中只需要一台 NIS 服务器，或者此服务器将充当更低级别 NIS 从属服务器的主服务器，请选择安装并设置 NIS 主服务器。YaST 将安装需要的程序包。



提示：已安装的 NIS 服务器软件

如果 NIS 服务器软件已安装在机器上，请通过单击创建 NIS 主服务器来创建 NIS 主服务器。

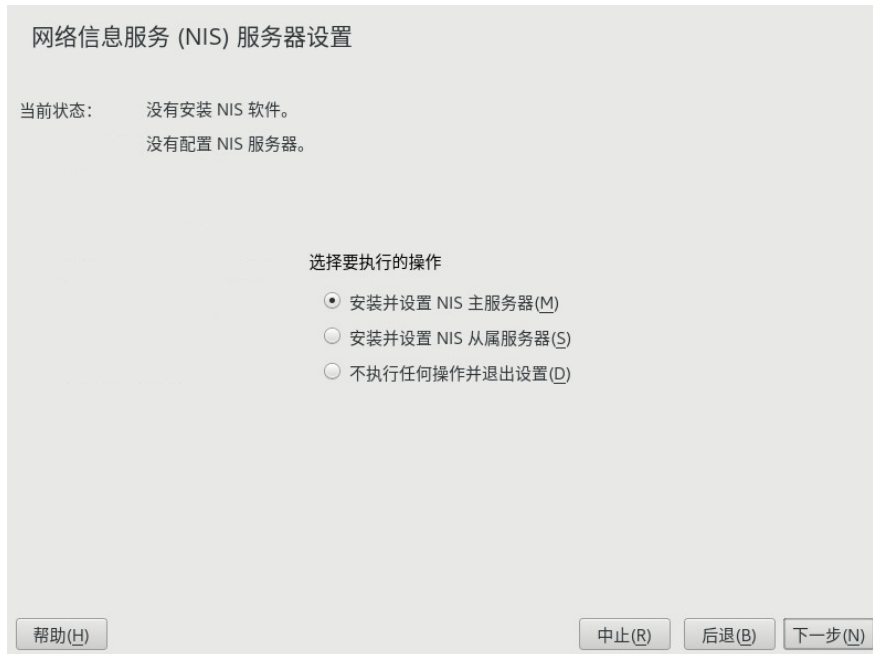


图 4.1：NIS 服务器设置

3. 确定基本 NIS 设置选项：

- a. 输入 NIS 域名。
- b. 通过选择这台主机也是一个 NIS 客户端，定义该主机是否也应该是 NIS 客户端（使用户能够从该 NIS 服务器登录并访问数据）。
- c. 如果 NIS 服务器需要充当其他子网中的 NIS 从属服务器的主服务器，请选择活动的从属 NIS 服务器存在。
快速映射分发选项只能与活动的从属 NIS 服务器存在搭配使用。它可以加速将映射传输到从属服务器的过程。

- d. 选择允许更改口令以允许网络中的用户（本地用户和通过 NIS 服务器管理的用户）更改其在 NIS 服务器上的口令（使用命令 `yppasswd`）。这会相应激活允许更改 GECOS 字段和允许更改登录 Shell 选项。“GECOS”意味着用户还可以使用命令 `ypchfn` 更改其名称和地址设置。“外壳”允许用户使用命令 `ypchsh` 更改其默认外壳（例如从 Bash 切换到 sh）。新外壳必须是 `/etc/shells` 中预定义的其中一项。
- e. 选择在防火墙中打开端口使 YaST 适应 NIS 服务器的防火墙设置。

图 4.2：主服务器设置

- f. 单击下一步退出此对话框，或单击其他全局设置进行其他设置。
- 其他全局设置包括更改 NIS 服务器的源目录（默认为 `/etc`）。此外，也可以在此合并口令。该设置应设为是才能基于系统身份验证文件 `/etc/passwd`、`/etc/shadow` 和 `/etc/group` 创建用户数据库。此外，请确定 NIS 应提供的最小用户和组 ID。单击确定来确认设置并返回上一个屏幕。



图 4.3：更改 NIS 服务器目录并同步文件

4. 如果先前启用了存在活动的从属 NIS 服务器，则输入用作从属服务器的主机的名称，然后单击下一步。如果不存在任何从属服务器，将跳过此配置步骤。
5. 继续在对话框中进行数据库配置。指定 **NIS 服务器映射**，即要从 NIS 服务器传输至客户端的部分数据库。默认设置通常就足够了。选择下一步可退出此对话框。
6. 选择可用映射，然后单击下一步继续。



图 4.4：NIS 服务器映射设置

7. 确定可以查询 NIS 服务器的主机。可通过单击相应的按钮来添加、编辑或删除主机。指定可以向 NIS 服务器发送来自哪个网络的请求。通常为内部网络。在当前情况下，应有以下两项：

255.0.0.0	127.0.0.0
0.0.0.0	0.0.0.0

第一项允许从您自己的主机（即 NIS 服务器）连接。第二项允许所有主机向服务器发送请求。

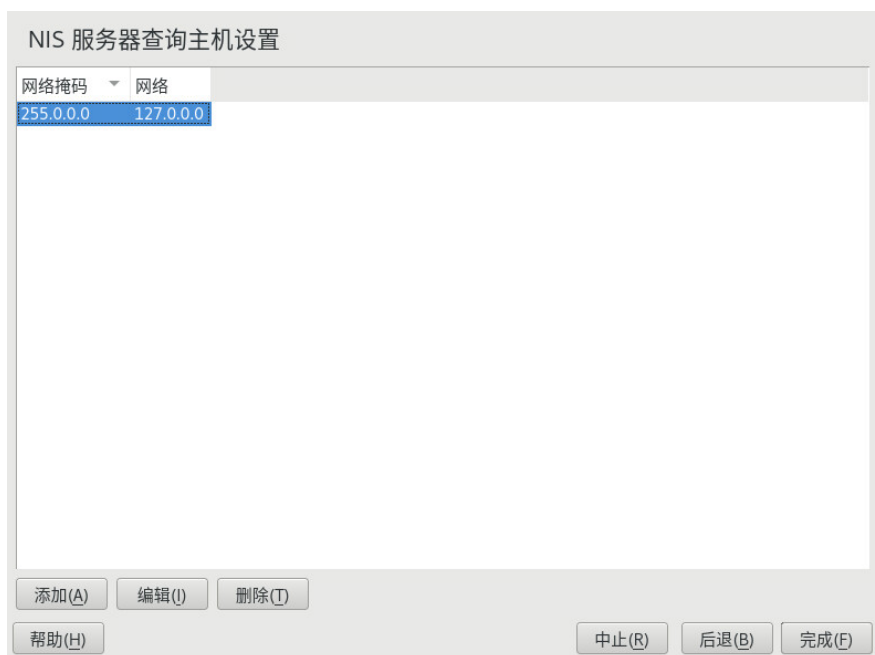


图 4.5：为 NIS 服务器设置请求权限

8. 单击完成以保存更改并退出设置。

4.1.2 配置 NIS 从属服务器

要在网络中配置其他 NIS **从属服务器**，请按如下所示继续：

1. 启动 YaST > 网络服务 > NIS 服务器。
2. 选择安装并设置 NIS 从属服务器，然后单击下一步。



提示

如果 NIS 服务器软件已安装在计算机上，则请通过单击创建 NIS 从属服务器来创建 NIS 从属服务器。

3. 完成 NIS 从属服务器的基本设置：
 - a. 输入 NIS 域。
 - b. 输入主服务器的主机名或 IP 地址。

- c. 如果要允许用户登录此服务器，请设置这台主机也是一个 NIS 客户端。
 - d. 通过打开防火墙中的端口来调整防火墙设置。
 - e. 单击“下一步”。
4. 输入可以查询 NIS 服务器的主机。可通过单击相应的按钮来添加、编辑或删除主机。指定可从中向 NIS 服务器发送请求的所有网络。如果可从所有网络发送请求，请使用以下配置：
- | | |
|-----------|-----------|
| 255.0.0.0 | 127.0.0.0 |
| 0.0.0.0 | 0.0.0.0 |
- 第一项允许从您自己的主机（即 NIS 服务器）连接。第二项允许所有可访问同一网络的主机向服务器发送请求。
5. 单击完成，保存更改并退出设置。

4.2 配置 NIS 客户端

要在工作站上使用 NIS，请执行以下操作：

1. 启动 YaST > 网络服务 > NIS 客户端。
2. 激活使用 NIS 按钮。
3. 输入 NIS 域。这通常是由管理员指定的域名或 DHCP 收到的静态 IP 地址。有关 DHCP 的信息，请参见《管理指南》，第 32 章“DHCP”。

图 4.6：设置 NIS 服务器的域和地址

4. 输入您的 NIS 服务器并以空格分隔其地址。如果您不知道 NIS 服务器的地址，请单击查找让 YaST 搜索您域中的所有 NIS 服务器。根据本地网络的大小，此过程有可能会耗费很长时间。广播可在指定的服务器没有响应后，在本地网络中寻找 NIS 服务器。
5. 根据本地安装，您可能还想激活 automounter。如果需要，此选项还会安装其它软件。
6. 如果您不希望其他主机能够查询您的客户端正在使用的服务器，请转到专家设置并禁用回答远程主机。通过选中断开的服务器，客户端将能够接收通过非特权端口通讯的服务器的答复。有关进一步信息，请参见 [manypbind](#)。
7. 单击完成以保存配置并返回到 YaST 控制中心。现在，您的客户端上已配置好 NIS。

5 使用 YaST 设置身份验证客户端

Kerberos 用于身份验证，而 LDAP 用于授权和标识。两者可以配合工作。有关 LDAP 的详细信息，请参见第 6 章 “389 LDAP Directory Server”；有关 Kerberos 的详细信息，请参见第 7 章 “使用 Kerberos 进行网络身份验证”。

5.1 使用 YaST 配置身份验证客户端

YaST 允许使用不同的模块设置客户端身份验证：

5.2 SSSD

有两个 YaST 模块基于 SSSD：用户登录管理及 LDAP 和 Kerberos 身份验证。

SSSD 指系统安全服务守护程序。SSSD 会与提供用户数据的远程目录服务通讯，并提供各种身份验证方法（例如 LDAP、Kerberos 或 Active Directory (AD)）。它还提供 NSS（名称服务切换）和 PAM（可插入身份验证模块）接口。

SSSD 可在本地缓存用户数据并让用户使用这些数据，即使实际的目录服务（暂时）不可访问时也是如此。

5.2.1 检查状态

运行某个 YaST 身份验证模块后，您可以使用以下命令检查 SSSD 是否正在运行：

```
root # systemctl status sssd
sssd.service - System Security Services Daemon
   Loaded: loaded (/usr/lib/systemd/system/sssd.service; enabled)
   Active: active (running) since Thu 2015-10-23 11:03:43 CEST; 5s ago
   [...]

```

5.2.2 缓存

为了允许用户在身份验证后端不可用时登录，SSSD 将使用其缓存，即使缓存已失效。这种情况会一直持续到后端再次可用。

要使缓存失效，请运行 `sss_cache -E` (`sss_cache` 命令是软件包 `sssd-tools` 的一部分)。

要彻底去除 SSSD 缓存，请运行：

```
tux > sudo systemctl stop sssd
tux > sudo rm -f /var/lib/sss/db/*
tux > sudo systemctl start sssd
```

6 389 LDAP Directory Server

轻量级目录访问协议 (LDAP) 是设计用来访问和维护信息目录的协议。LDAP 可用于执行多种任务，例如用户和组管理、系统配置管理和地址管理。在 SUSE Linux Enterprise Server 15.3 中，LDAP 服务由取代了 OpenLDAP 的 389 Directory Server 提供。

理想情况下，中心服务器会将数据储存在一个目录中，并使用明确定义的协议将其分发到所有客户端。结构化数据使各种应用程序都能对其进行访问。中心储存库减少了必要的管理工作。利用 LDAP 这样的标准化开放协议可以确保尽可能多的客户端应用程序都能访问这些信息。

这里所说的目录实际上是指一种经过优化能够快速有效地读取和搜索的数据库。储存在此目录中的数据类型往往会长期保留，且不会经常更改。这样，便可以针对高性能的并发读取优化 LDAP 服务，而传统数据库的优化目的是在短时间内接受大量的数据写入。

6.1 LDAP 目录树的结构

本节介绍 LDAP 目录树的布局，并提供 LDAP 相关的基本术语。如果您熟悉 LDAP，请继续阅读第 6.2.1 节“设置新的 389 Directory Server 实例”。

LDAP 目录具有树形结构。目录中的所有项（称为对象）在此层次结构中都有确定的位置。此层次结构称为**目录信息树** (DIT)。所需项的完整路径（可以明确标识该项）称为**判别名** (DN)。树中的对象由其**相对判别名** (RDN) 标识。判别名是基于项路径中的所有项的 RDN 构建的。

LDAP 目录树中的关系在下例中尤为明显，如图 6.1 “LDAP 目录的结构”所示。

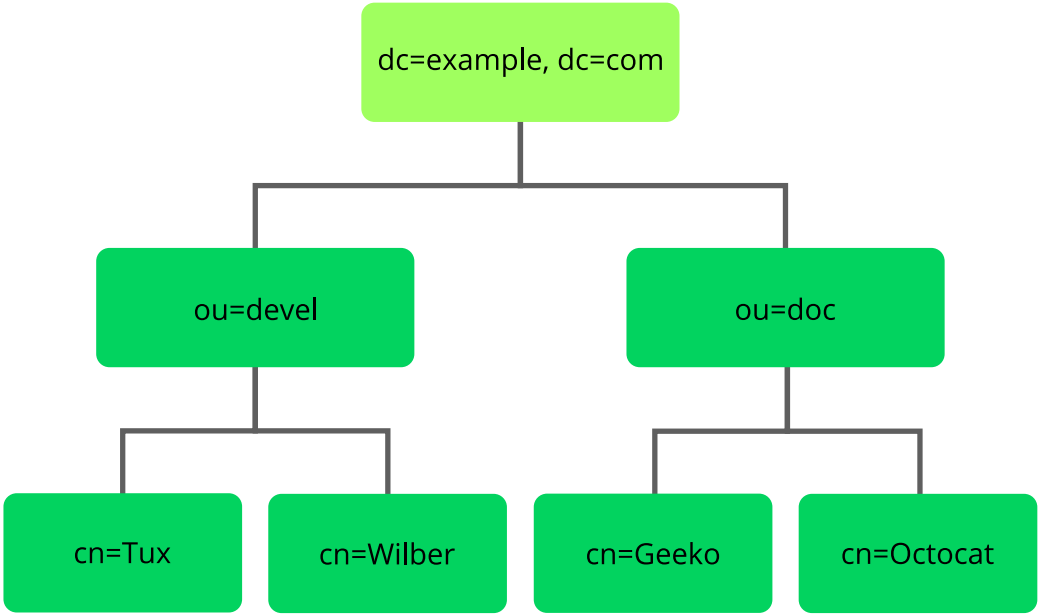


图 6.1：LDAP 目录的结构

完整的图是一个虚构的目录信息树。其中描述了三个层次上的项。每个项对应于图中的一个框。在本例中，虚构的雇员 **Geeko Linux** 的完整有效判别名为 `cn=Geeko Linux,ou=doc,dc=example,dc=com`。该名称是通过将 RDN `cn=Geeko linux` 添加到前一项的 DN `ou=doc,dc=example,dc=com` 而构成的。

可储存在 DIT 中的对象类型是按照**纲要**全局确定的。对象类型由**对象类**决定。对象类确定必须或可以指派给相关对象的属性。纲要包含 LDAP 服务器可以使用的所有对象类和属性。属性是结构化数据类型。其语法、顺序和其他行为由纲要定义。LDAP 服务器会提供一组可在各种环境中工作的核心纲要。如果您需要使用自定义纲要，可以将其上载到 LDAP 服务器。

表 6.1 “常用对象类和属性”提供了示例中所用 `00core.ldif` 和 `06inetorgperson.ldif` 中的对象类的简单概述，包括必需的属性（必需属性）和有效的属性值。安装 389 Directory Server 后，可在 `usr/share/dirsrv/schema` 中找到这些信息。

表 6.1：常用对象类和属性

对象类	含义	示例项	必需属性
域	域的名称组成部分	示例	displayName
<code>organizationalUnit</code>	组织单元	doc	ou

对象类	含义	示例项	必需属性
<u>nsPerson</u>	内部网或因特网中与个人相关的数据	Geeko Linux	cn

例 6.1 “CN=schema 的摘录” 显示了某个纲要指令的摘录及其解释。

例 6.1：CN=SCHEMA 的摘录

```

attributetype (1.2.840.113556.1.2.102 NAME 'memberOf' ❶
    DESC 'Group that the entry belongs to' ❷
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 ❸
    X-ORIGIN 'Netscape Delegated Administrator') ❹

objectclass (2.16.840.1.113730.3.2.333 NAME 'nsPerson' ❺
    DESC 'A representation of a person in a directory server' ❻
    SUP top STRUCTURAL ❼
    MUST ( displayName $ cn ) ❽
    MAY ( userPassword $ seeAlso $ description $ legalName $ mail \
        $ preferredLanguage ) ❾
    X-ORIGIN '389 Directory Server Project'
    ...

```

- ❶ 属性的名称、其唯一**对象标识符**（OID，数字格式）和属性的缩写。
- ❷ 通过 DESC 对该属性提供的简要说明。在此还可以指出定义所基于的相应 RFC。
- ❸ 可保存在该属性中的数据类型。在本例中，它是一个不区分大小写的目录字符串。
- ❹ 纲要元素（例如项目的名称）的来源。
- ❺ 对象类 nsPerson 的定义以 OID 以及该对象类的名称开头（与属性的定义类似）。
- ❻ 对象类的简要说明。
- ❼ SUP top 项指示此对象类不从属于另一个对象类。
- ❽ MUST 列出必须对 nsPerson 类型的对象使用的所有属性类型。
- ❾ MAY 列出可选择对此对象类使用的所有属性类型。

6.2 安装 389 Directory Server

使用以下命令进行安装：

```
tux > sudo zypper install 389-ds
```

此命令将安装 `389-ds` 和 `lib389` packages. 安装后，按照第 6.2.1 节中所述设置服务器。

6.2.1 设置新的 389 Directory Server 实例

您需使用 **dscreate** 命令来创建新的 389 Directory Server 实例，并使用 **dsctl** 命令彻底去除这些实例。

可以基于自定义配置文件以及基于自动生成的模板文件这两种方式来配置和创建新实例。对于测试实例，您可以使用自动生成的模板，而无需进行任何更改，不过，对于生产系统，则必须仔细检查该模板并进行任何必要的更改。

然后，您需要设置管理身份凭证，管理用户和组，并配置身份服务。

执行以下步骤可设置一个用于测试和开发的简单实例，并在其中填充少量的示例项。

1. 使用自定义配置文件创建 389 Directory Server 实例
2. 基于模板创建 389 Directory Server 实例
3. 配置用于本地管理的管理员身份凭证
4. 管理 LDAP 用户和组
5. 设置 SSSD
6. 管理模块
7. 使用 TLS 的 CA 证书

389 目录服务器由三个主要命令控制：

dsctl

管理本地实例，需要 `root` 权限。要求您连接到运行目录服务器实例的终端。用于启动、停止和备份数据库以及进行其他操作。

dsconf

用于管理和配置服务器的主要工具。可通过实例的外部接口管理其配置。这样，您便可以在该实例上远程更改配置。

dsidm

用于身份管理（管理用户、组、口令等）。权限由访问控制授予，因此，举例而言，用户可以重置自己的口令或更改自己帐户的细节。

6.2.2 使用自定义配置文件创建 389 Directory Server 实例

您可以基于一个简单的自定义配置文件创建新的 389 Directory Server 实例。此文件必须采用 INF 格式，您可以对其随意命名。

默认实例名称为 `localhost`。创建实例后，便无法更改实例名称。您最好创建自己的实例名称，而不要使用默认名称，这样可避免混淆并更容易理解实例的工作方式。

例 6.2 显示了一个可用于创建新 389 Directory Server 实例的示例配置文件。您可以复制并使用此文件，不过，请务必创建您自己的口令。

1. 将下面的示例文件 `ldap1.inf` 复制到您的主目录：

例 6.2：最小 389 DIRECTORY SERVER 实例的配置文件

```
# ldap1.inf

[general]
config_version = 2 ❶

[slapd]
root_password = password ❷
self_sign_cert = True ❸
instance_name = ldap1

[backend-userroot]
sample_entries = yes ❹
suffix = dc=ldap1,dc=com
```

- ❶ 此行为必需的指令，指示这是版本 2 的 INF 设置文件。

- ② 为 LDAP 用户 `cn=Directory Manager` 创建 `root_password`。此用户用于连接（绑定）到目录。
- ③ 在 `/etc/dirsrv/slapd-instance-name` 中创建自我签名的服务器证书。
- ④ 在新实例中填充示例用户和组项。

2. 要根据例 6.2 创建 389 Directory Server 实例，请运行以下命令：

```
tux > sudo dscreate -v from-file ldap1.inf | tee ldap1-output.txt
```

这会显示创建实例期间发生的所有活动，将所有消息储存在 `ldap1-output.txt` 中，并在大约一分钟内创建一个正常工作的 LDAP 服务器。详细输出包含大量有用的信息。如果您不想保存这些信息，请删除该命令的 `| tee ldap1-output.txt` 部分。

3. 如果 `dscreate` 命令失败，会显示消息告诉您原因。更正所有问题后，去除该实例（参见步骤 5）并创建新实例。
4. 成功安装后，系统会报告“已完成 `ldap1` 的安装”。检查新服务器的状态：

```
tux > sudo dsctl ldap1 status
Instance "ldap1" is running
```

5. 以下命令可用于彻底去除实例。第一条命令执行试运行，而不会去除实例。当您确定要去除时，请结合 `--do-it` 选项使用第二条命令：

```
tux > sudo dsctl ldap1 remove
Not removing: if you are sure, add --do-it

tux > sudo dsctl ldap1 remove --do-it
```

此命令还会去除未完整安装的或已损坏的实例。您可以放心地按所需的频率创建和去除实例。

如果您忘记了实例的名称，可使用 `dsctl` 列出所有实例：

```
tux > /usr/sbin/dsctl -l
slapd-ldap1
```

6.2.3 基于模板创建 389 Directory Server 实例

您可以使用 **dscreate** 命令为新 389 Directory Server 实例自动创建模板。这会创建一个无需进行任何更改即可使用的模板，您也可以根据自己的要求更改此模板。所有默认值在模板文件中都有说明，并已注释掉。要进行更改，请取消注释默认值并输入您自己的值。所有选项都有详细的说明。

下面的示例会将模板输出到 stdout：

```
tux > dscreate create-template
```

这有助于快速检查模板，但是，您必须创建一个文件用于创建新 389 Directory Server 实例。您可以对此文件随意命名：

```
tux > dscreate create-template ldap1-template.txt
```

下面是新文件中的代码段：

```
# full_machine_name (str)
# Description: Sets the fully qualified hostname (FQDN) of this system. When
# installing this instance with GSSAPI authentication behind a load balancer,
# set
# this parameter to the FQDN of the load balancer and, additionally, set
# "strict_host_checking" to "false".
# Default value: ldapserver1.test.net
;full_machine_name = ldapserver1.test.net

# selinux (bool)
# Description: Enables SELinux detection and integration during the
# installation
# of this instance. If set to "True", dscreate auto-detects whether SELinux is
# enabled. Set this parameter only to "False" in a development environment.
# Default value: True
;selinux = True
```

从中可以了解它是如何根据现有环境自动配置默认值的。按原样使用此文件来创建一个新实例：

```
tux > sudo dscreate from-file ldap1-template.txt
```

这会创建一个名为 localhost 的新实例，并在创建后自动启动该实例：

```
tux > sudo dsctl localhost status
Instance "localhost" is running
```

使用默认值会创建一个完全可正常运行的实例，不过您也可以更改某些值。

创建实例后，便无法更改实例名称。您最好创建自己的实例名称，而不要使用默认名称，这样可避免混淆并更容易理解实例的工作方式。为此，请取消注释 `;instance_name = localhost` 行，并将 `localhost` 更改为所需的名称。在以下示例中，实例名称为 `ldap1`。

另一项有用的更改是在新实例中填充示例用户和组。取消注释 `;sample_entries = no`，并将 `no` 更改为 `yes`。

通过取消注释 `;root_password` 并将默认口令替换为您自己的口令来设置口令。

模板不会创建默认后缀，因此您应在 `suffix` 行中配置自己的后缀，如下例所示：

```
suffix = dc=ldap1,dc=com
```

可以使用 `dsctl` 彻底去除任何实例，然后重新开始创建：

```
tux > sudo dsctl ldap1 remove --do-it
```

6.2.4 停止和启动 389 Directory Server

使用 `systemd` 管理 389 Directory Server 实例。获取服务器状态：

```
tux > systemctl status --no-pager --full dirsrv@ldap1.service
● dirsrv@ldap1.service - 389 Directory Server ldap1.
   Loaded: loaded (/usr/lib/systemd/system/dirsrv@.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2021-03-11 08:55:28 PST; 2h 7min ago
     Process: 4451 ExecStartPre=/usr/lib/dirsrv/ds_systemd_ask_password_acl /etc/dirsrv/slapd-ldap1/dse.ldif (code=exited, status=0/SUCCESS)
    Main PID: 4456 (ns-slapd)
      Status: "slapd started: Ready to process requests"
        Tasks: 26
       CGroup: /system.slice/system-dirsrv.slice/dirsrv@ldap1.service
               └─4456 /usr/sbin/ns-slapd -D /etc/dirsrv/slapd-ldap1 -i /run/dirsrv/slapd-ldap1.pid
```

启动和停止服务器：

```
tux > sudo systemctl start dirsrv@ldap1.service
tux > sudo systemctl stop dirsrv@ldap1.service
```

有关使用 **systemctl** 的详细信息，请参见《管理指南》，第 15 章 “systemd 守护程序”。

dsctl 命令还可启动和停止服务器：

```
tux > sudo dsctl ldap1 status
tux > sudo dsctl ldap1 stop
tux > sudo dsctl ldap1 restart
tux > sudo dsctl ldap1 start
```

6.2.5 配置用于本地管理的管理员身份凭证

要对 389 Directory Server 进行本地管理，您可以在 `/root` 目录中创建一个 `.dsrc` 配置文件，这样 `root` 和 `sudo` 用户管理服务器时就不必每运行一条命令都要键入连接细节。例 6.3 显示了在服务器上进行本地管理的示例，其中使用了 `ldap1` 和 `com` 作为基本 DN。

创建 `/root/.dsrc` 文件后，请尝试运行几条管理命令，例如创建新用户（参见第 6.4 节 “管理 LDAP 用户和组”）。

例 6.3：用于本地管理的 `.dsrc` 文件

```
# /root/.dsrc file for administering the ldap1 instance

[ldap1] ❶

uri = ldapi://%%2fvar%%2frun%%2fslapd-ldap1.socket ❷
basedn = dc=ldap1,dc=com
binddn = cn=Directory Manager
```

- ❶ 此处必须指定确切的实例名称。
- ❷ `ldapi` 将检测尝试登录到服务器的用户的 UID 和 GID。如果 UID/GID 为 `0/0` 或 `dirsrv:dirsrv`，则 `ldapi` 会将该用户绑定为目录服务器根 DN，即 `cn=Directory Manager`。

在 URI 中，斜杠将替换为 `%%2f`，因此在本示例中，路径为 `/var/run/slapd-ldap1.socket`。

6.3 防火墙配置

389 Directory Server 的默认 TCP 端口为 389 和 636。TCP 端口 389 用于建立未加密连接，以及用于 STARTTLS。端口 636 用于通过 TLS 建立加密连接。

`firewalld` 是 SUSE Linux Enterprise 的默认防火墙管理器。以下规则会激活 `ldap` 和 `ldaps` 防火墙服务：

```
tux > sudo firewall-cmd --add-service=ldap --zone=internal
tux > sudo firewall-cmd --add-service=ldaps --zone=internal
tux > sudo firewall-cmd --runtime-to-permanent
```

请将 `zone` 替换为您服务器的相应区域。有关使用 TLS 保护连接的信息，请参见第 6.8 节“使用 TLS 的 CA 证书”；有关 `firewalld` 的信息，请参见第 24.3 节“防火墙基础知识”。

6.4 管理 LDAP 用户和组

通过 `dsidm` 命令来创建并管理用户和组。该命令以交互方式运行，或者，您也可以在命令行上运行，并在一条命令中输入所有选项。

列出现有的用户和组：

```
tux > sudo dsidm ldap1 user list
tux > sudo dsidm ldap1 group list
```

列出有关单个用户的所有信息：

```
tux > sudo dsidm ldap1 user get username
```

列出有关单个组的所有信息：

```
tux > sudo dsidm ldap1 group get groupname
```

列出组的成员：

```
tux > sudo dsidm ldap1 group members demo_group
```

在下面的示例中，我们通过命令行参数指定相应数据来添加两个用户：wilber 和 geeko。示例服务器实例名为 ldap1，该实例的后缀为 dc=ldap1,dc=com。

过程 6.1：创建 LDAP 用户

LDAP 用户是已经存在的用户。他们应具有 Linux 系统帐户、Linux 登录名和主目录。将这些用户添加到您的 LDAP 服务器可集中管理您网络中的用户。您需要输入准确的用户数据，这些信息可通过以下方式来获取：访问存放这些用户帐户的计算机，然后运行 id 命令，如以下针对 Wilber Fox 的命令示例所示：

```
tux > id geeko
uid=1001(wilber) gid=101(users) groups=101(users)
```

1. 创建 LDAP 用户 wilber：

```
tux > sudo dsidm ldap1 user create --uid wilber \
--cn wilber --displayName 'Wilber Fox' --uidNumber 1001 --gidNumber 101 \
--homeDirectory /home/wilber
```

2. 通过查询新用户的判别名（目录对象的完全限定名称，能够确保唯一）进行校验：

```
tux > sudo dsidm ldap1 user get wilber
dn: uid=wilber,ou=people,dc=ldap1,dc=com
[...]
```

执行更改用户口令等操作时需要用到判别名。

3. 为新用户 wilber 创建口令：

a. tux > **sudo** dsidm ldap1 account reset_password \
uid=wilber,ou=people,dc=ldap1,dc=com

b. 输入 wilber 的新口令两次。

如果操作成功，您将看到以下消息：

```
reset password for uid=wilber,ou=people,dc=ldap1,dc=com
```

使用相同的命令更改现有口令。

4. 创建用户 geeko:

```
tux > sudo dsidm ldap1 user create --uid geeko\  
--cn geeko --displayName 'Suzanne Geeko' \  
--uidNumber 1002 --gidNumber 102 --homeDirectory /home/geeko  
  
tux > sudo dsidm ldap1 account reset_password \  
uid=geeko,ou=people,dc=ldap1,dc=com
```

5. 校验用户的口令是否有效:

```
tux > ldapwhoami -D uid=geeko,ou=people,dc=ldap1,dc=com -W  
Enter LDAP Password:  
dn: uid=geeko,ou=people,dc=ldap1,dc=com
```

以下示例会删除用户 wilber:

```
tux > sudo dsidm ldap1 user delete  
Enter dn to delete : uid=wilber,ou=people,dc=ldap1,dc=com  
Deleting nsUserAccount uid=wilber,ou=people,dc=ldap1,dc=com :  
Type 'Yes I am sure' to continue: Yes I am sure  
Successfully deleted uid=wilber,ou=people,dc=ldap1,dc=com
```

过程 6.2：创建 LDAP 组并将用户指派到其中

在以下示例中，我们将创建 server_admins 组，并将 wilber 用户指派到此组。示例服务器实例名为 ldap1，该实例的后缀为 dc=ldap1,dc=com。

1. 创建组:

```
tux > sudo dsidm ldap1 group create
```

系统会提示您输入组名。输入所选的组名:

```
Enter value for cn : server_admins
```

2. 将用户 wilber 添加到该组：

```
tux > sudo dsidm ldap1 group add_member server_admins
uid=wilber,ou=people,dc=ldap1,dc=com
added member: uid=wilber,ou=people,dc=ldap1,dc=com
```

6.5 设置 SSSD

SSSD（系统安全服务守护程序）是一个与远程身份提供者通讯并允许 pam 和 nsswitch 使用该数据的守护程序。SSSD 可以有多个后端，可缓存用户和组并提供 SSH 密钥分发等功能。

1. 在单独的服务器上安装 sssd 和 sssd-ldap 软件包：

```
tux > sudo zypper in sssd sssd-ldap
```

2. 禁用并停止 nscd 守护程序，因为它与 sssd 相冲突：

```
tux > sudo systemctl disable nscd && systemctl stop nscd
```

3. 创建 SSSD 配置，并仅允许我们在[过程 6.2](#) 中创建的 server_admins 组的成员登录：



提示

需要启用 memberOf 插件，以使客户端能够登录和授权（参见[第 6.6 节 “管理模块”](#)）。

```
tux > sudo dsidm localhost client_config sssd.conf server_admins
```

4. 查看输出并将其粘贴（或重定向）到 /etc/sss/sss.conf。必要时，请根据需要编辑配置文件。
5. 要在您的客户端上配置证书，请将 LDAP 服务器中的 ca.crt 复制到您的客户端：

```
tux > sudo mkdir -p /etc/openldap/certs
cp [...]>ca.crt /etc/openldap/certs/
/usr/bin/c_rehash /etc/openldap/certs
```

6. 启用并启动 SSSD:

```
tux > sudo systemctl enable sssd
systemctl start sssd
```

7. 为确保 SSSD 成为 PAM 和 NSS 的一部分，请遵照 <https://www.port389.org/docs/389ds/howto/howto-sssd.html> 上 “Configure PAM (SUSE)”（配置 PAM (SUSE)）和 “Configure NSS (SUSE)”（配置 NSS (SUSE)）章节中的说明操作。
8. 您的用户必须拥有自己的 SSH 密钥对，并可通过 SSH 访问您的 SSSD 服务器。如果所有设置均正确无误，wilber 可以通过与安装并配置了 SSSD 的计算机所建立的 SSH 连接访问 389 目录服务器实例。但 geeko 做不到这一点，因为 geeko 不属于我们在过程 6.2 中配置的 server_admins 组。

6.6 管理模块

使用以下命令可列出所有已启用和已禁用的可用模块：

```
tux > sudo dsconf -D "cn=Directory Manager" ldap://ldapserver1 plug-in list
Enter password for cn=Directory Manager on ldap://ldapserver1:
7-bit check
Account Policy Plugin
Account Usability Plugin
ACL Plugin
ACL preoperation
[...]
```

例如，下面指出了如何启用第 6.5 节 “设置 SSSD” 中提到的 MemberOf 插件：

```
tux > sudo dsconf -D "cn=Directory Manager" ldap://ldapserver1 plugin memberof
enable
```

请注意，插件名称区分大小写，与列出插件时显示的名称不同。启用插件后，需要重新启动服务器。要避免重新启动服务器，请将 nsslapd-dynamic-plugins 参数设置为 on：

```
tux > sudo dsconf -D "cn=Directory Manager" ldap://ldapserver1 config replace
nsslapd-dynamic-plugins=on
```

```
Enter password for cn=Directory Manager on ldap://ldapserver1:
Successfully replaced "nsslapd-dynamic-plugins"
```

6.7 从 OpenLDAP 迁移到 389 Directory Server

从 SUSE Linux Enterprise 15.3 开始，OpenLDAP 已弃用且不再受支持，它已由 389 Directory Server 取代。SUSE 提供了 **openldap_to_ds** 实用程序用于帮助迁移，该实用程序包含在 **389 Directory Server** 软件包中。

openldap_to_ds 实用程序旨在将尽可能多的迁移工作自动化。但是，每个 LDAP 部署各不相同，因此我们无法编写出适合所有情况的工具。您可能需要执行一些手动步骤，并在尝试进行生产迁移之前全面测试您的迁移过程。

6.7.1 测试从 OpenLDAP 迁移

OpenLDAP 与 389 Directory Server 的差异相当大，可能需要对迁移进行反复测试和调整。执行快速迁移测试可能会很有帮助，这样可大致了解需要执行哪些步骤才能确保迁移成功。

先决条件：

- 一个正在运行的 389 Directory Server 实例。
- 一个采用动态 ldif 格式的 OpenLDAP **slapd** 配置文件或目录。
- OpenLDAP 数据库的 ldif 文件备份。

如果您的 slapd 配置不是动态 ldif 格式，请使用 **slaptest** 创建动态副本。创建 **slapd.d** 目录，然后运行以下命令：

```
tux > sudo slaptest -f /etc/openldap/slapd.conf -F /root/slapd.d
```

这会生成类似于以下示例的内容：

```
tux > sudo ls  slapd.d/*
slapd.d/cn=config.ldif

slapd.d/cn=config:
cn=module{0}.ldif  cn=schema.ldif          olcDatabase={0}config.ldif
```

```
cn=schema          olcDatabase={-1}frontend.ldif  olcDatabase={1}mdb.ldif
```

为每个后缀创建一个 ldif 文件。在以下示例中，后缀为 `dc=ldap1,dc=com`。如果您使用的是 `/etc/openldap/slapd.conf` 格式，请运行以下命令创建 ldif 备份文件：

```
tux > sudo slapcat -f /etc/openldap/slapd.conf -b dc=ldap1,dc=com -l /  
root/ldap1-com.ldif
```

对于 `/etc/openldap/slapd.d` 格式，请使用以下命令：

```
tux > sudo slapcat -f /etc/openldap/slapd.conf -b dc=ldap1,dc=com -l /  
root/ldap1-com.ldif
```

使用 `openldap_to_ds` 分析配置和文件，并显示迁移计划而不更改任何内容：

```
tux > sudo openldap_to_ds ldap1 /root/slapd.d ldap1-com.ldif
```

这会执行试运行，但不更改任何内容。输出如下所示：

```
Examining OpenLDAP Configuration ...  
Completed OpenLDAP Configuration Parsing.  
Examining Ldifs ...  
Completed Ldif Metadata Parsing.  
The following migration steps will be performed:  
* Schema Skip Unsupported Attribute -> otherMailbox  
(0.9.2342.19200300.100.1.22)  
* Schema Skip Unsupported Attribute -> dSAQuality (0.9.2342.19200300.100.1.49)  
* Schema Skip Unsupported Attribute -> singleLevelQuality  
(0.9.2342.19200300.100.1.50)  
* Schema Skip Unsupported Attribute -> subtreeMinimumQuality  
(0.9.2342.19200300.100.1.51)  
* Schema Skip Unsupported Attribute -> subtreeMaximumQuality  
(0.9.2342.19200300.100.1.52)  
* Schema Create Attribute -> suseDefaultBase (SUSE.YaST.ModuleConfig.Attr:2)  
* Schema Create Attribute -> suseNextUniqueId (SUSE.YaST.ModuleConfig.Attr:3)  
[...]  
* Schema Create ObjectClass -> suseDhcpConfiguration  
(SUSE.YaST.ModuleConfig.OC:10)  
* Schema Create ObjectClass -> suseMailConfiguration  
(SUSE.YaST.ModuleConfig.OC:11)
```

```
* Database Reindex -> dc=example,dc=com
* Database Import Ldif -> dc=example,dc=com from example.ldif - excluding entry
  attributes = [{'structuralobjectclass', 'entrycsn'}]
No actions taken. To apply migration plan, use '--confirm'
```

以下示例会执行迁移，其输出与试运行后的输出不同：

```
tux > sudo openldap_to_ds ldap1 /root/slapd.d ldap1-com.ldif --confirm
Starting Migration ... This may take some time ...
migration: 1 / 40 complete ...
migration: 2 / 40 complete ...
migration: 3 / 40 complete ...
[...]
Index task index_all_05252021_120216 completed successfully
post: 39 / 40 complete ...
post: 40 / 40 complete ...
# Migration complete!
-----
You should now review your instance configuration and data:
* [ ] - Create/Migrate Database Access Controls (ACI)
* [ ] - Enable and Verify TLS (LDAPS) Operation
* [ ] - Schedule Automatic Backups
* [ ] - Verify Accounts Can Bind Correctly
* [ ] - Review Schema Inconsistent ObjectClass -> pilotOrganization
  (0.9.2342.19200300.100.4.20)
* [ ] - Review Database Imported Content is Correct -> dc=ldap1,dc=com
```

迁移完成后，**openldap_to_ds** 会创建必须完成的迁移后任务核对清单。最好记录所有迁移后步骤，以便可以在后期生产过程中再现这些步骤。然后测试客户端和应用程序与迁移后 389 Directory Server 实例的集成。

❗ 重要：制定回滚计划

必须制定一个回滚计划，以防发生任何失败。此计划应该定义成功迁移的要素、用于确定哪些方面正常以及哪些方面需要修复的测试、至关重要的步骤、可以推迟的事项、如何确定何时撤消更改、如何在尽量减少服务中断的情况下撤消更改，以及其他哪些团队需要参与迁移。

由于部署的可变性，很难提供成功进行生产迁移的通用方法。一旦您全面测试了迁移过程并确认可以获得较好的结果，就可以采取一些有用的常规步骤：

- 在做出更改之前的 48 小时将所有主机名/DNS TTL 减少至 5 分钟，以便能够快速回滚到现有的 OpenLDAP 部署。
- 暂停所有数据同步和传入数据进程，以确保 OpenLDAP 环境中的数据在迁移过程中不会更改。
- 在迁移之前准备好所有要部署 389 Directory Server 的主机。
- 准备好测试迁移文档。

6.7.2 规划迁移

由于 OpenLDAP 如同一个“零件箱”且高度可自定义，因此无法制定出放之四海皆准的迁移计划。有必要针对 OpenLDAP 和其他集成评估您的当前环境和配置。这包括但不限于：

- 复制拓扑。
- 高可用性和负载均衡器配置。
- 外部数据流（IGA、HR、AD 等）。
- 配置的叠加组件（389 Directory Server 中的插件）。
- 客户端配置和预期的服务器功能。
- 自定义纲要。
- TLS 配置。

规划 389 Directory Server 部署的最终大致结果。此项规划包含的内容与上面的列表相同，不过需要将叠加组件替换为插件。评估当前环境并规划您的 389 Directory Server 环境将会是什么样之后，便可以制定迁移计划。建议构建与 OpenLDAP 环境并行的 389 Directory Server 环境，以便可以在两者之间切换。

从 OpenLDAP 迁移到 389 Directory Server 属于单向迁移。两者之间的差异相当大，因此它们不可互操作，并且不存在从 389 Directory Server 到 OpenLDAP 的迁移路径。下表重点指出了两者的主要相似和不同之处。

功能	OpenLDAP	389 Directory Server	兼容
双向复制	SyncREPL	特定于 389 Directory Server 的系统	否
MemberOf	叠加组件	插件	是，仅限简单配置
外部身份验证	代理	-	否
Active Directory 同步	-	Winsync 插件	否
内置纲要	OLDAP 纲要	389 Directory Server 纲要	是，受迁移工具支持
自定义纲要	OLDAP 纲要	389 Directory Server 纲要	是，受迁移工具支持
数据库导入	LDIF	LDIF	是，受迁移工具支持
口令哈希	不确定	不确定	是，支持除 Argon2 以外的所有格式
从 OpenLDAP 复制到 389 Directory Server	-	-	不提供用于复制到 389 Directory Server 的机制
基于时间的一次性口令 (TOTP)	TOTP 叠加组件	-	否，将来可能受支持
entryUUID	OpenLDAP 的组成部分	插件	是

6.8 使用 TSL 的 CA 证书

您可以使用以下命令行工具管理 389 目录服务器的 CA 证书：[`certutil`](#)、[`openssl`](#) 和 [`pk12util`](#)。

如要进行测试，您可以使用 **dscreate** 创建一个自我签名证书。该证书储存在 `/etc/dirsrv/slapd-localhost/ca.crt` 中。如要进行远程管理，需将该证书复制到某个可读取的位置。对于生产环境，请联系您的组织所选的 CA 机构，并请求服务器证书、客户端证书和根证书。在执行以下过程之前，请确保符合下述要求：

- 您拥有可用于建立 TLS 连接的服务器证书和私用密钥。
- 已设置 NSS（网络安全服务）数据库（例如，使用 **certutil** 命令设置）。

在可将现有私用密钥和证书导入 NSS（网络安全服务）数据库之前，需要创建私用密钥和服务器证书的捆绑包。这会生成一个 `*.p12` 文件。

重要：*.p12 文件和友好名称

在创建 PKCS12 捆绑包时，必须将 `*.p12` 文件中的友好名称进行编码。

请务必使用 `Server-Cert` 作为友好名称，否则 TLS 连接将会失败，因为 389 目录服务器只会搜索此字符串。

请注意，将 `*.p12` 文件导入 NSS 数据库后，就无法更改该友好名称。

1. 使用下面的命令可创建包含所需友好名称的 PKCS12 捆绑包：

```
root # openssl pkcs12 -export -in SERVER.crt \  
-inkey SERVER.key -out SERVER.p12 \  
-name Server-Cert
```

请将 `SERVER.crt` 替换为服务器证书，将 `SERVER.key` 替换为要捆绑的私用密钥。使用 `-out` 指定 `*.p12` 文件的名称。使用 `-name` 设置要使用的友好名称：`Server-Cert`。

2. 在可将该文件导入 NSS 数据库之前，需获取该文件的口令。为此，请使用下面的命令：

```
pk12util -i PATH_TO_SERVER.p12 -d sql:PATH_TO_NSS_DB -n Server-cert -  
W SERVER.p12_PASSWORD
```

然后，您便可在 `PATH_TO_NSS_DB` 目录下的 `pwdfile.txt` 文件中找到该口令。

3. 现在，将 `SERVER.p12` 文件导入 NSS 数据库：

```
pk12util -i SERVER.p12 -d PATH_TO_NSS_DB
```

6.9 更多信息

有关 389 目录服务器的详细信息，请参见 <https://www.port389.org/docs/389ds/documentation.html> 中的上游文档。

7 使用 Kerberos 进行网络身份验证

Kerberos 是一个网络身份验证协议，同时还提供加密。本章介绍如何设置 Kerberos 以及集成 LDAP 和 NFS 等服务。

7.1 概念概述

除了通常的口令机制外，开放网络没有提供任何其他方法来确保工作站能够正确识别其用户。在一般的安装中，用户每次访问网络中的服务时都必须输入口令。Kerberos 提供了一种身份验证方法，采用这种方法，用户只要注册一次，就可在整个网络中获得信任以完成会话的剩余操作。要拥有安全的网络，必须满足以下要求：

- 使所有用户可以对每个所需服务证明他们自己的身份，并确保任何用户都不能使用其他用户的身份。
- 确保每个网络服务器也能证明其身份。否则攻击者就可能冒充服务器并获取传送给服务器的敏感信息。这种概念被称为**相互身份验证**，因为在客户端和服务器之间进行了相互身份验证。

Kerberos 通过提供严格加密的认证来帮助您满足这些要求。这里仅讨论 Kerberos 的基本原理。有关详细技术说明，请参见 Kerberos 文档。

7.2 Kerberos 术语

以下词汇表定义了一些 Kerberos 术语。

身份凭证

用户或客户端需要提供某种身份凭证才能获得授权来请求服务。Kerberos 支持两种身份凭证 — 票据和身份验证器。

票据

票据是随服务器而不同的身份凭证，客户端使用票据向它请求提供服务的服务器进行身份验证。它包含服务器的名称、客户端的名称、客户端的因特网地址、时戳、有效期和随机会话密钥。所有这些数据都使用服务器的密钥进行了加密。

身份验证器

身份验证器与票据结合使用，可用于证明提供票据的客户端确实与其声称的身份相符。身份验证器是使用客户端的名称、工作站的 IP 地址和当前工作站的时间（所有这些信息都通过只有客户端和相关服务器知道的会话密钥加密）构建的。与票据不同，身份验证器只能使用一次。客户端可以自己构建身份验证器。

主体

Kerberos 主体是可以对其指派票据的独特实体（用户或服务）。主体包含以下部分：

```
USER/INSTANCE@REALM
```

- **primary:** 主体的第一个部分。对于用户，这通常与用户名相同。
- **instance（可选）：** 描述 **primary** 特征的附加信息。此字符串与 **primary** 之间通过一个 `/` 分隔。
`tux@example.org` 和 `tux/admin@example.org` 可以存在于同一个 Kerberos 系统上，它们被视为不同的主体。
- **realm:** 指定 Kerberos 领域。通常情况下，领域就是您的大写域名。

相互身份验证

Kerberos 确保客户端和服务端都可以确认对方的身份。它们共享一个可用来安全通讯的会话密钥。

会话密钥

会话密钥是由 Kerberos 生成的临时私用密钥。客户端知道这些密钥。当客户端向服务器请求并收到票据后，将使用这些密钥来加密客户端和服务端之间的通讯。

重放

几乎所有在网络中发送的消息都能够被窃听、盗取和重发送。在使用 Kerberos 的情况下，如果攻击者获取了包含您的票据和身份验证器的服务请求，则会非常危险。攻击者随后可能会试图重新发送此请求（**重放**）来冒充您。然而，Kerberos 实施了多种机制来应对此问题。

服务器或服务

服务用来指要执行的特定操作。此操作幕后的进程称为**服务器**。

7.3 Kerberos 的工作原理

Kerberos 常常被称为第三方可信身份验证服务，这意味着其所有客户端都信任 Kerberos 对另一个客户端身份的判断。Kerberos 保存着一个包含它的所有用户及其私用密钥的数据库。

为确保 Kerberos 正常工作，请在专用计算机上运行身份验证和票据授权服务器。确保只有管理员能直接或通过网络访问此计算机。将此计算机上运行的（网络）服务数目降到最低 — 甚至不要运行 `sshd`。

7.3.1 首次联系

在首次接触 Kerberos 时，您的操作与在常规网络系统进行的任何登录过程类似。输入您的用户名。这一信息和票据授权服务的名称被发送到身份验证服务器 (Kerberos)。如果身份验证服务器知道您的身份，它会生成一个随机会话密钥，供以后在客户端和票据授权服务器之间使用。身份验证服务器现在将为票据授权服务器准备一个票据。该票据包含以下信息 — 仅认证服务器和票据授权服务器知道的、由会话密钥加密的所有信息：

- 客户端和票据授权服务器的名称
- 当前时间
- 为此票据指派的有效期
- 客户端的 IP 地址
- 新生成的会话密钥

随后，还是以加密形式将此票据与会话密钥一起发送回客户端，但这次使用的是客户端的私用密钥。只有 Kerberos 和客户端知道此私用密钥，因为它是从您的用户口令派生的。由于客户端已经收到了此响应，计算机将提示您输入口令。此口令被转换为一个密钥，利用它可解密身份验证服务器所发送的包。然后“拆封”此包，并将口令和密钥从工作站的内存中删除。只要没有超过为用于获取其他票据的那个票据指定的有效期，工作站就能证明您的身份。

7.3.2 请求服务

要从网络中的任何服务器请求服务，客户端应用程序都需要向服务器证明其身份。因此，此应用程序生成一个身份验证器。身份验证器包含以下部分：

- 客户端的主体
- 客户端的 IP 地址
- 当前时间
- 校验和（由客户端选择）

所有这些信息都使用客户端为这个特殊服务器接收到的会话密钥进行了加密。用于服务器的身份验证器和票据会被发送到该服务器。该服务器使用自身的会话密钥副本来解密身份验证器，而身份验证器为它提供与请求其服务的客户端相关的全部所需信息，然后服务器将这些信息与票据中包含的信息进行对比。服务器将检查票据和身份验证器是否来自同一客户端。

如果在服务器端没有采取任何安全措施，则这个阶段的过程将成为重放攻击的理想目标。某些人可能试图重发先前从网络上窃取请求。为防止出现这种情况，服务器将不接受具有先前已收到过的时间戳和票据的任何请求。此外，忽略时间戳与接收请求时的时间相差太大的请求。

7.3.3 相互身份验证

Kerberos 身份验证可以双向使用。它不仅可以验证客户端是否为其所声称的客户端，服务器本身也应能够向请求其服务的客户端身份验证自己。因此，它本身会发送身份验证器。它将在客户端的身份验证器中接收的校验和加 1，然后使用它和客户端共享的会话密钥对其加密。客户端将此响应作为对服务器的真实性的校验，然后它们开始协作。

7.3.4 票据授予 — 联系所有服务器

票据每次仅供一个服务器使用。因此，每当您请求另一个服务时，就需要获取一个新票据。Kerberos 实施了一种机制来获取用于各个服务器的票据。这种服务被称为“票据授权服务”。票据授权服务与前面提到的任何服务一样，也使用已介绍过的相同访问协议。当应用程序需要一个尚未请求过的票据时，就会联系票据授权服务器。此请求包含以下部分：

- 被请求的主体
- 票据授权票据
- 认证器

与任何其他服务器一样，票据授权服务器现在将检查票据授权票据和身份验证器。如果确定它们有效，票据授权服务器将构建一个将在原始客户端和新服务器之间使用的新会话密钥。然后构建用于新服务器的票据，其中包含以下信息：

- 客户端的主体
- 服务器的主体
- 当前时间
- 客户端的 IP 地址
- 新生成的会话密钥

新票据具有一个有效期，该有效期是票据授权票据的剩余有效期，或服务的默认有效期。系统将指派这两个值中较小的一个。客户端会接收票据授权服务发送的此票据和会话密钥。但这一次，响应已通过原始票据授权票据附带的会话密钥加密。当联系新服务时，客户端可以解密此响应而不需要用户的口令。因此，Kerberos 无需烦扰用户就能获取客户端的一个又一个票据。

7.4 Kerberos 的用户视图

理想情况下，用户与 Kerberos 的唯一接触是在工作站登录时发生的。登录进程包括获得一个票据授权票据。注销时，用户的 Kerberos 票据会自动损坏，这样其他人就不能模仿该用户。

当用户的登录会话持续时间超过为票据授权票据指定的最长时间限制（合理的设置是 10 小时）时，票据的自动失效可能会造成某种不便。但用户可以通过运行 **kinit** 来获得一个新的票据授权票据。再次输入口令，Kerberos 无需附加身份验证即可获得对所需服务的访问。要获得由 Kerberos 为您静默获取的所有票据的列表，请运行 **klist**。

下面的短列表列出了使用 Kerberos 身份验证的应用程序。安装 `krb5-apps-clients` 软件包后，可在 `/usr/lib/mit/bin` 或 `/usr/lib/mit/sbin` 下找到这些应用程序。它们拥有普通 Unix 和 Linux 应用程序的所有功能，同时具有 Kerberos 管理的透明身份验证的优势：

- **telnet**、**telnetd**
- **rlogin**
- **rsh**、**rcp**、**rshd**

- [ftp](#)、[ftpd](#)
- [ksu](#)

您不再需要输入口令即可使用这些应用程序，因为 Kerberos 已证明您的身份。如果为其编译了 Kerberos 支持，[ssh](#) 甚至可以将为一个工作站获取的所有票据转发到另一个工作站。如果您使用 [ssh](#) 登录到另一个工作站，[ssh](#) 将确保票据的加密内容会根据新情况而调整。仅在工作站之间复制票据是不够的，因为票据中包含工作站特定信息（IP 地址）。XDM 和 GDM 也提供 Kerberos 支持。请阅读 <https://web.mit.edu/kerberos> 上的《Kerberos V5 UNIX User's Guide》（Kerberos V5 UNIX 用户指南）中有关 Kerberos 网络应用程序的详细信息。

7.5 安装和管理 Kerberos

Kerberos 环境由多个组件构成。密钥分发中心 (KDC) 用于容纳中心数据库，该数据库包含所有 Kerberos 相关数据。所有客户端均依赖于 KDC 在整个网络中正确进行身份验证。KDC 和客户端都需要配置为与您的设置相匹配：

一般准备工作

检查网络设置，确保它符合第 7.5.1 节“Kerberos 网络拓扑”中所述的最低要求。为 Kerberos 设置选择适当的领域，请参见第 7.5.2 节“选择 Kerberos 领域”。谨慎设置要充当 KDC 的计算机并应用严格的安全策略，请参见第 7.5.3 节“设置 KDC 硬件”。在网络中设置可靠的时间源以确保所有票据都包含有效时戳，请参见第 7.5.4 节“配置时间同步”。

基本配置

配置 KDC 和客户端，请参见第 7.5.5 节“配置 KDC”和第 7.5.6 节“配置 Kerberos 客户端”。启用 Kerberos 服务的远程管理，这样您就无需对 KDC 计算机进行物理访问，请参见第 7.5.7 节“配置远程 Kerberos 管理”。为领域中的每个服务创建服务主体，请参见第 7.5.8 节“创建 Kerberos 服务主体”。

启用 Kerberos 身份验证

网络中的各种服务都可以使用 Kerberos。要使用 PAM 在应用程序中添加 Kerberos 口令检查，请按第 7.5.9 节“对 Kerberos 启用 PAM 支持”中所述操作。要使用 Kerberos 身份验证配置 SSH 或 LDAP，请按第 7.5.10 节“配置 SSH 进行 Kerberos 身份验证”和第 7.5.11 节“使用 LDAP 和 Kerberos”中所述操作。

7.5.1 Kerberos 网络拓扑

任何 Kerberos 环境都必须符合以下要求才能完全正常运行：

- 提供用于在网络中进行名称解析的 DNS 服务器，以便客户端和服务端能够找到彼此。有关 DNS 设置的信息，请参见《管理指南》，第 31 章“域名系统”。
- 在网络中提供一个时间服务器。使用准确的时戳对于 Kerberos 设置而言至关重要，因为有效的 Kerberos 票据必须包含正确的时戳。有关 NTP 设置的信息，请参见《管理指南》，第 30 章“使用 NTP 同步时间”。
- 提供一个密钥分发中心 (KDC) 作为 Kerberos 体系结构的中心组件。KDC 用于容纳 Kerberos 数据库。在此计算机上使用尽可能最严格的安全策略，以防止此计算机上的任何攻击破坏整个基础结构。
- 将客户端计算机配置为使用 Kerberos 身份验证。

下图描绘了一个简单的示例网络，其中仅包含构建 Kerberos 基础结构至少所需的组件。根据您的部署大小和拓扑，您的设置可能与此不同。

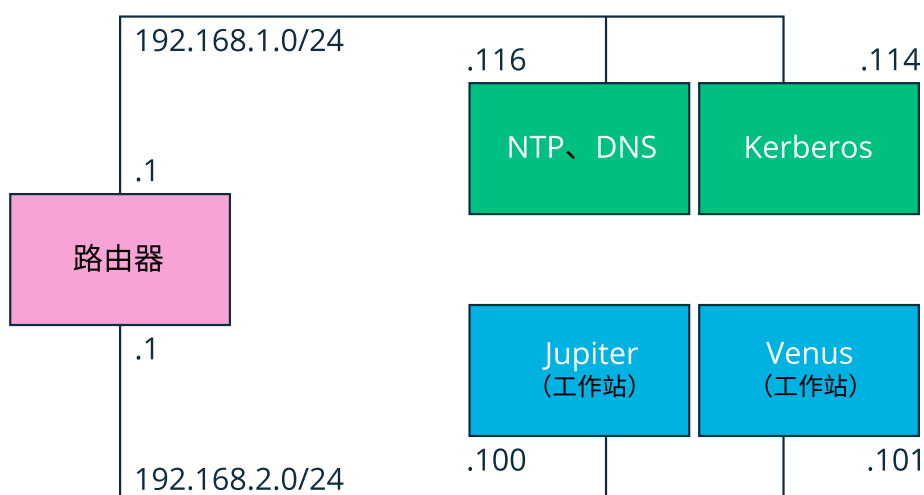


图 7.1：KERBEROS 网络拓扑



提示：配置子网路由

对于类似于图 7.1 “Kerberos 网络拓扑” 中所示的设置，需在两个子网（192.168.1.0/24 和 192.168.2.0/24）之间配置路由。有关使用 YaST 配置路由的详细信息，请参见《管理指南》，第 19 章“基本网络知识”，第 19.4.1.5 节“配置路由”。

7.5.2 选择 Kerberos 领域

Kerberos 安装的域称为领域，通过一个名称（如 `EXAMPLE.COM` 或简单的 `ACCOUNTING`）来标识。Kerberos 区分大小写，因此 `example.com` 实际上是与 `EXAMPLE.COM` 不同的领域。您可根据自己的偏好选择使用大小写。但通常的做法是使用大写领域名。

使用您的 DNS 域名（或子域，如 `ACCOUNTING.EXAMPLE.COM`）也是个不错的选择。如下所示，如果将 Kerberos 客户端配置为通过 DNS 来查找 KDC 和其他 Kerberos 服务，则系统管理员的工作就轻松多了。要做到这一点，将领域名设为 DNS 域名的子域会很有帮助。

与 DNS 名称空间不同，Kerberos 是不分级的。因此，如果您有一个名为 `EXAMPLE.COM` 的领域，该领域包含名为 `DEVELOPMENT` 和 `ACCOUNTING` 的两个“子领域”，这些从属领域不会从 `EXAMPLE.COM` 继承主体。相反，您拥有的是三个独立的领域，并需要为每个领域配置跨领域的身份验证，使一个领域中的用户能够与另一个领域中的服务器或其他用户交互。

为了便于说明，假设您只为整个组织设置一个领域。在本章剩余部分中，将在所有示例中使用领域名 `SAMPLE.COM`。

7.5.3 设置 KDC 硬件

要使用 Kerberos，首先需要一台用作密钥分发中心（或简称 KDC）的计算机。这台计算机将储存整个 Kerberos 用户数据库，其中包含口令和所有信息。

KDC 是安全基础设施中最重要的部分 — 如果有人侵入，则 Kerberos 保护的所有用户帐户和基础设施都会受到损害。能够访问 Kerberos 数据库的攻击者可以冒充数据库中的任何主体。所以应尽可能提高此计算机的安全性：

1. 将服务器计算机放在一个以物理方式保护的位置，如只有极少数人才可进入的加锁服务器室。
2. 除了 KDC 以外，不要在其上运行任何网络应用程序。其中包括服务器和客户端 — 例如，KDC 不应通过 NFS 导入任何文件系统或使用 DHCP 检索其网络配置。
3. 首先安装最小系统，然后检查已安装的软件包列表并除去任何不需要的软件包。其中包括服务器，如 `inetd`、`portmap`、`CUPS` 以及任何基于 X 的软件。甚至安装 SSH 服务器也应该视为潜在的安全风险。

4. 在此计算机上不提供图形登录，因为 X 服务器具有潜在的安全风险。Kerberos 提供它自己的管理界面。
5. 将 `/etc/nsswitch.conf` 配置为仅使用本地文件来进行用户和组查找。将 `passwd` 和 `group` 的行进行如下更改：

```
passwd:      files
group:       files
```

编辑 `/etc` 中的 `passwd`、`group` 和 `shadow` 文件，并去除任何以 `+` 字符开头的行（这些行用于 NIS 查找）。

6. 禁用 `root` 帐户以外的所有其他用户帐户，方法是编辑 `/etc/shadow` 并将哈希口令替换为 `*` 或 `!` 字符。

7.5.4 配置时间同步

要成功使用 Kerberos，应确保您的组织内的所有系统时钟都在给定范围内同步。这一点非常重要，因为 Kerberos 需要防范重放的身份凭证。攻击者可能会在网络上获取 Kerberos 身份凭证，并再使用它们来攻击服务器。Kerberos 采取多种保护措施进行防范。其中之一是在它们的票据中放入时间戳。如果服务器收到的票据的时间戳与当前时间不同，就会拒绝此票据。

Kerberos 在比较时间戳时允许一定的偏差。但计算机时钟的走时可能非常不准确 — 在一周内快或慢半个小时并不罕见。因此，应对网络上的所有主机进行配置，使它们的时钟与中央时间源同步。

要实现此目的，一种简单的方法就是在一台计算机上安装 NTP 时间服务器，并使所有客户端的时钟与该服务器同步。为此，请在所有这些计算机上以客户端的身份运行 NTP 守护程序 `chronyd`。KDC 本身也需要与公用时间源同步。由于在此计算机上运行 NTP 守护程序会有安全风险，通过 cron 作业运行 `chronyd -q` 执行此操作可能是个不错的选择。要将您的机器配置成 NTP 客户端，如《管理指南》，第 30 章 “使用 NTP 同步时间”，第 30.1 节 “使用 YaST 配置 NTP 客户端” 中概述的那样继续操作。

另一种保护时间服务并仍旧使用 NTP 守护程序的做法是，将一个硬件参考时钟挂接到专用的 NTP 服务器，并将另一个硬件参考时钟挂接到 KDC。

也可以调整 Kerberos 在检查时间戳时所允许的最大偏差。此值（称为**时钟扭斜**）可以在 `krb5.conf` 文件中进行设置，请参见第 7.5.6.3 节“[调整时钟偏差](#)”一节。

7.5.5 配置 KDC

本节介绍 KDC 的初始配置和安装，包括创建管理主体。此过程包括几个步骤：

- 1. 安装 RPM：** 在指定用作 KDC 的计算机上安装 `krb5`、`krb5-server` 和 `krb5-client` 软件包。
- 2. 调整配置文件：** 必须根据您的方案调整 `/etc/krb5.conf` 和 `/var/lib/kerberos/krb5kdc/kdc.conf` 配置文件。这些文件 KDC 上的所有信息。请参见第 7.5.5.1 节“[配置服务器](#)”。
- 3. 创建 Kerberos 数据库：** Kerberos 保持所有主体标识和需要被认证的所有主体密码的数据库。有关详细信息，请参考第 7.5.5.2 节“[设置数据库](#)”。
- 4. 调整 ACL 文件：添加管理员：** 可以远程管理 KDC 上的 Kerberos 数据库。要防止未授权的主体篡改数据库，Kerberos 使用访问控制列表。必须为管理员主体显式启用远程访问，以使其能够管理数据库。Kerberos ACL 文件位于 `/var/lib/kerberos/krb5kdc/kadm5.acl` 下。有关详细信息，请参考第 7.5.7 节“[配置远程 Kerberos 管理](#)”。
- 5. 调整 Kerberos 数据库：添加管理员：** 您至少需要一个管理主体来运行和管理 Kerberos。必须在启动 KDC 之前添加了该主体。有关详细信息，请参考第 7.5.5.3 节“[创建主体](#)”。
- 6. 启动 Kerberos 守护程序：** 安装并正确配置 KDC 软件后，启动 Kerberos 守护程序以便为您的领域提供 Kerberos 服务。有关详细信息，请参考第 7.5.5.4 节“[启动 KDC](#)”。
- 7. 为您自己创建主体：** 您自己需要一个主体。有关详细信息，请参考第 7.5.5.3 节“[创建主体](#)”。

7.5.5.1 配置服务器

Kerberos 服务器的配置存在很多的变数，涉及到您的网络体系结构、DNS 和 DHCP 配置、领域及其他考虑因素。您必须准备好一个默认领域以及域到领域的映射。下面的示例演示了一个极简配置。这并非复制粘贴而来的示例；有关 Kerberos 配置的详细信息，请参见 https://web.mit.edu/kerberos/krb5-latest/doc/admin/conf_files/index.html。

例 7.1：示例 KDC 配置 /etc/krb5.conf

```
[libdefaults]
    dns_canonicalize_hostname = false
    rdns = false
    default_realm = example.com
    ticket_lifetime = 24h
    renew_lifetime = 7d

[realms]
    example.com = {
        kdc = kdc.example.com:88
        admin_server = kdc.example.com
        default_domain = example.com
    }

[logging]
    kdc = FILE:/var/log/krb5kdc.log
    admin_server = FILE:/var/log/kadmind.log
    default = SYSLOG:NOTICE:DAEMON

[domain_realm]
    .example.com = example.com
    example.com = example.com
```

7.5.5.2 设置数据库

下一步是初始化 Kerberos 用来保存所有主体信息的数据库。设置数据库主密钥，此密钥用于防范数据库意外泄漏（特别是当它备份到磁带时）。主密钥是从通行口令派生的，储存在称为暂存文件的文件中。这就是您每次重启动 KDC 时无需键入口令的原因。确保选择一个适当的通行密码，如从一本书随机打开的一页中找出的一句话。

在对 Kerberos 数据库 (`/var/lib/kerberos/krb5kdc/principal`) 进行磁带备份时，切勿备份暂存文件（它位于 `/var/lib/kerberos/krb5kdc/.k5.EXAMPLE.COM` 中）。否则，能够读到此磁带的所有人都可以解密数据库。因此，请将通行口令副本保存在安全位置，因为在系统崩溃后从备份磁带恢复数据库时将用到它。

要创建暂存文件和数据库，请运行：

```
tux > sudo kdb5_util create -r EXAMPLE.COM -s
```

您将看到以下输出：

```
Initializing database '/var/lib/kerberos/krb5kdc/principal' for realm
'EXAMPLE.COM',
master key name 'K/M@EXAMPLE.COM'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key: ❶
Re-enter KDC database master key to verify: ❷
```

❶ Type the master password.

❷ 再次键入口令。

要进行校验，请使用 `list` 命令：

```
tux > kadmin.local

kadmin> listprincs
```

您将看到数据库中的多个主体，这些主体供 Kerberos 内部使用：

```
K/M@EXAMPLE.COM
kadmin/admin@EXAMPLE.COM
kadmin/changepw@EXAMPLE.COM
krbtgt/EXAMPLE.COM@EXAMPLE.COM
```

7.5.5.3 创建主体

为自己创建两个 Kerberos 主体，一个常规主体用于日常工作，另一个用于与 Kerberos 相关的管理任务。假设您的登录名是 geeko，请执行以下操作：

```
tux > kadmin.local  
  
kadmin> ank geeko
```

您将看到以下输出：

```
geeko@EXAMPLE.COM's Password: ❶  
Verifying password: ❷
```

❶ 键入 geeko 的口令。

❷ 再次键入 geeko 的口令。

接下来，在 **kadmin** 提示符处键入 **add geeko/admin** 以创建名为 geeko/admin 的另一个主体。您的用户名的 admin 后缀指定了您的**角色**。稍后在管理 Kerberos 数据库时将使用此角色。一个用户可以有用于不同目的的多个角色。角色的作用就像名称类似但彼此完全不同的帐户。

7.5.5.4 启动 KDC

启动 KDC 守护程序和 Kadmin 守护程序。要手动启动守护程序，请输入：

```
tux > sudo systemctl start krb5kdc  
sudo systemctl start kadmind
```

另请确保在服务器计算机重引导时，默认会启动服务 KDC (krb5kdc) 和 kadmind (kadmind)。请输入以下命令启用这些服务：

```
tux > sudo systemctl enable krb5kdc kadmind
```

或使用 YaST 服务管理器来启用。

7.5.6 配置 Kerberos 客户端

当已部署好支持基础结构（DNS、NTP）且已配置并启动 KDC 时，请配置客户端计算机。要配置 Kerberos 客户端，请使用下面所述的两种手动方法之一。

在配置 Kerberos 时，可以采用两种方法 — 在 `/etc/krb5.conf` 文件中进行静态配置或通过 DNS 进行动态配置。采用 DNS 配置时，Kerberos 应用程序会尝试使用 DNS 记录查找 KDC 服务。采用静态配置时，将您的 KDC 服务器的主机名添加到 `krb5.conf`（并在移动 KDC 或以其他方式重配置领域时更新文件）。

基于 DNS 的配置通常比较灵活，而且每台计算机的配置工作量也少得多，但要求您的领域名与您的 DNS 域相同或是它的子域。通过 DNS 配置 Kerberos 也会产生安全问题：攻击者能够通过 DNS 严重破坏您的基础结构（通过使名称服务器无效、窃取 DNS 记录等），但这最多只会造成拒绝服务攻击。除非在 `krb5.conf` 中输入 IP 地址而非主机名，否则静态配置也会发生相同的情况。

7.5.6.1 静态配置

一种配置 Kerberos 的方法是编辑 `/etc/krb5.conf`。默认安装的文件中包含多个示例项。在开始之前，请擦除所有这些项。`krb5.conf` 由多个部分（段落）构成，每个部分通过括在方括号中的部分名称引入，类似于 `[this]`。

要配置 Kerberos 客户端，请将下面的段落添加到 `krb5.conf`（其中 `kdc.example.com` 是 KDC 的主机名）中：

```
[libdefaults]
    default_realm = EXAMPLE.COM

[realms]
    EXAMPLE.COM = {
        kdc = kdc.example.com
        admin_server = kdc.example.com
    }
```

`default_realm` 行设置了 Kerberos 应用程序的默认领域。如果您有多个领域，请在 `[realms]` 部分添加附加语句。

此外还要向此文件添加一个语句，指示应用程序如何将主机名映射到领域。例如，当连接到远程主机时，Kerberos 库需要知道此主机位于哪个领域中。必须在 `[domain_realms]` 节中对此进行配置：

```
[domain_realm]
```

```
.example.com = EXAMPLE.COM  
www.example.org = EXAMPLE.COM
```

此语句向库说明 `example.com` DNS 域中的所有主机均位于 `SAMPLE.COM` Kerberos 领域中。此外，还应将一个名为 `www.example.org` 的外部主机视为 `EXAMPLE.COM` 领域的成员。

7.5.6.2 基于 DNS 的配置

基于 DNS 的 Kerberos 配置大量使用 SRV 记录。请参见 **(RFC2052) 用于指定服务位置的 DNS RR**，网址是 <http://www.ietf.org>。

对于 Kerberos 而言，SRV 记录的名称始终采用 `_service._proto.realm` 格式，其中 `realm` 是 Kerberos 领域。DNS 中的域名不区分大小写，因此当使用这种配置方法时，Kerberos 领域将无法再区分大小写。`_service` 是一个服务名（例如当尝试联系 KDC 或密码服务时会使用不同的名称）。`_proto` 可以是 `_udp` 或 `_tcp`，但不是所有服务都支持这两种协议。

SRV 资源记录的数据部分包括一个优先级值、一个权重、一个端口号和一个主机名。优先级确定了主机被尝试的顺序（值越低则优先级越高）。权重值用于支持在优先级相同的服务器之间实现一定程度的负载平衡。您也许不需要它们，所以将它们设为 0 即可。

MIT Kerberos 当前查找服务时将查找以下名称：

`_kerberos`

它定义了 KDC 守护程序（身份验证和票据授权服务器）的位置。典型记录如下：

```
_kerberos._udp.EXAMPLE.COM.  IN  SRV    0 0 88 kdc.example.com.  
_kerberos._tcp.EXAMPLE.COM.  IN  SRV    0 0 88 kdc.example.com.
```

`_kerberos-adm`

它描述了远程管理服务的位置。典型记录如下：

```
_kerberos-adm._tcp.EXAMPLE.COM. IN  SRV    0 0 749 kdc.example.com.
```

因为 `kadmind` 不支持 UDP，所以没有 `_udp` 记录。

与静态配置文件一样，这里也提供了一种机制来向客户端指示特定主机位于 `EXAMPLE.COM` 领域中，即使它不是 `example.com` DNS 的一部分。通过将一个 TXT 记录附加到 `_kerberos.host_name` 即可做到这一点，如下所示：

```
_kerberos.www.example.org. IN TXT "EXAMPLE.COM"
```

7.5.6.3 调整时钟偏差

时钟偏差是容许票据时戳与主机系统时钟相差的范围，超过此范围将不接受此票据。时钟偏差通常设置为 300 秒（5 分钟）。这意味着，票据中的时戳最多可以比服务器时钟慢五分钟，并且最多可以快五分钟。

当使用 NTP 同步所有主机时，可以将此值减少为大约一分钟。可以在 `/etc/krb5.conf` 中设置时钟偏差值，如下所示：

```
[libdefaults]
    clockskew = 60
```

7.5.7 配置远程 Kerberos 管理

为了无需直接访问 KDC 控制台即可从 Kerberos 数据库添加和去除主体，请编辑 `/var/lib/kerberos/krb5kdc/kadm5.acl` 以告知 Kerberos 管理服务器要允许哪些主体执行哪些操作。ACL（访问控制列表）文件可让您以精确的控制度指定特权。有关详细信息，请使用 **man 8 kadmind** 来参考手册页。

目前请通过在该文件中插入以下一行向您自己授予管理数据库的特权：

```
geeko/admin *
```

请将用户名 `geeko` 替换为您自己的用户名。重启动 `kadmind` 以使更改生效。

您现在应该能够使用 `kadmin` 工具远程执行 Kerberos 管理任务。首先，为您的 `admin` 角色获得一个票据，并在连接到 `kadmin` 服务器时使用此票据：

```
tux > kadmin -p geeko/admin
Authenticating as principal geeko/admin@EXAMPLE.COM with password.
Password for geeko/admin@EXAMPLE.COM:
kadmin: getprivs
current privileges: GET ADD MODIFY DELETE
kadmin:
```

使用 **getprivs** 命令验证您有哪些特权。上面的列表中列出了全部特权。

例如，修改主体 geeko：

```
tux > kadmin -p geeko/admin
Authenticating as principal geeko/admin@EXAMPLE.COM with password.
Password for geeko/admin@EXAMPLE.COM:

kadmin: getprinc geeko
Principal: geeko@EXAMPLE.COM
Expiration date: [never]
Last password change: Wed Jan 12 17:28:46 CET 2005
Password expiration date: [none]
Maximum ticket life: 0 days 10:00:00
Maximum renewable life: 7 days 00:00:00
Last modified: Wed Jan 12 17:47:17 CET 2005 (admin/admin@EXAMPLE.COM)
Last successful authentication: [never]
Last failed authentication: [never]
Failed password attempts: 0
Number of keys: 2
Key: vno 1, Triple DES cbc mode with HMAC/sha1, no salt
Key: vno 1, DES cbc mode with CRC-32, no salt
Attributes:
Policy: [none]

kadmin: modify_principal -maxlife "8 hours" geeko
Principal "geeko@EXAMPLE.COM" modified.
kadmin: getprinc geeko
Principal: geeko@EXAMPLE.COM
Expiration date: [never]
Last password change: Wed Jan 12 17:28:46 CET 2005
Password expiration date: [none]
Maximum ticket life: 0 days 08:00:00
Maximum renewable life: 7 days 00:00:00
Last modified: Wed Jan 12 17:59:49 CET 2005 (geeko/admin@EXAMPLE.COM)
Last successful authentication: [never]
Last failed authentication: [never]
Failed password attempts: 0
Number of keys: 2
```

```
Key: vno 1, Triple DES cbc mode with HMAC/sha1, no salt
Key: vno 1, DES cbc mode with CRC-32, no salt
Attributes:
Policy: [none]
kadmin:
```

这将票据的最长生命周期更改为 8 小时。有关 **kadmin** 命令和可用选项的详细信息，请参见 [krb5-doc](#) 软件包或 [man 8 kadmin](#) 手册页。

7.5.8 创建 Kerberos 服务主体

到目前为止，我们仅讨论了用户身份凭证。但与 Kerberos 兼容的服务通常也需要将它们自己认证到客户端。因此，对于领域中提供的每个服务，Kerberos 数据库中必须存在特殊的服务主体。例如，如果 `ldap.example.com` 提供 LDAP 服务，您将需要一个服务主体 `ldap/ldap.example.com@EXAMPLE.COM`，以使此服务向所有客户端验证身份。

服务主体的命名约定是 `SERVICE/HOSTNAME@REALM`，其中 `HOSTNAME` 是主机的完全限定主机名。

有效的服务描述符为：

服务描述符	服务
<code>主机</code>	Telnet、RSH、SSH
<code>nfs</code>	NFSv4（提供 Kerberos 支持）
<code>HTTP</code>	HTTP（提供 Kerberos 身份验证）
<code>imap</code>	IMAP
<code>pop</code>	POP3
<code>ldap</code>	LDAP

服务主体类似于用户主体，但存在一些重大差别。用户主体与服务主体之间的主要差别在于，前者的密钥受口令保护。当用户从 KDC 获取票据授权票据时，他们需要键入其口令，以使 Kerberos 能够解密该票据。如果系统管理员大约每 8 小时就必须为 SSH 守护程序获取一次新的票据，将会很不方便。

实际上，用来解密服务主体的初始票据的密钥是由管理员从 KDC 一次性提取的，并储存在名为 **keytab** 的本地文件中。SSH 守护程序等服务将读取此密钥并在需要时使用它来自动获取新票据。默认 keytab 文件位于 `/etc/krb5.keytab` 中。

要为 `jupiter.example.com` 创建主机服务主体，请在您的 `kadmin` 会话期间输入以下命令：

```
tux > kadmin -p geeko/admin
Authenticating as principal geeko/admin@EXAMPLE.COM with password.
Password for geeko/admin@EXAMPLE.COM:
kadmin: addprinc -randkey host/jupiter.example.com
WARNING: no policy specified for host/jupiter.example.com@EXAMPLE.COM;
defaulting to no policy
Principal "host/jupiter.example.com@EXAMPLE.COM" created.
```

`-randkey` 标志没有为新主体设置口令，而是指示 **kadmin** 生成一个随机密钥。之所以在这里使用这个标志，是因为此主体不需要用户交互。它是计算机的一个服务器帐户。

最后，抽取密钥并将其储存在本地 keytab 文件 `/etc/krb5.keytab` 中。这个文件由超级用户拥有，所以您必须是 root 用户才能在 `kadmin` shell 中执行以下命令：

```
kadmin: ktadd host/jupiter.example.com
Entry for principal host/jupiter.example.com with kvno 3, encryption type Triple
DES cbc mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal host/jupiter.example.com with kvno 3, encryption type DES
cbc mode with CRC-32 added to keytab WRFILE:/etc/krb5.keytab.
kadmin:
```

完成后，应确保使用 **kdestroy** 命令销毁通过 `kinit` 获得的 `admin` 票据。

7.5.9 对 Kerberos 启用 PAM 支持



警告：不完整的配置会锁定用户

不完整的 Kerberos 配置可能会将您（包括 root 用户）完全锁定在系统之外。要防止出现这种情况，请在根据下面所述将 `pam_krb5` 模块添加到现有的 PAM 配置文件之后，将 `ignore_unknown_principals` 指令添加到 `pam_krb5` 模块。

```
tux > sudo pam-config --add --krb5-ignore_unknown_principals
```

这会指示 `pam_krb5` 模块忽略某些错误，如不忽略，帐户阶段将会失败。

SUSE® Linux Enterprise Server 随附了一个名为 `pam_krb5` 的 PAM 模块，该模块支持 Kerberos 登录和口令更新。`su` 等控制台登录应用程序以及 GDM 等图形登录应用程序可以使用此模块。也就是说，在希望用户输入口令后由身份验证应用程序代表其获取初始 Kerberos 票据的所有场合，都可以使用此模块。要为 Kerberos 配置 PAM 支持，请使用下面的命令：

```
tux > sudo pam-config --add --krb5
```

上述命令将 `pam_krb5` 模块添加到现有的 PAM 配置文件，并确保以正确的顺序调用该模块。要精确调整 `pam_krb5` 的使用方式，请编辑 `/etc/krb5.conf` 文件并将默认应用程序添加到 PAM。有关详细信息，请使用 `man5 pam_krb5` 来参考手册页。

`pam_krb5` 模块的设计特意不用于接受 Kerberos 票据作为部分用户身份验证的网络服务。这一点很特别，后面将会讨论。

7.5.10 配置 SSH 进行 Kerberos 身份验证

OpenSSH 在协议版本 1 和 2 中均支持 Kerberos 身份验证。在版本 1 中，会通过特殊协议消息传输 Kerberos 票据。版本 2 不再直接使用 Kerberos，而是依赖于 GSSAPI，即通用安全服务 API。这是一种不特定于 Kerberos 的编程接口 — 其设计目的是隐藏基础身份验证系统的特性，无论它是 Kerberos、公共密钥认证系统（如 SPKM）还是其他系统。但是，包含的 GSSAPI 库仅支持 Kerberos。

要将 `sshd` 与 Kerberos 身份验证一起使用，请编辑 `/etc/ssh/sshd_config` 并设置如下选项：

```
# These are for protocol version 1
#
# KerberosAuthentication yes
# KerberosTicketCleanup yes

# These are for version 2 - better to use this
GSSAPIAuthentication yes
GSSAPICleanupCredentials yes
```

然后使用 **sudo systemctl restart sshd** 重新启动您的 SSH 守护程序。

要将 Kerberos 认证与协议版本 2 一起使用，需要在客户端也启用它。此操作可在整个系统范围的配置文件 `/etc/ssh/ssh_config` 中执行，也可通过编辑 `~/.ssh/config` 在每个用户级别上执行。在这两种情况下，均应添加选项 `GSSAPIAuthentication yes`。

您现在应该能够使用 Kerberos 身份验证进行连接。使用 **klist** 校验您是否拥有有效票据，然后连接到 SSH 服务器。要强制使用 SSH 协议版本 1，请在命令行上指定选项 `-1`。



提示：更多信息

文件 `/usr/share/doc/packages/openssh/README.kerberos` 中详细讨论了 OpenSSH 和 Kerberos 的交互。



提示：协议版本 2 的其他指令

支持 `GSSAPIKeyExchange` 机制 (RFC 4462)。此指令指定如何交换主机密钥。有关详细信息，请参见 `sshd_config` 手册页 (**man sshd_config**)。

7.5.11 使用 LDAP 和 Kerberos

Kerberos 提供身份验证，而 LDAP 则用于授权和标识。这两个服务可以配合工作。

为建立安全连接，389 目录服务器支持以不同的方式加密数据：SSL/TLS 连接、启动 TLS 连接和 SASL 身份验证。简单身份验证和安全层 (SASL) 是用于进行身份验证的网络协议。在 SUSE Linux Enterprise Server 中使用的 SASL 实现是 `cyrus-sasl`。Kerberos 身份验证是通过 GSS-API（常规安全服务 API）执行的，该 API 由 `cyrus-sasl-gssapi` 软件包提供。389 目录服务器通过 GSS-API 使用 Kerberos 票据对会话进行身份验证和加密数据。

借助 SASL 框架，您可以使用不同的机制向服务器验证用户的身份。在 Kerberos 中，身份验证永远是相互的。这表示您不仅向 389 目录服务器验证了您自己的身份，389 目录服务器本身也向您验证了它的身份。具体而言，这表示您是与所需的服务器而不是攻击者设置的随机服务进行通讯。

为了使 Kerberos 能够绑定到 389 目录服务器，请创建一个主体 `ldap/ldap.example.com` 并将其添加到 keytab。389 目录服务器用来进行身份验证的身份凭证将由 keytab 提供给其他服务器。389 目录服务器通过 `KRB5_KTNAME` 环境变量指派 keytab。

要设置该变量，请执行以下操作：

1. `tux > sudo systemctl edit dirsrv@INSTANCE`

如果您为 389 目录服务器实例使用了默认名称，请将 `INSTANCE` 替换为 `localhost`。

2. 添加以下命令：

```
[Service]
Environment=KRB5_KTNAME=/etc/dirsrv/slapd-INSTANCE/krb5.keytab
```

3. keytab 文件需可供用于运行 389 目录服务器的帐户（例如 `dirserv`）读取：

```
tux > sudo chown dirsrv:dirsrv /etc/dirsrv/slapd-INSTANCE/krb5.keytab
tux > sudo chmod 600 /etc/dirsrv/slapd-INSTANCE/krb5.keytab
```

7.5.11.1 将 Kerberos 身份验证与 LDAP 搭配使用

要获取并缓存初始票据授权票据，请使用在第 7.5.5.3 节“创建主体”中创建的主体：

```
tux > kinit geeko@EXAMPLE.COM
```

要检查 GSSAPI 身份验证是否正常工作，请运行：

```
tux > ldapwhoami -Y GSSAPI -H ldap://ldapkdc.example.com
dn: uid=testuser,ou=People,dc=example,dc=com
```

GSSAPI 使用 `ccache` 向 389 目录服务器验证用户身份，无需用户提供其口令。

7.5.11.2 配置 SASL 身份映射

在处理 SASL 绑定请求时，389 目录服务器会将 SASL 身份验证 ID（用于向目录服务器进行身份验证）映射到服务器中储存的 LDAP 项。使用 Kerberos 时，SASL 用户 ID 通常采用以下格式：userid@REALM，例如 tux@example.com。必须将此 ID 转换为用户目录服务器项的 DN，例如 uid=tux,ou=people,dc=example,dc=com。389 目录服务器为大多数常用配置随附了一些默认映射。不过，您可以创建自定义的映射。[过程 7.1 “管理映射”](#) 说明了如何列出和显示映射、如何删除映射，以及如何创建自定义映射。

过程 7.1：管理映射

1. 要列出现有的 SASL 映射，请运行以下命令：

```
tux > dsconf INSTANCE sasl list
Kerberos uid mapping
rfc 2829 dn syntax
rfc 2829u syntax
uid mapping
```

2. 要显示映射，请运行以下命令：

```
tux > sudo dsconf INSTANCE sasl get "Kerberos uid mapping"
dn: cn=Kerberos uid mapping,cn=mapping,cn=sasl,cn=config
cn: Kerberos uid mapping
nsSaslMapBaseDNTemplate: dc=\2,dc=\3
nsSaslMapFilterTemplate: (uid=\1)
nsSaslMapRegexString: \(.*\)\@(\.*\)\.\\(.*\)
objectClass: top
objectClass: nsSaslMapping
```

3. 仅当您的 dc 包含两个组件时，默认映射才起作用。要删除映射（如果它不适合您），请运行以下命令：

```
tux > sudo dsconf INSTANCE sasl delete "Kerberos uid mapping"
Deleting SaslMapping cn=Kerberos uid mapping,cn=mapping,cn=sasl,cn=config :
Successfully deleted cn=Kerberos uid mapping,cn=mapping,cn=sasl,cn=config
```

4. 要创建新映射，请运行以下命令：

```
tux > sudo dsconf localhost sasl create --cn=bhgssapi --
nsSaslMapRegexString "\
(.*)@EXAMPLE.NET.DE" --nsSaslMapBaseDNTemplate="dc=example,dc=net,dc=de"
--nsSaslMapFilterTemplate="(uid=\1)"
tux > sudo Enter value for nsSaslMapPriority :
Successfully created bhgssapi
```

5. 使用以下命令显示新创建的映射：

```
tux > sudo dsconf localhost sasl get "bhgssapi"
dn: cn=bhgssapi,cn=mapping,cn=sasl,cn=config
cn: bhgssapi
nsSaslMapBaseDNTemplate: dc=example,dc=net,dc=de
nsSaslMapFilterTemplate: (uid=\1)
nsSaslMapPriority: 100
nsSaslMapRegexString: \(.*)@EXAMPLE.NET.DE
objectClass: top
objectClass: nsSaslMapping
```

使用这些命令，您可以仅检查特定领域的用户，并将其重新映射到不同的 dc 库。可以看到，新映射包含 3 个 dc 组件，因此默认映射不适合此领域 (EXAMPLE.NET.DE)，而只适合 EXAMPLE.NET 这样的领域。

7.6 Kerberos 和 NFS

大多数 NFS 服务器都可以使用默认“信任网络”形式的安全性（称为 sec=sys）与基于 Kerberos 的三种不同级别的安全性（sec=krb5、sec=krb5i 和 sec=krb5p）的任意组合来导出文件系统。sec 选项设置为客户端上的装入选项。一种常见的情况是先配置 NFS 并将其与 sec=sys 配合使用，然后便可以实施 Kerberos。在这种情况下，服务器有可能会配置为同时支持 sec=sys 以及某种 Kerberos 级别，在转换所有客户端后，将会去除 sec=sys 支持，从而实现真正的安全性。转换到 Kerberos 的过程应该非常透明（如果有序进行）。但是，如果使用了 Kerberos，NFS 行为的一个微小细节的工作方式会有所不同，您需要了解并可能需要解决这种差异造成的影响。请参见第 7.6.1 节“组成员资格”。

三种 Kerberos 级别表示不同的安全级别。安全性越高，加密和解密消息所需的处理器资源就越多。在计划对 NFS 实施 Kerberos 时，选择适当的平衡是一个重要考虑因素。

krb5 仅提供身份验证。服务器知道谁发送了请求，而客户端知道服务器发送了答复。它不会为请求或答复的内容提供安全性，因此获得物理网络访问权限的攻击者可能会以各种方式转换请求和/或答复，以欺骗服务器或客户端。他们不能直接读取或更改经过身份验证的用户所不能读取或更改的任何文件，但从理论上说，任何事情几乎都有可能发生。

krb5i 添加了对所有消息的完整性检查。使用 krb5i 时，攻击者无法修改任何请求或答复，但可以查看所有交换的数据，因此可能会看到所读取的任何文件的内容。

krb5p 在协议中添加了隐私保护。除了可靠的身份验证和完整性检查外，消息将完全加密，这样攻击者只能知道在客户端与服务器之间交换了消息，但不能直接从消息中提取其他信息。能否从消息计时中提取信息是 Kerberos 无法解决的另一个问题。

7.6.1 组成员资格

sec=sys 与各种 Kerberos 安全级别之间的一个可以察觉到的行为差异与组成员资格相关。在 Unix 和 Linux 中，每个文件系统访问请求都来自某个进程，该进程由特定的用户拥有，并具有特定的组拥有者和多个补充组。对文件的访问权限因拥有者和各个组而异。

在每个请求中，使用 sec=sys 将 user-id、group-id 以及最多包含 16 个补充组的列表发送到服务器。

如果某个用户是 16 个以上的补充组的成员，超额的组将会丢失，并且在正常情况下用户本应可以访问的某些文件可能无法通过 NFS 访问。因此，使用 NFS 的大多数站点会通过某种方法将所有用户限制为最多 16 个补充组。

如果用户运行 **newgrp** 命令或运行 set-group-id 程序，并且该命令或程序可以更改用户所属的组列表，则这些更改会立即生效，并提供 NFS 上的不同访问权限。

使用 Kerberos 时，请求中不会发送组信息。只会标识用户（使用 Kerberos “主体”），服务器将执行查找来确定该主体的用户 ID 和组列表。这意味着，如果用户是 16 个以上的组的成员，则会使用所有这些组成员资格来确定文件访问权限。但也意味着，如果用户在客户端上以某种方式更改 group-id，服务器将不会注意到这种更改，并且在确定访问权限时也不会将其纳入考量。

通常，在提供对更多组的访问方面所做的改进能够带来真正的好处，而无法更改组所带来的损失不会被注意到，因为这种做法不太常用。不过，考虑使用 Kerberos 的站点管理员应该了解这种差异，并确保它不会真正造成问题。

7.6.2 性能和可伸缩性

利用 Kerberos 提高安全性需要使用额外的 CPU 资源来加密和解密消息。需要多少额外的 CPU 资源以及差异是否明显取决于所用的硬件和应用程序。如果服务器或客户端已用尽了可用的 CPU 资源，在从 `sec=sys` 切换到 Kerberos 时，可能会出现相当严重的性能下降。如果还有富余的 CPU 容量，则这种过渡很可能不会导致任何吞吐量变化。确定使用 Kerberos 所造成的影响大小的唯一方式是在硬件上测试您的负载。

可以减轻负载的配置选项同时也会降低提供的保护质量。`sec=krb5` 产生的负载应该比 `sec=krb5p` 要小得多，但如上文所述，它不能提供很高的安全性。类似地，您可以调整可供 Kerberos 从中选择的口令列表，而这可能会改变 CPU 的要求。但是，默认值是经过精心选择的，如未同样经过谨慎考虑，不应更改这些值。

将 NFS 配置为使用 Kerberos 时可能存在的另一个性能问题涉及到 Kerberos 身份验证服务器（称为 KDC 或密钥分发中心）的可用性。

使用 NFS 会增大此类服务器的负载，程度与对任何其他服务使用 Kerberos 时所增大的负载相同。每当给定的用户（Kerberos 主体）与服务建立会话时（例如，通过访问特定 NFS 服务器导出的文件），客户端就需要与 KDC 协商。协商会话密钥后，客户端与服务器在许多个小时内（此时段取决于 Kerberos 配置的细节，具体而言取决于 `ticket_lifetime` 设置）无需进一步的帮助即可通讯。

最有可能影响 Kerberos KDC 服务器供应的因素是可用性和峰值用量。

与其他核心服务（例如 DNS、LDAP）或类似的名称查找服务一样，使用两个距离每个客户端都比较“近”的服务器能够在资源有限时提供较佳的可用性。Kerberos 允许使用多个具有灵活模型的 KDC 服务器来进行数据库传播，因此，在校园、建筑物甚至机柜周围按需排布服务器的工作相当简单。确保每个客户端都查找附近的 Kerberos 服务器的最佳机制是对每个建筑物（或类似设施）使用水平分割 DNS 来从 DNS 服务器获取不同的细节。如果这种方法不可行，也可采用在不同的位置管理不同的 `/etc/krb5.conf` 文件这种替代做法。

由于对 Kerberos KDC 的访问并不频繁，只有在高峰时间，负载才可能会成为一个问题。如果数千人都在 9:00 到 9:05 登录，则服务器每分钟收到的请求数就会比在午夜收到的要多得多。Kerberos 服务器上的负载可能会超过 LDAP 服务器，但不会有数量级的差异。较为合理的准则是采用供应 LDAP 复本的相同方式来供应 Kerberos 复本，然后监视性能以确定需求是否超过容量。

7.6.3 主 KDC、多个域和信任关系

Kerberos KDC 的一个不容易分发的服务是更新处理，例如口令更改和新用户的创建。这些操作必须在单个主 KDC 上进行。

这些更新不太可能会以很高的频率发生，因而不会产生任何繁重负载，但可能出现可用性方面的问题。创建新用户或更改口令可能很麻烦，并且世界另一端的主 KDC 有时会暂时不可用。

如果组织分布于不同地理位置并且其政策规定在每个站点本地处理管理任务，创建多个 Kerberos 域（为每个管理中心创建一个）可能会是比较好的做法。这样每个域都会有位于本地的自己的主 KDC。通过在域之间设置信任关系，一个域中的用户仍可访问另一个域中的资源。

要安排多个域，最轻松的方式是使用一个全局域（例如 EXAMPLE.COM）和不同的本地域（例如 ASIA.EXAMPLE.COM、EUROPE.EXAMPLE.COM）。如果全局域配置为信任每个本地域，并且每个本地域配置为信任全局域，则任何一对域之间都将具有完全可传递的信任，并且任何主体都可以与任何服务建立安全连接。如何确保对资源（例如该服务提供的文件）的适当访问权限将取决于所用的用户名查找服务，以及 NFS 文件服务器的功能，这不在本文档的范畴内。

7.7 更多信息

MIT Kerberos 的官方网站是 <https://web.mit.edu/kerberos>。因此，找出任何有关 Kerberos（包括 Kerberos 安装、用户和管理指南）的任何其他相关资源的链接。

Brian Tung 编著的 **Kerberos — 网络认证系统** 一书 (ISBN 0-201-37924-4) 提供了深入和全面的信息。

8 Active Directory 支持

Active Directory* (AD) 是一项基于 LDAP、Kerberos 和其他服务的目录服务。Microsoft* Windows* 使用它来管理资源、服务和人员。在 Microsoft Windows 网络中，Active Directory 会提供有关这些对象的信息，限制对其的访问，并强制执行策略。SUSE® Linux Enterprise Server 可让您加入现有的 Active Directory 域，并将您的 Linux 计算机集成到 Windows 环境中。

8.1 集成 Linux 和 Active Directory 环境

使用已加入现有 Active Directory 域的 Linux 客户端（配置为 Active Directory 客户端），可以受益于纯粹的 SUSE Linux Enterprise Server Linux 客户端所不能提供的各种功能：

使用 SMB 浏览共享文件和目录

GNOME Files（以前称为 Nautilus）支持通过 SMB 浏览共享资源。

使用 SMB 共享文件和目录

GNOME Files 支持如同在 Windows 中那样共享目录和文件。

访问并操作 Windows 服务器上的用户数据

通过 GNOME Files，用户可以访问其 Windows 用户数据，并可以在 Windows 服务器上编辑、创建和删除文件与目录。用户无需多次输入其口令便能访问其数据。

脱机鉴定

即使用户脱机或者 Active Directory 服务器出于其他原因而无法使用，用户也仍可在 Linux 计算机上登录并访问其本地数据。

Windows 口令更改

Linux 中的此 Active Directory 支持端口强制执行储存在 Active Directory 中的公司口令策略。显示管理器和控制台支持口令更改消息并接受您的输入。甚至可以使用 Linux `passwd` 命令设置 Windows 口令。

通过 Kerberos 化应用程序单点登录

许多桌面应用程序都支持 Kerberos（**Kerberos 化**），这意味着它们可以透明地为用户处理身份验证，而无需在 Web 服务器、代理、群件应用程序或其他位置重新输入口令。



注意：通过 Windows Server* 2016 和更高版本管理 Unix 属性

在 Windows Server 2016 和更高版本中，Microsoft 去除了 **IDMU/NIS 服务器** 角色，并一并去除了 **Active Directory 用户和计算机** MMC 管理单元的 **Unix 属性** 插件。

但是，如果在 **Active Directory 用户和计算机** MMC 管理单元中启用了高级选项，则仍可以手动管理 Unix 属性。有关详细信息，请参见<https://blogs.technet.microsoft.com/activedirectoryua/2016/02/09/identity-management-for-unix-idmu-is-deprecated-in-windows-server/>。

或者，可以使用[过程 8.1 “使用用户登录管理加入 Active Directory 域”](#)中所述的方法在客户端完成属性设置（具体而言，请参见[步骤 6.c](#)）。

下一节包含前面所述大多数功能的技术背景。

8.2 有关 Linux Active Directory 支持的背景信息

许多系统组件需要无故障交互，以便将 Linux 客户端集成到现有的 Windows Active Directory 域。以下几节重点讲述 Active Directory 服务器和客户端交互中关键事件的底层进程。

为了与目录服务进行通信，客户端至少需要与服务器共享两个协议。

LDAP

LDAP 是一种为管理目录信息而优化的协议。具有 Active Directory 的 Windows 域控制器可以使用 LDAP 协议来与客户端交换目录信息。有关 LDAP 的更多一般信息，请参见[第 6 章 “389 LDAP Directory Server”](#)。

Kerberos

Kerberos 是可信的第三方身份验证服务。其所有客户端均信任 Kerberos 对另一个客户端的身份授权，从而支持 Kerberos 化单点登录 (SSO) 解决方案。Windows 支持 Kerberos 实施，因此即使是 Linux 客户端也可以使用 Kerberos SSO。有关 Linux 中 Kerberos 的详细信息，请参阅[第 7 章 “使用 Kerberos 进行网络身份验证”](#)。

根据您要使用哪个 YaST 模块设置 Kerberos 身份验证，将由不同的客户端组件处理帐户和身份验证数据：

基于 SSSD 的解决方案

- sssd 守护程序是此解决方案的核心部分。它处理与 Active Directory 服务器之间的所有通讯。
- 要收集名称服务信息，可使用 sssd_nss。
- 要对用户进行身份验证，可使用 PAM 的 pam_sss 模块。Linux 客户端上 Active Directory 用户的用户主目录创建由 pam_mkhomedir 处理。
有关 PAM 的详细信息，请参见第 3 章 “通过 PAM 进行身份验证”。

基于 Winbind (Samba) 的解决方案

- winbindd 守护程序是此解决方案的核心部分。它处理与 Active Directory 服务器之间的所有通讯。
- 要收集名称服务信息，可使用 nss_winbind。
- 要对用户进行身份验证，可使用 PAM 的 pam_winbind 模块。Linux 客户端上 Active Directory 用户的用户主目录创建由 pam_mkhomedir 处理。
有关 PAM 的详细信息，请参见第 3 章 “通过 PAM 进行身份验证”。

图 8.1 “基于 Winbind 的 Active Directory 身份验证的纲要” 突出显示了基于 Winbind 的 Active Directory 身份验证的最重要组件。

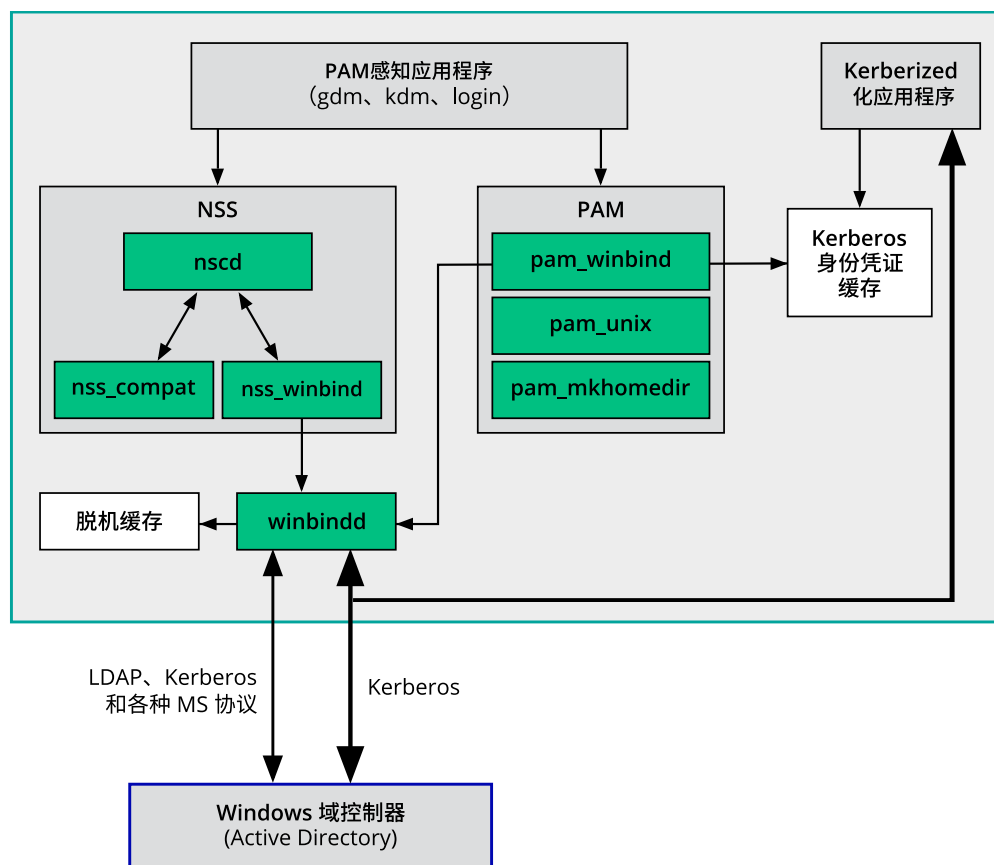


图 8.1：基于 WINBIND 的 ACTIVE DIRECTORY 身份验证的概要

可感知 PAM 的应用程序（如登录例程和 GNOME 显示管理器）会与 PAM 及 NSS 层交互，以便对 Windows 服务器进行身份验证。支持 Kerberos 身份验证的应用程序（如文件管理器、网页浏览器或电子邮件客户端）使用 Kerberos 身份凭证缓存来访问用户的 Kerberos 票据，因此是 SSO 框架的组成部分。

8.2.1 域加入

在域加入过程中，服务器和客户端确立安全关系。在客户端上，需要执行以下任务来加入 Windows 域控制器提供的现有 LDAP 和 Kerberos SSO 环境。整个加入过程由 YaST 域成员资格模块来处理，该模块可以在安装过程中运行或在已安装系统中运行：

1. 找到了提供 LDAP 和 KDC（密钥发布中心）服务的 Windows 域控制器。
2. 加入客户端的计算机帐户是在目录服务中创建的。

3. 客户端的初始票据授予票据 (TGT) 已经获得并储存于其本地 Kerberos 身份凭证缓存。客户端需要此 TGT 来获得进一步的票据，使其可以联系其他服务，如联系目录服务器进行 LDAP 查询。
4. NSS 和 PAM 配置要进行调整，使客户端能对域控制器进行身份验证。

客户端引导过程中，将启动 winbind 守护程序并检索计算机帐户的初始 Kerberos 票据。winbindd 自动刷新计算机票据以保持其有效。为了跟踪当前的帐户策略，winbindd 定期查询域控制器。

8.2.2 域登录和用户主目录

GNOME 的登录管理器 (GDM) 已经过扩展，允许处理 Active Directory 域登录。用户可以选择登录其计算机已加入的主域或主域的域控制器已经与之确立信任关系的可信域之一。

如第 8.2 节 “有关 Linux Active Directory 支持的背景信息” 中所述，用户身份验证由多个 PAM 模块调解。如果出现错误，错误代码将转换为用户易于理解的错误消息，这些消息是 PAM 通过任意支持的方法（GDM、控制台和 SSH）在登录时提供的：

口令已失效

用户看到一条消息，说明口令已经失效，需要更改。系统会提示输入新口令，并在新口令不符合公司口令策略（例如口令太短、太简单或已用过）时告知用户。如果用户的口令更改失败，会显示原因，提示输入新口令。

帐户被禁用

用户会看到一条错误消息，告知其帐户已禁用，需与系统管理员联系。

帐户已锁定

用户会看到一条错误消息，告知其帐户已锁定，需与系统管理员联系。

口令必须更改

用户可以登录，但会收到警告说口令很快就必须更改了。该警告会在口令失效前三天发出。失效后，用户便无法登录。

工作站无效

如果仅允许用户登录特定的工作站，而当前 SUSE Linux Enterprise Server 计算机并不在此列，则会出现一条消息，告知此用户无法从此工作站登录。

登录时段无效

如果仅允许用户在工作时间登录，当该用户尝试在非工作时间登录时，会出现一条消息，告知用户在此时间无法登录。

帐户已失效

管理员可为特定用户帐户设置失效时间。如果该用户尝试在失效后登录，将会看到一条消息，告知其帐户已失效，不能用于登录。

在成功的身份验证期间，客户端从 Active Directory 的 Kerberos 服务器中获得票据授权票据 (TGT) 并将其储存在用户的身份凭证缓存中。它还可以在后台续订 TGT，而无需用户的交互。

SUSE Linux Enterprise Server 对 Active Directory 用户提供本地主目录支持。如果按[第 8.3 节“为 Active Directory 配置 Linux 客户端”](#)中所述通过 YaST 进行了配置，当 Windows/Active Directory 用户首次登录到 Linux 客户端时，系统会创建用户主目录。这些主目录的外观与标准的 Linux 用户主目录相同，可独立于 Active Directory 域控制器工作。

使用本地用户主目录可以访问此计算机上的用户数据（即使 Active Directory 服务器断开连接），前提是 Linux 客户端已配置为执行脱机身份验证。

8.2.3 办公服务和策略支持

公司环境中的用户必须能够成为漫游用户（例如，切换网络，甚至在断开连接的情况下工作一段时间）。为使用户能够登录断开连接的计算机，已经将大量的缓存集成到 winbind 守护程序。winbind 守护程序即使在脱机状态下都可强制实施口令策略。它跟踪失败的登录尝试次数并根据 Active Directory 中配置的策略做出反应。脱机支持默认处于禁用状态，必须在 YaST 域成员资格模块中显式启用。

当域控制器变成不可用状态时，用户仍可使用断开连接之前获得的有效 Kerberos 票据访问网络资源（不包括 Active Directory 服务器本身），这与在 Windows 中一样。域控制器联机时才能处理口令更改。与 Active Directory 服务器断开连接时，用户无法访问储存在此服务器上的任何数据。当工作站与网络完全断开连接并于稍后再次连接到公司网络时，SUSE Linux Enterprise Server 会在用户锁定再解锁桌面（例如使用桌面屏幕保护程序）时获得新的 Kerberos 票据。

8.3 为 Active Directory 配置 Linux 客户端

在客户端加入 Active Directory 域之前，需要对网络设置进行一些调整以确保客户端和服务器的正常交互。

DNS

将您的客户端计算机配置为使用可将 DNS 请求转发到 Active Directory DNS 服务器的 DNS 服务器。或者，将您的计算机配置为使用 Active Directory DNS 服务器作为名称服务数据源。

NTP

要成功进行 Kerberos 身份验证，必须准确设置客户端的时间。为此，强烈建议使用中心 NTP 时间服务器（这也可以是 Active Directory 域控制器上运行的 NTP 服务器）。如果您的 Linux 主机和域控制器之间的时钟偏差超过特定限制，Kerberos 身份验证将会失败，客户端将使用较弱的 NTLM（NT LAN 管理器）身份验证登录。有关使用 Active Directory 进行时间同步的更多细节，请参见[过程 8.2 “使用 Windows 域成员资格加入 Active Directory 域”](#)。

防火墙

要浏览您的网上邻居，请完全禁用防火墙，或将用于浏览的接口标记为内部区域的一部分。

要更改客户端上的防火墙设置，请以 `root` 身份登录并启动 YaST 防火墙模块。选择接口。从接口列表选择网络接口并单击更改。选择内部区域并单击确定应用您的设置。单击下一步 > 完成退出防火墙设置。要禁用防火墙，请选中禁用防火墙自动启动选项，然后单击下一步 > 完成退出防火墙模块。

Active Directory 帐户

除非 Active Directory 管理员为您提供了对 Active Directory 域有效的用户帐户，否则您无法登录到该域。在您的 Linux 客户端上使用 Active Directory 用户名和口令登录到 Active Directory 域。

8.3.1 选择用于连接 Active Directory 的 YaST 模块

YaST 包含多个可连接 Active Directory 的模块：

8.3.2 使用用户登录管理加入 Active Directory

YaST 模块用户登录管理支持在 Active Directory 上进行身份验证。此外，它还支持以下相关身份验证和标识提供程序：

标识提供程序

- **委派到第三方软件库：** 通过代理提供传统 NSS 提供程序支持。
- **FreeIPA：** FreeIPA 和 Red Hat Enterprise 身份管理提供程序。
- **通用目录服务 (LDAP)：** 一个 LDAP 提供程序。有关配置 LDAP 的详细信息，请参见 [`man 5 sssd-ldap`](#)。
- **本地 SSSD 文件数据库：** 面向本地用户的 SSSD 内部提供程序。

身份验证提供程序

- **委派到第三方软件库：** 通过代理将身份验证中继到另一个 PAM 目标。
- **FreeIPA：** FreeIPA 和 Red Hat Enterprise 身份管理提供程序。
- **通用 Kerberos 服务：** 一个 LDAP 提供程序。
- **通用目录服务 (LDAP)：** Kerberos 身份验证。
- **本地 SSSD 文件数据库：** 面向本地用户的 SSSD 内部提供程序。
- **此域不提供身份验证服务：** 显式禁用身份验证。

要使用 SSSD 以及 YaST 的用户登录管理模块加入 Active Directory 域，请执行以下操作：

过程 8.1：使用用户登录管理加入 ACTIVE DIRECTORY 域

1. 打开 YaST。
2. 如果希望以后能够使用 DNS 自动发现，请将 Active Directory 域控制器（Active Directory 服务器）设置为客户端的名称服务器。
 - a. 在 YaST 中单击网络设置。
 - b. 选择主机名/DNS，然后在名称服务器 1 文本框中输入 Active Directory 域控制器的 IP 地址。
单击确定保存设置。

3. 在 YaST 主窗口中，启动用户登录管理模块。

该模块随即打开，其中的概述显示了您计算机的不同网络属性，以及当前使用的身份验证方法。

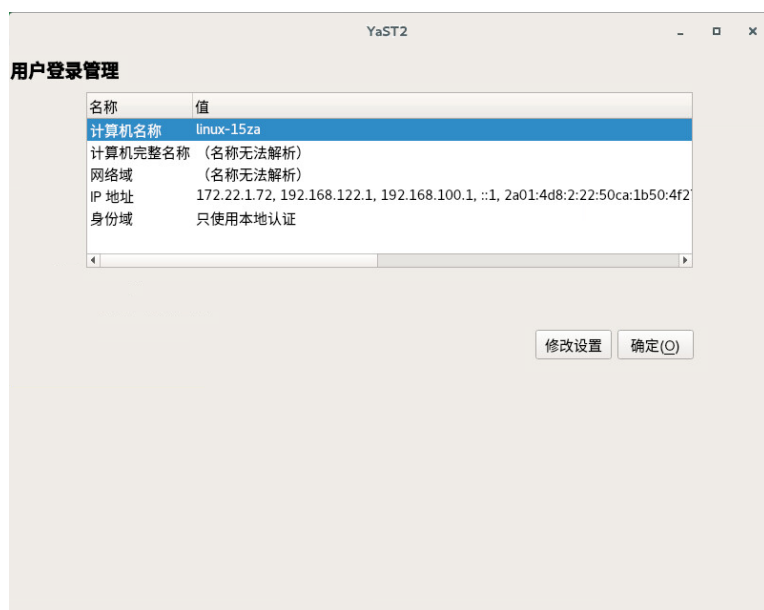


图 8.2：用户登录管理的主窗口

4. 要开始编辑，请单击更改设置。

5. 现在加入该域。

- a. 单击加入域。
- b. 在出现的对话框中，指定正确的域名。然后指定用于处理身份数据和身份验证的服务：为两者均选择 Microsoft Active Directory。
确保已选中启用该域。
单击确定。
- c. （可选）在下一个对话框中，通常可保留默认设置。不过，在以下情况下将需要做出更改：
 - **如果本地主机名与域控制器上设置的主机名不匹配：** 确定您计算机的主机名是否与 Active Directory 域控制器所获悉的您的计算机名称相匹配。在终端中运行 `hostname` 命令，然后将其输出与 Active Directory 域控制器的配置进行比较。

如果值不相同，请在 AD 主机名下指定 Active Directory 配置中的主机名。否则，请将相应的文本框留空。

- **如果您不想使用 DNS 自动发现：** 指定您要使用的 Active Directory 服务器主机名。如果有多个域控制器，请以逗号分隔其主机名。

d. 要继续操作，请单击确定。

如果尚未安装所有软件，计算机现在将安装缺少的软件。然后，它会检查配置的 Active Directory 域控制器是否可用。

e. 如果一切正常，下一个对话框现在应会显示它已发现一个 Active Directory 服务器，但您尚未注册。

在对话框中，指定 Active Directory 管理员帐户（通常为 Administrator）的用户名和口令。

为了确保为 Samba 启用当前域，请选中覆盖要与此 AD 搭配使用的 Samba 配置。要进行注册，请单击确定。



The image shows the 'Active Directory 注册' (Active Directory Registration) window in YaST2. It displays the current status of the system's Active Directory configuration. The status table shows that the Active Directory Server has been discovered via DNS, but the domain, workgroup, and enrollment status are not yet configured. Below the table, there are fields for entering the AD user credentials (Administrator) and a password. There are also checkboxes for updating DNS records and covering Samba configuration.

名称	值
Active Directory Server	(已通过 DNS 自动发现)
Active Directory Domain	
Workgroup	
Enrollment Status	尚未注册

输入 AD 用户身份凭证（如 Administrator）以注册或重新注册此计算机：

用户名
Administrator

口令
.....

☒ 亦更新活动目录的 DNS 记录

可选组织单位，例如“Headquarter/HR/BuildingA”
.....

☐ 覆盖要与此 AD 搭配使用的 Samba 配置

确定(O)

图 8.3：注册到域中

f. 现在，您应该会看到一条确认您已成功注册的消息。单击确定完成注册。

6. 注册后，使用管理域用户登录窗口配置客户端。

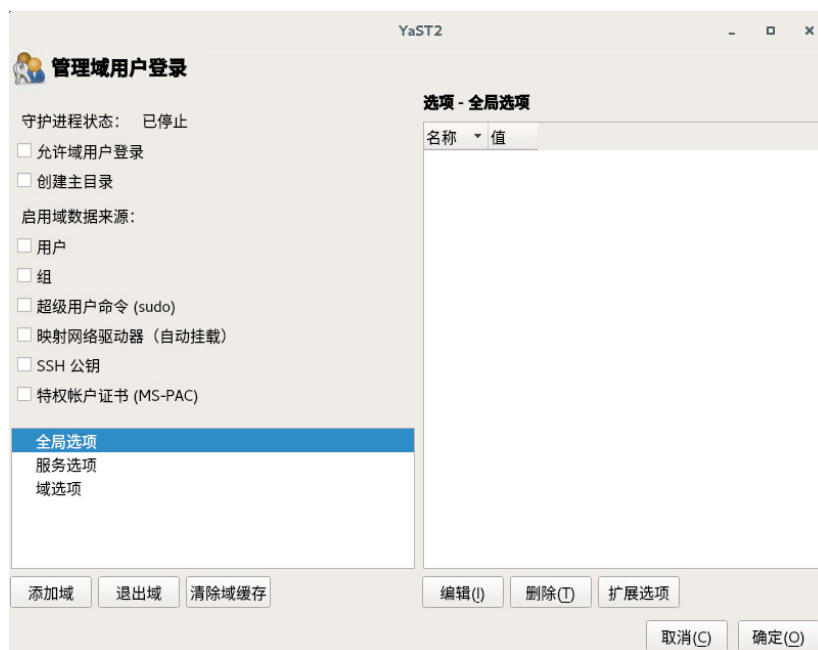


图 8.4：用户登录管理的配置窗口

- a. 要允许使用 Active Directory 提供的登录数据登录到计算机，请选中允许域用户登录。
- b. （可选）（可选）在启用域数据来源下，激活其他数据源，例如，有关允许哪些用户使用 `sudo` 或哪些网络驱动器可用的信息。
- c. 要允许为 Active Directory 用户创建主目录，请选中创建主目录。可通过多种方式设置主目录的路径 — 在客户端上、在服务器上，或将两种方式结合使用：
 - 要在域控制器上配置主目录路径，请为每个用户的 `UnixHomeDirectory` 属性设置相应的值。此外，请确保将此属性复制到全局目录。有关在 Windows 中存档该内容的信息，请参见 <https://support.microsoft.com/en-us/kb/248717>。
 - 要在客户端上配置主目录路径并指定域控制器上设置的路径具有优先权，请使用选项 `fallback_homedir`。
 - 要在客户端上配置主目录路径并指定客户端设置将覆盖服务器设置，请使用 `override_homedir`。

由于域控制器上的设置超出了本文档的范畴，下面仅介绍客户端选项的配置。在侧边栏中选择服务选项 > 名称切换，然后单击扩展选项。在该窗口中选择 `fallback_homedir` 或 `override_homedir`，然后单击添加。指定一个值。要让主目录采用 `/home/USER_NAME` 格式，请使用 `/home/%u`。有关可能的变量的详细信息，请参见 `sssd.conf` 手册页 (`man 5 sssd.conf`) 的“`override_homedir`”部分。单击确定。

7. 单击“确定”保存更改。确保现在显示的值正确无误。要退出对话框，请单击取消。

8.3.3 使用 Windows 域成员资格加入 Active Directory

要使用 `winbind` 以及 YaST 的 Windows 域成员资格模块加入 Active Directory 域，请执行以下操作：

过程 8.2：使用 WINDOWS 域成员资格加入 ACTIVE DIRECTORY 域

1. 作为 `root` 登录并启动 YaST。
2. 启动网络服务 > Windows 域成员。
3. 在 Windows 域成员资格屏幕中的域或工作组，输入域以加入（请参见图 8.5 “确定 Windows 域成员资格”）。如果您主机上的 DNS 设置与 Windows DNS 服务器正确集成，请以 DNS 格式 (`mydomain.mycompany.com`) 输入 Active Directory 域名。如果您输入简短域名（也称为 Windows 2000 之前的域名），YaST 必须依赖 NetBIOS 名称解析（而不是 DNS）来查找正确的域控制器。



图 8.5：确定 WINDOWS 域成员资格

4. 要将 SMB 源用于 Linux 身份验证，请选中同时使用 SMB 信息进行 Linux 身份验证。
5. 要自动为 Linux 计算机上的 Active Directory 用户创建本地主目录，请选中在登录时创建主目录。
6. 选中脱机身份验证，让域用户即使在 Active Directory 服务器暂时不可用或者无网络连接的情况下也能够登录。
7. 要更改 Samba 用户和组的 UID 与 GID 范围，请选择专家设置。仅在需要时让 DHCP 检索 WINS 服务器。当一些计算机仅通过 WINS 系统解析时，就需要这么做。
8. 选择 NTP 配置并输入相应的服务器名称或 IP 地址，来为 Active Directory 环境配置 NTP 时间同步。如果您已在独立的 YaST NTP 配置模块中输入了相应的设置，则不需要执行此步骤。
9. 单击确定并在提示时确认域连接。
10. 在 Active Directory 服务器上提供 Windows 管理员的口令并单击确定（请参见图 8.6 “提供管理员身份凭证”）。



请输入用户名和口令以加入域。

要匿名加入域，请将文本条目留空。

用户名(U)

Administrator

口令(P)

计算机帐户 OU(M)

(默认) 获取列表(B)

取消(C) 确定(O)

图 8.6：提供管理员身份凭证

加入 Active Directory 域之后，使用桌面上的显示管理器或控制台从工作站登录到该域。

！ 重要：域名

如果域名以 `.local` 结尾，可能无法成功加入域。以 `.local` 结尾的名称可能导致与多路广播 DNS (MDNS) 相冲突，在 MDNS 中，`.local` 是为链路本地主机名保留的。

🔑 注意：只有管理员能够注册计算机

只有域管理员帐户（例如 `Administrator`）能够将 SUSE Linux Enterprise Server 加入 Active Directory。

8.3.4 检查 Active Directory 连接状态

要检查您是否已成功在 Active Directory 域中注册，请使用以下命令：

- `klist` 显示当前用户是否具有有效的 Kerberos 票据。
- `getent passwd` 显示针对所有用户发布的 LDAP 数据。

8.4 登录到 Active Directory 域

如果您的计算机已配置为对 Active Directory 进行身份验证且您拥有有效的 Windows 用户 ID，您便可以使用 Active Directory 身份凭证登录到计算机。支持通过 GNOME、控制台、SSH 和任何其他可感知 PAM 的应用程序登录。

! 重要：脱机鉴定

SUSE Linux Enterprise Server 支持脱机身份验证，这样即使客户端计算机处于脱机状态，您也可以登录到其中。有关详细信息，请参见第 8.2.3 节“办公服务和策略支持”。

8.4.1 GDM

要对 Active Directory 服务器进行 GNOME 客户端计算机身份验证，请执行以下操作：

1. 单击未列出。
2. 在用户名文本框中，以 `DOMAIN_NAME\USER_NAME` 格式输入域名和 Windows 用户名。
3. 输入您的 Windows 口令。

如果已进行相应的配置，SUSE Linux Enterprise Server 会在已经过身份验证的每个用户通过 Active Directory 首次登录时，在本地计算机上创建一个用户主目录。这样，您便可以获享 SUSE Linux Enterprise Server 的 Active Directory 支持，同时确保您的 Linux 计算机完全正常运行且任您操控。

8.4.2 控制台登录

除了使用图形前端登录到 Active Directory 客户端计算机以外，您还可以使用基于文本的控制台登录，甚至是使用 SSH 远程登录。

要从控制台登录到 Active Directory 客户端，请在 `login:` 提示符处输入 `DOMAIN_NAME\USER_NAME`，并提供口令。

要使用 SSH 远程登录到 Active Directory 客户端计算机，请执行以下操作：

1. 在登录提示符处，输入：

```
tux > ssh DOMAIN_NAME\\USER_NAME@HOST_NAME
```

\ 域和登录分界符用另一个 \ 号转义了。

2. 提供用户密码。

8.5 更改口令

SUSE Linux Enterprise Server 可帮助用户选择一个符合公司安全策略的适当的新口令。底层 PAM 模块会从域控制器检索当前的口令策略设置，并在登录时以消息的形式告知用户其帐户通常需要满足的具体口令质量要求。与 Windows 操作系统一样，SUSE Linux Enterprise Server 也会显示一条描述以下信息的信息：

- 口令历史设置
- 口令最短长度要求
- 口令最短时限
- 口令复杂度

只有成功满足了所有要求后，口令更改过程才会成功。口令状态的反馈会同时通过显示管理器和控制台提供。

GDM 提供有关口令失效的反馈，并以交互模式提示输入新口令。要通过显示管理器更改口令，请按提示提供口令信息。

要更改 Windows 口令，可以使用标准 Linux 实用程序 **passwd** 而无需在服务器上操作该数据。要更改 Windows 口令，请执行以下操作：

1. 登录控制台。
2. 输入 **passwd**。
3. 出现提示时输入当前口令。
4. 输入新口令。
5. 重输入新的口令进行确认。如果新口令不符合 Windows 服务器上的策略，此信息将反馈给您并提示您输入另一个口令。

要从 GNOME 桌面更改 Windows 口令，请按以下步骤操作：

1. 单击面板左边缘的计算机图标。
2. 选择控制中心。
3. 在个人部分选择关于我 › 更改口令。
4. 输入旧口令。
5. 输入并确认新口令。
6. 保留对话框中的关闭，应用设置。

9 设置 freeRADIUS 服务器

RADIUS（远程身份验证拨入用户服务）协议一直以来都是用于管理网络访问的标准服务。它为特大型企业（例如因特网服务提供商和手机网络提供商）执行身份验证、授权和统计 (AAA) 协议，在小型网络中应用也很广泛。它对用户和设备进行身份验证，授权这些用户和设备使用特定的网络服务，并跟踪服务的使用以进行计费 and 审计。您不需要使用全部三个 AAA 协议，只需使用所需的协议。例如，您可能不需要统计功能，而只需要客户端身份验证功能，或者，您可能只需要统计功能，客户端授权交由其他某个系统来管理。

此协议极其高效，使用普通配置的硬件就能管理数千个请求。当然，虽然它称为“拨入”协议，但适合用于所有网络协议，而不仅仅是拨号网络。

RADIUS 在分布式体系结构中运行，与网络访问服务器 (NAS) 相隔离。用户访问数据储存在可供多个 NAS 使用的中心 RADIUS 服务器上。NAS 提供对受管以太网交换机或无线接入点等网络的物理访问。

FreeRADIUS 是 RADIUS 的开源实现，并且是使用最广泛的 RADIUS 服务器。在本章中，您将了解如何安装和测试 FreeRADIUS 服务器。由于可能的用例众多，在初始设置可以正常工作后，接下来请查看内容详尽的官方文档（参见 <https://freeradius.org/documentation/>）。

9.1 在 SUSE Linux Enterprise 上安装和测试

以下步骤将设置一个简单的测试系统。在确认服务器可以正常运行并且您已准备好创建生产配置后，需要执行几个撤消步骤再开始生产配置。

首先请安装 `freeradius-server` 和 `freeradius-server-utils` 软件包。然后进入 `/etc/raddb/certs` 并运行 `bootstrap` 脚本，以创建一组测试证书：

```
root # zypper in freeradius-server
root # cd /etc/raddb/certs
root # ./bootstrap
```

`certs` 目录中的 README 文件包含了大量有用信息。`bootstrap` 脚本完成后，以调试模式启动服务器：

```
root # radiusd -X
[...]
```

```
Listening on auth address * port 1812 bound to server default
Listening on acct address * port 1813 bound to server default
Listening on auth address :: port 1812 bound to server default
Listening on acct address :: port 1813 bound to server default
Listening on auth address 127.0.0.1 port 18120 bound to server inner-tunnel
Listening on proxy address * port 54435
Listening on proxy address :: port 58415
Ready to process requests
```

如果看到“正在监听”和“准备处理请求”行，表示服务器已正常启动。如果服务器未启动，请仔细阅读输出，因为其中告知了问题出在哪里。您可以使用 **tee** 将输出复制到文本文件：

```
tux > radiusd -X | tee radiusd.text
```

下一步是用某个测试客户端和用户来测试身份验证。该客户端是 RADIUS 服务器的客户端，例如无线接入点或交换机。客户端在 `/etc/raddb/client.conf` 中配置。人类用户在 `/etc/raddb/mods-config/files/authorize` 中配置。

打开 `/etc/raddb/mods-config/files/authorize` 并取消注释以下几行：

```
bob    Cleartext-Password := "hello"
Reply-Message := "Hello, %{User-Name}"
```

`/etc/raddb/client.conf` 中提供了测试客户端 `client localhost`，其机密的为 `testing123`。打开另一个终端，并以非特权用户 `bob` 的身份使用 **radtest** 命令登录：

```
tux > radtest bob hello 127.0.0.1 0 testing123
Sent Access-Request Id 241 from 0.0.0.0:35234 to 127.0.0.1:1812 length 73
    User-Name = "bob"
    User-Password = "hello"
    NAS-IP-Address = 127.0.0.1
    NAS-Port = 0
    Message-Authenticator = 0x00
    Cleartext-Password = "hello"
Received Access-Accept Id 241 from 127.0.0.1:1812 to 0.0.0.0:0 length 20
```

成功登录后，`radius -X` 终端中会显示如下所示的信息：

```
(3) pap: Login attempt with password
```

```
(3) pap: Comparing with "known good" Cleartext-Password
(3) pap: User authenticated successfully
(3)      [pap] = ok
[...]
(3) Sent Access-Accept Id 241 from 127.0.0.1:1812 to 127.0.0.1:35234 length 0
(3) Finished request
Waking up in 4.9 seconds.
(3) Cleaning up request packet ID 241 with timestamp +889
```

现在，通过网络中的另一台计算机再次运行登录测试。通过取消注释并修改 `clients.conf` 中的以下项，在服务器上创建一个客户端配置：

```
client private-network-1 {
    ipaddr      = 192.0.2.0/24
    secret      = testing123-1
}
```

输入测试客户端计算机的 IP 地址。在客户端计算机上安装 `freeradius-server-utils`，它可以提供一些有用的测试命令。尝试使用 `radtest` 命令以 bob 的身份从客户端登录。最好使用 RADIUS 服务器的 IP 地址而非主机名，因为 IP 地址的访问速度更快：

```
tux > radtest bob hello 192.168.2.100 0 testing123-1
```

如果测试登录失败，请查看所有输出以了解问题出在哪里。其中提供了多个测试用户和测试客户端。配置文件中包含大量有用信息，我们建议研究这些文件。对测试结果感到满意并准备好创建生产配置时，请去除 `/etc/raddb/certs` 中的所有测试证书并将其替换为您自己的证书，注释掉所有测试用户和客户端，然后按 `Ctrl-C` 停止 `radiusd`。可以使用 `systemctl` 管理 `radiusd.service`，就像管理任何其他服务一样。

要了解如何在网络中安装 FreeRADIUS 服务器，请参见 <https://freeradius.org/documentation/> 和 <https://networkradius.com/freeradius-documentation/>，其中提供了深入的参考信息和操作指南。

II 本地安全性

- 10 物理安全性 102
- 11 使用 seccheck 进行自动安全检查 108
- 12 软件管理 112
- 13 文件管理 117
- 14 加密分区和文件 122
- 15 使用 cryptctl 对托管应用程序的储存区加密 126
- 16 用户管理 134
- 17 Spectre/Meltdown Checker 150
- 18 使用 YaST 配置安全设置 153
- 19 Linux 中的访问控制列表 157
- 20 证书存储区 168
- 21 使用 AIDE 进行入侵检测 170

10 物理安全性

物理安全应予以最大限度的关注。Linux 生产服务器应安放于加锁的数据中心之内，只有通过安全检查的人员才能访问。您也可以考虑使用引导加载程序口令，具体视环境和情况而定。

此外，还要考虑如下问题：

- 哪些人拥有主机的直接物理访问权限？
- 他们是否应拥有这些权限？
- 是否可以保护主机不被篡改，以及是否应进行此保护？

特定系统上所需的物理安全措施数量视情况而定，并且根据可用资金，安全措施也可能会有很大差别。

10.1 系统锁

数据中心内的大部分服务器机架都包含锁定功能。通常是位于机架正面的搭扣锁/圆筒锁，可让您转动插入锁定（或未锁定）位置的钥匙，以允许（或拒绝）进入。笼锁可防止有人篡改或窃取服务器的设备/介质，或者开箱直接操作/破坏硬件。防止系统重引导或从替代设备（例如 CD、DVD、闪存盘等）引导也很重要。

一些服务器还配有箱锁。根据系统供应商的设计和构造，这些锁可以发挥不同的作用。许多系统都设计为当尝试打开未开锁的系统时进行自我禁用。其他配有设备保护盖的系统将不允许插入或拔下键盘或鼠标。虽然有时锁是一项很实用的功能，但它们通常质量较差，很容易会被怀有不良意图的攻击者破坏。

10.2 锁定 BIOS



提示：安全引导

本节仅介绍确保引导进程安全的基本方法。要了解使用 UEFI 和安全引导功能的更高级引导保护的相关信息，请参见《管理指南》，第 13 章“UEFI（统一可扩展固件接口）”，第 13.1 节“安全引导”。

BIOS（基本输入/输出系统）或其继承者 UEFI（统一可扩展固件接口）是 PC 类系统上最低级别的软件/固件。运行 Linux 的其他硬件类型（POWER、IBM Z）也配有执行与 PC BIOS 类似功能的低级别固件。当本文档提及 BIOS 时，通常指的是 BIOS 和/或 UEFI。BIOS 指示系统配置，使系统处于一个定义良好的状态，并提供访问低级别硬件的例程。BIOS 执行已配置的 Linux 引导加载程序（例如 GRUB 2）来引导主机。

大部分 BIOS 实施都可配置为阻止未经授权的用户操作系统及引导设置。通常通过设置 BIOS 管理员或引导口令来完成。只有更改系统配置时才需要输入管理员口令，但在每次正常引导时都需要引导口令。对于大多数用例，设置管理员口令并将引导限制为内置硬盘便已足够。这样，攻击者便无法仅仅引导 Linux live CD 或闪存盘等设备。虽然这并不会提供高级别的安全（BIOS 可以被重设置、去除或修改 — 假设用例访问权限），但它可以作为另一种保护措施。

许多 BIOS 固件实现都提供其他各种安全相关设置。您可以咨询系统供应商，查阅系统文档或在系统引导时检查 BIOS，来了解更多信息。

！ 重要：在设置了 BIOS 引导口令的情况下引导

如果为系统设置了引导口令，主机将不会在无人照管的情况下引导（例如当系统重引导或发生电源故障时）。这是一种权衡。

！ 重要：丢失 BIOS 管理员口令

首次设置系统时，通常不会要求提供 BIOS 管理员口令。请勿忘记该口令，否则您将需要通过硬件操作清除 BIOS 内存来再次获得访问权限。

10.3 通过引导加载程序提供的安全性

SUSE Linux Enterprise Server 中默认使用的 Linux 引导加载程序 GRUB 2 可设置引导口令。它还提供了口令功能，以便只有管理员才能启动交互操作（例如编辑菜单项和进入命令行界面）。如果指定了口令，在您按 **C** 键和 **E** 键并输入正确的口令之前，GRUB 2 将不允许任何交互控制。

有关示例，请参见 GRUB 2 手册页。

设置这些口令时请务必记住它们！此外，启用这些口令可能只会减缓入侵，而不一定能阻止入侵。同样，有些人可能会从移动设备引导，并装入您的根分区。如果您使用的是 BIOS 级别安全性和引导加载程序，较好的做法是禁用从您计算机 BIOS 中的可移动设备进行引导的功能，然后通过口令来保护 BIOS 本身。

另请注意，需要将引导加载程序配置文件的模式更改为 `600`（仅限 `root` 读取/写入）以对其进行保护，否则其他人将能够读取您的口令或哈希！

10.4 淘汰包含敏感数据的 Linux 服务器

安全策略通常包含即将被淘汰或被处置的储存媒体的一些处理过程。常常采用磁盘和媒体擦除过程，因为这会彻底销毁媒体。您可以在因特网上找到多个免费工具。搜索“`dod` 磁盘擦除实用程序”将返回多个搜索结果。要淘汰包含敏感数据的服务器，请务必确保无法从硬盘恢复数据。要确保已去除所有数据痕迹，可以使用 `scrub` 等擦除实用程序。许多擦除实用程序都会多次重写数据。这样可确保即使使用复杂的方法，也无法取回已擦除数据的任何部分。一些工具甚至可通过可引导移动设备进行操作，并根据美国国防部 (DoD) 标准去除数据。请注意，许多政府机构都会指定自己的数据安全标准。一些标准可能强于其他标准，但可能需要更多的实施时间。

！ 重要：擦除耗损均衡设备

一些设备（例如 SSD）使用耗损均衡功能，不一定会在同一物理位置写入新数据。此类设备通常会提供自己的删除功能。

10.4.1 `scrub`：磁盘重写实用程序

`scrub` 利用重复模式来重写硬盘、文件和其他设备，旨在让从这些设备恢复数据变得更困难。它有三种基本操作模式：针对字符或块设备、针对文件，或者针对指定目录。有关详细信息，请参见 `man 1 scrub` 手册页。

支持的擦除方法

nnsa

用于清理可移动和不可移动硬盘的 4 轮 NNSA Policy Letter NAP-14.1-C (XVI-8)，需要使用伪随机模式重写所有位置两次，然后使用一种已知模式：random(x2)、0x00、verify。

dod

用于清理可移动和不可移动 rigid 磁盘的 4 轮 DoD 5220.22-M 第 8-306 节程序 (d)，需要使用一个字符、其补充、一个随机字符重写所有可寻址位置，然后进行校验。注意：scrub 首先执行一轮随机，以使校验更容易：random、0x00、0xff、verify。

bsi

德国信息技术安全中心 (<http://www.bsi.bund.de>) 建议的 9 轮方法：
0xff、0xfe、0xfd、0xfb、0xf7、0xef、0xdf、0xbf、0x7f。

gutmann

下面引述的 Gutmann 文献中对规范化 35 轮序列进行了说明。

schneier

Bruce Schneier 在 “Applied Cryptography”（应用密码学，1996）中介绍的 7 轮方法：0x00、0xff、random(x5)

pfitzner7

Roy Pfitzner 的 7 轮随机方法：random(x7)。

pfitzner33

Roy Pfitzner 的 33 轮随机方法：random(x33)。

usarmy

US Army AR380-19 方法：0x00、0xff、random。（注意：与 DoD 522.22-M 第 8-306 节的过程 (e) 相同，用于清理磁芯内存）。

fillzero

1 轮模式：0x00。

fillff

1 轮模式：0xff。

random

1 轮模式：random(x1)。

random2

2 轮模式：random(x2)。

old

6 轮预发行版 1.7 擦除方法：0x00、0xff、0xaa、0x00、0x55、verify。

fastold

5 轮模式：0x00、0xff、0xaa、0x55、verify。

custom=string

1 轮自定义模式。字符串可能包含 C 样式数字转义符：\nnn（八进制）或 \xnn（十六进制）。

10.5 限制对可移动媒体的访问

在某些环境中，需要对可移动媒体的访问，例如 USB 储存设备或光学设备。[udisks2](#) 软件包随附的工具可帮助进行此类配置。

1. 创建允许用户装入和弹出移动设备的用户组，例如 [mmedia_all](#)：

```
tux > sudo groupadd mmedia_all
```

2. 向新组添加特定用户 [tux](#)：

```
tux > sudo usermod -a -G mmedia_all tux
```

3. 创建包含以下内容的 [/etc/polkit-1/rules.d/10-mount.rules](#) 文件：

```
tux > cat /etc/polkit-1/rules.d/10-mount.rules
polkit.addRule(function(action, subject) {
  if (action.id == "org.freedesktop.udisks2.eject-media"
    && subject.isInGroup("mmedia_all")) {
    return polkit.Result.YES;
  }
});

polkit.addRule(function(action, subject) {
```

```
if (action.id == "org.freedesktop.udisks2.filesystem-mount"
    && subject.isInGroup("mmedia_all")) {
    return polkit.Result.YES;
}
});
```

❗ 重要：规则文件命名

规则文件的名称必须以数字开头，否则将忽略该名称。

规则文件按字母顺序进行处理。函数按其添加的顺序进行调用，直到其中一个函数返回值为止。因此，要添加在其他规则之前处理的授权规则，请将其放入 `/etc/polkit-1/rules.d` 中名称排序在其他规则文件之前的某个文件，例如 `/etc/polkit-1/rules.d/10-mount.rules`。每个函数应从 `polkit.Result` 返回值。

4. 重新启动 `udisks2`:

```
root # systemctl restart udisks2
```

5. 重新启动 `polkit`

```
root # systemctl restart polkit
```

11 使用 seccheck 进行自动安全检查

seccheck SUSE 安全检查器是一组外壳脚本，旨在定期自动检查系统的本地安全，并通过电子邮件向 root 用户或管理员配置的任何用户发送报告。

如果 **seccheck** 未安装（在您的系统上），请使用 **sudo zypper in seccheck** 进行安装。这些脚本由 **systemd** 计时器进行控制，该计时器默认未启用，必须由管理员进行启用。

11.1 Seccheck 计时器

有以下四种 **seccheck** 计时器：

- `/usr/lib/systemd/system/seccheck-daily.timer`
- `/usr/lib/systemd/system/seccheck-monthly.timer`
- `/usr/lib/systemd/system/seccheck-weekly.timer`
- `/usr/lib/systemd/system/seccheck-autologout.timer`

`seccheck-daily.timer`、`seccheck-monthly.timer` 和 `seccheck-weekly.timer` 运行多项检查，如第 11.3 节“每日、每周和每月检查”中所述。`seccheck-autologout.timer` 可注销非活动用户，请参见第 11.4 节“自动注销”。

您可以将 **seccheck** 邮件的收件人由 root 更改为 `/etc/sysconfig/seccheck` 中的任何用户。以下示例将其更改为名为 `firewall` 的管理员用户：

```
SECCHK_USER="firewall"
```

11.2 启用 seccheck 计时器

使用 **systemctl** 管理您的计时器，如同任何其他 **systemd** 计时器一样。以下示例启用并启动 `seccheck-daily.timer`：

```
tux > sudo systemctl enable --now seccheck-daily.timer
```

列出所有活动计时器：

```
tux > sudo systemctl list-timers
```

列出所有已启用的活动和非活动计时器：

```
tux > sudo systemctl list-timers --all
```

11.3 每日、每周和每月检查

seccheck 执行下列每日检查：

<u>/etc/passwd</u> 检查	字段的长度/数量/内容，具有相同 UID 的帐户，UID/GID 为 0 或 1 的帐户（root 和 bin 除外）
<u>/etc/shadow</u> 检查	字段的长度/数量/内容，没有口令的帐户
<u>/etc/group</u> 检查	字段的长度/数量/内容
用户 root 检查	安全 umask 和 <u>PATH</u>
<u>/etc/ftpusers</u>	检查是否将重要的系统用户置于此处
<u>/etc/aliases</u>	检查执行程序邮件别名
<u>.rhosts</u> 检查	检查用户的 <u>.rhosts</u> 文件是否包含 + 符号
用户主目录	检查主目录是否可写入或被其他用户拥有
点文件检查	检查主目录中的众多点文件是否具有正确的所有权和权限
邮箱检查	检查用户邮箱是否由正确的用户拥有，以及是否可读
NFS 导出检查	不应进行全局导出

NFS 导入检查	NFS 装入应设置 <code>nosuid</code> 选项
混杂检查	检查网卡是否处于混杂模式
列表模块	列出装载的模块
列表套接字	列出打开的端口

下表列出了每周检查：

口令检查	运行 john 以破解口令文件；用户将收到更改其口令的电子邮件通知
RPM md5 检查	通过 RPM 的 MD5 校验和功能检查已更改文件
suid/sgid 检查	列出所有 suid 和 sgid 文件
可执行文件组写入	列出组可写/全局可写的所有可执行文件
可写入检查	列出全局可写的所有文件（包括可执行文件）
设备检查	列出所有设备

重要：使用 **john** 审计口令

要启用口令审计，必须先安装软件包 `john`，即 John the Ripper 快速口令破解程序。该软件包可从 <https://build.opensuse.org/package/show/security/john> 上的 openSUSE Build Service 获取。

每月检查会列显一份完整的报告，每日检查和每周检查的列显内容则不相同。

11.4 自动注销

`seccheck-autologout.timer` 计时器每隔 10 分钟运行一次，检查远程和本地终端会话是否处于非活动状态，并在超过空闲时间后终止会话。

在 /etc/security/autologout.conf 文件中配置您需要的超时。参数包括默认的空闲和注销延迟时间，以及限制特定于用户、组、TTY 设备和 SSH 会话的最大空闲时间的配置。/etc/security/autologout.conf 包含多个配置示例。

12 软件管理

12.1 去除不需要的软件包 (RPM)

保护 Linux 系统的一个非常重要的步骤就是确定 Linux 服务器的主要功能或角色。否则，就很难了解需要保护哪些方面，并且对这些 Linux 系统的保护可能是无效的。因此，请务必查看默认的软件包列表，并去除任何与您定义的安全策略不符的不需要的软件包。

一般而言，RPM 软件包包含以下各项：

- 在安装时写入到 RPM 数据库的软件包元数据。
- 软件包的文件和目录。
- 安装和去除之前和之后执行的脚本。

除非软件包包含以下项目，否则一般不会给系统带来任何安全风险：

1. 任何已安装文件上的 setuid 或 setgid 位
2. 组可写或全局可写的文件或目录
3. 在安装时激活或默认激活的服务

假设上述三个条件均不适用，软件包只是文件集合。安装或卸载此类软件包对系统的安全价值不会产生影响。

但无论如何，将您系统中的已安装软件包限制为最少数量都会很有用。当发布安全警报和补丁时，限制最少数量将使得需要更新的软件包更少，并将简化维护工作。最好的做法是不要在生产服务器上安装开发软件包或桌面软件包（例如 X 服务器）。如果您不需要这些软件包，您也不应该安装它们，例如 Apache Web 服务器或 Samba 文件共享服务器。

重要：第三方安装程序的要求

Oracle 和 IBM 等许多第三方供应商需要桌面环境和开发库来运行安装程序。为防止它对生产服务器的安全产生影响，许多组织都会通过在开发实验室中创建静默安装（响应文件）来解决此问题。

此外，除非有正当的理由，否则不应安装 FTP 和 Telnet 守护程序等其他软件包。应改用 **ssh**、**scp** 或 **sftp** 替代方案。

最先执行的操作中的一项操作应该是创建仅包含系统和应用程序所需 RPM 以及维护和查错目的所需 RPM 的 Linux 映像。较好的做法是从 RPM 最少列表开始，然后根据需要添加软件包。



提示：Just Enough Operating System (JeOS)

SUSE Appliance Program 包含一个称为 JeOS（最精简操作系统，Just Enough Operating System）的组件。JeOS 的占用量非常小，并且可以对其进行自定义，以满足系统开发人员的特定需求。JeOS 主要用于硬件/软件设备或虚拟机开发。JeOS 的关键优点是高效、高性能、高安全性以及管理简单。

如果您无法选择 JeOS，较好的选择是最小安装模式。

要生成所有已安装软件包的列表，请使用以下命令：

```
root # zypper packages -i
```

要检索有关特定软件包的细节，请运行：

```
root # zypper info PACKAGE_NAME
```

要在删除某个软件包时检查并报告可能的冲突或依赖项，请运行：

```
root # zypper rm -D PACKAGE_NAME
```

此命令非常有用，因为在未测试的情况下就运行去除命令常常会产生大量投诉，并且需要手动寻找递归依赖项。



重要：基本系统软件包的去除

去除软件包时，请小心不要去除任何基本的系统软件包。这可能会使您的系统受损，无法再引导或修复。如果您对此不确定，最好是先对您的系统进行完整的备份，然后再去除任何软件包。

要最终去除一个或多个软件包，请结合 “-u” 开关（可去除所有未使用的依赖项）使用以下 **zypper** 命令：

```
root # zypper rm -u PACKAGE_NAME
```

12.2 修补 Linux 系统

构建用于进行补丁管理的基础结构是构成主动型安全 Linux 生产环境的另一个非常重要的部分。

建议您实施书面安全策略和程序，以处理 Linux 安全更新和问题。例如，安全策略应详细指出评估、测试和发布补丁的时间范围。与网络有关的安全漏洞应具有最高优先级，并应在短时间内立即予以解决。评估阶段应发生于测试实验室内，并且初始发布应首先在开发系统中进行。

独立的安全日志文件应包含有关已接收的 Linux 安全公告、已研究和已评估的补丁、补丁的应用时间等细节。

SUSE 会发布以下三类补丁：安全、推荐和可选。我们提供了一些选项用于确保系统获得修补、保持最新且是安全的。每个系统都可以注册，然后使用附带的 YaST 工具（YaST 联机更新）通过 SUSE 更新网站来检索更新。SUSE 还创建了储存库镜像工具 (RMT)，它可以有效地维护可用/已发布补丁/更新/修复的本地储存库，以便系统随后可从该储存库中提取相应补丁/更新/修复，减少了因特网流量。SUSE 还提供了 SUSE Manager，用于维护、修补、报告和集中管理 Linux 系统，不局限于 SUSE，其他发行套件也包括在内。

12.2.1 YaST 联机更新

可使用 YaST 联机更新工具基于每台服务器安装重要的更新和改进。可通过包含补丁的产品特定更新目录获取 SUSE Linux Enterprise 系列的当前更新。您可以使用 YaST 并选择软件组中的联机更新来安装更新和改进。已将您系统当前可用的所有新补丁（可选补丁除外）标记为待安装。单击接受将会自动安装这些补丁。

12.2.2 自动联机更新

YaST 还可设置自动更新。选择软件 > 自动联机更新。配置每日或每周更新。有些补丁（如内核更新）需要用户交互，交互可能会导致自动更新停止。请选中跳过交互补丁使更新过程自动进行。

在这种情况下，请手动运行联机更新安装需要交互的补丁。

选中仅下载补丁后，将在指定时间下载补丁但不会进行安装。必须使用 `rpm` 或 `zypper` 对其进行手动安装。

12.2.3 储存库镜像工具 — RMT

适用于 SUSE Linux Enterprise 的储存库镜像工具比联机更新进程更进一步，建立了具有储存库和注册目标的代理系统。这有助于客户在每个系统上的防火墙内集中管理软件更新，同时维护其公司安全政策和法规合规性。

可下载的 RMT (<http://download.suse.com/>) 与 SUSE Customer Center (<https://scc.suse.com/>) 集成在一起，并提供与其同步的储存库和注册目标。这对于跟踪大型部署中的权利非常有帮助。RMT 可维护 SUSE Customer Center 的所有功能，同时允许更安全的集中部署。它随每个 SUSE Linux Enterprise 订购提供，因此完全受支持。

RMT 提供了默认配置的替代配置，需要打开出站连接的防火墙，然后每个设备才能接收更新。此要求通常会违反公司安全策略，并且可能被某些组织视为对法规遵从的威胁。RMT 与 SUSE Customer Center 集成，可确保每个设备都可以接收相应的更新，而无需打开防火墙，也没有冗余带宽要求。

通过 RMT，客户还能在本地跟踪整个企业中的 SUSE Linux Enterprise 设备（即服务器、桌面或服务点终端）。现在，客户可以轻松确定帐单周期结束时有多少权利需要续约，而不必走到数据中心来手动更新电子表格。

RMT 会向 SUSE Linux Enterprise 设备告知任何可用的软件更新。每个设备随后可从 RMT 获取所需的软件更新。RMT 的引入改进了网络中的 SUSE Linux Enterprise 设备之间的交互，并简化了这些设备接收系统更新的方式。RMT 支持每个安装实例均拥有一个适用于数百个 SUSE Linux Enterprise 设备的基础结构（取决于特定的使用配置文件）。这使服务器跟踪更精确、更有效。

简单来说，适用于 SUSE Linux Enterprise 的储存库镜像工具向客户提供：

- 防火墙和法规遵从保证
- 软件更新期间缩减的带宽使用率
- 来自 SUSE 的完全支持（依照活动订阅）
- 对与 SUSE Customer Center 的现有客户接口的维护
- 准确的服务器授权跟踪以及对订购使用情况的有效测量

- 自动化的过程以便轻松计算权利总数（不再需要电子表格！）
- 简单的安装过程，该过程会自动与 SUSE Customer Center 同步服务器权利

12.2.4 SUSE Manager

SUSE Manager 可自动进行 Linux 服务器管理，可让您更快且更准确地对服务器进行配置和维护。它会通过单个控制台监视每台 Linux 服务器的运行状况，以便您可以在服务器性能问题对业务产生影响之前识别出该问题。SUSE Manager 可让您在物理、虚拟和云环境中全面管理您的 Linux 服务器，同时还能提高数据中心效率。SUSE Manager 为 Linux 提供全面的生命周期管理：

- 资产管理
- 置备
- 软件包管理
- 补丁管理
- 配置管理
- 重新部署

有关 SUSE Manager 的详细信息，请参见 <https://www.suse.com/products/suse-manager/> 。

13 文件管理

13.1 磁盘分区

服务器至少应该具有以下独立文件系统：/、/boot、/usr、/var、/tmp 和 /home。如此可避免一些问题，例如，避免 /var 和 /tmp 下的日志记录空间和临时空间填满根分区。第三方应用程序也应位于独立的文件系统，例如在 /opt 下。

独立文件系统的另一项优势是可以选择只适合文件系统层次结构中某些区域的特殊挂载选项。一些相关的装入选项包括：

- noexec：阻止文件执行。
- nodev：阻止使用字符或块特殊设备。
- nosuid：阻止 set-user-ID 或 set-group-ID 位生效。
- ro：以只读方式装入文件系统。

在对分区装入应用上述选项之前，需要仔细考虑每个选项。否则应用程序可能会停止工作，或者可能会违反支持状态。如果正确应用，装入选项有助于抵御某些类型的安全攻击或避免错误配置。例如，无需将 set-user-ID 二进制文件置于 /tmp 中。

建议您查看第 2 章“通用准则”。请务必了解是否需要将可能影响运行中系统的分区进行分隔（例如，日志文件会填满 /var/log，因此有必要将 /var 从 / 分区分隔出来）。另一个注意事项是您可能需要利用 LVM 或其他卷管理器，或者至少是扩展分区类型，来解决 PC 类系统上的四个主分区的限制。

SUSE Linux Enterprise Server 中的另一项功能是将分区甚至是充当容器的单个目录或文件进行加密。有关详细信息，请参见第 14 章“加密分区和文件”。

13.2 检查文件权限和所有权

以下几节介绍修改默认权限和文件设置以提高主机安全性的一些方法。请务必注意，虽然您可以运行 **seccheck** 等默认 SUSE Linux Enterprise Server 实用程序锁定系统并改进一般文件安全性和用户环境，但最好了解如何修改这些项目。

SUSE Linux Enterprise Server 主机包含以下三个默认文件权限设置：`permissions.easy`、`permissions.secure` 和 `permissions.paranoid`，这三者均位于 `/etc` 目录中。这些文件用于定义特殊权限（如全局可写目录），或为文件定义 `setuser` ID 位（设置了 `setuser` ID 位的程序未以启动该程序的用户权限运行，而是以文件拥有者权限运行，通常为 `root`）。

管理员可以使用文件 `/etc/permissions.local` 添加自己的设置。实施上述其中一个默认权限规则集最简单的方法就是使用 YaST 中的本地安全模块。

将由某个选定规则集修改以下每一个主题，但请务必了解其自身信息。

13.3 默认的 `umask`

`umask`（用户文件创建模式掩码）命令是一个外壳内置命令，决定了新建文件和目录的默认文件权限。它可被系统调用重写，但许多程序和实用程序都使用 **`umask`**。默认情况下，**`umask`** 设置为 `022`。您可以更改 `/etc/profile` 中的值以全局修改此设置，也可以分别针对外壳启动文件中的每个用户进行更改。

要确定活动的 `umask`，请使用 **`umask`** 命令：

```
tux > umask
022
```

如果至少设置了一个位，则将从访问模式 `777` 去除 `umask`。

使用默认的 `umask`，您将会看到大多数用户预期可在 Linux 系统上看到的行为。

```
tux > touch a
tux > mkdir b
tux > ls -on
total 16
-rw-r--r--. 1 17086    0 Nov 29 15:05 a
drwxr-xr-x. 2 17086 4096 Nov 29 15:05 b
```

您可以指定任意 `umask` 值，具体取决于您的需求。

```
tux > umask 111
tux > touch c
```

```
tux > mkdir d
tux > ls -on
total 16
-rw-rw-rw-. 1 17086    0 Nov 29 15:05 c
drw-rw-rw-. 2 17086 4096 Nov 29 15:05 d
```

根据您的线程模型，您可以使用 037 等更严格的 umask，以防止意外的数据泄漏。

```
tux > umask 037
tux > touch e
tux > mkdir f
tux > ls -on
total 16
-rw-r-----. 1 17086    0 Nov 29 15:06 e
drwxr-----. 2 17086 4096 Nov 29 15:06 f
```

13.4 SUID/SGID 文件

如果在可执行文件上设置 SUID（设置用户 ID）位或 SGID（设置组 ID）位，它会以可执行文件拥有者的 UID 或 GID 来执行，而不是以执行人员的 UID 或 GID 来执行。举例来说，这意味着会以 root 的 UID 执行设置了 SUID 位且由 root 拥有的所有可执行文件。**passwd** 命令就是一个典型的示例，它允许普通用户更新由 root 拥有的 /etc/shadow 文件中的口令字段。

但当可执行文件存在安全漏洞时，SUID/SGID 位可能会被滥用。因此，您应该搜索并记录整个系统中的 SUID/SGID 可执行文件。要搜索整个系统中的 SUID 或 SGID 文件，您可以运行以下命令：

```
root # find /bin /boot /etc /home /lib /lib64 /opt /root /sbin /srv /tmp /usr /
var -type f -perm '/6000' -ls
```

如果您有其他文件系统结构，则需要扩展搜索的目录列表。

SUSE 仅在确实需要时才会会在二进制文件上设置 SUID/SGID 位。不是绝对必要的情况下，请确保代码开发人员未在其程序上设置 SUID/SGID 位。通常，您可以使用诸如去除世界/其他可执行文件位等解决方法。但更好的方法是更改软件设计或使用权限。

SUSE Linux Enterprise Server 支持文件权限，可授予程序更精细的特权，而不是 root 的全部权限：

```
root # getcap -v /usr/bin/ping
/usr/bin/ping = cap_new_raw+eip
```

以上命令只会为执行 **ping** 的人员授予 **CAP_NET_RAW** 权限。如果 **ping** 的内部存在漏洞，攻击者最多可以获得此权限，而不是 **root** 的全部权限。可能的情况下，应选择文件权限以支持 SUID 位。但仅当二进制文件的 SUID 设置为 **root**，而不是其他 **news**、**lp** 等类似用户时，这一条才适用。

13.5 全局可写文件

全局可写文件存在安全风险，因为系统上的任何用户都可以对其进行修改。此外，全局可写目录允许任何人添加或删除文件。要查找全局可写文件和目录，您可以使用以下命令：

```
root # find /bin /boot /etc /home /lib /lib64 /opt /root /sbin /srv /tmp /usr /
var -type f -perm -2 ! -type l -ls
```

如果您有其他文件系统结构，则需要扩展搜索的目录列表。

! -type l 参数可跳过所有符号链接，因为符号链接始终为全局可写。但只要链接的目标不是全局可写（由上述查找命令进行检查），就不会发生问题。

带有粘滞位的全局可写目录（例如 **/tmp** 目录）不允许任何人（文件拥有者除外）在此目录中执行文件删除或重命名操作。粘滞位使文件与其创建用户相关联，可防止其他用户删除或重命名这些文件。因此，带有粘滞位的全局可写目录通常都不会发生问题，具体视目录的用途而定。例如 **/tmp** 目录：

```
tux > ls -ld /tmp
drwxrwxrwt 18 root root 16384 Dec 23 22:20 /tmp
```

输出中的 **t** 模式位表示粘滞位。

13.6 孤立文件或无拥有者的文件

未被任何用户或组拥有的文件本身不一定会发生安全问题。但无拥有者的文件可能会在将来发生安全问题。例如，如果创建了新用户，并且该新用户获得的 UID 刚好与无拥有者文件的 UID 相同，则该新用户将自动成为这些文件的拥有者。

要查找未被任何用户或组拥有的文件，请使用以下命令：

```
root # find /bin /boot /etc /home /lib /lib64 /opt /root /sbin /srv /tmp /usr /  
var -nouser -o -nogroup
```

如果您有其他文件系统结构，则需要扩展搜索的目录列表。

另一个问题是某些文件不是通过打包系统安装的，因而不会收到更新。您可以使用以下命令检查此类文件：

```
tux > find /bin /lib /lib64 /usr -path /usr/local -prune -o -type f -a -exec /  
bin/sh -c "rpm -qf {} &> /dev/null || echo {}" \;
```

因为经过设计的文件名可能会导致命令执行，会以不可信用户（例如“没有任何用户”）运行此命令。这不应成为一个问题，因为这些目录只能由 root 进行写入，但仍不失为一项良好的安全预防措施。

这将会显示 /bin、/lib、/lib64 和 /usr（/usr/local 中的文件除外）下软件包管理器未跟踪的所有文件。这些文件可能不代表安全问题，但您应注意未跟踪的文件，并执行必要的预防措施确保这些文件是最新的。

14 加密分区和文件

加密文件、分区和整个磁盘可以防止有人未经授权访问您的数据，并保护您的机密文件和文档。

您可以选择的加密方案如下：

加密硬盘分区

可在安装期间或在已安装的系统中使用 YaST 创建加密分区。有关更多信息，请参见第 14.1.1 节“在安装过程中创建加密分区”和第 14.1.2 节“在正在运行的系统上创建加密分区”。此选项还可用于可移动媒体（如外部硬盘），如第 14.1.3 节“加密可移动媒体的内容”中所述。

使用 GPG 加密单个文件

要快速加密一个或多个文件，可以使用 GPG 工具。有关更多信息，请参见第 14.2 节“使用 GPG 加密文件”。



警告：加密提供的保护有限

本章中所述的加密方法无法防范运行中系统的安全性受到损害。成功装入加密卷后，具有适当权限的任何人都可以访问它。不过，加密媒体在计算机丢失或被窃的情况下会很有用，它还可以防止未经授权的人员读取您的机密数据。

14.1 使用 YaST 设置加密文件系统

使用 YaST 在安装期间或已安装的系统中加密分区或部分文件系统。不过，在已安装的系统中加密分区更加困难，因为这需要重新调整分区大小并更改现有分区。在此情况下，创建一个指定大小的加密文件来**储存**其他文件或文件系统的某些部分可能更加方便。要加密整个分区，需要在分区布局中提供一个专用于加密的分区。默认情况下，YaST 的标准分区建议并不包括加密分区。请在分区对话框中手动添加加密分区。

14.1.1 在安装过程中创建加密分区



警告：口令输入

确保牢记加密分区的口令。没有这个口令将无法访问或恢复加密数据。

用于进行分区的 YaST 专家对话框提供了创建加密分区所需的选项。要创建新的加密分区，请执行以下操作：

1. 单击系统 > 分区程序运行 YaST 专家分区程序。
2. 选择一个硬盘，单击添加，然后选择主分区或扩展分区。
3. 选择分区大小或者要在磁盘上使用的区域。
4. 选择文件系统以及此分区的安装点。
5. 选中加密设备复选框。



注意：需要其他软件

选中加密设备后，可能会出现一个弹出窗口，要求您安装其他软件。请确认安装全部所需的软件包，以确保加密分区正常工作。

6. 如果仅在必要的情况下才需要装入加密文件系统，请在 `Fstab` 选项中启用不装入分区，否则请启用装入分区并输入安装点。
7. 单击下一步，并输入用于加密此分区的口令。不会显示该口令。为防止键入错误，您需要输入口令两次。
8. 单击完成以完成该过程。新加密的分区现在即创建成功。

在引导过程中，操作系统会在装入 `/etc/fstab` 中设置为自动装入的任何加密分区之前要求您输入口令。此类分区在装入后可供所有用户使用。

要在启动期间跳过装入加密分区的步骤，请在提示输入口令时按 **Enter**。然后，再次拒绝输入口令的提示。在此情况下，不装入加密文件系统，并且操作系统继续引导并阻止对您的数据的访问。

要想装入引导期间未装入的加密分区，请打开文件管理器，然后在列出文件系统上公用位置的窗格中单击该分区项。系统会提示您输入口令，然后将装入该分区。

在已存在分区的计算机上安装系统时，您还可以决定是否在安装期间加密现有分区。在此情况下，请遵循第 14.1.2 节“在正在运行的系统上创建加密分区”中的说明并注意此操作将会损坏现有分区中的所有数据。

14.1.2 在正在运行的系统上创建加密分区



警告：在正在运行的系统上激活加密

还可以在正在运行的系统上创建加密分区。但是，加密现有分区会损坏分区中的所有数据，并需要重新调整现有分区的大小及结构。

在运行中的系统上，于 YaST 控制中心选择系统 > 分区程序。单击是继续。在 Expert Partitioner 中选择要加密的分区，并单击编辑。其余过程与第 14.1.1 节“在安装过程中创建加密分区”中描述的过程相同。

14.1.3 加密可移动媒体的内容

YaST 将可移动媒体（例如外部硬盘或闪存盘）当作任何其他储存设备一样处理。您可按上述方法加密外部媒体上的虚拟磁盘或分区，但应禁止引导时装入，因为通常系统仅在已启动并运行时才会连接可移动媒体。

如果您已使用 YaST 对可移动设备进行加密，GNOME 桌面会自动识别加密分区并在检测到该设备时提示输入口令。如果您在运行 GNOME 时插入 FAT 格式的可移动设备，输入口令的桌面用户将自动成为该设备的拥有者。对于文件系统不是 FAT 的设备，请显式更改非 root 用户的所有权，以授予其对设备的读写访问权限。

14.2 使用 GPG 加密文件

可以使用 GPG 加密软件来加密单个文件和文档。

要使用 GPG 加密文件，首先需生成一个密钥对。为此，请运行 **gpg --gen-key** 并遵循屏幕上的指导操作。生成密钥对时，GPG 将会基于您的真实姓名、注释和电子邮件地址创建一个用户 ID (UID)，以用于标识密钥。指定用于加密文件的密钥时需要用到此 UID（或只是它的一部分，例如您的名字或电子邮件地址）。要查找现有密钥的 UID，请使用 **gpg --list-keys** 命令。要加密文件，请使用以下命令：

```
tux > gpg -e -r UID  
FILE
```

请将 UID 替换为 UID 的一部分（例如，您的名字），并将 FILE 替换为要加密的文件。例如：

```
tux > gpg -e -r Tux secret.txt
```

此命令会创建可通过 .gpg 文件扩展名识别的指定文件（在本示例中为 secret.txt.gpg）的加密版本。

要解密加密文件，请使用以下命令：

```
tux > gpg -d -o DECRYPTED_FILE  
ENCRYPTED_FILE
```

请将 DECRYPTED_FILE 替换为想让解密的文件使用的名称，并将 ENCRYPTED_FILE 替换为要解密的加密文件。

请记住，只能使用加密时所用的相同密钥来解密加密文件。如果您要与其他人共享加密文件，必须使用此人的公共密钥来加密该文件。

15 使用 cryptctl 对托管应用程序的储存区加密

数据库和类似的应用程序常常托管在由第三方工作人员管理的外部服务器上。某些数据中心维护任务需要第三方工作人员直接访问受影响的系统。在此类情况下，为了满足隐私要求，就必须进行磁盘加密。

cryptctl 可让您使用 LUKS 加密敏感目录，并提供以下附加功能：

- 加密密钥位于中心服务器上，而中心服务器可位于客户本地。
- 系统会在计划外重引导后自动重新装入加密分区。

cryptctl 包括以下组件：

- 客户端是一台包含一个或多个加密分区的计算机，但不永久储存解密这些分区所必需的密钥。例如，客户端可以是云或托管计算机。
- 由服务器来保存加密密钥，客户端可以请求这些密钥来解锁加密分区。

您也可以设置 **cryptctl** 服务器，以便在与 KMIP（密钥管理互操作性协议）1.3 兼容的服务器上储存加密密钥。在这种情况下，**cryptctl** 服务器将不储存客户端的加密密钥，而是依赖与 KMIP 兼容的服务器提供这些密钥。



警告：cryptctl 服务器维护

由于 **cryptctl** 服务器负责管理加密磁盘超时，而且还可以保存加密密钥（具体取决于配置），因此它应该由您自己直接控制，并仅由可信的人员管理。

此外，应定期对其进行备份。丢失服务器的数据意味着会失去客户端上加密分区的访问权限。

为了处理加密，**cryptctl** 将 LUKS 与 aes-xts-256 加密法和 512 位密钥结合使用。可使用 TLS 通过证书校验来传输加密密钥。

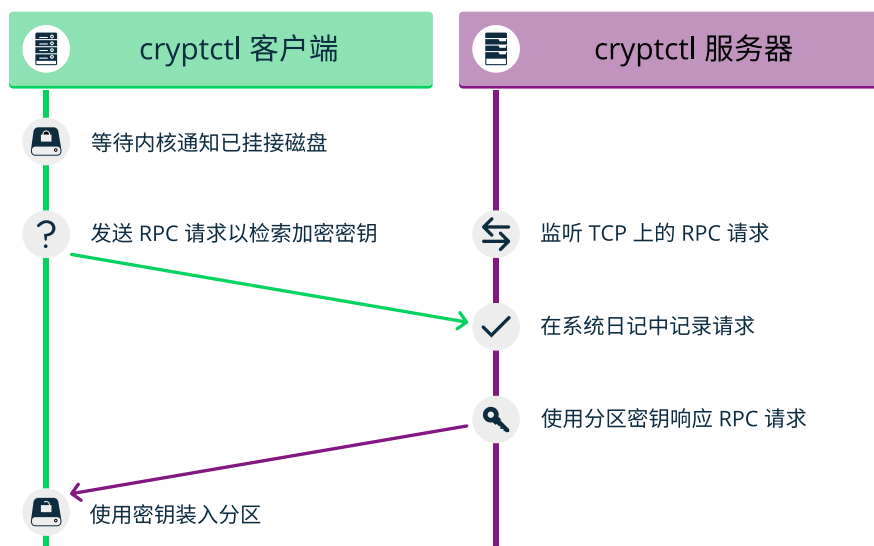


图 15.1：使用 cryptctl 检索密钥（不连接 KMIP 服务器的模式）

注意：安装 cryptctl

在继续之前，请确保软件包 `cryptctl` 已安装在您要设置为服务器或客户端的所有计算机上。

15.1 设置 cryptctl 服务器

在您可以将一台计算机定义为 `cryptctl` 客户端之前，需将一台计算机设置为 `cryptctl` 服务器。

在开始之前，请选择是否要使用自我签名证书来保护服务器与客户端之间的通讯。如果不使用，请为服务器生成 TLS 证书，并通过证书颁发机构为该证书签名。

此外，可以让客户端使用由证书颁发机构签名的证书向服务器进行身份验证。要采取这项额外的安全措施，请务必在开始执行此过程之前准备好 CA 证书。

1. 以 `root` 身份运行：

```
root # cryptctl init-server
```

2. 回答随后出现的每个提示问题，并在每回答一个问题后按 `Enter`。如果有默认的答案，提示末尾的方括号中会显示此答案。

- a. 选择使用至少包含 10 个字符的口令并确认该口令。此口令将充当主口令，可以解锁服务器上注册的所有分区。
- b. 指定 PEM 编码的 TLS 证书或证书链文件的路径，或者将该字段留空以创建自我签名证书。如果指定了路径，请使用绝对路径。
- c. 如果您不想使用所显示的默认主机名来标识服务器，请指定主机名。**cryptctl** 随后将生成包含该主机名的证书。
- d. 指定 IP 地址，该地址属于您要在其上监听来自客户端的解密请求的网络接口；然后设置端口号（默认端口为 3737）。
默认 IP 地址设置 0.0.0.0 表示 **cryptctl** 将使用 IPv4 在所有网络接口上监听客户端请求。
- e. 指定服务器上用于保存客户端解密密钥的目录。
- f. 指定客户端是否需要使用 TLS 证书向服务器进行身份验证。如果您选择否，则表示客户端仅使用磁盘 UUID 进行身份验证。（不过，在所有情况下，都将使用服务器证书加密通讯。）
如果您选择是，请选取一个用于对客户端证书进行签名的 PEM 编码的证书颁发机构。
- g. 指定是否使用一台与 KMIP 1.3 兼容的服务器（或多台此类服务器）来储存客户端的加密密钥。如果您选择此选项，请提供一台或多台与 KMIP 兼容的服务器的主机名和端口。
此外，请提供 KMIP 服务器的用户名、口令、CA 证书，以及 **cryptctl** 服务器的客户端身份证书。



重要：无法轻松重新配置 KMIP 设置

以后将无法轻松更改有关使用 KMIP 服务器的设置。要更改此设置，需要从头开始配置 **cryptctl** 服务器及其客户端。

- h. 最后，配置一台 SMTP 服务器用于发送加密和解密请求的电子邮件通知，或者将提示留空以跳过电子邮件通知的设置。



注意：受口令保护的服务器

cryptctl 目前无法使用受身份验证保护的 SMTP 服务器发送电子邮件。如果必须发送电子邮件，请设置本地 SMTP 代理。

i. 当系统询问是否要启动 **cryptctl** 服务器时，请输入 y。

3. 要检查服务 **cryptctl-server** 的状态，请使用：

```
root # systemctl status cryptctl-server
```

如果以后要重新配置服务器，请执行以下操作之一：

- 再次运行 **cryptctl init-server** 命令。**cryptctl** 随后会建议将现有设置用作默认设置，因此您只需指定要更改的值。
- 直接在配置文件 `/etc/sysconfig/cryptctl-server` 中进行更改。
不过，为了避免出现问题，请不要手动更改 `AUTH_PASSWORD_HASH` 和 `AUTH_PASSWORD_SALT` 设置。您需要正确计算这些选项的值。

15.2 设置 **cryptctl** 客户端

目前仅支持下述 **cryptctl** 交互式设置方法。

确保满足以下先决条件：

- 可通过网络使用 **cryptctl** 服务器。
- 存在一个要加密的目录。
- 客户端计算机包含一个可用的空分区，该分区足以容纳要加密的目录。
- 使用自我签名证书时，在服务器上生成的证书（*.crt 文件）可在客户端本地使用。否则，服务器证书的证书颁发机构必须受客户端的信任。
- 如果您将服务器设置为要求客户端使用客户端证书进行身份验证，请为客户端准备一个由您为服务器选择的 CA 证书签名的 TLS 证书。

1. 以 `root` 身份运行：

```
root # cryptctl encrypt
```

2. 回答随后出现的每个提示问题，并在每回答一个问题后按 `Enter`。如果有默认的答案，提示末尾的方括号中会显示此答案。

- a. 指定要在 `cryptctl` 服务器上连接到的主机名和端口。
- b. 如果您已将服务器配置为要求客户端使用 TLS 证书进行身份验证，请指定客户端的证书和密钥文件。客户端证书必须由设置服务器时选择的证书颁发机构签名。
- c. 指定服务器证书 (`*.crt` 文件) 的绝对路径。
- d. 输入设置服务器时指定的加密口令。
- e. 指定要加密的目录的路径。指定用于包含目录加密内容的空分区的路径。
- f. 指定允许同时解密该分区的计算机数目。

然后指定在从一个或多个客户端收到最后一个活跃信号之后到允许其他计算机解密分区之前必须经过的超时（以秒为单位）。

当计算机意外停止工作然后再重引导时，它需要能够再次解锁其分区。这意味着，此超时应设置为比客户端重引导时长略短的时间。

重要：超时时长

如果设置的时间太长，计算机首次尝试解密加密分区时将会失败。虽然 `cryptctl` 之后会继续定期检查加密密钥是否可用，但这会造成一定的延迟。

如果超时设置得太短，包含加密分区副本的计算机首先解锁该分区的可能性将会提高。

3. 要开始加密，请输入 `yes`。

`cryptctl` 现在会将指定目录加密到先前为空的分区中，然后装入这个新加密的分区。文件系统类型将与原始的未加密文件系统的类型相同。

在创建加密分区之前，`cryptctl` 会将原始目录的未加密内容移至带有 `cryptctl-moved-` 前缀的位置。

4. 要检查是否确实正确装入了该目录，请使用：

```
tux > lsblk -o NAME,MOUNTPOINT,UUID
NAME                                MOUNTPOINT          UUID
[...]
sdc
└─sdc1                                PARTITION_UUID
   └─cryptctl-unlocked-sdc1  /secret-partition  UNLOCKED_UUID
```

cryptctl 通过加密分区的 UUID 来识别该分区。在上一示例中，其为 sdc1 旁边显示的 UUID。

在服务器上，您可以使用 **cryptctl** 来检查目录是否已解密。

```
root # cryptctl list-keys
```

如果分区已成功解密，您将看到如下所示的输出：

```
2019/06/06 15:50:00 ReloadDB: successfully loaded database of 1 records
Total: 1 records (date and time are in zone EDT)
Used By      When                UUID  Max.Users  Num.Users  Mount Point
IP_ADDRESS   2019-06-06 15:00:50  UUID  1          1          /secret-
partition
```

如果分区未成功解密，您将看到如下所示的输出：

```
2019/06/06 15:50:00 ReloadDB: successfully loaded database of 1 records
Total: 1 records (date and time are in zone EDT)
Used By      When                UUID  Max.Users  Num.Users  Mount Point
              2019-06-06 15:00:50  UUID  1          1          /secret-
partition
```

在空的 Used by 列中可以看到差异。

校验显示的 UUID 是否属于先前加密的分区。

5. 确认加密分区可正常工作后，从客户端中删除未加密内容。例如，使用 **rm**。为了提高安全性，请在删除文件内容之前将其重写（例如，使用 **shred -u**）。

❗ 重要：shred 不保证完全擦除该数据

使用 **shred** 不能保证完全去除所有数据，具体取决于储存媒体的类型。具体而言，SSD 通常采用耗损均衡策略，这使得 **shred** 的效率不高。

从客户端到服务器的连接配置储存在 `/etc/sysconfig/cryptctl-client` 中，可以手动编辑。

服务器将客户端分区的加密密钥储存在 `/var/lib/cryptctl/keydb/PARTITION_UUID` 中。

15.3 使用服务器端命令检查分区解锁状态

当 **cryptctl** 客户端处于活动状态时，它每 10 秒会向 **cryptctl** 服务器发送一次“检测信号”。如果在设置客户端期间配置的超时时长内服务器未收到来自客户端的检测信号，服务器将认为客户端已脱机。然后，服务器将允许另一个客户端与其连接（或允许同一客户端在重引导后重新连接）。

要查看所有密钥的使用状态，请使用：

```
root # cryptctl list-keys
```

`Num.Users` 下的信息显示该密钥当前是否已使用。要查看单个密钥的更多细节，请使用：

```
root # cryptctl show-key UUID
```

此命令将显示有关安装点、装入选项、用法选项、上次检索密钥的时间，以及来自客户端的最后三个检测信号的信息。

此外，您可以使用 **journalctl** 来查找检索密钥时的日志。

15.4 手动解锁加密分区

可通过两种方法手动解锁分区，这两种方法都在客户端上运行：

- **联机解锁：** 联机解锁允许规避超时或用户限制。当客户端与服务器之间已建立网络连接，但客户端（目前）无法自动解锁分区时，可以使用此方法。此方法将解锁计算机上的所有加密分区。
要使用此方法，请运行 `cryptctl online-unlock`。准备好输入在设置服务器时指定的口令。
- **脱机解锁：** 当客户端无法或者不得联机与其服务器通讯时，可以使用此方法。服务器中的加密密钥必须仍然可用。此方法只能在万不得已的情况下才使用，每次只能解锁一个分区。
要使用此方法，请运行 `cryptctl offline-unlock`。服务器的必备分区 (`/var/lib/cryptctl/keydb/PARTITION_UUID`) 的密钥文件需在客户端上可用。

15.5 维护停机过程

为确保在维护停机期间不能解密分区，请关闭客户端并禁用 `cryptctl` 服务器。您可以通过以下方式来实现此目的：

- 停止服务 `cryptctl-server`：

```
root # systemctl stop cryptctl-server
```

- 断开 `cryptctl` 服务器的网络连接。

15.6 更多信息

有关详细信息，另请参见项目主页 <https://github.com/SUSE/cryptctl/>。

16 用户管理

16.1 各种帐户检查

16.1.1 未锁定的帐户

请务必锁定未用于登录的所有系统帐户和供应商帐户。要获取您系统中未锁定帐户的列表，您可以查找不含以 ! 或 * 开头（在 /etc/shadow 文件中）的加密口令字符串的帐户。如果使用 passwd -l 锁定帐户，将会在加密口令之前添加 !!，以有效禁用该口令。如果使用 usermod -L 锁定帐户，将会在加密口令之前添加 !。默认情况下，许多系统和共享帐户通常都由口令字段中的 * 或 !! 进行锁定，这会让加密口令变为无效字符串。因此，要获取所有未锁定（可加密）帐户的列表，请运行以下命令：

```
root # egrep -v ':\*|:!' /etc/shadow | awk -F: '{print $1}'
```

同时确保所有帐户在 /etc/passwd 中的口令字段中都有一个 x。以下命令列出口令字段中没有 x 的所有帐户：

```
root # grep -v ':x:' /etc/passwd
```

口令字段中的 x 表示该口令已阴影化，例如，需要在 /etc/shadow 文件中查找加密口令。如果 /etc/passwd 中的口令字段为空，则系统将不会查找阴影文件，并且不会在出现登录提示时提示用户提供口令。

16.1.2 未使用的帐户

应从系统中去除未被用户、应用程序、系统或守护程序使用的所有系统帐户或供应商帐户。您可以使用以下命令，查找特定帐户是否拥有任何文件：

```
root # find / -path /proc -prune -o -user ACCOUNT -ls
```

在此示例中，-prune 选项用于跳过 /proc 文件系统。如果您确定可以删除某个帐户，可以使用以下命令去除该帐户：

```
root # userdel -r ACCOUNT
```

如果没有 `-r` 选项，**userdel** 将不会删除用户的主目录和邮件假脱机目录 (`/var/spool/mail/USER`)。请注意，许多系统帐户都没有主目录。

16.2 启用口令时效

口令失效是一种普遍采用的最佳实践，但对于一些系统帐户和共享帐户（例如 Oracle）而言，可能需要将其排除。如果应用程序帐户失效，这些帐户的失效口令可能会导致系统服务中断。通常情况下，应针对系统帐户和共享帐户的口令更改规则/程序制定相应的公司政策。但常规用户帐户口令应该会自动失效。以下示例显示如何针对各个用户帐户设置口令失效。

使用 **useradd** 命令创建新帐户时，可以使用下表中的文件和参数。将在 `/etc/shadow` 文件中为每个用户帐户储存此类设置。如果使用 YaST 工具（用户和组管理）添加用户，则会为每个用户提供该设置。下面是各种不同的设置，其中一些也可能是系统范围的设置（例如，`/etc/login.defs` 和 `/etc/default/useradd` 的修改）：

<code>/etc/login.defs</code>	<code>PASS_MAX_DAYS</code>	口令保持有效的最大天数。
<code>/etc/login.defs</code>	<code>PASS_MIN_DAYS</code>	自上次更改到用户下次可更改口令之前的最小天数。
<code>/etc/login.defs</code>	<code>PASS_WARN_AGE</code>	从上次更改口令到下次提醒更改口令间隔的天数。
<code>/etc/default/useradd</code>	<code>INACTIVE</code>	口令失效后帐户处于禁用状态的天数。
<code>/etc/default/useradd</code>	<code>EXPIRE</code>	帐户失效日期（采用 YYYY-MM-DD 格式）。



注意

在这些修改之前创建的用户不受影响。

请确保在 `/etc/login.defs` 和 `/etc/default/useradd` 文件中更改上述参数。审阅 `/etc/shadow` 文件将显示在添加用户之后如何储存这些设置。

要创建新用户帐户，请执行以下命令：

```
root # useradd -c "TEST_USER" -g USERS TEST
```

`-g` 选项指定此帐户的主组：

```
root # id TEST
uid=509(test) gid=100(users) groups=100(users)
```

为 `/etc/shadow` 文件中的测试用户在 `/etc/login.defs` 和 `/etc/default/useradd` 中记录的设置如下所示：

```
root # grep TEST /etc/shadow
test:!!:12742:7:60:7:14:::
```

可使用 **chage** 命令随时修改口令时效。要禁用系统帐户和共享帐户的口令时效，您可以运行以下 **chage** 命令：

```
root # chage -M -1 SYSTEM_ACCOUNT_NAME
```

要获取口令失效信息，请运行：

```
root # chage -l SYSTEM_ACCOUNT_NAME
```

例如：

```
root # chage -l TEST
Minimum: 7
Maximum: 60
Warning: 7
Inactive: 14
Last Change: Jan 11, 2015
Password Expires: Mar 12, 2015
Password Inactive: Mar 26, 2015
Account Expires: Never
```

16.3 实施更强的口令

在经审计的系统上，请务必限制用户使用可被轻松破解的简单口令。可以将复杂的口令记下来，只要妥善保管即可。有些人主张通过强口令来保护您免受字典攻击，并可通过数次失败尝试之后锁定帐户来防御此类攻击。但此方法并非始终有效。如果进行此类设置，锁定系统帐户可能会使应用程序和系统服务中断，这会产生另一个同样棘手的问题 — 拒绝服务攻击。

但无论如何，实施有效的口令管理安全措施都很重要。大多数公司会要求口令至少包含一个数字、一个小写字母和一个大写字母。虽然政策各不相同，但要在口令强度/复杂性和管理难易度之间保持平衡可能并不容易。

16.4 使用 PAM 进行口令和登录管理

Linux-PAM（适用于 Linux 的可插入身份验证模块）是一组共享库，可让本地系统管理员选择应用程序该如何对用户进行身份验证。

强烈建议您熟悉 PAM 的功能以及如何利用此体系结构为某个环境提供最佳身份验证设置。您可以完成此配置一次并在所有系统中实施此配置（标准），也可以针对各个主机进行增强（增强安全性 — 按主机/服务/应用程序进行增强）。重点了解该体系结构的灵活性。

要了解有关 PAM 体系结构的详细信息，请参见 [/usr/share/doc/packages/pam](#) 目录中的 PAM 文档（提供多种格式）。

下面讨论的是如何修改默认 PAM 堆栈的示例（特别是围绕口令政策展开），例如口令强度、口令重复使用和帐户锁定。虽然只涉及了少数几种可能性，但它们是一个良好的开端，向您展示了 PAM 的灵活性。

❗ 重要：pam-config 限制

pam-config 工具可用于配置包含全局选项的 common-

{account,auth,password,session} PAM 配置文件。这些文件包含以下注释：

```
# This file is autogenerated by pam-config. All changes  
# will be overwritten.
```

个别服务文件（例如 `login`、`password`、`sshd` 和 `su`）必须直接编辑。您可以选择直接编辑所有文件，而不使用 **pam-config**，虽然 **pam-config** 包含转换较旧配置、更新当前配置和健全性检查等有用功能。有关详细信息，请参见 [man 8 pam-config](#)。

16.4.1 口令强度

SUSE Linux Enterprise Server 可利用 `pam_cracklib` 库测试弱口令，并在确定口令明显较弱时建议使用较强的口令。以下参数示例可能属于公司口令策略的一部分，或者由于审计约束而需要的某些设置。

PAM 库遵循定义的流程。通常，设计完美堆栈的最佳方式是考虑所有要求和策略并绘制流程图。

表 16.1：口令强制执行的示例规则/约束

<code>pam_cracklib.so</code>	<code>minlen=8</code>	口令的最小长度为 8
<code>pam_cracklib.so</code>	<code>lcredit=-1</code>	小写字母的最少数量为 1
<code>pam_cracklib.so</code>	<code>ucredit=-1</code>	大写字母的最少数量为 1
<code>pam_cracklib.so</code>	<code>dcredit=-1</code>	数字的最少数量为 1
<code>pam_cracklib.so</code>	<code>ocredit=-1</code>	其他字符的最少数量为 1

要设置这些口令限制，请使用 **pam-config** 工具指定您要配置的参数。例如，可使用如下命令修改最小长度参数：

```
tux > sudo pam-config -a --cracklib-minlen=8 --cracklib-retry=3 \
--cracklib-lcredit=-1 --cracklib-ucredit=-1 --cracklib-dcredit=-1 \
--cracklib-ocredit=-1 --cracklib
```

现在校验新口令限制是否适用于新口令。登录非 `root` 帐户并使用 **passwd** 命令更改口令。请注意，如果您在 `root` 下运行 **passwd** 命令，则不会强制执行上述要求。

16.4.2 限制使用先前的口令

pam_pwhistory 模块可用于配置无法重复使用的先前口令数量。以下命令可在系统上实施口令限制，如此至少六个月内无法重复使用某个口令。

```
tux > sudo pam-config -a --pwhistory --pwhistory-remember=26
```

回想一下，在第 16.2 节“启用口令时效”中我们将 `PASS_MIN_DAYS` 设置为 7，该选项指定了两次口令更改之间的最少天数。因此，如果 `pam_unix` 配置为记住 26 个口令，则至少六个月（ 26×7 天）内无法重复使用先前曾使用过的口令。

pam-config 命令生成的 PAM 配置 (`/etc/pam.d/common-auth`) 如下所示：

```
auth      required    pam_env.so
auth      required    pam_unix.so      try_first_pass
account   required    pam_unix.so      try_first_pass
password  requisit     pam_cracklib.so
password  required    pam_pwhistory.so      remember=26
password  optional    pam_gnome_keyring.so  use_authtok
password  required    pam_unix.so      use_authtok nullok shadow try_first_pass
session   required    pam_limits.so
session   required    pam_unix.so      try_first_pass
session   optional    pam_umask.so
```

16.4.3 登录失败次数太多后锁定用户帐户

在到达所定义的 `ssh`、`login`、`su` 或 `sudo` 失败尝试次数后锁定帐户是一种常见的安全做法。但如果应用程序、管理员或 `root` 用户被锁定，则可能会导致服务中断。实际上，如果故意大量制造登录失败，就很容易会导致拒绝服务攻击。幸好可以直接通过 PAM 来控制此情况。

默认情况下，PAM 允许所有 `root` 登录。使用 **pam_tally2** 可控制所有其他用户（包括人类用户和系统用户）的失败登录行为。将下面一行添加到 `/etc/pam.d/login` 上方可在六次登录失败之后锁定所有用户（`root` 用户除外），并在十分钟后自动解锁帐户：

```
auth required pam_tally2.so deny=6 unlock_time=600
```

下面是一个完整的 `/etc/pam.d/login` 文件示例：

```
##PAM-1.0
```

```

auth    requisite    pam_nologin.so
auth    include      common-auth
auth    required     pam_tally2.so deny=6 unlock_time=600
account include      common-account
account required     pam_tally2.so
password include     common-password
session required     pam_loginuid.so
session include      common-session
#session optional    pam_lastlog.so nowtmp showfailed
session optional     pam_mail.so standard

```

您也可以锁定 root 用户，不过显然您必须非常确定要执行此操作：

```
auth required pam_tally2.so deny=6 even_deny_root unlock_time=600
```

您可以为 root 用户定义不同的锁定时间：

```
auth required pam_tally2.so deny=6 root_unlock_time=120 unlock_time=600
```

如果您希望必须由管理员来解锁帐户，请不要使用 `unlock_time` 选项。下面两个示例命令显示失败的登录尝试次数以及如何锁定用户帐户：

```

root # pam_tally2 -u username
Login          Failures Latest failure    From
username              6    12/17/19 13:49:43 pts/1

root # pam_tally2 -r -u username

```

尝试访问的默认位置记录在 `/var/log/tallylog` 中。

如果用户在登录超时失效后或在管理员重设置其帐户之后成功登录，计数器将重设置为 0。

配置其他登录服务，以在 `/etc/pam.d/` 下其各自的配置文件中使用的

pam_tally2：`sshd`、`su`、`sudo`、`sudo-i` 和 `su-l`。

16.5 限制 root 登录

默认情况下，为 `root` 用户分配了口令，并且该用户可以使用各种方法进行登录 — 例如，在本地终端上、在图形会话中，或者通过 SSH 远程登录。应尽可能限制使用这些方法登录。应避免共享使用 root 帐户。而是由各个管理员使用 `su` 或 `sudo` 等工具（有关详细信息，请键入 `man`

1 `su` 或 `man 8 sudo`) 获取提升的特权。如此可将 `root` 登录与特定用户相关联。同时还可增加了另一层安全保护；要获得完全的 `root` 访问权限，需要破解的不仅仅是 `root` 口令，而是 `root` 口令以及管理员普通帐户的口令。本节说明如何限制在不同级别系统上的直接 `root` 登录。

16.5.1 限制本地文本控制台登录

TTY 设备通过控制台提供文本模式的系统访问权限。对于桌面系统，通过本地键盘进行访问；如果是服务器系统，则通过连接到 KVM 交换机或远程管理卡（例如 ILO 和 DRAC）的输入设备进行访问。Linux 默认会提供 6 个不同的控制台，在文本模式下运行时，可通过组合键 `Alt - F1` 到 `Alt - F6` 切换它们；在图形会话中运行时，可通过组合键 `Ctrl - Alt - F1` 到 `Ctrl - Alt - F6` 进行切换。关联的终端设备命名为 `tty1` 至 `tty6`。

下面的步骤限制对第一个 TTY 的 `root` 访问。此访问方法仅作为系统的紧急访问方式，绝不应将其用于日常系统管理任务。



注意

此处显示的步骤根据 PC 体系结构（x86 和 AMD64/Intel 64）进行定制。在 POWER 等体系结构上，可以使用 `tty1` 以外的其他终端设备名称。请小心不要因为指定了错误的终端设备名称而将您自己完全锁定。您可以通过运行 `tty` 命令确定当前登录的终端的设备名称。请注意不要在虚拟终端（例如通过 SSH）或图形会话（设备名称 `/dev/pts/N`）中执行此操作，而只通过实际登录终端（可通过 `Alt - FN` 访问）进行操作。

过程 16.1：限制在本地 TTY 上进行 ROOT 登录

1. 请确保 PAM 堆栈配置文件 `/etc/pam.d/login` 包含 `auth` 块中的 `pam_securetty` 模块：

```
auth      requisite      pam_nologin.so
auth      [user_unknown=ignore success=ok ignore=ignore auth_err=die
default=bad] pam_securetty.so noconsole
auth      include        common-auth
```

这样系统会在本地控制台处理身份验证期间包含 `pam_securetty` 模块，该模块会将 `root` 限制为仅可在文件 `/etc/securetty` 中列出的 TTY 设备上登录。

2. 去除 `/etc/securetty` 中除这一项外的其他所有项。这将限制对 TTY 设备的 `root` 访问。

```
#
# This file contains the device names of tty lines (one per line,
# without leading /dev/) on which root is allowed to login.
#
tty1
```

3. 检查是否会拒绝 `root` 登录其他终端。应立即拒绝在 `tty2` 等终端上登录，甚至无需查询帐户口令。同时确保您仍可以成功登录 `tty1`，因而不会将 `root` 完全锁定在系统之外。

! 重要

请不要将 `pam_securetty` 模块添加到 `/etc/pam.d/common-auth` 文件中。这将破坏 `su` 和 `sudo` 命令，因为这些工具也会拒绝 `root` 身份验证。

! 重要

这些配置更改还将导致拒绝在 `/dev/ttyS0` 等串行控制台上进行 `root` 登录。如果您需要此类用例，则需要将 `/etc/securetty` 文件中额外列出相应的 TTY 设备。

16.5.2 限制图形会话登录

要提高您服务器的安全性，请完全避免使用图形环境。通常，图形程序不会设计为以 `root` 身份运行，因此与控制台程序相比更可能包含安全问题。如果您需要图形登录，请使用非 `root` 登录。配置您的系统，禁止通过 `root` 登录图形会话。

为了防止通过 `root` 登录图形会话，您可以采用第 16.5.1 节“限制本地文本控制台登录”中概述的相同基本步骤。只需将 `pam_securetty` 模块添加到属于显示管理器的 PAM 堆栈文件 — 例如，GDM 的 `/etc/pam.d/gdm`。图形会话还会在 TTY 设备上运行：默认为 `tty7`。因此，如果您限制 `root` 登录到 `tty1`，将拒绝通过 `root` 登录图形会话。

16.5.3 限制 SSH 登录

默认情况下，还允许 `root` 用户通过 SSH 网络协议远程登录计算机（如果 SSH 端口未被防火墙阻止）。要限制此类登录，请对 OpenSSH 配置进行以下更改：

1. 编辑 `/etc/ssh/sshd_config` 并调整以下参数：

```
PermitRootLogin no
```

2. 重新启动 `sshd` 服务以使更改生效：

```
systemctl restart sshd.service
```



注意

对于 OpenSSH，不适合使用 PAM `pam_securetty` 模块，因为在授权期间并非所有 SSH 登录都会通过 PAM 堆栈进行（例如使用 SSH 公共密钥身份验证时）。此外，攻击者能够区分错误口令和策略只能在稍后予以拒绝的成功登录。

16.6 为交互式外壳会话设置无活动超时

最好在一段无活动时间之后终止交互式外壳会话。例如，通过此方式可以阻止打开的无人监管会话或避免浪费系统资源。

默认情况下，外壳没有无活动超时。如此当外壳处于打开状态且在几天甚至几年内未被使用时，也不会有任何反应。不过，您可以将大多数外壳配置为在一段时间后自动终止空闲会话。以下示例显示如何为一些常见类型的外壳设置无活动超时。

可仅为登录外壳配置无活动超时，也可以为所有交互式外壳配置无活动超时。在后一种情况下，将针对每个外壳实例单独运行无活动超时。这意味着超时将会累积。当启动子外壳时，会为子外壳开始新的超时，并且仅在此超时过后，才会继续运行父外壳的超时。

下表包含 SUSE Linux Enterprise Server 随附的常见外壳选集的配置详细信息：

软件包	外壳特点	外壳变量	时间单位	只读设置	配置路径 (仅登录外壳)	配置路径 (所有外壳)
bash	bash 、 sh	TMOUT	秒	read-only TMOUT=	/etc/profile.d/ /etc/profile.d/	/etc/bash.bashrc
mksh	ksh 、 lksh 、 nsh	TMOUT	秒	read-only TMOUT=	/etc/profile.d/ /etc/profile.d/	/etc/mkshrc.local
tcsh	csh 、 tcsh	autologout	分钟	set -r autologout	/etc/ /etc/	/etc/tcshrc.local
zsh	zsh	TMOUT	秒	readonly TMOUT=	/etc/profile.d/ /etc/profile.d/	/etc/zshrc.local

每个列出的外壳支持使用一个内部超时外壳变量，可将该变量设置为某个特定时间值以触发无活动超时。如果您要防止用户覆盖超时设置，可以将相应的外壳超时变量标记为只读。上表中还提供了相应的变量声明语法。

注意

此功能仅有助于避免因用户疏忽或遵循不安全做法而导致的风险。而无法防范恶意用户。超时只适用于外壳的交互式等待状态。恶意用户总是能找到绕过超时的方法，使得无论在任何情况下，都可以让其会话保持打开状态。

要配置无活动超时，需要为每个外壳的启动脚本都添加匹配的超时变量声明。可仅为登录外壳使用一种路径，或为所有外壳使用一种路径，如表中所列。以下示例使用适合 **bash** 和 **ksh** 的路径和设置，设置了无法被用户覆盖的只读登录外壳超时。使用以下内容创建文件 `/etc/profile.d/timeout.sh`：

```
# /etc/profile.d/timeout.sh for SUSE Linux
#
# Timeout in seconds until the bash/ksh session is terminated
# in case of inactivity.
# 24h = 86400 sec
readonly TMOUT=86400
```

提示

建议使用 **screen** 工具在注销前分离会话。**screen** 会话不会终止，一旦有需要便可重新挂接。可以在未注销的情况下锁定活动会话（有关细节，请阅读 `man screen` 中的 `Ctrl - a - x / lockscreen`）。

16.7 防止意外拒绝服务

Linux 允许您对用户和组可以使用的系统资源量设置限制。如果因为程序中的 bug 导致用户和组耗尽太多资源（例如内存泄漏）、降低计算机速度，甚至使系统无法使用，此项设置也非常有用。错误的设置可能会允许程序使用过多资源，这可能会导致服务器无法响应新连接，甚至无法响应本地登录请求（例如，如果某个程序用掉主机上的所有可用文件句柄）。这也会成为一个安全问题，例如，如果允许某人用掉所有系统资源，将会导致拒绝服务攻击（无论是无意的还是更糟的恶意攻击）。设置用户和组的资源限制可能是一种有效的系统保护方法，具体视环境而定。

16.7.1 限制系统资源示例

以下示例演示了设置或限制 Oracle 用户帐户的系统资源使用的实际用法。有关系统资源设置列表，请参见 [/etc/security/limits.conf](#) 或 [man limits.conf](#)。

Bash 等大多数外壳都会提供基于用户的各种资源控制（例如允许的打开文件描述符数量上限或最大进程数）。要检查外壳中的所有当前限制，请执行以下命令：

```
root # ulimit -a
```

有关 Bash 外壳的 [ulimit](#) 的详细信息，请查看 Bash 手册页。

！ 重要：设置 SSH 会话的限制

使用 SSH 会话时，设置“硬”限制和“软”限制可能不会起到预期的效果。要查看有效行为，可能需要以 root 身份登录，然后使用 [su](#) 命令切换到权限受限的身份（例如，在这些示例中为 [Oracle](#)）。假设应用程序在引导过程中已自动启动，资源限制也应生效。如果通过 SSH 对资源限制进行的更改似乎未生效，可能需要将 [/etc/ssh/sshd_config](#) 中的 [UsePrivilegeSeparation](#) 设置为 [no](#)，并重新启动 SSH 守护程序（[systemctl restart sshd](#)）。但一般不建议这样做，因为这会降低系统安全性。

💡 提示：禁用通过 ssh 进行口令登录功能

您可以通过禁用 SSH 的口令身份验证来为服务器增加一定程度的安全性。请记住，您需要配置 SSH 密钥，否则无法访问服务器。要禁用口令登录，请在 [/etc/ssh/sshd_config](#) 中添加下面几行：

```
UseLogin no
UsePAM no
PasswordAuthentication no
PubkeyAuthentication yes
```

在此示例中，可通过以 [root](#) 身份编辑 [/etc/security/limits.conf](#) 并进行以下更改，来更改用户 [oracle](#) 可使用的文件句柄数或打开的文件数：

```
oracle          soft    nofile          4096
```

oracle	hard	nofile	63536
--------	------	--------	-------

第一行中的软限制定义了登录后 oracle 用户将拥有的文件句柄（打开的文件）数目限制。如果用户看到有关文件句柄用尽的错误消息，则可以执行以下命令，将此示例中所示的文件句柄数增加到硬限制（在此示例中为 63536）：

```
root # ulimit -n 63536
```

必要时，您可以设置更高的软限制和硬限制。



注意

请务必合理利用 `ulimit`。允许用户的 `nofile` 的“硬”限制（相当于内核限制（`/proc/sys/fs/file-max`））是非常糟糕的设置！如果用户占用了所有可用文件句柄，系统将无法启动新的登录，因为它无法访问执行登录所需的 PAM 模块。

您还需要确保在 `/etc/pam.d/common-auth` 中全局配置 `pam_limits`，或者针对以下文件中的 SSH、su、login 和 telnet 等个别服务进行配置：

`/etc/pam.d/sshd`（用于 SSH）

`/etc/pam.d/su`（用于 su）

`/etc/pam.d/login`（本地登录和 telnet）

如果您不想为所有登录启用该配置，会有一个特定 PAM 模块读取 `/etc/security/limits.conf` 文件。PAM 配置指令中的项将如下所示：

session	required	/lib/security/pam_limits.so
session	required	/lib/security/pam_unix.so

请务必注意，更改不会立即生效，需要新的登录会话：

```
root # su - oracle
tux > ulimit -n
4096
```

请注意，这些示例特定于 Bash 外壳；其他外壳的 `ulimit` 选项会有所不同。用户 oracle 的默认限制为 4096。要将用户 oracle 可使用的文件句柄数增加至 63536，请执行以下命令：

```
root # su - oracle
tux > ulimit -n
4096
tux > ulimit -n 63536
tux > ulimit -n
63536
```

要使其成为永久设置，需要向用户配置文件（`~/.bashrc` 或 `~/.profile` 文件）添加设置 **`ulimit -n 63536`**（仍是适用于 Bash），该配置文件为 SUSE Linux Enterprise Server 上的 Bash 外壳的用户启动文件（要校验外壳，请运行 **`echo $SHELL`**）。为此，可以针对用户 `oracle` 的 Bash 外壳运行以下命令：

```
root # su - oracle
tux > cat >> ~oracle/.bash_profile << EOF
ulimit -n 63536
EOF
```

16.8 显示登录标题

出于法律/审计政策原因，或者为了向用户提供安全说明，通常有必要在所有服务器上的登录屏幕上设置一个标题。

如果您想在用户登录文本型终端（例如，使用 SSH 或在本地控制台上）**后**列显登录标题，可以使用 `/etc/motd` 文件（motd = 当天的消息）。默认情况下，该文件存在于 SUSE Linux Enterprise Server 上，但是空文件。只需向适用的/组织所需的文件中添加内容。



注意：标题长度

请尽量将登录标题内容放在一个终端页面（或更少）中，如果一页容纳不下而需要滚动屏幕，会使阅读变得更加困难。

您也可以在用户登录文本型终端**前**列显登录标题。如果是本地控制台登录，您可以编辑 `/etc/issue` 文件，这会使标题在出现登录提示前显示。如果是通过 SSH 进行登录，您可以编辑 `/etc/ssh/sshd_config` 文件中的“Banner”参数，这会在出现 SSH 登录提示前相应地显示标题文本。

如果是通过 GDM 进行的图形登录，您可以遵循 [GNOME 管理员指南 \(https://help.gnome.org/admin/system-admin-guide/stable/login-banner.html.en\)](https://help.gnome.org/admin/system-admin-guide/stable/login-banner.html.en) 设置登录标题。此外，您可以进行以下更改，以要求用户通过选择是或否来对法律标题进行确认。编辑 `/etc/gdm/Xsession` 文件，并在脚本**开头**添加下面几行：

```
if ! /usr/bin/gdialog --yesno '\nThis system is classified...\n' 10 10; then
    /usr/bin/gdialog --infobox 'Aborting login'
    exit 1;
fi
```

需要将文本 `This system is classified...` 替换为所需的标题文本。请务必注意，此对话框不会阻止登录进行。有关 GDM 脚本的详细信息，请参见 [GDM 管理员手册 \(https://help.gnome.org/admin/gdm/stable/configuration.html.en#scripting\)](https://help.gnome.org/admin/gdm/stable/configuration.html.en#scripting)。

16.9 连接统计实用程序

下面是您可以用于获取有关用户登录的数据的命令列表：

who： 列出当前登录的用户。

w： 显示登录者及其执行的操作。

last： 显示最近登录的用户列表，包括登录时间、注销时间、登录 IP 地址等。

lastb： 与 **last** 相同（除了默认显示 `/var/log/btmp`），包含所有无效登录尝试。

lastlog： 此命令报告 `/var/log/lastlog` 中维护的数据，该文件是用户上次登录的记录。

ac： 安装 `acct` 软件包之后提供。按每个用户或每天等条件列显连接时间（以小时为单位）。此命令读取 `/var/log/wtmp`。

dump-utmp： 将原始数据从 `/var/run/utmp` 或 `/var/log/wtmp` 转换为 ASCII 可分析格式。

此外，如果未运行任何日志记录工具，请检查 `/var/log/messages` 文件或 **journalctl** 的输出。有关 `systemd` 日记的详细信息，请参见《管理指南》，第 17 章 “**journalctl**：查询 `systemd` 日志”。

17 Spectre/Meltdown Checker

spectre-meltdown-checker 是一个外壳脚本，用于测试您的系统是否容易受到多种推测执行漏洞的影响，这些漏洞在过去 20 年制造的所有 CPU 中普遍存在。这是一种硬件缺陷，攻击者可能会利用它来读取系统上的所有数据。在云计算服务中，如果多个虚拟机位于一台物理主机上，攻击者可以获取对所有虚拟机的访问权限。修复这些漏洞需要重新设计并更换 CPU。在采取此措施之前，可以通过多个软件补丁来缓解这些漏洞。如果您经常在更新 SUSE 系统，应该已安装了所有这些补丁。

spectre-meltdown-checker 会生成详细的报告。它不能为您的系统提供安全保证，但会显示采取了哪些缓解措施以及潜在的漏洞。

17.1 使用 spectre-meltdown-checker

安装该脚本，然后不指定任何选项以 root 身份运行它：

```
root # zypper in spectre-meltdown-checker
root # spectre-meltdown-checker.sh
```

您将看到如图 17.1 “spectre-meltdown-checker 的输出” 中所示的彩色输出：

```

dreamer:/home/carla # spectre-meltdown-checker.sh
Spectre and Meltdown mitigation detection tool v0.40

Checking for vulnerabilities on current system
Kernel is Linux 4.12.14-lp151.28.13-default #1 SMP Wed Aug 7 07:20:16 UTC 2019 (0c09ad2) x86_64
CPU is Intel(R) Xeon(R) CPU E5-1620 v3 @ 3.50GHz

Hardware check
* Hardware support (CPU microcode) for mitigation techniques
  * Indirect Branch Restricted Speculation (IBRS)
    * SPEC_CTRL MSR is available: YES
    * CPU indicates IBRS capability: YES (SPEC_CTRL feature bit)
  * Indirect Branch Prediction Barrier (IBPB)
    * PRED_CMD MSR is available: YES
    * CPU indicates IBPB capability: YES (SPEC_CTRL feature bit)
  * Single Thread Indirect Branch Predictors (STIBP)
    * SPEC_CTRL MSR is available: YES
    * CPU indicates STIBP capability: YES (Intel STIBP feature bit)
  * Speculative Store Bypass Disable (SSBD)
    * CPU indicates SSBD capability: YES (Intel SSBD)
  * L1 data cache invalidation
    * FLUSH_CMD MSR is available: YES
    * CPU indicates L1D flush capability: YES (L1D flush feature bit)
  * Enhanced IBRS (IBRS_ALL)
    * CPU indicates ARCH_CAPABILITIES MSR availability: NO
    * ARCH_CAPABILITIES MSR advertises IBRS_ALL capability: NO

```

图 17.1：SPECTRE-MELTDOWN-CHECKER 的输出

spectre-meltdown-checker.sh --help 会列出所有选项。此命令可用于将非彩色纯文本输出导出到文件中：

```
root # spectre-meltdown-checker.sh --no-color | tee filename.txt
```

上述示例是在运行中的系统上执行的情况，这是此脚本的默认运行方式。您也可以指定内核、配置和 System.map 文件的路径来脱机运行 **spectre-meltdown-checker**：

```

root # cd /boot
root # spectre-meltdown-checker.sh \
--no-color \
--kernel vmlinuz-4.12.14-lp151.28.13-default \
--config config-4.12.14-lp151.28.13-default \
--map System.map-4.12.14-lp151.28.13-default | tee filename.txt

```

其他有用的选项如下：

--verbose、-v

提高详细程度；重复指定可以不断提高详细程度，例如 **-v -v -v**

--explain

列显直观易懂的说明

--batch [short] [json] [nrpe] [prometheus]

以各种机器可读格式设置输出格式



重要：--disclaimer 选项

spectre-meltdown-checker.sh --disclaimer 提供有关该脚本能够和不能提供的功能的重要信息。

17.2 更多信息

有关详细信息，请参见以下参考：

- SUSE 知识库文章 #7022937 Security Vulnerability: Spectre Variant 4 (Speculative Store Bypass) aka CVE-2018-3639（安全漏洞：Spectre 变体 4（推测储存绕过），又称 CVE-2018-3639）：<https://www.suse.com/support/kb/doc/?id=7022937> ↗
- GitHub 上的 speed47/spectre-meltdown-checker 源代码，包括相关公共漏洞和暴露 (CVE) 的详细参考：<https://github.com/speed47/spectre-meltdown-checker> ↗
- SUSE 博客文章 Meltdown and Spectre Performance（Meltdown 和 Spectre 性能）：<https://www.suse.com/c/meltdown-spectre-performance/> ↗
- SUSE 知识库文章 #7022512，其中提供了有关体系结构、CVE 和缓解措施的信息：<https://www.suse.com/support/kb/doc/?id=7022512> ↗

18 使用 YaST 配置安全设置

YaST 的安全中心模块提供了一个中心控制面板，用于配置 SUSE Linux Enterprise Server 的安全相关设置。使用该模块可以配置与安全相关的各个方面，例如，有关登录过程、口令创建、引导权限、用户创建或默认文件权限的设置。在 YaST 控制中心内选择安全和用户 > 安全中心启动该模块。安全中心对话框即会打开并显示安全概览，左侧和右侧窗格中会显示其他配置对话框。

18.1 安全概览

安全概览显示系统最重要的安全设置的综合列表。列表中会清楚地列出每一项的安全状态。绿色对勾标记表示相应设置是安全的，而红色叉号则表示相应的项不安全。单击帮助可打开设置概述以及有关如何使其变得安全的信息。要更改某项设置，请单击“状态”列中相应的链接。根据具体的设置，会显示以下几项：

已启用/已禁用

单击此项可将设置状态切换为已启用或已禁用。

配置

单击此项可启动另一个 YaST 模块进行配置。退出该模块后，您会返回到“安全性概述”。

未知

未安装关联的服务时，相应设置的状态会设置为未知。此类设置不代表潜在的安全风险。



图 18.1 : YAST 安全和强化中心：安全概览

18.2 预定义安全配置

SUSE Linux Enterprise Server 包含三项预定义安全配置。这些配置会影响安全中心模块中提供的所有设置。在左侧窗格中单击预定义安全配置可查看预定义的配置。单击要应用的配置，模块随即会关闭。如果您要修改预定义设置，请重新打开安全中心模块，单击预定义安全配置，然后在右侧窗格中单击自定义设置。您做出的所有更改都将应用于所选的预定义配置。

工作站

使用任何网络连接类型（包括连接到因特网）的工作站的配置。

漫游设备

此设置适用于连接到不同网络的笔记本电脑或平板电脑。

网络服务器

适用于提供 Web 服务器、文件服务器、名称服务器等网络服务的计算机的安全性设置。此设置为预定义的设置提供最安全的配置。

自定义设置

选择自定义设置可以在应用三个预定义配置中的任何一个后对其进行修改。

18.3 口令设置

容易猜出的口令是一个重大的安全问题。可以通过口令设置对话框来确保只能使用安全的口令。

检查新口令

激活此选项后，如果新口令包含在某个字典中，或者口令是专有名词，系统将发出警告。

口令的最小可接受长度

如果用户所选口令的长度小于此处指定的值，系统将发出警告。

要记忆的口令数目

激活口令失效功能（通过口令有效期）后，此设置会储存给定数量的用户既往口令，以防止重复使用这些口令。

口令加密方法

选择一种口令加密算法。通常无需更改默认设置 (Blowfish)。

口令有效期

通过指定最小和最大时间限制（以天为单位）来激活口令失效功能。将最短有效期设置为大于 0 天的值可以防止用户立即更改其口令（这样做会绕过口令失效功能）。使用值 0 和 99999 可以停用口令失效功能。

口令失效前多少天发出警告

当口令即将失效时，用户会提前收到警告。指定应在失效日期前的多少天发出警告。

18.4 引导设置

在此对话框中配置哪些用户可以通过图形登录管理器关闭计算机。您还可以指定如何解释 **Ctrl - Alt - Del**，以及谁可以将系统休眠。

18.5 登录设置

此对话框可让您配置安全性相关的登录设置：

不正确登录尝试后的延迟

为了提高有人通过反复登录猜出用户口令的难度，建议在登录失败后延迟显示登录提示。请指定以秒为单位的值。确保错误键入口令的用户不需要等待太长时间。

允许远程图形登录

如果选中此项，则可以通过网络访问图形登录管理器 (GDM)。这会造成潜在的安全风险。

18.6 用户添加

设置用户与组 ID 的最小值和最大值。极少需要更改这些默认设置。

18.7 其他设置

此处列出了不属于上述类别的其他安全性设置：

文件权限

SUSE Linux Enterprise Server 随附了针对文件系统的三组预定义文件权限。这几组权限定义普通用户是否可以读取日志文件或启动特定的程序。容易文件权限适用于独立计算机。例如，这些设置允许普通用户读取大多数系统文件。有关完整配置，请参见 `/etc/permissions.easy` 文件。安全文件权限适用于可提供网络访问的多用户计算机。`/etc/permissions.secure` 中提供了这些设置的全面说明。非常安全设置是限制性最强的权限，请慎用。有关详细信息，请参见 `/etc/permissions.paranoid`。

启动 `updatedb` 的用户

`updatedb` 程序可扫描系统，并创建能够使用 `locate` 命令查询的所有文件的数据库。以 `nobody` 用户身份运行 `updatedb` 时，只会将全局可读文件添加到数据库。以 `root` 用户身份运行时，会添加几乎所有的文件（不允许 `root` 读取的文件除外）。

启用魔术 SysRq 键

魔术 SysRq 键是一个组合键，即使系统已崩溃，您也能借助它对系统进行一定程度的控制。<https://www.kernel.org/doc/html/latest/admin-guide/sysrq.html> 上提供了完整文档。

19 Linux 中的访问控制列表

可以将 POSIX ACL（访问控制列表）作为文件系统对象的传统权限概念的扩展来使用。相较于采用传统权限概念，利用 ACL 可以更灵活地定义权限。

POSIX ACL 这一术语表明它是一种真正的 POSIX（**可移植操作系统接口**）标准。由于多种原因，相应的标准草案 POSIX 1003.1e 和 POSIX 1003.2c 已被撤消。但是，在属于 Unix 系列的许多系统上使用的 ACL 都基于这两个草案，并且本章中介绍的文件系统 ACL 的实施也遵照这两个标准。

19.1 传统文件权限

SUSE Linux Enterprise Server 中包含的所有文件的权限都是精心选择的。在安装其他软件或文件期间，请在设置权限时格外小心。请始终在 **ls** 命令中使用 **-l** 选项，以立即检测出任何不正确的文件权限。错误的文件属性不仅意味着文件可能被更改或删除，修改的文件可能会由 **root** 执行，或者攻击者可能会通过修改配置文件来劫持服务。这会显著增加受到攻击的风险。

SUSE® Linux Enterprise Server 系统包含

`permissions`、`permissions.easy`、`permissions.secure` 和

`permissions.paranoid` 文件，它们全部位于 `/etc` 目录中。这些文件用于定义特殊权限，例如全局可写目录或针对文件的 `setuser ID` 位。设置了 `setuser ID` 位的程序不会使用启动它的用户的权限运行，而是使用文件拥有者（通常是 **root**）的权限运行。管理员可以使用 `/etc/permissions.local` 文件添加自己的设置。

要定义提供的其中一个配置文件，请在 YaST 的安全和用户部分选择本地安全。要了解该主题的详细信息，请阅读 `/etc/permissions` 中的注释，或查阅 **man chmod**。

可在 GNU Coreutils 信息页“节点**文件权限**” (**info coreutils "File permissions"**) 中找到有关传统文件权限的详细信息。更多高级功能有 `setuid`、`setgid` 和粘滞位。

19.1.1 **setuid** 位

在某些情况下，访问权限可能过于严格。因此，Linux 另有一些设置，允许为执行特定操作临时更改当前用户和组标识。例如，**passwd** 程序通常要求拥有根权限才能访问 `/etc/passwd`。此文件包含一些重要信息，如用户主目录及用户和组 ID。因此，普通用户将无法更改 **passwd**，因为授予所有用户直接访问此文件的权限太过危险。此问题的一种可行解决方法是使用 **setuid** 机制。setuid（设置的用户 ID）是一个特殊文件属性，它指示系统执行相应标记在特定用户 ID 下的程序。以 **passwd** 命令为例：

```
-rwsr-xr-x 1 root shadow 80036 2004-10-02 11:08 /usr/bin/passwd
```

您可以看见 **s**，它表示为用户许可设置了 setuid 位。通过设置 setuid 位，启动 **passwd** 命令的所有用户都以 `root` 身份执行该命令。

19.1.2 **setgid** 位

setuid 位适用于用户。而对组而言也有一个等价的属性：**setgid** 位。设置了此位的程序基于保存该程序的组 ID 运行，而不论是哪个用户启动了该程序。因此，在设置了 setgid 位的目录中，所有新建文件和子目录都被指派到该目录所属的组。请考虑下面的示例目录：

```
drwxrws--- 2 tux archive 48 Nov 19 17:12 backup
```

您可以看见 **s**，它表示为组许可设置了 setgid 位。目录的拥有者和 `archive` 组的成员可以访问此目录。不是该组成员的用户会“映射”到各自的组中。所有写入文件的有效组 ID 为 `archive`。例如，使用组 ID `archive` 运行的备份程序即便没有 root 特权也能访问此目录。

19.1.3 粘性位

另外还可以设置**粘滞位**。属于可执行程序粘滞位和属于目录粘滞位在作用上有所不同。如果属于某个程序，以这种方式标记的文件将被装入 RAM，而不必在每次使用时从硬盘读取。由于目前硬盘的速度已经足够快，此属性已经很少使用。如果为目录指派了此位，则可以防止用户删除彼此的文件。典型示例如 `/tmp` 目录和 `/var/tmp` 目录：

```
drwxrwxrwt 2 root root 1160 2002-11-19 17:15 /tmp
```

19.2 ACL 的优势

传统情况下，会为 Linux 系统上的每个文件对象定义三组权限。这三组权限包括用于每种类型用户（即文件拥有者、组和其它用户这三种用户）的读 (r)、写 (w) 和执行 (x) 权限。此外，还可以设置设置用户 ID、设置组 ID 和粘滞位。这种简缩概念完全适用于大多数实际情况。但对于较复杂的方案或高级应用程序，以前系统管理员需要采用多种变通方案来避开传统权限概念的限制。

可以将 ACL 作为传统文件权限概念的扩展来使用。它们可用于向单个用户或组指派权限，即使这些权限并不与原始拥有者或所属组相对应。访问控制列表是 Linux 内核的一项功能，目前受 Ext2、Ext3、Ext4、JFS 和 XFS 的支持。通过使用 ACL，无需在应用程序级别实施复杂的权限模型就可以实现复杂的方案。

如果您想用 Linux 服务器代替 Windows 服务器，则 ACL 的优势尤为明显。一些已连接的工作站即使在迁移后也仍可继续在 Windows 下运行。Linux 系统利用 Samba 向 Windows 客户端提供文件和打印服务。有了 Samba 支持访问控制列表，则既可以在 Linux 服务器上配置用户权限，也可以在具有图形用户界面的 Windows（仅限 Windows NT 和更高版本）中配置用户权限。利用 **winbindd**（Samba 套件的一部分），甚至可以向仅存在于 Windows 域中而在 Linux 服务器中没有任何帐户的用户指派权限。

19.3 定义

用户类别

传统的 POSIX 许可权限概念使用三类用户在文件系统中指派权限：拥有者、拥有的组和其他用户。可以为每个用户类别设置三个权限位，用于分配读 (r)、写 (w) 和执行 (x) 权限。

ACL

所有种类的文件系统对象（文件和目录）的用户和组访问权限均通过 ACL 来确定。

默认 ACL

默认 ACL 只能应用于目录。它们确定文件系统对象在创建时从其父目录继承的权限。

ACL 项

每个 ACL 都包含一组 ACL 项。ACL 项中包含一个类型、一个此项所关联的用户或组的限定符和一组权限。对于某些项类型，未定义组或用户的限定符。

19.4 处理 ACL

表 19.1 “ACL 项类型”总结了 ACL 项 6 种可能出现的类型，每种类型都定义了一个用户或一组用户的权限。**拥有者**项定义了拥有该文件或目录的用户的权限。**所属组**项定义了文件所属组的权限。超级用户可以使用 **chown** 或 **chgrp** 更改拥有者或所属组，而在这种情况下，拥有者和所属组项表示新的拥有者和所属组。每个**已命名用户**项定义了在该项的限定符字段中指定的用户的权限。每个**已命名组**项定义了在该项的限定符字段中指定的组的权限。只有已命名用户和已命名组项具有非空的限定符字段。**其他**项定义了所有其他用户的权限。

通过定义这些项中的有效权限和要屏蔽的权限，**掩码**项进一步限制了已命名用户、已命名组和所属组项授予的权限。如果权限同时存在于上述项之一和掩码中，它们就是有效的。仅包含在掩码或实际项中的权限是无效的，表示未授予这些权限。拥有者和所属组项中定义的所有权限始终有效。表 19.2 “屏蔽访问权限”中的示例说明了这种机制。

有两种基本的 ACL 类：一种是**最小** ACL，仅包含用于类型拥有者、所属组和其他的项，对应于文件和目录的传统权限位。另一种是**扩展** ACL，它比前一种要复杂得多。它必须包含一个掩码项，并可能包含若干已命名用户和已命名组类型的项。

表 19.1：ACL 项类型

类型	文本形式
拥有者	<u>user::rwx</u>
已命名用户	<u>user:name:rwx</u>
所属组	<u>group::rwx</u>
已命名组	<u>group:name:rwx</u>
掩码	<u>mask::rwx</u>
其他	<u>other::rwx</u>

表 19.2：屏蔽访问权限

项类型	文本形式	许可权限
已命名用户	<u>user:geeko:r-x</u>	<u>r-x</u>

项类型	文本形式	许可权限
掩码	<code>mask::rw-</code>	<code>rw-</code>
	有效权限:	<code>r--</code>

19.4.1 ACL 项和文件模式权限位

图 19.1 “最小 ACL：与权限位相比的 ACL 项”和图 19.2 “扩展 ACL：与权限位相比的 ACL 项”说明了最小 ACL 和扩展 ACL 这两种情况。这些图分为三块 — 左边一块显示 ACL 项的类型规范，中间一块显示一个示例 ACL，右边一块显示对应于传统权限概念的各个权限位（例如，如 `ls -l` 所显示）。在这两种情况下，**拥有者**权限均被映射到 ACL 拥有者项。**其他类别**权限也被映射到各自的 ACL 项。但是，**组类别**权限的映射在这两种情况中是不同的。

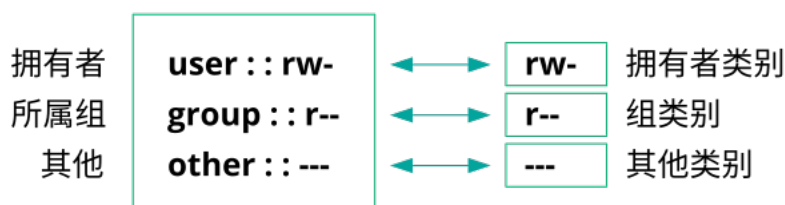


图 19.1：最小 ACL：与权限位相比的 ACL 项

对于最小 ACL（没有屏蔽），组类别许可权限被映射到 ACL 的所属组项。图 19.1 “最小 ACL：与权限位相比的 ACL 项”中显示了这一点。对于扩展 ACL（具有屏蔽），组类别许可权限被映射到屏蔽项。图 19.2 “扩展 ACL：与权限位相比的 ACL 项”中显示了这一点。

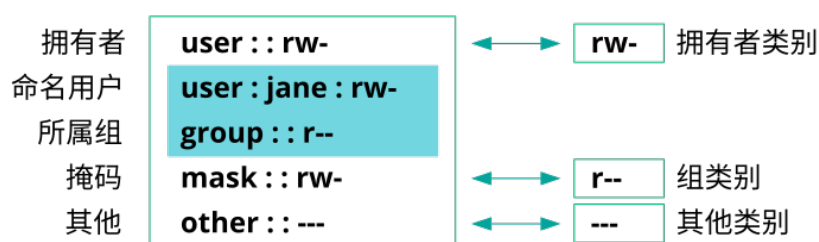


图 19.2：扩展 ACL：与权限位相比的 ACL 项

不管应用程序是否具有 ACL 支持，这种映射方式都可以确保应用程序的流畅交互。通过权限位方式分配的访问权限表示通过 ACL 所进行的所有其他“微调”的上限。对权限位的更改将由 ACL 反映出来，反之亦然。

19.4.2 具有 ACL 的目录

命令行上显示 **getfacl** 和 **setfacl** 的情况下，您可以访问 ACL。以下示例演示了这些命令的用法。

在创建目录之前，使用 **umask** 命令来定义每次创建文件对象时应屏蔽哪些访问权限。命令 **umask 027** 会设置以下默认权限：为拥有者授予全部权限 (0)、拒绝组的写入访问权限 (2)，以及不为其他用户提供权限 (7)。**umask** 实际上会屏蔽相应的许可权限位或将它们关闭。有关详细信息，请参考 **umask** 手册页。

mkdir mydir 创建具有由 **umask** 设置的默认权限的 **mydir** 目录。使用 **ls -dl mydir** 来检查是否已正确分配所有权限。该示例的输入为：

```
drwxr-x--- ... tux project3 ... mydir
```

使用 **getfacl mydir**，检查 ACL 的初始状态。这样会得出如下信息：

```
# file: mydir
# owner: tux
# group: project3
user::rwx
group::r-x
other::---
```

输出的前三行显示了目录的名称、拥有者和所属组。随后三行包含三个 ACL 项，即拥有者、所属组和其他。事实上，对于最小 ACL，**getfacl** 命令不会生成您使用 **ls** 所不能获得的任何信息。

使用以下命令修改 ACL，为附加用户 **geeko** 和附加组 **mascots** 指派读、写和执行权限：

```
root # setfacl -m user:geeko:rwx,group:mascots:rwx mydir
```

选项 **-m** 提示 **setfacl** 修改现有的 ACL。以下参数指示要修改的 ACL 项（各项之间用逗号隔开）。最后部分指定了应该对其应用这些修改的目录的名称。使用 **getfacl** 命令来查看所生成的 ACL。

```
# file: mydir
# owner: tux
```

```
# group: project3
user::rwx
user:geeko:rwx
group::r-x
group:mascots:rwx
mask::rwx
other:---
```

除了为用户 `geeko` 和组 `mascots` 创建的项外，还生成了一个掩码项。此掩码项是自动设置的，因此所有权限都是有效的。`setfacl` 自动使现有掩码项适应修改的设置，除非您使用 `-n` 停用此功能。该掩码项定义组类别中所有项的最大有效访问权限。其中包括已命名用户、已命名组和所属组。由 `ls -dl mydir` 显示的组类别权限位现在与 `mask` 项相对应。

```
drwxrwx---+ ... tux project3 ... mydir
```

输出的第一栏包含一个附加的 `+`，表明此项存在一个扩展 ACL。

根据 `ls` 命令的输出，掩码项的权限包含写访问权限。传统情况下，这样的权限位意味着所属组（这里是 `project3`）也具有对 `mydir` 目录的写访问权限。

但是，所属组的有效访问权限对应于为所属组和屏蔽定义的许可权限的重叠部分 — 在我们的示例中是 `r-x`（参见表 19.2 “屏蔽访问权限”）。对本例中的所属组的有效权限而言，即使是在添加了 ACL 项之后，也未发生任何改变。

用 `setfacl` 或 `chmod` 编辑掩码项。例如，使用 `chmod g-w mydir`。`ls -dl mydir` 即会显示：

```
drwxr-x---+ ... tux project3 ... mydir
```

`getfacl mydir` 提供以下输出：

```
# file: mydir
# owner: tux
# group: project3
user::rwx
user:geeko:rwx      # effective: r-x
group::r-x
group:mascots:rwx   # effective: r-x
mask::r-x
```

```
other:---
```

执行 **chmod** 将写入权限从组类别位中去除后，通过 **ls** 的输出就完全能够看出掩码位一定已相应更改：写入权限再次仅限 **mydir** 的拥有者拥有。**getfacl** 的输出证实了这一点。这个输出包含了对有效权限位与原始权限不对应的所有项的注释，因为已根据掩码项对它们进行了过滤。可以随时用 **chmod g+w mydir** 来恢复原始权限。

19.4.3 具有默认 ACL 的目录

目录可以具有默认 ACL，这是一种特殊的 ACL，它定义的是此目录下的对象在创建时继承的访问权限。默认 ACL 影响子目录和文件。

19.4.3.1 默认 ACL 的效果

将目录的默认 ACL 的权限传递到文件和子目录时，有两种方式：

- 子目录会继承父目录的默认 ACL 作为其默认 ACL 和 ACL。
- 文件会继承该默认 ACL 作为其 ACL。

创建文件系统对象的所有系统调用都使用 **mode** 参数，该参数定义新创建的文件系统对象的访问权限。如果父目录没有默认 ACL，则从 **mode** 参数传递的权限中去除 **umask** 定义的权限位，同时将结果分配到新对象。如果父目录存在默认 ACL，则分配到新对象的权限位对应于 **mode** 参数的权限和默认 ACL 中定义的权限的重叠部分。这种情况下忽略了 **umask**。

19.4.3.2 默认 ACL 的应用

以下三个示例说明了目录和默认 ACL 的主要操作：

1. 将默认 ACL 添加到现有目录 **mydir**，语句为：

```
tux > setfacl -d -m group:mascots:r-x mydir
```

setfacl 命令中的选项 **-d** 提示 **setfacl** 在默认 ACL 中执行以下修改（选项 **-m**）。

仔细查看此命令的结果：

```
tux > getfacl mydir

# file: mydir
# owner: tux
# group: project3
user::rwx
user:geeko:rwx
group::r-x
group:mascots:rwx
mask::rwx
other::---
default:user::rwx
default:group::r-x
default:group:mascots:r-x
default:mask::r-x
default:other::---
```

getfacl 会返回 ACL 和默认 ACL。默认 ACL 由以 `default` 开头的所有行组成。虽然您只是对 `mascots` 组的一个项执行了 **setfacl** 命令来创建默认 ACL，但为了创建有效的默认 ACL，**setfacl** 自动复制了 ACL 中的所有其他项。默认 ACL 对访问权限没有直接效果。它们只在创建文件系统对象时起作用。这些新对象只从其父目录的默认 ACL 中继承权限。

2. 在下一个示例中，我们将使用 **mkdir** 在 `mydir` 中创建一个子目录，它将继承默认 ACL。

```
tux > mkdir mydir/mysubdir

getfacl mydir/mysubdir

# file: mydir/mysubdir
# owner: tux
# group: project3
user::rwx
group::r-x
group:mascots:r-x
mask::r-x
```

```
other::---
default:user::rwx
default:group::r-x
default:group:mascots:r-x
default:mask::r-x
default:other::---
```

根据预期，新建的子目录 `mysubdir` 具有父目录的默认 ACL 的权限。`mysubdir` 的 ACL 准确反映了 `mydir` 的默认 ACL。该目录将向其从属对象传递的默认 ACL 也是相同的。

3. 使用 **touch** 在 `mydir` 目录中创建一个文件，例如 **touch mydir/myfile**。**ls -l mydir/myfile** 即会显示：

```
-rw-r-----+ ... tux project3 ... mydir/myfile
```

getfacl mydir/myfile 的输出是：

```
# file: mydir/myfile
# owner: tux
# group: project3
user::rw-
group::r-x          # effective:r--
group:mascots:r-x   # effective:r--
mask::r--
other::---
```

当创建新文件时，**touch** 使用值为 `0666` 的 `mode`，这意味所创建的新文件具有用于所有用户类别的读和写权限，前提是 **umask** 或默认 ACL 中不存在任何其他限制（请参见第 19.4.3.1 节“默认 ACL 的效果”）。实际上，这意味着 `mode` 值中不包含的所有访问权限均将从各自的 ACL 项中去除。虽然没有从组类别的 ACL 项中去除任何权限，但仍修改了掩码项来屏蔽不在 `mode` 中设置的权限。

这种方式确保应用程序（如编译器）与 ACL 的交互平稳流畅。您可以创建具有有限访问权限的文件，然后将其标记为可执行文件。**mask** 机制确保适当的用户和组可以在需要时执行它们。

19.4.4 ACL 检查算法

在为任何进程或应用程序授予访问受 ACL 保护的文件系统对象的权限之前，将应用检查算法。作为基本规则，按照以下序列检查 ACL 项：拥有者、命名用户、所属组或命名组以及其他组。访问将根据最适合进程的项进行处理。权限不能累加。

如果某个进程属于多个组并且潜在适合多个组项，情况会更为复杂。这时将从具有所需权限的合适项中随机选择一个。它与是哪些项触发了最终结果“已授权访问”无关。同样，如果没有任何适当组项包含所需的权限，则随机选择的项将触发最终结果“访问被拒绝”。

19.5 应用程序中的 ACL 支持

ACL 可用于实施非常复杂的权限方案以满足目前应用程序的要求。可以用一种智能方式将传统权限概念和 ACL 结合在一起。基本文件命令（**cp**、**mv**、**ls** 等）均支持 ACL，这与 Samba 和 Nautilus 相同。

Vi/Vim 和 emacs 都完全支持 ACL，它们会保留针对写入文件（包括备份）的权限。遗憾的是，许多编辑器和文件管理器仍缺少 ACL 支持。当用编辑器修改文件时，文件的 ACL 有时会被保留，有时则会丢失，这取决于所使用编辑器的备份方式。如果编辑器向原始文件写入更改，则会保留 ACL。如果编辑器将已更新的内容保存到一个新文件，然后将此文件重命名为旧文件名，则 ACL 可能会丢失，除非编辑器支持 ACL。除了 **star** 存档程序外，当前没有任何其他备份应用程序会保留 ACL。

19.6 更多信息

有关 ACL 的详细信息，请参见 **getfacl(1)**、**acl(5)** 和 **setfacl(1)** 的手册页。

20 证书存储区

证书在公司和个人身份验证中发挥着重要的作用。证书通常由应用程序本身管理。在某些情况下，在应用程序之间共享证书会给用户带来便利。证书存储区是 Firefox、Evolution 和 NetworkManager 的共同基础。本章将介绍一些相关细节。

目前，证书存储区是 Firefox、Evolution 和 NetworkManager 的公用数据库，未涵盖其他使用证书的应用程序，但将来可能会将它们纳入其中。如果您有此类应用程序，可以继续使用其私用的独立配置。

20.1 激活证书存储区

配置大部分都是在后台完成。要激活证书存储区，请执行以下操作：

1. 确定是要全局激活证书存储区（针对系统上的每个用户），还是专门针对特定的用户激活：

- 针对每个用户：使用 `/etc/profile.local` 文件
- 针对特定的用户：使用 `~/.profile` 文件

2. 打开上一步骤中所述的文件并插入如下一行：

```
export NSS_USE_SHARED_DB=1
```

保存文件

3. 注销然后登录到桌面。

所有证书都储存在 `$HOME/.local/var/pki/nssdb/` 下。

20.2 导入证书

要将证书导入证书存储区，请执行以下操作：

1. 启动 Firefox。

2. 选择编辑 > 首选项打开相应的对话框。切换到高级 > 加密，然后单击 查看证书。
3. 根据用户类型导入证书：使用服务器可以导入服务器证书，使用 人员可以标识其他人，使用您的证书可以标识您自己。

21 使用 AIDE 进行入侵检测

保护您的系统是任何一位任务关键型系统管理员必须完成的任务。由于无法始终保证系统的安全性不会受到损害，定期执行额外的检查（例如，使用 `cron` 进行检查）以确保系统仍受您的控制，就显得极为重要。这正是 AIDE（高级入侵检测环境）的用武之地。

21.1 为何要使用 AIDE?

可以通过 RPM 执行简单的检查，这往往可以发现一些不必要的更改。软件包管理器具有一项内置的校验功能，可以检查系统中所有受管文件发生的更改。要校验所有文件，请运行 `rpm -Va` 命令。不过，此命令还会显示配置文件中的更改，您需要进行过滤才能检测出重要的更改。

使用 RPM 进行检查的另一个问题在于，聪明的攻击者会修改 `rpm` 本身，以隐藏通过某种 root-kit 进行的任何更改，这样攻击者便可掩盖其入侵行为并获得 root 特权。要解决此问题，您应该实施另一项检查，这项检查也可以完全独立于安装的系统运行。

21.2 设置 AIDE 数据库



重要：安装后初始化 AIDE 数据库

在安装系统之前，请校验媒体的校验和（参见《部署指南》，第 12 章“查错”，第 12.1 节“检查媒体”），以确保您使用的不是受损安装源。安装系统后，初始化 AIDE 数据库。为确保在安装期间和之后一切正常，请在计算机未连到任何网络的情况下，直接在控制台上进行安装。在 AIDE 创建其数据库之前，请不要使计算机处于无人照管的状态或将其连接到任何网络。

SUSE Linux Enterprise Server 上默认不会安装 AIDE。要安装 AIDE，请使用计算机 > 安装软件，或者以 `root` 身份在命令行中输入 `zypper install aide`。

要告知 AIDE 应检查哪些文件的哪些属性，请使用 `/etc/aide.conf` 配置文件。此文件必须经过修改才能成为实际的配置。第一部分处理一般参数，例如 AIDE 数据库文件的位置。自定义规则以及目录和文件部分与本地配置更为相关。典型规则如下所示：

```
Binlib      = p+i+n+u+g+s+b+m+c+md5+sha1
```

定义 `Binlib` 变量后，将在文件部分使用相应的检查框。重要选项包括：

表 21.1：重要的 AIDE 检查框

选项	说明
p	检查选定文件或目录的文件权限。
i	检查 inode 编号。每个文件名都有一个不得更改的唯一 inode 编号。
n	检查指向相关文件的链接数。
u	检查文件拥有者是否已更改。
g	检查文件组是否已更改。
s	检查文件大小是否已更改。
b	检查文件使用的块计数是否已更改。
m	检查文件的修改时间是否已更改。
c	检查文件访问时间是否已更改。
S	检查更改的文件大小。
l	忽略文件名的更改。
md5	检查文件的 md5 校验和是否已更改。我们建议使用 sha256 或 sha512。

选项	说明
sha1	检查文件的 sha1（160 位）校验和是否已更改。我们建议使用 sha256 或 sha512。
sha256	检查文件的 sha256 校验和是否已更改。
sha512	检查文件的 sha512 校验和是否已更改。

此配置使用 [Binlib](#) 中定义的选项检查 [/sbin](#) 中的所有文件，但会忽略 [/sbin/conf.d/](#) 目录：

```
/sbin Binlib
!/sbin/conf.d
```

要创建 AIDE 数据库，请执行以下操作：

1. 打开 [/etc/aide.conf](#)。
2. 定义应使用哪些检查框检查哪些文件。有关可用检查框的完整列表，请参见 [/usr/share/doc/packages/aide/manual.html](#)。定义文件的选择需要掌握正则表达式方面的一些知识。保存修改内容。
3. 要检查配置文件是否有效，请运行：

```
root # aide --config-check
```

此命令的任何输出都是一条指出配置无效的提示。例如，如果您收到以下输出：

```
root # aide --config-check
35:syntax error:!!
35:Error while reading configuration:!!
Configuration error
```

该错误预期会在 [/etc/aide.conf](#) 的第 36 行中出现。请注意，错误消息包含上次成功读取的配置文件行。

4. 初始化 AIDE 数据库。运行以下命令：

```
root # aide -i
```

5. 将生成的数据库复制到某个保存位置（例如 CD-R、DVD-R、远程服务器或闪存盘），供以后使用。

! 重要

此步骤至关重要，因为它可以避免数据库的安全受到损害。建议使用只能写入一次的媒体，以防止数据库遭到修改。**切勿**将数据库保留在您要监视的计算机上。

21.3 本地 AIDE 检查

要进行文件系统检查，请执行以下操作：

1. 重命名数据库：

```
root # mv /var/lib/aide/aide.db.new /var/lib/aide/aide.db
```

2. 发生任何配置更改后，始终需要重新初始化 AIDE 数据库，随后移动新生成的数据库。备份此数据库也是个不错的选择。有关更多信息，请参见第 21.2 节“设置 AIDE 数据库”。
3. 使用以下命令执行检查：

```
root # aide --check
```

如果输出为空，则表示一切正常。如果 AIDE 发现了更改，会显示更改摘要，例如：

```
root # aide --check
AIDE found differences between database and filesystem!!

Summary:
Total number of files:      1992
Added files:                0
Removed files:              0
Changed files:              1
```

要了解实际的更改，请使用参数 `-V` 提高检查的详细级别。对于前面的示例，此参数的用法如下所示：

```
root # aide --check -V
AIDE found differences between database and filesystem!!
Start timestamp: 2009-02-18 15:14:10

Summary:
  Total number of files:      1992
  Added files:                0
  Removed files:              0
  Changed files:              1

-----
Changed files:
-----

changed: /etc/passwd

-----
Detailed information about changes:
-----

File: /etc/passwd
  Mtime    : 2009-02-18 15:11:02          , 2009-02-18 15:11:47
  Ctime    : 2009-02-18 15:11:02          , 2009-02-18 15:11:47
```

为了演示该结果，本示例中改动了文件 `/etc/passwd`。

21.4 独立于系统的检查

为了避免风险，建议同时从可信的来源运行 AIDE 二进制文件。这可以排除某些攻击者另外修改 AIDE 二进制文件以隐藏其行踪的风险。

要完成此任务，必须从独立于所安装系统的救援系统运行 AIDE。使用 SUSE Linux Enterprise Server 可以相对轻松地通过任意程序扩展救援系统，如此便可添加所需的功能。

在开始使用救援系统之前，需要将两个软件包提供给系统。包含这些软件包时所用的语法与将驱动程序更新磁盘添加到系统的语法相同。有关用于此目的的 `linuxrc` 可用功能的详细说明，请参见 <https://en.opensuse.org/SDB:Linuxrc>。下面介绍了一种完成此任务的可行方式。

过程 21.1：使用 AIDE 启动救援系统

1. 提供一台 FTP 服务器作为另一台计算机。
2. 将 `aide` 和 `mhash` 软件包复制到 FTP 服务器目录，在本例中为 `/srv/ftp/`。请将 `ARCH` 和 `VERSION` 占位符替换为相应的值：

```
root # cp DVD1/suse/ARCH/aideVERSION.ARCH.rpm /srv/ftp
root # cp DVD1/suse/ARCH/mhashVERSION.ARCH.rpm /srv/ftp
```

3. 创建信息文件 `/srv/ftp/info.txt`，用于提供救援系统所需的引导参数：

```
dud:ftp://ftp.example.com/aideVERSION.ARCH.rpm
dud:ftp://ftp.example.com/mhashVERSION.ARCH.rpm
```

请将您的 FTP 域名、`VERSION` 和 `ARCH` 替换为系统上使用的值。

4. 重新启动需要使用 DVD 中的救援系统完成整个 AIDE 检查的服务器。向引导参数添加以下字符串：

```
info=ftp://ftp.example.com/info.txt
```

此参数告知 `linuxrc` 还要读入 `info.txt` 文件中的所有信息。

救援系统引导后，AIDE 程序即可供使用。

21.5 更多信息

以下位置提供了有关 AIDE 的信息：

- AIDE 主页：<http://aide.sourceforge.net>
- 记录的模板配置 `/etc/aide.conf` 中。

- 安装 `aide` 软件包后 `/usr/share/doc/packages/aide` 下的多个文件中。
- <https://www.ipi.fi/mailman/listinfo/aide>  上的 AIDE 用户邮件列表中。

III 网络安全性

- 22 X Window 系统和 X 身份验证 178
- 23 SSH：安全网络操作 179
- 24 掩蔽和防火墙 191
- 25 配置 VPN 服务器 208
- 26 使用 XCA、X 证书和密钥管理器管理 PKI 221

22 X Window 系统和 X 身份验证

网络透明性是 Unix 系统最重要的特征之一。X（Unix 操作系统的窗口系统）能够鲜明地利用这一特性。使用 X 可以成功完成以下操作：登录到远程主机并启动一个图形程序，然后可以通过网络发送该程序，使其显示在您的计算机上。

如果需要使用 X 服务器远程显示 X 客户端，X 服务器应该防范有人未经授权访问它所管理的资源（显示内容）。更具体地说，必须给客户端指派特定权限。在 X 窗口系统中，有两种指派权限的方法，分别为基于主机的访问控制和基于 Cookie 的访问控制。前者依赖应该运行客户端的主机的 IP 地址。用于控制这种指派的程序为 **xhost**。**xhost** 将合法客户端的 IP 地址输入到属于 X 服务器的数据库中。不过，依赖 IP 地址进行身份验证不是十分安全。例如，如果有另一个用户也在发送客户端程序的主机上操作，该用户也可以访问 X 服务器 — 就像某人伪造了 IP 地址一样。由于存在这些缺点，在此不再详述这种身份验证方法，但您可以通过 **manxhost** 了解更多相关信息。

使用基于 Cookie 的访问控制时，将生成一个只有 X 服务器和合法用户才知道的字符串（类似于某种身份证）。登录时，此 Cookie 将储存在用户主目录中的 **.Xauthority** 文件内，可供想要使用 X 服务器来显示窗口的任何 X 客户端使用。用户可以使用工具 **xauth** 检查文件 **.Xauthority**。如果您重命名了 **.Xauthority** 或者在主目录中意外删除了该文件，将无法打开任何新窗口或 X 客户端。

SSH（安全外壳）可用于加密网络连接并以透明方式将其转发到 X 服务器。这也称为 X 转发。要实现 X 转发，需要在服务器端模拟 X 服务器，并在远程主机上为 shell 设置 DISPLAY 变量。有关 SSH 的更多详细信息，请参见 第 23 章 “SSH：安全网络操作”。



警告：X 转发可能不安全

如果您认为用于登录的计算机不是安全主机，请不要使用 X 转发。如果启用了 X 转发，攻击者可能会通过您的 SSH 连接进行身份验证。然后，攻击者可能会侵入您的 X 服务器，并读取您的键盘输入（举例而言）。

23 SSH：安全网络操作

在网络环境中，常常需要从远程位置访问主机。如果用户以纯文本形式发送用于身份验证的登录和口令字符串，攻击者可能会截获这些信息，并滥用它们来获取对该用户帐户的访问权限。这样，攻击者便可以打开该用户的所有文件，并可以利用非法帐户获取管理员或 `root` 访问权限，或侵入其他系统。过去常用 `telnet`、`rsh` 或 `rlogin` 建立远程连接，但这种方式不能采用加密形式或其他安全机制防止窃听。另外还存在其他几种不受保护的通讯通道，例如传统的 FTP 协议和某些远程复制程序（如 `rcp`）。

SSH 套件通过对身份验证字符串（通常由登录名和口令构成）及主机间交换的所有其他数据进行加密，能够提供必要的保护。使用 SSH，虽然第三方仍可以记录数据流，但内容是经过加密的，除非了解加密钥，否则无法将其还原为明文。这样，SSH 在不安全的网络（如因特网）上实现了安全通讯。SUSE Linux Enterprise Server 附带的 SSH 实现是 OpenSSH。

默认情况下，SUSE Linux Enterprise Server 会安装可提供 `ssh`、`scp` 和 `sftp` 命令的 OpenSSH 软件包。在默认配置中，要远程访问 SUSE Linux Enterprise Server 系统，只能使用 OpenSSH 实用程序来进行，并且仅当 `sshd` 正在运行且防火墙允许这种访问时才可以。

SUSE Linux Enterprise Server 上的 SSH 会使用加密硬件加速（如果可用）。因此，与不使用加密硬件相比，通过 SSH 连接传输大量数据的速度要快得多。另一个优势是，CPU 的负载也会大幅减少。

23.1 `ssh` — 安全外壳

使用 `ssh` 可以登录到远程系统并以交互方式工作。要以用户 `tux` 的身份登录到主机 `sun`，请输入以下命令之一：

```
tux > ssh tux@sun
tux > ssh -l tux sun
```

如果两台计算机上的用户名相同，您可以省略用户名。使用 `ssh sun` 便已足够。远程主机提示输入远程用户的口令。成功进行身份验证后，您可以通过远程命令行执行操作，或使用交互式应用程序（例如文本模式的 YaST）。

此外，**ssh** 可让您使用 **ssh HOST COMMAND** 在远程系统上运行非交互式命令。需要正确地将 **COMMAND** 括在引号中。可以像在本地外壳中一样串联多个命令。

```
tux > ssh root@sun "dmesg -T | tail -n 25"
tux > ssh root@sun "cat /etc/issue && uptime"
```

23.1.1 在远程主机上启动 X 应用程序

SSH 还简化了远程 X 应用程序的使用。如果您结合 **-X** 选项运行 **ssh**，远程计算机上会自动设置 **DISPLAY** 变量，而且所有 X 输出都将通过现有 SSH 连接导出到本地计算机。此外，未获授权的个人无法拦截远程启动的 X 应用程序。

23.1.2 代理转发

添加 **-A** 选项可将 **ssh-agent** 身份验证机制转移到下一台计算机。这样，您就可以在不同计算机上工作而无需输入口令，但前提是：已将公钥分发给目标主机并在其上正确保存。有关详细信息，请参考第 23.5.2 节“复制 SSH 密钥”。

默认设置中会停用此机制，但您可以在系统范围的配置文件 **/etc/ssh/sshd_config** 中设置 **AllowAgentForwarding yes** 随时将其永久激活。

23.2 scp — 安全复制

scp 可将文件复制到远程计算机或从中复制文件。如果 **jupiter** 上的用户名不同于 **sun** 上的用户名，请使用 **USER_NAME@host** 格式指定后者的用户名。如果应将文件复制到其他目录而不是远程用户的主目录，请以 **sun:DIRECTORY** 形式指定该目录。下列示例显示了如何将文件从本地计算机复制到远程计算机，以及反向复制。

```
tux > scp ~/MyLetter.tex tux@sun:/tmp ❶
tux > scp tux@sun:/tmp/MyLetter.tex ~ ❷
```

- ❶ 本地计算机到远程计算机

② 远程计算机到本地计算机



提示：-l 选项

在 **ssh** 命令中，可以使用 **-l** 选项指定远程用户（替代 `USER_NAME@host` 格式）。在 **scp** 中，**-l** 选项用于限制 **scp** 所使用的带宽。

输入正确的口令后，**scp** 将启动数据传输。它会显示复制的每个文件的进度条和剩余时间。使用 **-q** 选项可以隐藏所有输出。

scp 还提供了对整个目录的递归复制功能。命令

```
tux > scp -r src/ sun:backup/
```

会将目录 **src** 的全部内容（包括所有子目录）复制到主机 **sun** 上的 `~/backup` 目录中。如果此子目录不存在，系统会自动创建该子目录。

-p 选项告知 **scp** 不要更改文件的时戳。**-C** 将对传送数据进行压缩。这可以最大限度地减少要传输的数据量，但同时会增加两台计算机的处理器负担。

23.3 sftp — 安全文件传输

23.3.1 使用 sftp

如果您要将多个文件复制到其他位置或从中复制多个文件，使用 **sftp** 会很方便，它能够替代 **scp**。它会打开一个外壳，其中包含一组与普通 FTP 外壳类似的命令。在 **sftp** 提示符处键入 **help** 可获取可用命令的列表。**sftp** 手册页中提供了更多细节。

```
tux > sftp sun
Enter passphrase for key '/home/tux/.ssh/id_rsa':
Connected to sun.
sftp> help
Available commands:
bye                               Quit sftp
```

```
cd path                                Change remote directory to 'path'
[...]
```

23.3.2 设置文件上传权限

与使用普通的 FTP 服务器一样，用户不仅可以下载，而且可以使用 **put** 命令将文件上传到运行 SFTP 服务器的远程计算机。默认情况下，向远程主机上传文件时将使用与本地计算机上相同的权限。有两个选项可以自动更改这些权限：

设置 umask

umask 充当本地主机上原始文件的权限的过滤器。它还可以撤回权限：

表 23.1：UMASK

原始权限	umask	上传的权限
0666	0002	0664
0600	0002	0600
0775	0025	0750

要在 SFTP 服务器上应用 umask，请编辑文件 `/etc/ssh/sshd_config`。搜索以 `Subsystem sftp` 开头的行，并添加包含所需设置的 `-u` 参数，例如：

```
Subsystem sftp /usr/lib/ssh/sftp-server -u 0002
```

显式设置权限

显式设置权限会为通过 SFTP 上传的所有文件设置相同的权限。使用 `-u` 指定三位数模式，例如 `600`、`644` 或 `755`。如果同时指定 `-m` 和 `-u`，将忽略 `-u`。

要在 SFTP 服务器上为上传的文件应用显式权限，请编辑文件 `/etc/ssh/sshd_config`。搜索以 `Subsystem sftp` 开头的行，并添加包含所需设置的 `-m` 参数，例如：

```
Subsystem sftp /usr/lib/ssh/sftp-server -m 600
```

23.4 SSH 守护程序 (sshd)

要使用 SSH 客户端程序 **ssh** 和 **scp**，必须在后台运行一台服务器（SSH 守护程序）来监听 TCP/IP 端口 22 上的连接。首次启动该守护程序时将生成三个密钥对。每个密钥对由私用密钥和公共密钥组成。因此，此过程称为基于公共密钥。要保证通过 SSH 安全地通讯，必须限制只有系统管理员才能访问私钥文件。文件权限是在默认安装中相应设置的。只有在本地 SSH 守护程序才需要私钥，切勿将私钥提供给其他任何人。公钥组件（可通过扩展名 **.pub** 识别）将被发送到请求连接的客户端。所有用户都可以读取公钥组件。

连接请求是 SSH 客户端发出的。等待中的 SSH 守护程序将与请求方 SSH 客户端交换标识数据来比较协议和软件版本，以防止连接通过错误的端口。由于请求是由最初的 SSH 守护程序的子进程回复的，所以可以同时建立多个 SSH 连接。

对于 SSH 服务器和 SSH 客户端间的通讯，OpenSSH 支持使用版本 1 和版本 2 的 SSH 协议。默认情况下使用的是版本 2 的 SSH 协议。使用 **-1** 选项可以覆盖此默认设置，改为使用该协议的版本 1。

使用 SSH 版本 1 时，服务器将发送其公共主机密钥和服务器密钥，SSH 守护程序每小时就重新生成一次服务器密钥。这两个密钥都允许 SSH 客户端对自由选择的会话密钥加密（会话密钥会被发送到 SSH 服务器）。SSH 客户端还会通知服务器使用哪种加密方法（加密法）。版本 2 的 SSH 协议不需要服务器密钥。服务器端和客户端都使用基于 Diffie-Hellman 的算法来交换它们的密钥。

一定要使用私用主机密钥和服务器密钥对会话密钥解密，从公钥根本无法得出这些密钥。只有被联系的 SSH 守护程序能够使用其私用密钥解密会话密钥。使用 SSH 客户端的 **-v** 选项启用详细调试可以密切监测此初始连接阶段。



提示：查看 SSH 守护程序日志文件

要监测 **sshd** 的日志项，请使用以下命令：

```
tux > sudo journalctl -u sshd
```

23.4.1 维护 SSH 密钥

建议将储存在 `/etc/ssh/` 中的私钥和公钥备份到安全的外部位置。这样就可以检测密钥修改事件，或者在安装新系统后再次使用旧密钥。



提示：现有的 SSH 主机密钥

如果在已经装有 Linux 系统的计算机上安装 SUSE Linux Enterprise Server，安装例程会自动从现有安装导入最近访问的 SSH 主机密钥。

首次与远程主机建立安全连接时，客户端会在 `~/.ssh/known_hosts` 中储存所有公共主机密钥。这会防止各种中间人攻击 — 外部 SSH 服务器试图使用伪造名称和伪造 IP 地址侵入系统。如果 `~/.ssh/known_hosts` 未包含某个主机密钥，或是因未能提供正确的私用密钥致使服务器无法解密会话密钥时，就可以检测到此类攻击。

如果主机的公共密钥已更改（连接到此类服务器之前需要校验此情况），可以使用 `ssh-keygen -r HOSTNAME` 去除造成问题的密钥。

23.4.2 轮换主机密钥

从版本 6.8 开始，OpenSSH 随附了一个支持主机密钥轮换的协议扩展。如果您仍在使用弱密钥（例如 1024 位 RSA 密钥），更换密钥会有帮助。强烈建议更换此类密钥，改用 2048 位 DSA 密钥甚至更强的密钥。然后，客户端会使用“最佳的”主机密钥。



提示：重新启动 sshd

在服务器上安装新的主机密钥后，重新启动 sshd。

如果用户使用 `ssh` 发起连接，此协议扩展可向客户端告知服务器上的所有新主机密钥。然后，客户端上的软件会更新 `~/.ssh/known_hosts`，因而用户无需手动接受以前已知且可信的主机的新密钥。除了在此会话期间用于对主机进行身份验证的密钥以外，本地 `known_hosts` 文件还将包含远程主机的所有主机密钥。

在服务器管理员知道所有客户端已提取新密钥后，他们便可以去除旧密钥。该协议扩展还可确保从客户端的配置中去除已过时的密钥。密钥在发起 `ssh` 会话时去除。

有关更多信息，请参见：

- <http://blog.djm.net.au/2015/02/key-rotation-in-openssh-68.html>
- <http://heise.de/-2540907> („Endlich neue Schlüssel für SSH-Server “，仅提供德语版)

23.5 SSH 身份验证机制

最简单的身份验证方式是通过输入用户的口令来完成，就如同用户在本地登录一样。但记住远程计算机上多个用户的口令比较没有效率，而且这些口令将来有可能会更改。另一方面，在授予 `root` 访问权限时，管理员需要能够在不更改 `root` 口令的情况下快速撤消此类权限。

为了在不要求输入远程用户口令的情况下实现登录，SSH 将使用另一个密钥对，该密钥对需由用户生成。该密钥对由一个公共密钥 (`id_rsa.pub` 或 `id_dsa.pub`) 和一个私用密钥 (`id_rsa` 或 `id_dsa`) 组成。

要在不指定远程用户口令的情况下登录，“SSH 用户”的公共密钥必须位于 `~/.ssh/authorized_keys` 中。此方法还能确保远程用户获得完全控制权：添加密钥需要远程用户的口令，去除密钥会撤消远程登录权限。

为实现最大安全性，此类密钥应受通行口令的保护，每当您使用 `ssh`、`scp` 或 `sftp` 时，都需要输入此通行口令。与简单身份验证相反，此通行短语独立于远程用户，因此始终保持不变。

除了上述基于密钥的身份验证以外，SSH 还提供基于主机的身份验证。借助基于主机的身份验证，可信主机上的用户可以使用相同的用户名登录到启用了此功能的另一台主机。SUSE Linux Enterprise Server 设置为使用基于密钥的身份验证，有关在 SUSE Linux Enterprise Server 上设置基于主机的身份验证的内容不在本手册的范畴内。



注意：基于主机的身份验证的文件权限

如果要使用基于主机的身份验证，`/usr/lib/ssh/ssh-keysign` 中应该设置 `setuid` 位，但这不是 SUSE Linux Enterprise Server 中的默认设置。在这种情况下，请手动设置文件权限。应使用 `/etc/permissions.local` 来实现此目的，确保在对 `openssh` 进行安全更新后，能够保留 `setuid` 位。

23.5.1 生成 SSH 密钥

1. 要使用默认参数（RSA，2048 位）生成密钥，请输入 **ssh-keygen** 命令。
2. 按 **Enter** 接受密钥的默认储存位置 ~/.ssh/id_rsa（强烈建议），或输入其他位置。
3. 输入包含 10 到 30 个字符的通行口令。有关创建安全口令的规则在此同样适用。强烈建议不要省略指定通行口令的步骤。

应务必确保除您自己以外的任何人都不能访问私用密钥（始终将其权限设置为 0600）。私用密钥绝对不能落入其他人手中。

要更改现有密钥对的口令，请使用 **ssh-keygen -p** 命令。

23.5.2 复制 SSH 密钥

要将 SSH 公共密钥复制到远程计算机上用户的 ~/.ssh/authorized_keys，请使用 **ssh-copy-id** 命令。要复制 ~/.ssh/id_rsa.pub 下储存的您的个人密钥，可以使用简写格式。要复制 DSA 密钥或其他用户的密钥，需要指定路径：

```
tux > ~/.ssh/id_rsa.pub
ssh-copy-id -i tux@sun

tux > ~/.ssh/id_dsa.pub
ssh-copy-id -i ~/.ssh/id_dsa.pub tux@sun

tux > ~notme/.ssh/id_rsa.pub
ssh-copy-id -i ~notme/.ssh/id_rsa.pub tux@sun
```

要成功复制密钥，需要输入远程用户的口令。要去除现有密钥，请手动编辑 ~/.ssh/authorized_keys。

23.5.3 使用 **ssh-agent**

执行大量的安全外壳操作时，为每个此类操作键入 SSH 通行口令会很麻烦。因此，SSH 软件包提供了另一个工具 **ssh-agent**，用于在 X 会话或终端会话期间保留私用密钥。所有其他窗口或程序以 **ssh-agent** 客户端的形式启动。启动代理时，会设置一组环境变量，**ssh**、**scp** 或 **sftp** 将使用这些变量来查找用于自动登录的代理。有关细节，请参见 **ssh-agent** 手册页。

ssh-agent 启动后，您需要使用 **ssh-add** 添加自己的密钥。它会提示您输入通行口令。提供一次口令后，您便可在运行中的会话内部使用安全外壳命令，而无需再次进行身份验证。

23.5.3.1 在 X 会话中使用 **ssh-agent**

在 SUSE Linux Enterprise Server 上，**ssh-agent** 会由 GNOME 显示管理器自动启动。要在 X 会话开始时同时调用 **ssh-add** 向代理添加您的密钥，请执行以下操作：

1. 以所需用户的身份登录，并检查文件 `~/.xinitrc` 是否存在。
2. 如果不存在，请使用现有模板，或从 `/etc/skel` 复制该文件：

```
if [ -f ~/.xinitrc.template ]; then mv ~/.xinitrc.template ~/.xinitrc; \
else cp /etc/skel/.xinitrc.template ~/.xinitrc; fi
```

3. 如果您复制了该模板，请搜索以下几行并将其取消注释。如果 `~/.xinitrc` 已存在，请添加以下几行（不带注释符号）。

```
# if test -S "$SSH_AUTH_SOCK" -a -x "$SSH_ASKPASS"; then
#     ssh-add < /dev/null
# fi
```

4. 启动新的 X 会话时，系统会提示您输入 SSH 通行口令。

23.5.3.2 在终端会话中使用 **ssh-agent**

在终端会话中，您需要手动启动 **ssh-agent**，然后调用 **ssh-add**。可通过两种方式启动代理。下面的第一个示例在现有外壳之上启动新的 Bash 外壳。第二个示例在现有外壳中启动代理，并按需修改环境。

```
tux > ssh-agent -s /bin/bash
eval $(ssh-agent)
```

代理启动后，运行 **ssh-add** 以向代理提供您的密钥。

23.6 端口转发

ssh 还可用于重定向 TCP/IP 连接。此功能也称为 **SSH 隧道**，它通过加密的通道将定向到特定端口的 TCP 连接重定向到另一台计算机。

使用以下命令可将定向到 jupiter 端口 25 (SMTP) 的所有连接重定向到 sun 上的 SMTP 端口。如果用户所用的 SMTP 服务器不具备 SMTP-AUTH 或 POP-before-SMTP 功能，此命令特别有用。与网络相连的任意位置都可以将电子邮件传送到“家庭”邮件服务器进行递送。

```
root # ssh -L 25:sun:25 jupiter
```

同样，使用以下命令可将 jupiter 上的所有 POP3 请求（端口 110）转发到 sun 的 POP3 端口：

```
root # ssh -L 110:sun:110 jupiter
```

必须以 **root** 身份执行这两个命令，因为连接指向有特权的本地端口。普通用户通过现有 SSH 连接发送和检索电子邮件。为此，必须将 SMTP 和 POP3 主机设置为 **localhost**。上述每个程序的手册页以及 **/usr/share/doc/packages/openssh** 下的 OpenSSH 软件包文档中提供了更多信息。

23.7 在安装的系统上添加和去除公共密钥

在某些环境中，通过 SSH 登录会比较方便或者有此必要。在此情况下，用户需要提供 SSH 公共密钥。要添加或去除 SSH 密钥，请执行以下操作：

1. 打开 YaST。
2. 在安全和用户下，打开用户和组管理模块。
3. 选择您要更改的用户并按编辑。

4. 切换到 SSH 公共密钥选项卡。
5. 添加或去除您的公共密钥。如果您添加了 SSH 公共密钥，请检查文件扩展名 .pub。
6. 单击确定进行确认。

SSH 公共密钥保存在 ~/.ssh/authorized_keys 中。

23.8 更多信息

<https://www.openssh.com> ↗

OpenSSH 主页

<https://en.wikibooks.org/wiki/OpenSSH> ↗

OpenSSH Wikibook

man sshd

OpenSSH 守护程序的手册页

man ssh_config

OpenSSH SSH 客户端配置文件的手册页

man scp,

man sftp,

man slogin,

man ssh,

man ssh-add,

man ssh-agent,

man ssh-copy-id,

man ssh-keyconvert,

man ssh-keygen,

man ssh-keyscan

用于安全复制文件（scp、sftp）、用于登录（slogin、ssh）和用于管理密钥的多个二进制文件的手册页。

/usr/share/doc/packages/openssh/README.SUSE ,
/usr/share/doc/packages/openssh/README.FIPS

特定于 SUSE 软件包的文档；上游相关默认设置的更改、有关 FIPS 模式的说明，等等。

24 掩蔽和防火墙

只要在网络环境中使用 Linux，您就可以利用内核功能通过操纵网络包将内部网络区域和外部网络区域隔开。Linux `netfilter` 框架提供了一种建立有效防火墙的方法，可以将不同网络隔开。使用 `iptables`（用于定义规则集的通用表结构）可以精确控制哪些包能通过网络接口。可以使用 `firewalld` 及其图形界面 `firewall-config` 设置此类包过滤器。

SUSE Linux Enterprise Server 15 GA 引入了 `firewalld` 作为新的默认软件防火墙，以其取代了 `SuSEfirewall2`。`SuSEfirewall2` 尚未从 SUSE Linux Enterprise Server 15 GA 中去除，仍是主储存库的一部分，不过默认不会安装它。本章为已从旧版 SUSE Linux Enterprise Server 升级的用户提供有关配置 `firewalld` 以及从 `SuSEfirewall2` 进行迁移的指导。

24.1 使用 iptables 过滤包

本节介绍包过滤的具体细节。`netfilter` 和 `iptables` 组件负责过滤和操纵网络包以及进行网络地址转换 (NAT)。过滤准则及与过滤准则关联的所有操作均储存在链中；各个网络包在到达时，必须依次与这些链进行匹配。要匹配的链储存在表中。使用 `iptables` 命令可以更改这些表和规则集。

Linux 内核维护以下三个表，分别对应包过滤器的不同功能：

filter

此表储存大多数过滤规则，因为它执行严格意义上的**包过滤**机制，例如，决定是让包通过 (ACCEPT) 还是将包丢弃 (DROP)。

nat

此表定义对包的源地址和目标地址所做的任何更改。使用这些功能还能实现**伪装**，这是 NAT 的一个特例，用于将专用网络与因特网链接起来。

mangle

此表中的规则用于操纵 IP 报头中储存的值（如服务类型）。

这些表包含多个用于匹配包的预定义链：

PREROUTING

此链适用于所有传入包。

INPUT

此链适用于发往系统内部进程的包。

FORWARD

此链适用于仅在系统中路由的包。

OUTPUT

此链适用于从系统自身发出的包。

POSTROUTING

此链适用于所有出站包。

图 24.1 “iptables：包的可能路径”演示了网络包在特定系统中传送时可能经过的路径。为了便于说明，图中将表作为链的各个部分列出，但实际上表本身储存了这些链。

最简单的情况是，发往系统本身的传入包抵达 `eth0` 接口。数据包首先要转到 `mangle` 表的 `PREROUTING` 链，然后转到 `nat` 表的 `PREROUTING` 链。随后的步骤（涉及包的路由选择）确定包的最终目标，这是系统自身的过程。在包经过 `mangle` 和 `filter` 表的 `INPUT` 链后，只要 `filter` 表的规则允许，那么包最终将抵达目标。

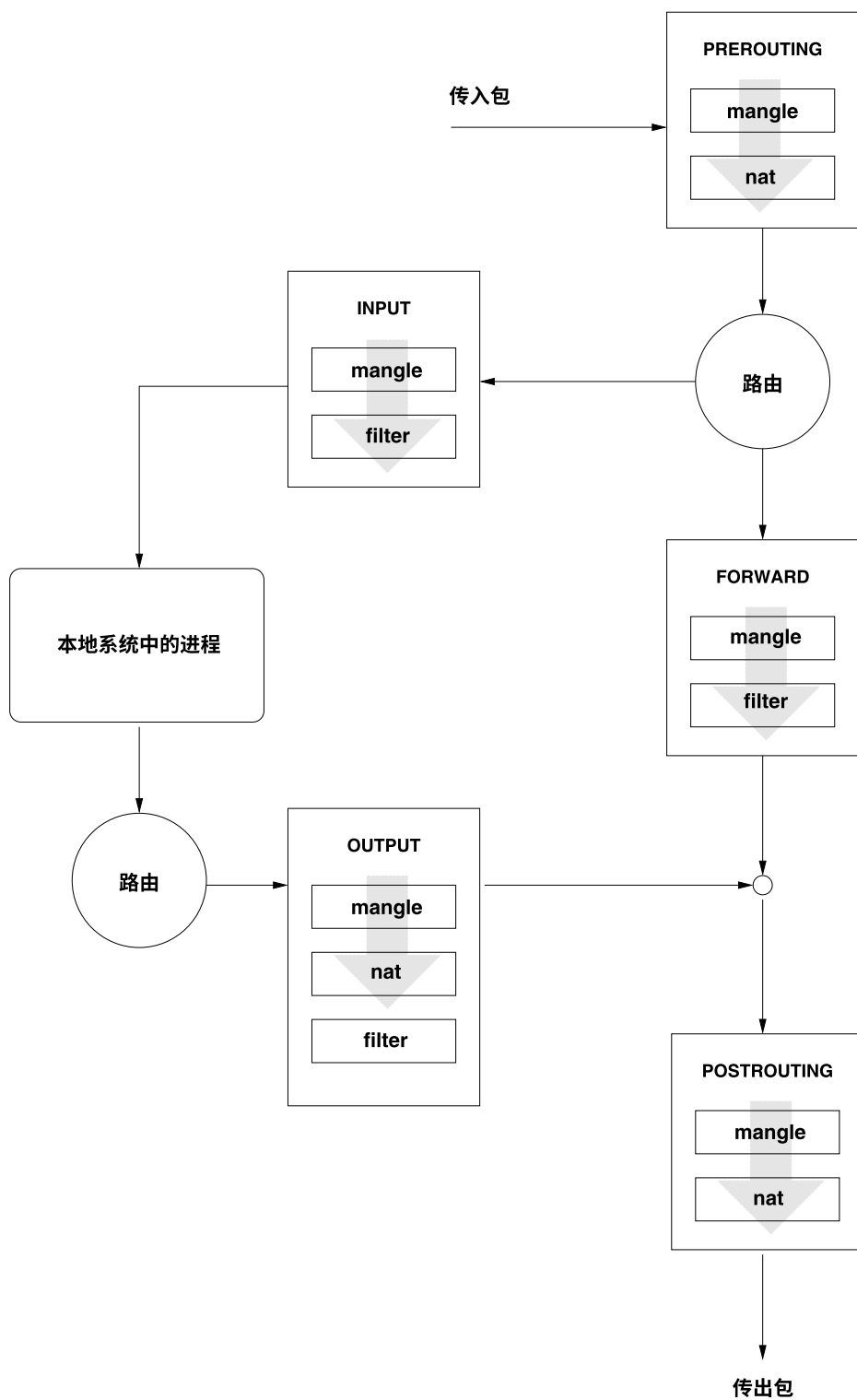


图 24.1 : IPTABLE: 包的可能路径

24.2 关于掩蔽的基础知识

掩蔽是 Linux 专用的 NAT（网络地址转换）形式，可用于将小型 LAN 连接到因特网。LAN 主机使用私用地址范围内的 IP 地址（请参见《管理指南》，第 19 章“基本网络知识”，第 19.1.2 节“网络掩码和路由”），而在因特网上，使用的是正式 IP 地址。要能够连接到因特网，LAN 主机的私用地址需转换为正式地址。这种转换是在路由器上完成的，路由器充当了 LAN 和因特网之间的网关。其中的原理只有简单的一条：路由器有多个网络接口，通常是一个网卡和与因特网连接的另一个接口。后者将路由器与外部世界链接起来，同时，还会有一个或多个其他网络接口将路由器与 LAN 主机链接起来。在本地网络中的这些主机连接到路由器的网卡（如 `eth0`）后，它们就可以将发往本地网络之外的所有包发送到其默认网关或路由器。

！ 重要：使用正确的网络掩码

在配置网络时，确保所有本地主机的广播地址和网络掩码都相同。做不到这一点就会导致无法正确路由数据包。

如上所述，只要有某台 LAN 主机要向因特网地址发送包，这个包就会发送到默认路由器。但是，必须先配置路由器，然后才能转发这些包。由于安全原因，默认安装中未启用它。要启用它，请在 `/etc/sysctl.conf` 文件中添加 `net.ipv4.ip_forward = 1` 一行。或者，您可以通过 YaST 实现此目的，例如，调用 **yast routing ip-forwarding on**。

连接的目标主机可以看到路由器，但对内部网络中发出包的那台主机却毫不知情。伪装技术就是因此而得名的。由于要进行地址转换，路由器自然成为所有回复包首先到达的目标。路由器必须能够识别这些入站包并转换其目标地址，这样才能将包转发给本地网络中的正确主机。

由于入站通讯数据的路由选择取决于伪装表，所以从外部根本无法打开与内部主机的连接。对于这种连接，伪装表中不会有任何对应项。此外，所有已建立的连接在该表中都被指派了一个状态项，所以其他连接无法再使用该项。

受以上各种因素影响，在使用多个应用程序协议，如 ICQ、cucme、IRC（DCC、CTCP）和 FTP（采用 PORT 模式）时，您可能会遇到一些问题。Web 浏览器、标准 FTP 程序和许多其他程序都使用 PASV 方式。就包过滤和伪装而言，这种被动方式不容易出问题。

24.3 防火墙基础知识

在描述用于控制网络间数据流的机制时，**防火墙**也许是用得最广泛的一个术语。严格地说，本节所述的机制应该叫做**包过滤器**。包过滤器根据特定准则（如协议、端口和 IP 地址）来控制数据流。这样您就可以根据包的地址来拦截不应该发送到您网络中的包。举例来说，若允许对 Web 服务器进行公共访问，应明确打开相应的端口。不过，包过滤器并不扫描有合法地址的包的内容（例如那些要发送到该 Web 服务器的包）。例如，即使是在入站包想要破坏 Web 服务器上的 CGI 程序的情况下，包过滤器仍然允许它们通过。

一种更有效但同时也更复杂的机制是将多种系统结合起来使用，例如让包过滤器与应用程序网关或代理进行交互。在这种情况下，包过滤器将拒绝所有发往禁用端口的包，而只接受发往应用程序网关的包。此网关或代理伪装成服务器的实际客户端。从某种意义上说，可以将这种代理视为应用程序使用的协议级的伪装主机。此类代理的一个示例就是 Squid（一种 HTTP 和 FTP 代理服务器）。要使用 Squid，必须将浏览器配置为通过代理通讯。代理缓存将提供请求的任何 HTTP 页面或 FTP 文件，在缓存中找不到的对象将由代理从因特网提取。

下一节重点介绍 SUSE Linux Enterprise Server 随附的包过滤器。有关包过滤和防火墙设置的更多信息，请阅读 [Firewall HOWTO（防火墙操作指南）](http://www.tldp.org/HOWTO/Firewall-HOWTO.html) (<http://www.tldp.org/HOWTO/Firewall-HOWTO.html>) 。

24.4 firewalld



注意：firewalld 取代了 SuSEfirewall2

SUSE Linux Enterprise Server 15 GA 引入了 `firewalld` 作为新的默认软件防火墙，以其取代了 `SuSEfirewall2`。`SuSEfirewall2` 尚未从 SUSE Linux Enterprise Server 15 GA 中去除，仍是主储存库的一部分，不过默认不会安装它。如果您是从早于 SUSE Linux Enterprise Server 15 GA 的版本升级，`SuSEfirewall2` 将不会有变化，并且您必须手动升级到 `firewalld`（请参见第 24.5 节“[从 SuSEfirewall2 迁移](#)”）。

`firewalld` 是一个守护程序，它可维护系统的 `iptables` 规则，并提供一个 D-Bus 接口用于操作这些规则。它随附了命令行实用程序 `firewall-cmd` 以及图形用户界面 `firewall-config` 与其交互。由于 `firewalld` 在后台运行并提供明确定义的接口，因此它允许其他应用程序请求对 `iptables` 规则进行更改，例如，设置虚拟机网络。

`firewalld` 实施了不同的安全区域。它提供了内部和公共等多个预定义区域。管理员可根据需要定义其他自定义区域。每个区域包含自身的 `iptables` 规则集。每个网络接口只能是一个区域的成员。也可以根据源地址将单个连接指派到某个区域。

每个区域代表一个特定的信任级别。例如，公共区域不受信任，因为此网络中的其他计算机不受您的控制（适合因特网或无线热点连接）。另一方面，内部区域用于受您控制的网络，类似于家庭或公司网络。以这种方式利用区域，主机能够以定义的方式向可信网络和不可信网络提供不同种类的服务。

有关 `firewalld` 中的预定义区域及其含义的详细信息，请参见其手册页：<http://www.firewalld.org/documentation/zone/predefined-zones.html>。



注意：不指派区域的行为

网络接口的初始状态是完全未指派到任何区域。在此情况下，将在默认区域（可通过调用 `firewall-cmd --get-default-zone` 来确定）中隐式处理网络接口。如果未配置为其他值，默认区域是公共区域。

`firewalld` 包过滤模型允许任何传出连接通过。传出连接是指由本地主机主动建立的连接。如果相关区域中不允许相应的服务，则会阻止远程主机建立的传入连接。因此，具有传入流量的每个接口必须放在适当的区域，以使所需的服务可供访问。对于每个区域，请定义所需的服务或协议。

`firewalld` 的一个重要概念是划分了两个不同的配置：**运行时配置**和**永久配置**。运行时配置代表当前处于活动状态的规则，而永久配置代表重新启动 `firewalld` 时将应用的已保存规则。这样，就可以添加在重新启动 `firewalld` 后将丢弃的临时规则，并且在试验新规则时能够还原到原始状态。当您更改配置时，需要知道您正在编辑哪个配置。第 24.4.2.2 节“运行时配置与永久配置”中介绍了如何做到这一点。

要使用图形用户界面 **firewall-config** 执行 **firewalld** 配置，请参见其文档 (<http://www.firewalld.org/documentation/utilities/firewall-config.html>) 。下一节将介绍如何在命令行上使用 **firewall-cmd** 执行典型的 **firewalld** 配置任务。

24.4.1 使用 NetworkManager 配置防火墙

NetworkManager 支持通过选择区域来对 **firewalld** 进行基本配置。

编辑有线或无线连接时，请在配置窗口中转到**身份**选项卡，然后使用**防火墙区域**下拉框。

24.4.2 在命令行上配置防火墙

24.4.2.1 防火墙启动

系统默认会安装并启用 **firewalld**。它是一个普通的 **systemd** 服务，可以通过 **systemctl** 或 YaST 服务管理器进行配置。

重要：自动配置防火墙

安装后，YaST 会自动启动 **firewalld**，并将所有接口保留在默认的**公共**区域中。如果在系统上配置并激活了某个服务器应用程序，YaST 可通过服务器配置模块中的在防火墙中打开所选接口上的端口或在防火墙中打开端口选项调整防火墙规则。某些服务器模块对话框包含防火墙细节按钮，用于激活其它服务和端口。

24.4.2.2 运行时配置与永久配置

默认情况下，所有 **firewall-cmd** 命令将对运行时配置运行。您可以通过添加 **--permanent** 参数来**仅**对永久配置应用大多数操作。如果这样做，更改将只会影响永久配置，而不会在运行时配置中立即生效。目前无法通过单次调用将规则同时添加到运行时配置和永久配置。要实现此目的，可将所有必要更改应用到运行时配置，并在一切符合预期时发出以下命令：

```
root # firewall-cmd --runtime-to-permanent
```

这会将所有当前运行时规则写入永久配置。您或其他程序在其他环境中可能对防火墙所做的任何临时修改都将以这种方式变成永久修改。如果您不确信这一点，保险起见，您也可以采取相反的方法：将新规则添加到永久配置，然后重新装载 `firewalld` 以使这些规则成为活动规则。



注意

某些配置项（例如默认区域）由运行时配置和永久配置共享。对这些项的更改会立即在这两个配置中反映出来。

要将运行时配置还原为永久配置并从而丢弃所有临时更改，可以采用以下两种做法：通过 `firewalld` 命令行界面或通过 `systemd`：

```
root # firewall-cmd --reload
```

```
root # systemctl reload firewalld
```

为简洁起见，下列章节中的示例始终对运行时配置运行（如果适用）。要使其适用于永久配置，请进行相应调整。

24.4.2.3 将接口指派到区域

您可按如下所示列出当前指派到某个区域的所有网络接口：

```
root # firewall-cmd --zone=public --list-interfaces
eth0
```

同样，您可以查询特定的接口指派到了哪个区域：

```
root # firewall-cmd --get-zone-of-interface=eth0
public
```

以下命令行将一个接口指派到某个区域。仅当 `eth0` 尚未指派到其他区域时，使用 `--add-interface` 的变体才起作用。使用 `--change-interface` 的变体始终起作用，在必要时会从其当前区域中去除 `eth0`：

```
root # firewall-cmd --zone=internal --add-interface=eth0
root # firewall-cmd --zone=internal --change-interface=eth0
```

任何不带显式 `--zone` 参数的操作将对默认区域隐式运行。此命令对可用于获取和设置默认的区域指派：

```
root # firewall-cmd --get-default-zone
dmz
root # firewall-cmd --set-default-zone=public
```

! 重要

未显式指派到区域的任何网络接口将自动成为默认区域的一部分。更改默认区域会立即为永久配置和运行时配置重新指派所有这些网络接口。切勿使用内部区域这样的可信区域作为默认区域，以免意外暴露于威胁之中。例如，在这种情况下，USB 以太网接口等热插拔网络接口将自动成为可信区域的一部分。

另请注意，不显式属于任何区域的接口不会显示在区域接口列表中。目前没有任何命令可列出未指派的接口。因此，在常规操作期间，最好避免使用未指派的网络接口。

24.4.2.4 使网络服务可供访问

`firewalld` 存在**服务**的概念。服务由端口和协议的定义构成。在给定网络服务（例如 Web 或邮件服务器协议）的环境中，这些定义在逻辑上合为一体。您可以使用以下命令获取有关预定义服务的信息及其细节：

```
root # firewall-cmd --get-services
[...] dhcp dhcpv6 dhcpv6-client dns docker-registry [...]
root # firewall-cmd --info-service dhcp
dhcp
  ports: 67/udp
  protocols:
  source-ports:
  modules:
  destination:
```

使用这些服务定义可以轻松使相关的网络功能在区域中可供访问。例如，下面的命令行将打开内部区域中的 HTTP Web 服务器端口：

```
root # firewall-cmd --add-service=http --zone=internal
```

要去除区域中的服务，则需使用对应的命令 `--remove-service`。您还可以使用 `--new-service` 子命令定义自定义的服务。有关如何执行此操作的更多细节，请参见 <http://www.firewalld.org/documentation/howto/add-a-service.html>。

如果您只想按编号打开单个端口，可使用以下方法。这会打开内部区域中的 TCP 端口 8000：

```
root # firewall-cmd --add-port=8000/tcp --zone=internal
```

要去除端口，请使用对应的命令 `--remove-port`。



提示：临时打开服务或端口

`firewalld` 支持使用 `--timeout` 参数将服务或端口打开一段有限的时间。在进行快速测试时，此参数可能很有用，它可以确保测试人员不会忘记关闭该服务或端口。要允许打开内部区域中的 `imap` 服务 5 分钟，可以调用

```
root # firewall-cmd --add-service=imap --zone=internal --timeout=5m
```

24.4.2.5 锁定模式

`firewalld` 提供了一种**锁定模式**来防止对处于活动状态的防火墙规则进行更改。由于应用程序可以通过 D-Bus 接口自动更改防火墙规则，并且普通用户也有可能可以执行该操作（取决于 PolicyKit 规则），因此，锁定在某些情况下有助于防止此类更改发生。<https://fedoraproject.org/wiki/Features/FirewalldLockdown> 上提供了有关此模式的详细信息。请务必注意，锁定模式功能不提供真正的安全性，而只是防范有人意外或者无恶意地尝试更改防火墙。目前，在 `firewalld` 中实施锁定模式无法针对恶意企图提供安全保护，<http://seclists.org/oss-sec/2017/q3/139> 上已指出了这一点。

24.4.2.6 添加自定义 `iptables` 规则

`firewalld` 声明对主机的 `netfilter` 规则拥有排它控制权。切勿使用其他工具（例如 `iptables`）修改防火墙规则。否则可能会造成 `firewalld` 的混乱并破坏安全性或功能。

如果您需要添加 `firewalld` 功能未涵盖的自定义防火墙规则，可通过两种方式实现目的。要直接传递原始 `iptables` 语法，可以使用 `--direct` 选项。此选项预期使用表、链和优先级作为初始参数，命令行的其余部分将按原样传递给 `iptables`。下面的示例将添加一个用于转发过滤器表的连接跟踪规则：

```
root # firewall-cmd --direct --add-rule ipv4 filter FORWARD 0 -i eth0 -o eth1 \
      -p tcp --dport 80 -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
```

此外，`firewalld` 实施了所谓的**富规则**，这是一种扩展语法，用于更轻松地指定 `iptables` 规则。<http://www.firewalld.org/documentation/man-pages/firewalld.richlanguage.html> 上提供了相应语法规范。下面的示例将丢弃来自特定源地址的所有 IPv4 包：

```
root # firewall-cmd --zone=public --add-rich-rule='rule family="ipv4" \
      source address="192.168.2.4" drop'
```

24.4.2.7 路由、转发和掩蔽

`firewalld` 并不是设计用来作为功能完备的路由器。它可提供典型家庭路由器设置的基本功能，但对于公司生产环境的路由器，则不应使用 `firewalld`，而应使用专用路由器和防火墙设备。下面仅提供了利用 `firewalld` 中的路由时需要考虑的几点提示：

- 首先，需要按第 24.2 节 “关于掩蔽的基础知识” 中所述启用 IP 转发。
- 要启用 IPv4 掩蔽（例如，在内部区域中），请发出以下命令。

```
root # firewall-cmd --zone=internal --add-masquerade
```

- `firewalld` 还可以启用端口转发。以下命令将端口 80 上的本地 TCP 连接转发到另一主机：

```
root # firewall-cmd --zone=public \
      --add-forward-port=port=80:proto=tcp:toport=80:toaddr=192.168.1.10
```

24.4.3 访问监听动态端口的服务

某些网络服务不会监听预定义的端口号，而是基于 `portmapper` 或 `rpcbind` 协议运行。从现在开始，我们将使用术语 `rpcbind`。当其中一项服务启动时，它会选择一个随机本地端口，并与 `rpcbind` 通讯以使端口号变为已知。`rpcbind` 本身正在监听已知的端口。然后，远程系统可以向 `rpcbind` 查询该协议已知的网络服务，以及它们正在监听哪些端口。现今还在使用此方法的程序不是很多。常见示例包括网络信息服务（NIS；`ypserv` 和 `ypbind`）以及网络文件系统（NFS）版本 3。



注意：关于 NFSv4

较新的 NFSv4 仅需要单个已知的 TCP 端口 2049。对于协议版本 4.0，可能需要将内核参数 `fs.nfs.nfs_callback_tcpport` 设置为静态端口（请参见例 24.1 “`/etc/modprobe.d/60-nfs.conf` 中 `nfs` 内核模块的回调端口配置”）。从协议版本 4.1 开始，此项设置也变得没有必要。

由于 `rpcbind` 协议具有动态性质，要使防火墙后面的受影响服务可供访问变得很困难。`firewalld` 本身并不支持这些服务。如需手动配置，请参见第 24.4.3.1 节“配置静态端口”。此外，SUSE Linux Enterprise Server 提供了一个助手脚本。有关详细信息，请参见第 24.4.3.2 节“使用 `firewall-rpcbind-helper` 配置静态端口”。

24.4.3.1 配置静态端口

一种可行的做法是将所有相关网络服务配置为使用固定端口号。完成此操作后，可以在 `firewalld` 中打开固定端口，然后一切会正常进行。使用的实际端口号由您决定，但不应与指派给其他服务的任何已知端口号相冲突。有关 NIS 和 NFSv3 服务的可用配置项列表，请参见表 24.1 “静态端口配置的重要 `sysconfig` 变量”。请注意，您的设置不一定需要所有这些端口，具体取决于您的实际 NIS 或 NFS 配置。

表 24.1：静态端口配置的重要 `SYSCONFIG` 变量

文件路径	变量名	示例值
<code>/etc/sysconfig/nfs</code>	<code>MOUNTD_PORT</code>	21001

文件路径	变量名	示例值
	STATD_PORT	21002
	LOCKD_TCPPORT	21003
	LOCKD_UDPPORT	21003
	RQUOTAD_PORT	21004
<u>/etc/sysconfig/ybind</u>	YPBIND_OPTIONS	-p24500
<u>/etc/sysconfig/ypserv</u>	YPXFRD_ARGS	-p24501
	YPSERV_ARGS	-p24502
	YPPASSWDD_ARGS	--port 24503

您需要重新启动受这些静态端口配置影响的所有相关服务才能使更改生效。可以使用 **rpcinfo -p** 命令查看当前指派的 rpcbind 端口。如果成功，输出中应该只会显示静态配置的端口。

使用 NFS 时，除了为用户空间中运行的网络服务配置端口以外，还需要配置 Linux 内核直接使用的端口。其中一个端口是 nfs_callback_tcpport。仅在早于 4.1 的 NFS 协议版本中才需要此端口。名为 fs.nfs.nfs_callback_tcpport 的 sysctl 可用于配置此端口。仅当 NFS 装入处于活动状态时，此 sysctl 节点才会动态显示。因此，最好通过内核模块参数来配置该端口。可以按例 24.1 “/etc/modprobe.d/60-nfs.conf 中 nfs 内核模块的回调端口配置” 中所示创建一个文件来实现此目的。

例 24.1： /etc/modprobe.d/60-nfs.conf 中 nfs 内核模块的回调端口配置

```
options nfs callback_tcpport=21005
```

要使此项更改生效，最简单的方法是重引导计算机。如果采用其他方法，将需要停止所有 NFS 服务并重新装载 nfs 内核模块。要校验活动的 NFS 回调端口，请检查 **cat /sys/module/nfs/parameters/callback_tcpport** 的输出。

要轻松处理现已静态配置的 RPC 端口，创建新的 firewalld 服务定义会很有用。例如，此服务定义可将所有相关端口分组，您轻而易举就可使这些端口在特定的区域中供用户访问。在例 24.2 “用来为 NFS 定义新 firewalld RPC 服务的命令” 中，对 NFS 端口就采取了这种做法，因为在配套的示例中已对这些端口进行配置。

例 24.2：用来为 NFS 定义新 firewalld RPC 服务的命令

```
root # firewall-cmd --permanent --new-service=nfs-rpc
root # firewall-cmd --permanent --service=nfs-rpc --set-description="NFS
      related, statically configured RPC ports"
# add UDP and TCP ports for the given sequence
root # for port in 21001 21002 21003 21004; do
      firewall-cmd --permanent --service=nfs-rpc --add-port ${port}/udp --add-port
      ${port}/tcp
done
# the callback port is TCP only
root # firewall-cmd --permanent --service=nfs-rpc --add-port 21005/tcp

# show the complete definition of the new custom service
root # firewall-cmd --info-service=nfs-rpc --permanent -v
nfs-rpc
  summary:
  description: NFS and related, statically configured RPC ports
  ports: 4711/tcp 21001/udp 21001/tcp 21002/udp 21002/tcp 21003/udp 21003/tcp
  21004/udp 21004/tcp
  protocols:
  source-ports:
  modules:
  destination:

# reload firewalld to make the new service definition available
root # firewall-cmd --reload

# the new service definition can now be used to open the ports for example in
  the internal zone
root # firewall-cmd --add-service=nfs-rpc --zone=internal
```

24.4.3.2 使用 firewall-rpcbind-helper 配置静态端口

您可以使用 SUSE 助手工具 `firewall-rpc-helper.py` 来简化上一节中所述的静态端口配置步骤。可使用 `zypper in firewall-rpcbind-helper` 安装该工具。

该工具允许以交互方式配置上一节中所述的服务模式。它还可以显示当前端口指派，并可用于编写脚本。有关细节，请参见 `firewall-rpc-helper.py --help`。

24.5 从 SuSEfirewall2 迁移



注意：为 AutoYaST 创建 firewalld 配置

请参见《AutoYaST 指南》的“防火墙配置”一节，了解如何为 AutoYaST 创建 `firewalld` 配置。

从 SUSE Linux Enterprise Server 12 的任何服务包升级到 SUSE Linux Enterprise Server 15 SP3 时，SuSEfirewall2 不会更改，并会保持活动状态。它不会自动迁移，因此您必须手动迁移到 `firewalld`。`firewalld` 包含助手迁移脚本 `susefirewall2-to-firewalld`。该脚本可能会完美执行迁移，也可能失败，具体取决于 SuSEfirewall2 配置的复杂性。它很可能会部分成功，在此情况下，您必须检查新的 `firewalld` 配置并做出调整。

最终的配置会使 `firewalld` 的行为在一定程度上类似于 SuSEfirewall2。要充分利用 `firewalld` 的功能，您可以选择创建新的配置，而不要尝试迁移旧配置。可以安全运行不带任何选项的 `susefirewall2-to-firewalld` 脚本，因为它不会对您的系统做出永久性更改。但是，如果您正在远程管理系统，则可能会被锁定。

安装并运行 `susefirewall2-to-firewalld`：

```
root # zypper in susefirewall2-to-firewalld
root # susefirewall2-to-firewalld
INFO: Reading the /etc/sysconfig/SuSEfirewall2 file
INFO: Ensuring all firewall services are in a well-known state.
INFO: This will start/stop/restart firewall services and it's likely
INFO: to cause network disruption.
INFO: If you do not wish for this to happen, please stop the script now!
5...4...3...2...1...Lets do it!
INFO: Stopping firewalld
INFO: Restarting SuSEfirewall2_init
INFO: Restarting SuSEfirewall2
```

```

INFO: DIRECT: Adding direct rule="ipv4 -t filter -A INPUT -p udp -m udp --dport
5353 -m pkttype
--pkt-type multicast -j ACCEPT"
[...]
INFO: Enabling direct rule=ipv6 -t filter -A INPUT -p udp -m udp --dport 546 -j
ACCEPT
INFO: Enabling direct rule=ipv6 -t filter -A INPUT -p udp -m udp --dport 5353 -m
pkttype
--pkt-type multicast -j ACCEPT
INFO: Enable logging for denied packets
INFO: #####
INFO:
INFO: The dry-run has been completed. Please check the above output to ensure
INFO: that everything looks good.
INFO:
INFO: #####
INFO: Stopping firewalld
INFO: Restarting SuSEfirewall2_init
INFO: Restarting SuSEfirewall2

```

这会生成大量输出，您可能需要将其复制到某个文件以方便查看：

```
root # susefirewall2-to-firewalld | tee newfirewallrules.txt
```

该脚本支持下列选项：

-c

提交更改。脚本将对系统进行更改，因此，请确保仅当您确实对建议的更改感到满意时才使用此选项。这会重置当前的 firewalld 配置，因此请务必进行备份！

-d

超级繁琐。请使用此选项来提交 bug 报告，但务必小心屏蔽敏感信息。

-h

此消息。

-q

无输出。也不会列显错误！

-v

冗长方式。会列显警告和其他信息性消息。

24.6 更多信息

`/usr/share/doc/packages/firewalld` 中提供了有关 `firewalld` 软件包的最新信息和其他文档。netfilter 和 iptables 项目的主页 <http://www.netfilter.org> 以多种语言提供了大量有关 iptables 一般信息的文档。

25 配置 VPN 服务器

现今，因特网连接费用低廉，几乎在任何地方都可以上网，但并非所有连接都是安全的。利用虚拟专用网 (VPN)，您可以在不安全的网络（例如因特网或 Wi-Fi）内创建安全网络。VPN 可通过不同的方式实现，并用于多种目的。本章重点介绍如何使用 [OpenVPN \(https://openvpn.net\)](https://openvpn.net) 来通过安全广域网 (WAN) 链接各分支办公室。

25.1 概念概述

本节定义了 VPN 相关的一些术语，并简要概述了一些方案。

25.1.1 术语

端点

隧道的两“端”：源客户端和目标客户端。

Tap 设备

Tap 设备可模拟以太网设备（OSI 模型中的第 2 层包，例如以太网帧）。Tap 设备用于创建网桥，可处理以太网帧。

Tun 设备

Tun 设备可模拟点对点网络（OSI 模型中的第 3 层包，例如 IP 包）。Tun 设备用于路由，可处理 IP 帧。

隧道

通过主公共网络链接两个位置。从技术角度看，隧道是客户端设备与服务器设备之间的连接。隧道通常会加密，但按定义它确实需要加密。

25.1.2 VPN 方案

每当您设置 VPN 连接时，您的 IP 包就会通过安全隧道传输。隧道可以使用 **tun** 或 **tap** 设备。这些设备是虚拟网络内核驱动程序，用于实现以太网帧或 IP 帧/包的传输。

任何用户空间程序（例如 OpenVPN）都可以将自身挂接到 tun 或 tap 设备，以接收操作系统发送的包。该程序还可以向此类设备写入包。

设置和构建 VPN 连接的解决方案有许多。本节重点介绍 OpenVPN 软件包。不像其他 VPN 软件，OpenVPN 可在两种模式下运行：

路由式 VPN

路由是可设置的简易解决方案。它的效率比桥接式 VPN 更高，缩放能力更强。此外，它允许用户调整 MTU（最大传输单元）以提高效率。但在异构环境中，如果您的网关上没有 Samba 服务器，NetBIOS 广播不会正常工作。如果您需要 IPv6，两端上 tun 设备的驱动程序必须显式支持此协议。图 25.1 “路由式 VPN” 中描绘了此方案。

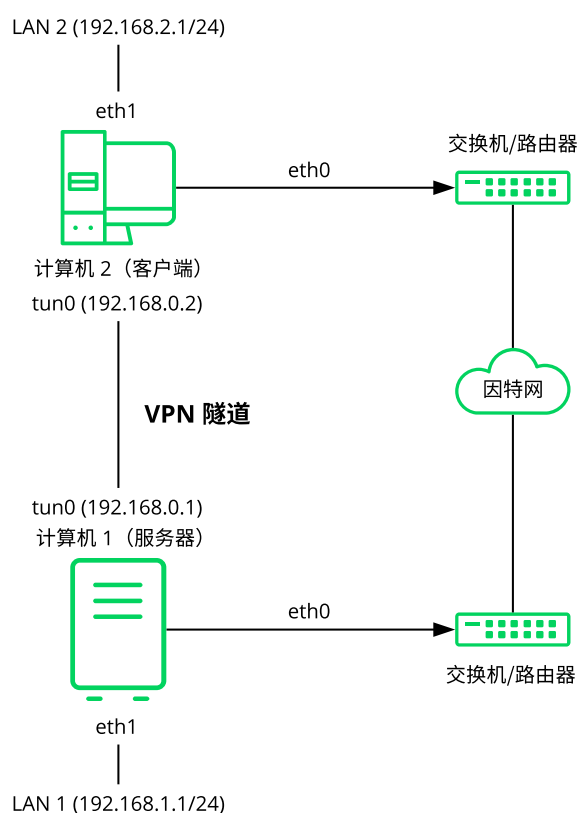


图 25.1：路由式 VPN

桥接式 VPN

桥接是更复杂的解决方案。如果您需要在不设置 Samba 或 WINS 服务器的情况下通过 VPN 浏览 Windows 文件共享，则建议使用桥接。使用非 IP 协议（例如 IPX）或依赖于网络广播的应用程序也需要用到桥接式 VPN。但是，桥接式 VPN 的效率比路由式 VPN 要低。另一项劣势是它的缩放能力不强。下列插图描绘了此方案。

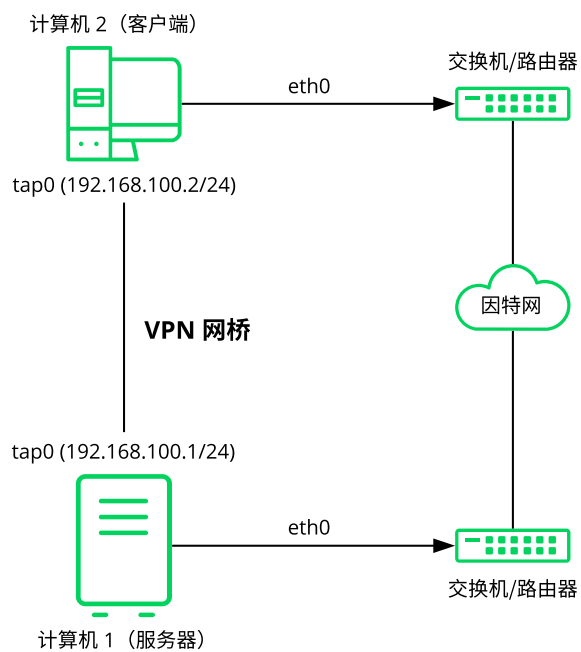


图 25.2：桥接式 VPN - 方案 1

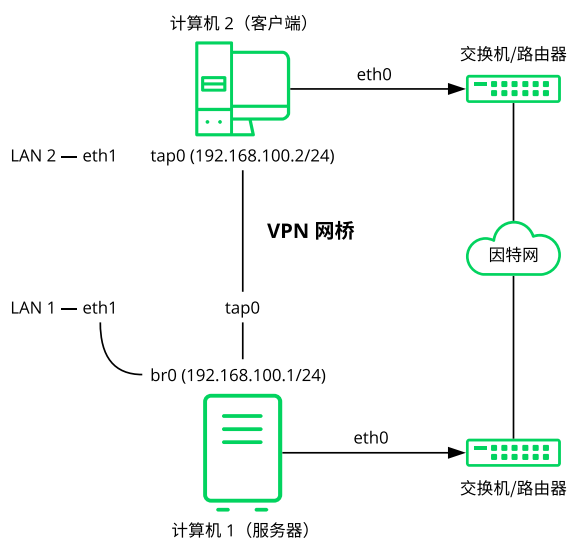


图 25.3：桥接式 VPN - 方案 2

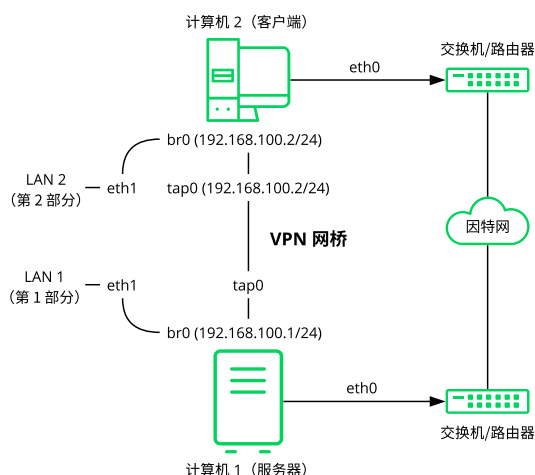


图 25.4：桥接式 VPN - 方案 3

桥接与路由之间的主要差别在于，路由式 VPN 无法进行 IP 广播，而桥接式 VPN 则可以。

25.2 设置简单测试方案

在下面的示例中，我们将创建一个点对点 VPN 隧道。该示例说明如何在一个客户端与某个服务器之间创建 VPN 隧道。假设您的 VPN 服务器将使用私用 IP 地址（例如 IP_OF_SERVER），客户端将使用 IP 地址 IP_OF_CLIENT。请确保选择的地址与其他 IP 地址不冲突。



警告：仅用于测试

下面的方案仅为示例，旨在帮助您熟悉 VPN 技术。**请勿**使用此示例作为真实方案，因为它可能会损害 IT 基础结构的安全性！



提示：配置文件的名称

为了简化 OpenVPN 配置文件的处理，我们建议采取以下做法：

- 将 OpenVPN 配置文件放在目录 `/etc/openvpn` 中。
- 将配置文件命名为 `MY_CONFIGURATION.conf`。
- 如果有多个文件属于同一配置，请将这些文件放在某个子目录（例如 `/etc/openvpn/MY_CONFIGURATION`）中。

25.2.1 配置 VPN 服务器

要配置 VPN 服务器，请执行以下操作：

过程 25.1：VPN 服务器配置

1. 在稍后要用作 VPN 服务器的计算机上安装 `openvpn` 软件包。

2. 在外壳上，以 `root` 身份创建 VPN 机密密钥：

```
root # openvpn --genkey --secret /etc/openvpn/secret.key
```

3. 将机密密钥复制到客户端：

```
root # scp /etc/openvpn/secret.key root@IP_OF_CLIENT:/etc/openvpn/
```

4. 创建包含以下内容的 `/etc/openvpn/server.conf` 文件：

```
dev tun
ifconfig IP_OF_SERVER IP_OF_CLIENT
secret secret.key
```

5. 通过创建包含以下内容的 `/etc/sysconfig/network/ifcfg-tun0` 文件设置 tun 设备配置：

```
STARTMODE='manual'
BOOTPROTO='static'
TUNNEL='tun'
TUNNEL_SET_OWNER='nobody'
TUNNEL_SET_GROUP='nobody'
```

```
LINK_REQUIRED=no
PRE_UP_SCRIPT='systemd:openvpn@server'
PRE_DOWN_SCRIPT='systemd:openvpn@service'
```

`openvpn@server` 表示法指向位于 `/etc/openvpn/server.conf` 的 OpenVPN 服务器配置文件。有关详细信息，请参见 `/usr/share/doc/packages/openvpn/README.SUSE`。

6. 如果您使用防火墙，请启动 YaST 并打开 UDP 端口 1194（安全和用户 > 防火墙 > 允许的服务）。
7. 通过将 tun 设备设置为 `up` 启动 OpenVPN 服务器服务：

```
tux > sudo wicked ifup tun0
```

此时应会看到确认消息：

```
tun0          up
```

25.2.2 配置 VPN 客户端

要配置 VPN 客户端，请执行以下操作：

过程 25.2：VPN 客户端配置

1. 在客户端 VPN 计算机上安装 `openvpn` 软件包。
2. 创建包含以下内容的 `/etc/openvpn/client.conf`：

```
remote DOMAIN_OR_PUBLIC_IP_OF_SERVER
dev tun
ifconfig IP_OF_CLIENT IP_OF_SERVER
secret secret.key
```

请将第一行中的占位符 `IP_OF_CLIENT` 替换为服务器的域名或公共 IP 地址。

3. 通过创建包含以下内容的 `/etc/sysconfig/network/ifcfg-tun0` 文件设置 tun 设备配置：

```
STARTMODE='manual'
BOOTPROTO='static'
TUNNEL='tun'
TUNNEL_SET_OWNER='nobody'
TUNNEL_SET_GROUP='nobody'
LINK_REQUIRED=no
PRE_UP_SCRIPT='systemd:openvpn@client'
PRE_DOWN_SCRIPT='systemd:openvpn@client'
```

4. 如果您使用防火墙，请按[过程 25.1 “VPN 服务器配置”](#)的[步骤 6](#)中所述启动 YaST 并打开 UDP 端口 1194。
5. 通过将 tun 设备设置为 up 启动 OpenVPN 服务器服务：

```
tux > sudo wicked ifup tun0
```

此时应会看到确认消息：

```
tun0          up
```

25.2.3 测试 VPN 示例方案

OpenVPN 成功启动后，使用以下命令测试 tun 设备的可用性：

```
ip addr show tun0
```

要校验 VPN 连接，请在客户端和服务端使用 ping 来确定它们能否相互连接。从客户端 ping 服务器：

```
ping -I tun0 IP_OF_SERVER
```

从服务器 ping 客户端：

```
ping -I tun0 IP_OF_CLIENT
```

25.3 使用证书颁发机构设置 VPN 服务器

第 25.2 节 中的示例用于测试，但不可用于日常工作。本节说明如何构建一个同时允许多个连接的 VPN 服务器。此过程使用公共密钥基础结构 (PKI) 完成。PKI 由以下组件构成：服务器和每个客户端的一对公共密钥和私用密钥，以及一个用来为每个服务器证书和客户端证书签名的主证书颁发机构 (CA)。

此设置涉及以下基本步骤：

1. 第 25.3.1 节 “创建证书”
2. 第 25.3.2 节 “配置 VPN 服务器”
3. 第 25.3.3 节 “配置 VPN 客户端”

25.3.1 创建证书

在可以建立 VPN 连接之前，客户端必须对服务器证书进行身份验证。相对地，服务器也必须对客户端证书进行身份验证。此过程称为**相互身份验证**。

SUSE Linux Enterprise Server 不支持创建证书。以下内容假设您已在另一个系统上创建了 CA 证书、服务器证书和客户端证书。

服务器证书需要采用 PEM 格式，未加密的密钥需采用 PEM 格式。将 PEM 版本复制到 VPN 服务器上的 `/etc/openvpn/server_cert.pem` 中。未加密版本需放到 `/etc/openvpn/server_key.pem` 中。

客户端证书需采用 PKCS12（首选）或 PEM 格式。PKCS12 格式的证书需要包含 CA 链，并且需要复制到 `/etc/openvpn/CLIENT.p12` 中。如果您有包含 CA 链的 PEM 格式客户端证书，请将其复制到 `/etc/openvpn/CLIENT.pem` 中。如果您已将 PEM 证书分割成客户端证书 (*.ca)、客户端密钥 (*.key) 和 CA 证书 (*.ca)，请将这些文件复制到每个客户端上的 `/etc/openvpn/` 中。

CA 证书需复制到服务器和每个客户端上的 `/etc/openvpn/vpn_ca.pem` 中。

！ 重要：分割客户端证书

如果您要将客户端证书分割成客户端证书、客户端密钥和 CA 证书，需要在相应客户端上的 OpenVPN 配置文件中提供相应的文件名（请参见例 25.1 “VPN 服务器配置文件”）。

25.3.2 配置 VPN 服务器

将 `/usr/share/doc/packages/openvpn/sample-config-files/server.conf` 复制到 `/etc/openvpn/` 中以作为配置文件的基础。然后根据需要对其进行自定义。

例 25.1：VPN 服务器配置文件

```
# /etc/openvpn/server.conf
port 1194 ❶
proto udp ❷
dev tun0 ❸

# Security ❹

ca    vpn_ca.pem
cert  server_cert.pem
key   server_key.pem

# ns-cert-type server
remote-cert-tls client ❺
dh    server/dh2048.pem ❻

server 192.168.1.0 255.255.255.0 ❼
ifconfig-pool-persist /var/run/openvpn/ipp.txt ❽

# Privileges ❾
user nobody
group nobody

# Other configuration ❿
keepalive 10 120
```

```
comp-lzo
persist-key
persist-tun
# status /var/log/openvpn-status.tun0.log ❾
# log-append /var/log/openvpn-server.log ❿
verb 4
```

- ❶ OpenVPN 监听的 TCP/UDP 端口。需要在防火墙中打开该端口，具体请参见第 24 章“[掩蔽和防火墙](#)”。VPN 的标准端口为 1194，因此您通常可以将此设置保持不变。
- ❷ 协议 UDP 或 TCP。
- ❸ Tun 或 tap 设备。有关两者的差异，请参见第 25.1.1 节“[术语](#)”。
- ❹ 下面的行包含根服务器 CA 证书 ([ca](#))、根 CA 密钥 ([cert](#)) 和服务器私用密钥 ([key](#)) 的相对或绝对路径。这些项是在第 25.3.1 节“[创建证书](#)”中生成的。
- ❺ 要求基于 RFC3280 TLS 规则使用显式密钥和扩展密钥为对等证书签名。
- ❻ Diffie-Hellman 参数。使用以下命令创建所需的文件：

```
openssl dhparam -out /etc/openvpn/dh2048.pem 2048
```

- ❼ 提供 VPN 子网。可通过 [192.168.1.1](#) 访问该服务器。
- ❽ 在给定文件中记录客户端及其虚拟 IP 地址的映射。当服务器关闭以及客户端（在重新启动后）获取以前指派的 IP 地址时很有用。
- ❾ 出于安全原因，请以降级的特权运行 OpenVPN 守护程序。为此，请指定应使用组和用户 [nobody](#)。
- ❿ 多个配置选项 — 请参见示例配置文件 [/usr/share/doc/packages/openvpn/sample-config-files](#) 中的注释。
- ❾ 启用此选项可将包含统计数据的简短状态更新（“操作状态转储”）写入命名的文件。默认不会启用此选项。
所有输出将写入到可通过 [journalctl](#) 显示的系统日记中。如果您有多个配置文件（例如，一个在家里使用，一个在工作时使用），我们建议在文件名中包含设备名。这可以避免意外重写输出文件。在本例中，设备名是 [tun0](#)（取自 [dev](#) 指令）— 请参见 ❸。
- ❿ 默认情况下，日志消息将写入 syslog。去除井号字符可重写此行为。在这种情况下，所有消息将写入 [/var/log/openvpn-server.log](#)。不要忘记配置 logrotate 服务。有关更多细节，请参见 [man 8 logrotate](#)。

完成此配置后，可以在 `/var/log/openvpn.log` 下查看 OpenVPN 服务器的日志消息。首次启动此配置后，最后应会显示：

```
... Initialization Sequence Completed
```

如果未看到此消息，请仔细检查日志，确定其中是否有任何提示指出了配置文件中的错误。

25.3.3 配置 VPN 客户端

将 `/usr/share/doc/packages/openvpn/sample-config-files/client.conf` 复制到 `/etc/openvpn/` 中以作为配置文件的基础。然后根据需要对其进行自定义。

例 25.2：VPN 客户端配置文件

```
# /etc/openvpn/client.conf
client ❶
dev tun ❷
proto udp ❸
remote IP_OR_HOST_NAME 1194 ❹
resolv-retry infinite
nobind

remote-cert-tls server ❺

# Privileges ❻
user nobody
group nobody

# Try to preserve some state across restarts.
persist-key
persist-tun

# Security ❼
pkcs12 client1.p12

comp-lzo ❽
```

❶ 指定此计算机是客户端。

- ② 网络设备。客户端和服务端必须使用相同的设备。
- ③ 协议。使用与服务端上相同的设置。
- ⑤ 这是客户端的一个安全选项，确保客户端连接到的主机是指定的服务器。
- ④ 请将占位符 `IP_OR_HOST_NAME` 替换为 VPN 服务器的相应主机名或 IP 地址。主机名后面提供了服务器端口。您可以设置指向不同 VPN 服务器的多行 `remote` 项。此设置可用在不同 VPN 服务器之间进行负载平衡。
- ⑥ 出于安全原因，请以降级的特权运行 OpenVPN 守护程序。为此，请指定应使用组和用户 `nobody`。
- ⑦ 包含客户端文件。出于安全原因，请为每个客户端单独使用一对文件。
- ⑧ 开启压缩。请仅在服务器上也启用了压缩时，才使用此参数。

25.4 使用 YaST 设置 VPN 服务器或客户端

您还可以使用 YaST 来设置 VPN 服务器，但 YaST 模块不支持 OpenVPN，却提供对 IPsec 协议的支持（在软件 StrongSwan 中实现）。与 OpenVPN 一样，IPsec 是广受支持的 VPN 模式。

过程 25.3：设置 IPSEC 服务器

1. 要启动 YaST VPN 模块，请选择应用程序 > VPN 网关和客户端。
2. 在全局配置下，选中启用 VPN 守护程序。
3. 要创建新 VPN，请单击新建 VPN，然后输入连接名称。
4. 在类型下，选择网关（服务器）。
5. 然后选择方案：
 - 使用预共享密钥的安全通讯和使用证书的安全通讯方案最适合 Linux 客户端设置。
 - 提供 Android、iOS、MacOS X 客户端的访问权方案会设置现代版本的 Android、iOS 和 macOS 原生支持的配置。此方案以使用附加用户名和口令身份验证的预共享密钥设置为基础。
 - 提供 Windows 7、Windows 8 客户端的访问权方案是 Windows 和 BlackBerry 设备原生支持的配置。此方案以使用附加用户名和口令身份验证的证书设置为基础。

对于本示例，请选择使用预共享密钥的安全通讯。

6. 要指定密钥，请单击编辑身份凭证。选中显示密钥，然后键入机密密钥。单击确定进行确认。
7. 在提供以下项目的 VPN 客户端访问权下选择是否以及如何限制 VPN 内部的访问权限。要仅启用特定的 IP 范围，请在有限的 CIDR 中以 CIDR 格式指定这些范围并以逗号分隔。有关 CIDR 格式的详细信息，请参见 https://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing。
8. 在客户端的地址池下，指定 VPN 应向其客户端提供的 IP 地址的格式。
9. 要完成该过程，请单击确定。现在，YaST VPN 模块会自动添加并启用防火墙规则，以允许客户端连接到新 VPN。
要查看连接状态，请在随后的确认窗口中单击是。然后，您将看到对 VPN 运行 `systemctl status` 后的输出，并可在其中检查 VPN 是否正在运行且已正确配置。

25.5 更多信息

有关 VPN 的一般详细信息，请参见：

- <https://openvpn.net>：OpenVPN 主页
- `man openvpn`
- </usr/share/doc/packages/openvpn/sample-config-files/>：不同方案的示例配置文件。
- </usr/src/linux/Documentation/networking/tuntap.txt>，用于安装 `kernel-source` 软件包。

26 使用 XCA、X 证书和密钥管理器管理 PKI

传统情况下，使用 **openssl** 实用程序来管理自己的公共密钥基础结构 (PKI)。对于偏好使用图形工具的管理员，SUSE Linux Enterprise 15.3 提供了 XCA，即 X 证书和密钥管理工具 (<http://hohnstaedt.de/xca>)。

XCA 可创建和管理 X.509 证书、证书请求、RSA、DSA 和 EC 私用密钥、智能卡以及证书吊销列表 (CRL)。XCA 会为您提供创建和管理自己的证书颁发机构 (CA) 所需的一切支持。XCA 包含可用于生成证书或请求的可自定义模板。本章将介绍基本设置。

26.1 安装 XCA

XCA 由 **xca** 软件包提供：

```
tux > sudo zypper in xca
```

26.2 创建新 PKI

XCA 将所有加密数据都储存在数据库中。当您首次使用 XCA 创建新 PKI 时，首先必须单击文件 > 新建数据库创建一个新数据库（图 26.1 “创建新 XCA 数据库”）。

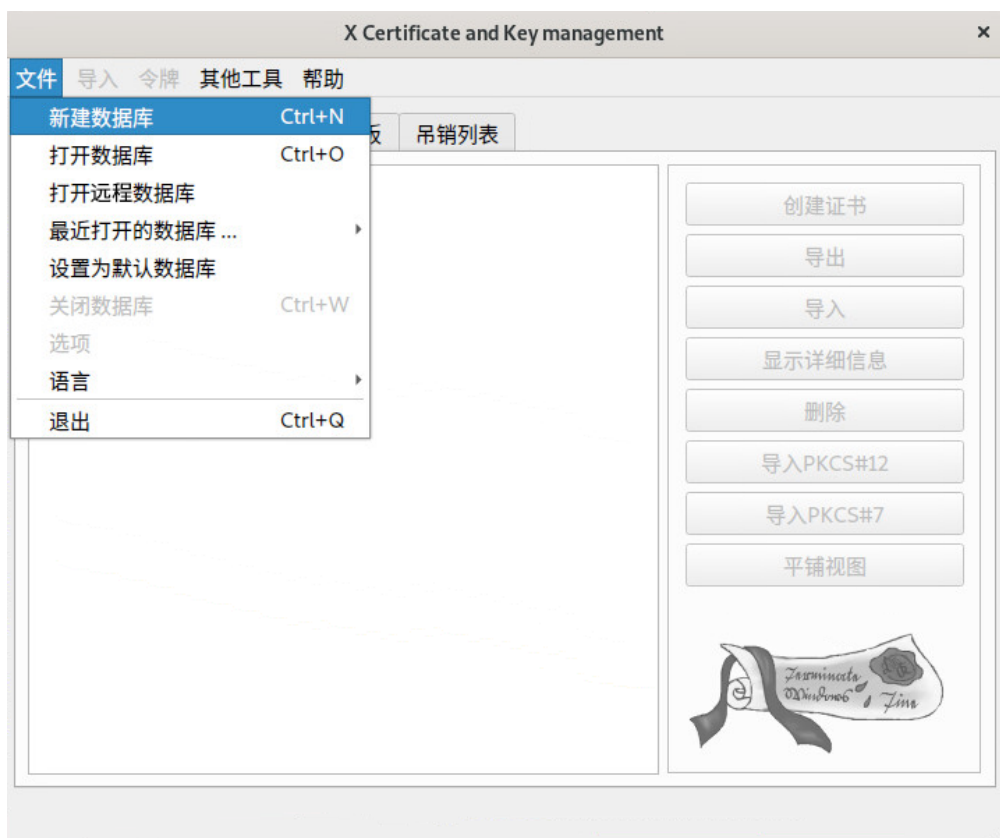


图 26.1：创建新 XCA 数据库

26.2.1 创建新的根 CA

以下步骤说明如何创建新的根 CA。

1. 单击证书选项卡。
2. 单击新建证书按钮。
3. 单击源选项卡。在窗口底部的新证书的模板下，选择 [默认] CA 模板，然后单击全部应用。
4. 单击主体选项卡。创建一个内部名称用于在内部标识新的根 CA（仅在 XCA 内部）。填写判别名部分中的字段。使用添加按钮添加任何其他元素（如果需要）。
5. 在私用密钥下拉列表中选择您的首选私用密钥（如果有），或者生成一个新密钥。

6. 单击扩展选项卡。根据需要编辑任何属性。默认时间范围为 10 年。证书吊销列表分发点将成为颁发的证书的一部分。最好为所有证书使用一个通用 URL，例如 <http://www.example.com/crl/crl.der>。完成后，单击确定按钮。

26.2.2 创建已签名的主机证书

下一步是创建由新证书颁发机构签名的主机证书。

1. 单击证书选项卡，然后单击新建证书按钮。
2. 在源选项卡上选择 [默认] TLS_server，然后单击全部应用按钮。这样会在扩展、密钥用途和 Netscape 选项卡中输入相应的值。在签名部分，选择在第 26.2.1 节 “创建新的根 CA” 中创建的证书。
3. 单击主体选项卡。创建一个用于在 XCA 中显示的内部名称。最好使用主机名或完全限定的域名。然后填写判别名部分中的字段。对于主机证书，通用名必须是您的用户将使用的 FQDN。此名称可以是主机的规范名称，也可以是别名。例如，如果 jupiter.example.com 是您的 Web 服务器，并且它具有 DNS CNAME 项 www.example.com，那么，您也许可以使用 www.example.com 作为证书中的 commonName 值。如果您要在判别名中添加任何其他组成部分，请使用下拉框和 “添加” 按钮。选择所需的私用密钥，或生成一个新密钥。
4. 单击扩展选项卡。默认时间范围为 1 年。如果您更改了此设置，请单击应用按钮。
5. 建议指定证书吊销列表位置。此根证书的位置必须是唯一的。XCA 以 PEM 或 DER 格式导出 CRL 并会添加相应的后缀，因此在选择 URL 时应考虑到这点，例如，选择 <http://www.example.com/crl/crl.der> 这样的 URL。在 CRL 分发点行中，单击编辑按钮。键入您的 URI，然后单击添加。单击验证和应用。单击确定按钮。

26.2.3 吊销证书

1. 单击证书选项卡。
2. 右键单击您要吊销的证书，然后单击吊销。

3. 右键单击为您要吊销的证书签名的 CA 证书。单击 CA > 生成 CRL。
在创建 CRL 对话框中单击确定按钮。
4. 在主窗口中单击吊销列表选项卡。右键单击刚刚生成的 CRL 并选择导出。选择所需的格式（例如 DER），然后单击确定。
将导出的 CRL 复制到所颁发证书的 CRL 分发点中发布的位置。

IV 通过 AppArmor 限制特权

- 27 AppArmor 简介 226
- 28 入门 228
- 29 使程序免疫 233
- 30 配置文件组件和语法 242
- 31 AppArmor 配置文件储存库 274
- 32 使用 YaST 构建和管理配置文件 275
- 33 从命令行构建配置文件 285
- 34 使用 ChangeHat 构建 Web 应用程序的配置文件 312
- 35 使用 pam_apparmor 限制用户 323
- 36 管理已构建配置文件的应用程序 324
- 37 支持 326
- 38 AppArmor 术语表 335

27 AppArmor 简介

许多安全漏洞是**可信赖**程序中的错误产生的。可信赖程序是使用攻击者想要拥有的特权运行的。如果该程序中存在的 bug 导致攻击者获得了此特权，则该程序将丧失可信赖性。

AppArmor® 是一套应用程序安全解决方案，专门用于针对可疑程序应用特权限制。AppArmor 允许管理员通过开发安全**配置文件**来指定程序可执行的活动域。安全配置文件是程序可访问的文件以及可执行的操作的列表。AppArmor 不依赖攻击特征，而是以强制方式使应用程序保持良好行为，从而保障应用程序的安全，因此即使是以前未知的漏洞遭到恶意利用，它也能预防攻击。

27.1 AppArmor 组件

AppArmor 由以下部分组成：

- 常用 Linux* 应用程序的 AppArmor 配置文件库，描述程序需要访问的文件。
- 进行常见的应用程序活动（如 DNS 查找和用户身份验证）所需的 AppArmor 配置文件基础类（配置文件构建基块）库。
- 用于开发和增强 AppArmor 配置文件的工具套件，使用它可以对现有配置文件进行更改以适应您的需要，还可以为您自己的本地和自定义应用程序创建新的配置文件。
- 若干经过特别修改的应用程序，这些应用程序支持 AppArmor，能够通过独特的子进程限制方式来提升安全性，其中包括 Apache。
- AppArmor 相关的内核代码和关联的控制脚本，用于在 SUSE® Linux Enterprise Server 系统上强制实施 AppArmor 策略。

27.2 有关 AppArmor 配置文件构建的背景信息

有关 AppArmor 的科学和安全性的详细信息，请参见以下文献：

SubDomain: Parsimonious Server Security, 作者: Crispin Cowan、Steve Beattie、Greg Kroah-Hartman、Calton Pu、Perry Wagle 和 Virgil Gligor

介绍 AppArmor 的初始设计和实施。在 2000 年 12 月在路易斯安娜州新奥尔良召开的 USENIX LISA 会议期间出版。此文献目前已过时，介绍的语法和功能与最新的 AppArmor 产品不同。此文献仅可用于了解背景知识，不能用作技术文档。

Defcon Capture the Flag: Defending Vulnerable Code from Intense Attack, 作者: Crispin Cowan、Seth Arnold、Steve Beattie、Chris Wright 和 John Viega

是很好的战略和战术上的 AppArmor 使用指导，可以帮助您在很短的时间内解决严重的安全问题。2003 年 4 月在华盛顿召开 DARPA Information Survivability Conference and Expo (DISCEX III) 会议期间出版

AppArmor for Geeks, 作者: Seth Arnold

此文档的目标是让读者更好地了解 AppArmor 的技术细节。https://en.opensuse.org/SDB:AppArmor_geeks 上提供了该文档。

28 入门

请仔细考虑以下事项，以便为在系统上成功部署 AppArmor 做好准备：

1. 确定要构建配置文件的应用程序。有关详细信息，请参见第 28.3 节 “选择要构建配置文件的应用程序”。
2. 根据第 28.4 节 “构建和修改配置文件” 中的简要说明构建需要的配置文件。检查结果并在必要时调整配置文件。
3. 每当环境发生变化或者您需要对 AppArmor 的报告工具记录的安全事件作出反应时，请更新您的配置文件。有关详细信息，请参见第 28.5 节 “更新您的配置文件”。

28.1 安装 AppArmor

在任何安装的 SUSE® Linux Enterprise Server 上，无论安装了哪些模式，默认都会安装并运行 AppArmor。AppArmor 的完整功能实例需要下面列出的软件包：

- [apparmor-docs](#)
- [apparmor-parser](#)
- [apparmor-profiles](#)
- [apparmor-utils](#)
- [audit](#)
- [libapparmor1](#)
- [perl-libapparmor](#)
- [yast2-apparmor](#)



提示

如果您的系统上未安装 AppArmor，请安装 [apparmor](#) 模式以安装完整的 AppArmor。请使用 YaST 软件管理模块进行安装，或者在命令行上使用 Zypper：

```
tux > sudo zypper in -t pattern apparmor
```

28.2 启用和禁用 AppArmor

在任何全新安装的 SUSE Linux Enterprise Server 上，默认都会将 AppArmor 配置为运行状态。可以通过两种方式切换 AppArmor 的状态：

使用 YaST 服务管理器

通过在系统引导时所执行的脚本序列中去除或添加引导脚本来禁用或启用 AppArmor。重引导时将应用状态更改。

使用 AppArmor 配置窗口

可以使用 YaST AppArmor 控制面板关闭或打开 AppArmor，以在运行中的系统上切换其状态。在控制面板中所执行的更改将即时应用。控制面板会触发 AppArmor 停止或启动事件，并在系统引导序列中去除或添加它的引导脚本。

要通过从系统引导时所执行的脚本序列中去除 AppArmor 永久将其禁用，请执行以下操作：

1. 启动 YaST。
2. 选择系统 > 服务管理器。
3. 在服务列表中单击 `apparmor` 所在的行将其选中，然后在窗口的下半部分单击启用/禁用。在 `apparmor` 行中检查已启用是否已更改为已禁用。
4. 单击确定进行确认。

AppArmor 在重引导时不会初始化，并会保持非活动状态，直到您重新将其启用。使用 YaST 服务管理器工具重新启用服务的操作与禁用服务类似。

使用“AppArmor 配置”窗口在运行中的系统上切换 AppArmor 的状态。应用这些更改并重引导系统后，这些更改将生效。要切换 AppArmor 的状态，请执行以下操作：

1. 启动 YaST，选择 AppArmor 配置，然后在主窗口中单击设置。
2. 选中启用 AppArmor 以启用 AppArmor，或取消选中该选项以禁用 AppArmor。
3. 单击 AppArmor 配置窗口中的完成。

28.3 选择要构建配置文件的应用程序

您只需保护在您的特定设置中会受到攻击的程序，因此只需为实际运行的程序使用配置文件。使用以下列表来确定最可能的候选程序：

网络代理

Web 应用程序

Cron 作业

要了解哪些进程当前以开放网络端口运行并且可能需要配置文件来进行限制，请作为 root 运行 **aa-unconfined**。

例 28.1：AA-UNCONFINED 的输出

```
19848 /usr/sbin/cupsd not confined
19887 /usr/sbin/sshd not confined
19947 /usr/lib/postfix/master not confined
1328 /usr/sbin/smbd confined by '/usr/sbin/smbd (enforce)'
```

上例中标注为 not confined 的每个进程都可能需要定制的配置文件来进行限制。标有 confined by 的进程已受 AppArmor 保护。



提示：更多信息

有关如何选择要构建配置文件的正确应用程序的详细信息，请参见第 29.2 节“确定要使其免疫的程序”。

28.4 构建和修改配置文件

SUSE Linux Enterprise Server 上的 AppArmor 随附了预配置的配置文件集，用于最重要的应用程序。此外，您也可使用 AppArmor 来为所需的任何应用程序创建您自己的配置文件。

管理配置文件有两种方式。一种是使用 YaST AppArmor 模块提供的图形前端，另一种是使用 AppArmor 套件自身提供的命令行工具。主要差别是，YaST 仅支持 AppArmor 配置文件的基本功能，而命令行工具可让您以更细微的方式更新/调整配置文件。

对每个应用程序执行以下步骤以创建配置文件：

1. 以 root 身份运行 **aa-genprof PROGRAM_NAME**，让 AppArmor 创建应用程序配置文件的大致轮廓。

或

通过运行 YaST > 安全和用户 > AppArmor 配置 > 手动添加配置文件并指定要构建配置文件的应用程序的完整路径，来创建基本配置文件的轮廓。

系统会创建新的基本配置文件的轮廓并将其置于学习模式，这意味着，它会记录您正在执行的程序的每个活动，但目前还不会对程序进行限制。

2. 运行应用程序的所有操作，让 AppArmor 完全了解程序的每个活动。
3. 在 aa-genprof 中键入 **s**，以便让 AppArmor 分析在步骤 2 中生成的日志文件。
AppArmor 扫描在程序运行期间记录的日志，然后请求您为每个记录的事件设置访问权限。请对每个文件进行设置或使用通配。
4. 依据应用程序的复杂性，可能必须重复步骤 2 和步骤 3。限制应用程序，在限制条件下执行应用程序并处理任何新的日志事件。要准确限制应用程序功能的完整范围，您可能必须经常重复此过程。
5. 完成 **aa-genprof** 后，您的配置文件即设置为强制模式。系统会应用该配置文件，而 AppArmor 将根据该配置文件限制应用程序。
如果某应用程序的现有配置文件处于控诉模式，对此应用程序启动 **aa-genprof** 时，此配置文件在退出此学习周期后仍会处于学习模式。有关更改配置文件模式的更多信息，请参见第 33.7.3.2 节 “aa-complain — 进入控诉或学习模式” 和第 33.7.3.6 节 “aa-enforce — 进入强制模式”。

使用您限制的应用程序执行所需的每一项任务，以测试您的配置文件设置。正常情况下，受限制的应用程序会顺利运行，您完全不会察觉到 AppArmor 活动。但是，如果您注意到应用程序的某些行为异常，请检查系统日志以查看 AppArmor 对应用程序的限制是否过于严格。根据系统上所使用的日志机制，可从以下几个位置查找 AppArmor 日志项：

/var/log/audit/audit.log

命令 **journalctl | grep -i apparmor**

命令 **dmesg -T**

要调整配置文件，可按第 33.7.3.9 节 “aa-logprof — 扫描系统日志” 所述再次分析与此应用程序相关的日志消息。发出提示时，请确定访问权限或限制。



提示：更多信息

有关配置文件构建和修改的更多信息，请参见第 30 章 “配置文件组件和语法”、第 32 章 “使用 YaST 构建和管理配置文件” 和第 33 章 “从命令行构建配置文件”。

28.5 更新您的配置文件

软件和系统配置会随着时间的流逝而更改。因此，可能需要不定期对 AppArmor 的配置文件设置进行一定的微调。AppArmor 会检查系统日志以查找策略违例或其他 AppArmor 事件，并使您能够相应地调整配置文件。不在任何配置文件定义范围内的任何应用程序行为均可通过 **aa-logprof** 解决。有关更多信息，请参见第 33.7.3.9 节 “**aa-logprof** — 扫描系统日志”。

29 使程序免疫

要有效强化计算机系统，您需要将可调解特权的程序数量降至最低，然后尽可能保护程序的安全。利用 AppArmor，您只需为环境中曝露于攻击下的程序构建配置文件，这极大地减少了强化计算机所需执行的工作量。AppArmor 配置文件可强制策略以确保程序仅执行所限定操作。

AppArmor 提供的免疫技术可以保护应用程序，免于其固有的漏洞所带来的风险。安装 AppArmor、设置 AppArmor 配置文件并重引导计算机后，您的系统即变为免疫系统，因为它已开始强制执行 AppArmor 安全策略。使用 AppArmor 保护程序的过程称为**免疫**。

管理员自己只需关注那些容易受到攻击的应用程序，并为它们生成配置文件。这样，系统的强化就简化为构建和维护 AppArmor 配置文件集，以及监视 AppArmor 报告功能记录的任何策略违规或异常。

用户应该察觉不到 AppArmor 的运行。它在“后台”运行，无需任何用户交互操作。AppArmor 不会对性能造成明显的影响。如果应用程序的某些活动没有包含在 AppArmor 配置文件中或者被 AppArmor 阻止，管理员需要调整此应用程序的配置文件。

AppArmor 建立一个默认应用程序配置文件集以保护标准的 Linux 服务。要保护其他应用程序，请使用 AppArmor 工具为您要保护的应用程序创建配置文件。本章介绍使程序免疫的基本原理。如果您已做好构建和管理 AppArmor 配置文件的准备，请转到[第 30 章“配置文件组件和语法”](#)、[第 32 章“使用 YaST 构建和管理配置文件”](#)或[第 33 章“从命令行构建配置文件”](#)。

AppArmor 会指定每个程序可以读取、写入和执行哪些文件，以及允许访问的网络类型，从而为网络服务提供优化的访问控制。这确保了每个程序只会执行它们应该执行的操作，而不会执行其他操作。AppArmor 会对程序进行检疫，以防止系统的其他部分被入侵的进程损坏。

AppArmor 是一套主机入侵防御或强制访问控制方案。以前，访问控制方案以用户为中心，因为它们是针对大型的分时共享系统而构建的。然而，现代网络服务器大多不允许用户登录，只是为用户提供各种网络服务（例如 Web、邮件、文件和打印服务器）。AppArmor 对给予网络服务和其它程序的访问进行控制，以防御对其缺陷的攻击。



提示：AppArmor 的背景信息

要更深入地全面了解 AppArmor 及其背后的总体概念，请参见[第 27.2 节“有关 AppArmor 配置文件构建的背景信息”](#)。

29.1 AppArmor 框架简介

本节提供当您运行 AppArmor 时“幕后”（以及 YaST 界面之下）所发生的情况的基本知识。

AppArmor 配置文件是包含路径项和访问权限的纯文本文件。有关详细的参考配置文件，请参见第 30.1 节“[分解 AppArmor 配置文件](#)”。AppArmor 例程会强制执行此文本文件中包含的指令来隔离进程或程序。

以下工具会参与 AppArmor 配置文件和策略的构建与强制执行：

aa-status

aa-status 可报告运行中 AppArmor 限制当前状态的各个方面。

aa-unconfined

aa-unconfined 可检测系统上正在运行并会监听网络连接且不受 AppArmor 配置文件保护的任何应用程序。有关此工具的详细信息，请参见第 33.7.3.12 节“[aa-unconfined — 识别不受保护的进程](#)”。

aa-autodep

aa-autodep 可为投放到生产环境之前需要充实的配置文件创建基本框架。生成的配置文件将被装载并置于控诉模式，将报告 AppArmor 规则（尚）未涵盖的应用程序的任何行为。有关此工具的详细信息，请参见第 33.7.3.1 节“[aa-autodep — 创建大概的配置文件](#)”。

aa-genprof

aa-genprof 可生成基本配置文件，并请求您通过执行应用程序并生成需要由 AppArmor 策略处理的日志事件来优化此配置文件。系统会通过一系列问题引导您处理应用程序执行期间触发的日志事件。生成配置文件后，系统会装载此配置文件并将其置于强制模式。有关此工具的详细信息，请参见第 33.7.3.8 节“[aa-genprof — 生成配置文件](#)”。

aa-logprof

aa-logprof 会以交互方式扫描和检查处于控诉与强制模式的 AppArmor 配置文件所限制的应用程序生成的日志项。它可以帮助您在相关配置文件中生成新的项。有关此工具的详细信息，请参见第 33.7.3.9 节“[aa-logprof — 扫描系统日志](#)”。

aa-easyprof

aa-easyprof 提供了便于使用的界面来生成 AppArmor 配置文件。**aa-easyprof** 支持使用模板和策略组来快速构建应用程序的配置文件。请注意，尽管此工具有助于生成策略，但其实用程序依赖于所用模板、策略组和抽象的质量。**aa-easyprof** 在创建配置文件方面的限制比使用 **aa-genprof** 和 **aa-logprof** 创建配置文件要少一些。

aa-complain

aa-complain 可将 AppArmor 配置文件从强制模式切换到控诉模式。系统会记录对配置文件中所设规则的违规，但不强制执行配置文件。有关此工具的详细信息，请参见第 33.7.3.2 节 “**aa-complain** — 进入控诉或学习模式”。

aa-enforce

aa-enforce 可将 AppArmor 配置文件从控诉模式切换到强制模式。系统会记录且不允许对配置文件中所设规则的违规 — 将强制执行配置文件。有关此工具的详细信息，请参见第 33.7.3.6 节 “**aa-enforce** — 进入强制模式”。

aa-disable

aa-disable 可禁用一个或多个 AppArmor 配置文件的强制模式。此命令将从内核中卸载配置文件，并防止在 AppArmor 启动时装载该配置文件。使用 **aa-enforce** 和 **aa-complain** 实用程序可更改此行为。

aa-exec

aa-exec 可启动指定的 AppArmor 配置文件和/或名称空间所限制的程序。如果同时指定了配置文件和名称空间，命令将由新策略名称空间中的配置文件限制。如果仅指定了名称空间，将使用当前限制的配置文件名称。如果配置文件和名称空间均未指定，将使用标准的配置文件附件运行命令 — 如同不结合 **aa-exec** 运行一样。

aa-notify

aa-notify 是一个便利的实用程序，它可在桌面环境中显示 AppArmor 通知。您还可以对它进行配置，以显示指定的最近几天的通知摘要。有关更多信息，请参见第 33.7.3.13 节 “**aa-notify**”。

29.2 确定要使其免疫的程序

现在您已熟悉 AppArmor，请开始选择要为其构建配置文件的应用程序。需要构建配置文件的程序是那些调解权限的程序。以下程序可以访问使用此程序的用户所不能访问的资源，因此使用这些程序时可以授予用户权限：

cron 作业

cron 定期运行的程序。此类程序会读取来自多个来源的输入，可以使用特权运行，有时甚至可以使用 root 特权运行。例如，cron 可以每日运行 /usr/sbin/logrotate 来轮换、压缩系统日志，甚至可以通过邮件发送系统日志。要了解如何查找此类程序，请参见第 29.3 节“使 cron 作业免疫”。

Web 应用程序

网页浏览器可以调用的程序，包括 CGI Perl 脚本、PHP 页面以及更复杂的 Web 应用程序。要了解如何查找此类程序，请参见第 29.4.1 节“使 Web 应用程序免疫”。

网络代理

具有开放网络端口的程序（服务器端和客户端）。邮件客户端和网页浏览器等用户客户端会调解特权。这些程序在运行时具有书写用户主目录的权限，而且他们会处理来自恶意的远程来源的输入，如恶意的网站和通过电子邮件发送的恶意代码。要了解如何查找此类程序，请参见第 29.4.2 节“使网络代理免疫”。

相反，您不必为非特权程序构建配置文件。例如，外壳脚本可以调用 cp 程序来复制文件。由于 cp 默认没有自身的配置文件或子配置文件，它将继承父外壳脚本的配置文件。因此，cp 可以复制父外壳脚本的配置文件能够读取和写入的任何文件。

29.3 使 cron 作业免疫

要查找由 cron 运行的程序，请检查您的本地 cron 配置。遗憾的是，cron 配置非常复杂，因此需要检查大量的文件。定期的 cron 作业是基于以下文件运行的：

```
/etc/crontab
/etc/cron.d/*
/etc/cron.daily/*
/etc/cron.hourly/*
```

```
/etc/cron.monthly/*  
/etc/cron.weekly/*
```

crontab 命令会列出/编辑当前用户的 crontab。要操作 root 的 cron 作业，请先转变为 root 用户，然后使用 **crontab -e** 编辑任务或使用 **crontab -l** 列出任务。

29.4 使网络应用程序免疫

使用 **aa-unconfined** 工具可以自动查找应构建配置文件的网络服务器守护程序。

aa-unconfined 工具使用 **netstat -nlp** 命令来检查计算机内部的开放端口、检测与这些端口相关联的程序，以及检查已装载的 AppArmor 配置文件集。然后，**aa-unconfined** 工具会报告这些程序以及与每个程序相关联的 AppArmor 配置文件，如果程序不受限制，则报告 “none”。



注意

如果您要创建新配置文件，必须重新启动已构建配置文件的程序，使其受到 AppArmor 的有效限制。

下面是 **aa-unconfined** 输出示例：

```
3702 ① /usr/sbin/sshd ② confined  
    by '/usr/sbin/sshd ③ (enforce)'  
4040 /usr/sbin/smbd confined by '/usr/sbin/smbd (enforce)'  
4373 /usr/lib/postfix/master confined by '/usr/lib/postfix/master (enforce)'  
4505 /usr/sbin/httpd2-prefork confined by '/usr/sbin/httpd2-prefork (enforce)'  
646  /usr/lib/wicked/bin/wickedd-dhcp4 not confined  
647  /usr/lib/wicked/bin/wickedd-dhcp6 not confined  
5592 /usr/bin/ssh not confined  
7146 /usr/sbin/cupsd confined by '/usr/sbin/cupsd (complain)'
```

- ① 第一部分是编号。此编号是监听程序的进程 ID 编号 (PID)。
- ② 第二部分是一个字符串，代表监听程序的绝对路径
- ③ 最后部分表示限制此程序的配置文件（如果存在）。



注意

aa-unconfined 需要 root 特权，且不应通过 AppArmor 配置文件限制的外壳运行。

aa-unconfined 不区分网络接口，因此会报告所有未受限的进程，甚至是可能正在监听内部 LAN 接口的进程。

用户网络客户端应用程序的查找视用户的自选设置而定。**aa-unconfined** 工具会检测并报告客户端应用程序打开的网络端口，但仅限于执行 **aa-unconfined** 分析时正在运行的客户端应用程序。这是一个问题，因为网络服务一般不间断运行，而网络客户端应用程序通常只在用户有兴趣时运行。

向用户网络客户端应用程序应用 AppArmor 配置文件的方式还取决于用户的偏好。因此，我们将用户网络客户端应用程序的配置文件构建作为留给用户的练习。

为了更主动地限制桌面应用程序，**aa-unconfined** 命令支持 `--paranoid` 选项，该选项会报告所有正在运行的进程，以及可能与各个进程相关联或不关联的对应 AppArmor 配置文件。这样用户就可以确定各个程序是否需要 AppArmor 配置文件。

如果您有新的或修改过的配置文件，可连同您使用的应用程序的行为用例，提交到 apparmor@lists.ubuntu.com 邮件列表。AppArmor 团队将审查该配置文件，并可能会将最终成果提交到 SUSE Linux Enterprise Server 中。我们无法保证包含每个配置文件，但会尽力包含尽可能多的配置文件。

29.4.1 使 Web 应用程序免疫

要查找 Web 应用程序，请检查您的 Web 服务器配置。Apache Web 服务器的可配置性比较高，您可以将 Web 应用程序保存在多个目录中，这取决于本地配置。默认情况下，SUSE Linux Enterprise Server 将 Web 应用程序储存在 /srv/www/cgi-bin/ 中。应尽最大可能使每个 Web 应用程序都有一个 AppArmor 配置文件。

找到这些程序后，可以使用 **aa-genprof** 和 **aa-logprof** 工具创建或更新其 AppArmor 配置文件。

由于 CGI 程序通过 Apache Web 服务器执行，因此您必须对 Apache 自身的配置文件 usr.sbin.httpd2-prefork（适用于 SUSE Linux Enterprise Server 上的 Apache 2）进行修改，以添加对每个 CGI 程序的执行权限。例如，添加 /srv/www/cgi-bin/

`my_hit_counter.pl rPx` 一行可授予 Apache 执行 Perl 脚本 `my_hit_counter.pl` 的权限，而且要求存在专用于 `my_hit_counter.pl` 的配置文件。如果 `my_hit_counter.pl` 不具备与之关联的专用配置文件，则规则应为 `/srv/www/cgi-bin/my_hit_counter.pl rix`，从而让 `my_hit_counter.pl` 继承 `usr.sbin.httpd2-prefork` 配置文件。

某些用户可能感觉为 Apache 可能调用的每个 CGI 脚本指定执行权限比较繁琐。管理员可以将一定的访问权限授予 CGI 脚本的集合，这是一种替代方法。例如，添加 `/srv/www/cgi-bin/*.{pl,py,pyc} rix` 一行将允许 Apache 执行 `/srv/www/cgi-bin/` 中所有以 `.pl`（Perl 脚本）和 `.py` 或 `.pyc`（Python 脚本）结尾的文件。如上所示，规则的 `ix` 部分将使 Python 脚本继承 Apache 配置文件，这适用于您不想为每个 CGI 脚本编写单独的配置文件的情况。



注意

在 Web 应用程序处理 Apache 模块（`mod_perl` 和 `mod_php`）时，如果您需要子进程限制模块（`apache2-mod-apparmor`）功能，请在于 YaST 或命令行中添加配置文件时使用 `ChangeHat` 功能。要利用子进程限制，请参见第 34.2 节“管理 `ChangeHat` 感知型应用程序”。

对于使用 `mod_perl` 和 `mod_php` 的 Web 应用程序，构建其配置文件所需要的处理略有不同。在这种情况下，“program”是 Apache 进程内的模块直接解释的脚本，因此不进行执行。而 AppArmor 版的 Apache 使用与所请求 URI 的名称对应的子配置文件（“帽子”）来调用 `change_hat()`。



注意

要执行的脚本所呈现的名称可能不是 URI，取决于 Apache 被配置为在何处查找模块脚本。如果您之前将 Apache 配置为将脚本放置在其他位置，当 AppArmor 指出访问违规事件时，日志文件中会出现不同的名称。请参见第 36 章“管理已构建配置文件的应用程序”。

对于 `mod_perl` 和 `mod_php` 脚本，这是请求的 Perl 脚本或 PHP 页面的名称。例如，添加以下子配置文件将允许 `localtime.php` 页面执行并访问本地系统时间和区域设置文件：

```
/usr/bin/httpd2-prefork {
```

```
# ...
^/cgi-bin/localtime.php {
    /etc/localtime                r,
    /srv/www/cgi-bin/localtime.php r,
    /usr/lib/locale/**            r,
}
}
```

如果尚未定义子配置文件，AppArmor 版的 Apache 将应用 `DEFAULT_URI` 帽子。要显示网页，使用此子配置文件便足以满足需求。AppArmor 在默认情况下提供的 `DEFAULT_URI` 帽子如下所示：

```
^DEFAULT_URI {
    /usr/sbin/suexec2             mixr,
    /var/log/apache2/**           rwl,
    @{HOME}/public_html          r,
    @{HOME}/public_html/**       r,
    /srv/www/htdocs               r,
    /srv/www/htdocs/**           r,
    /srv/www/icons/*.{gif,jpg,png} r,
    /srv/www/vhosts               r,
    /srv/www/vhosts/**           r,
    /usr/share/apache2/**        r,
    /var/lib/php/sess_*          rwl
}
```

要将单个 AppArmor 配置文件用于 Apache 处理的所有网页和 CGI 脚本，编辑 `DEFAULT_URI` 子配置文件是个不错的方法。有关使用 Apache 限制 Web 应用程序的详细信息，请参见第 34 章 “使用 ChangeHat 构建 Web 应用程序的配置文件”。

29.4.2 使网络代理免疫

要查找需要构建配置文件的网络服务器守护程序和网络客户端（例如 **fetchmail** 或 Firefox），您应检查计算机上的开放端口。另外，还请考虑通过这些端口响应的程序，并为其尽量多的程序提供配置文件。如果您为具有开放网络端口的所有程序提供了配置文件，那么攻击者不突破 AppArmor 配置文件策略，就无法进入计算机上的文件系统。

使用扫描程序（例如 nmap）从计算机外部手动扫描服务器上的开放网络端口，或以 root 身份使用 **netstat --inet -n -p** 命令从计算机内部扫描。然后检查计算机，以确定哪些程序通过所发现的开放端口进行响应。



提示

有关所有可能的选项的详细参考信息，请参见 netstat 命令的手册页。

30 配置文件组件和语法

构建 AppArmor 配置文件来限制应用程序的操作非常简单且直观。AppArmor 附带了多种工具来帮助创建配置文件。您无需编程或处理脚本。管理员唯一需要执行的任务是为每个需要强化的应用程序确定最严格的访问和执行权限策略。

只有在软件配置或所需的活动范围发生变化时才有必要更新或修改应用程序配置文件。AppArmor 提供了直观的工具来处理配置文件的更新和修改。

选择要构建配置文件的程序后，您就作好了构建 AppArmor 配置文件的准备。要执行此操作，必须了解配置文件的组件和语法。AppArmor 配置文件包含多个可帮助您构建简单且可重用配置文件代码的构建基块：

Include 文件

Include 语句用于提取其他 AppArmor 配置文件的组成部分，可简化新配置文件的结构。

抽象

抽象是按常见应用程序任务分组的 include 语句。

程序块

程序块是包含专用于程序套件的配置文件块的 include 语句。

功能项

功能项是任何 POSIX.1e <http://en.wikipedia.org/wiki/POSIX#POSIX.1> Linux 功能的配置文件项，可用于精细控制允许受限制进程通过需要特权的系统调用执行哪些操作。

网络访问控制项

网络访问控制项基于地址类型和地址族调解网络访问。

局部变量定义

局部变量定义路径的快捷方式。

文件访问控制项

文件访问控制项指定应用程序可访问的文件集。

rlimit 项

rlimit 项设置和控制应用程序的资源限制。

如需有关确定要构建配置文件的程序的帮助，请参见第 29.2 节 “确定要使其免疫的程序”。要开始使用 YaST 构建 AppArmor 配置文件，请转到第 32 章 “使用 YaST 构建和管理配置文件”。要使用 AppArmor 命令行界面构建配置文件，请转到第 33 章 “从命令行构建配置文件”。

有关创建 AppArmor 配置文件的更多细节，请参见 [man 5 apparmor](#)。

30.1 分解 AppArmor 配置文件

要介绍配置文件的构成以及创建配置文件的过程，最简单的方法就是显示示例配置文件的细节，本例使用了名为 `/usr/bin/foo` 的虚构应用程序的配置文件：

```
#include <tunables/global> ❶

# a comment naming the application to confine
/usr/bin/foo ❷ { ❸
    #include <abstractions/base> ❹

    capability setgid ❺,
    network inet tcp ❻,

    link /etc/sysconfig/foo -> /etc/foo.conf, ❼

    /bin/mount                ux,
    /dev/{,u} ❽ random        r,
    /etc/ld.so.cache          r,
    /etc/foo/*                 r,
    /lib/ld-*.so*             mr,
    /lib/lib*.so*             mr,
    /proc/[0-9]**             r,
    /usr/lib/**               mr,
    /tmp/                     r, ❾
    /tmp/foo.pid              wr,
    /tmp/foo.*                lrw,
    /@{HOME} ❿ /.foo_file     rw,
    /@{HOME}/.foo_lock        kw,
    owner ❿ /shared/foo/**     rw,
    /usr/bin/foobar           Cx, ⓫
```

```

/bin/**                                Px -> bin_generic, ⑬

# a comment about foo's local (children) profile for /usr/bin/foobar.

profile /usr/bin/foobar ⑭ {
    /bin/bash                        rmix,
    /bin/cat                         rmix,
    /bin/more                       rmix,
    /var/log/foobar*                rwl,
    /etc/foobar                     r,
}

# foo's hat, bar.
^bar ⑮ {
    /lib/ld-*.so*                  mr,
    /usr/bin/bar                   px,
    /var/spool/*                   rwl,
}
}

```

- ① 此语句装载包含变量定义的文件。
- ② 受限制程序的规范化路径。
- ③ 花括号 ({}) 充当 include 语句、子配置文件、路径项、功能项和网络项的容器。
- ④ 此指令提取 AppArmor 配置文件的组件以简化配置文件。
- ⑤ 功能项语句可启用每个 29 POSIX.1e 草案功能。
- ⑥ 确定允许应用程序进行哪种网络访问的指令。有关详细信息，请参考 第 30.5 节 “网络访问控制”。
- ⑦ 链接对规则，指定链接的源和目标。有关更多信息，请参见 第 30.7.6 节 “链接对”。
- ⑧ 此处的花括号 ({}) 允许所列的每个可能的值，其中一个可能的值为空字符串。
- ⑨ 路径项，指定程序可以访问文件系统的哪些区域。路径项的第一部分指定文件的绝对路径（包括正则表达式通配），第二部分指示允许的访问模式（例如 r 表示读，w 表示写，x 表示执行）。路径名的开头可以包含任何类型的空白字符（空格或制表符），但必须以空格分隔路径名和模式说明符。可以选择用空格分隔各访问模式并在末端包含尾随逗号。第 30.7 节 “文件权限访问模式” 中提供了可用访问模式的综合概述。

- 10 此变量将扩展为一个无需更改整个配置文件即可更改的值。
- 11 拥有者条件规则，授予对用户所拥有文件的读写权限。有关更多信息，请参考第 30.7.8 节“拥有者条件规则”。
- 12 此项定义转换到本地配置文件 `/usr/bin/foobar`。第 30.12 节“执行模式”中提供了可用执行模式的综合概述。
- 13 目标为位于全局范围内的 `bin_generic` 配置文件的命名配置文件转换。有关详细信息，请参见第 30.12.7 节“命名配置文件转换”。
- 14 本地配置文件 `/usr/bin/foobar` 在此部分中定义。
- 15 此部分引用了应用程序的“帽子”子配置文件。有关 AppArmor ChangeHat 功能的更多细节，请参见第 34 章“使用 ChangeHat 构建 Web 应用程序的配置文件”。

为程序创建配置文件后，此程序只可访问配置文件中指定的文件、模式和 POSIX 功能。这些限制是对本机 Linux 访问控制的补充。

示例： 要获得 `CAP_CHOWN` 功能，程序必须能够在常规 Linux 访问控制下访问 `CAP_CHOWN`（通常为 `root` 拥有的进程），并且其配置文件中必须设置 `chown` 功能。与此类似，要能够写入 `/foo/bar` 文件，程序的文件属性中必须设置正确的用户 ID 和模式位，并且其配置文件中必须设置 `/foo/bar w`。

违反 AppArmor 规则的尝试将记录在 `/var/log/audit/audit.log`（如果已安装 `audit` 软件包）、`/var/log/messages` 中，或仅记录在 `journalctl` 中（如果未安装传统的 `syslog`）。AppArmor 规则通常可防止攻击发挥作用，因为必要的文件不可访问。在任何情况下，AppArmor 限制都可以禁止攻击者可能对 AppArmor 所允许的文件集进行的破坏。

30.2 配置文件类型

AppArmor 可识别四种不同类型的配置文件：标准配置文件、未关联的配置文件、本地配置文件和帽子。标准配置文件和未关联的配置文件是独立的配置文件，各自储存在 `/etc/apparmor.d/` 下的某个文件中。本地配置文件和帽子是在父配置文件内部嵌入的子配置文件，用于针对应用程序的子任务提供更严格或备选的限制。

30.2.1 标准配置文件

默认的 AppArmor 配置文件将按其名称关联到程序，因此，配置文件的名称必须与其要限制的应用程序的路径相匹配。

```
/usr/bin/foo {  
...  
}
```

每当未受限的进程执行 `/usr/bin/foo` 时，就会自动使用此配置文件。

30.2.2 未关联的配置文件

未关联的配置文件不会驻留在文件系统名称空间中，因此不会自动关联到应用程序。未关联的配置文件的名称前面带有关键字 `profile`。您可以自由选择配置文件名称，但存在以下限制：名称不得以 `:` 或 `.` 字符开头。如果名称包含空格，必须将它括在引号中。如果名称以 `/` 开头，则将该配置文件视为标准配置文件，因此以下两个配置文件是相同的：

```
profile /usr/bin/foo {  
...  
}  
/usr/bin/foo {  
...  
}
```

系统永远不会自动使用未关联的配置文件，也不能通过 `Px` 规则将其他配置文件转换为未关联的配置文件。需使用命名配置文件转换（请参见第 30.12.7 节“命名配置文件转换”）或 `change_profile` 规则（请参见第 30.2.5 节“更改规则”）将其关联到程序。

一般不应由系统范围的配置文件（例如 `/bin/bash`）限制的系统实用程序的专用配置文件适合采用未关联的配置文件。它们还可用于设置角色或限制用户。

30.2.3 本地配置文件

本地配置文件可让您方便地针对受限制应用程序启动的实用程序提供专门的限制。其指定方式类似于标准配置文件，不过，它们嵌入于父配置文件中，并以 `profile` 关键字开头：

```
/parent/profile {  
    ...  
    profile /local/profile {  
        ...  
    }  
}
```

要转换为本地配置文件，请使用 `cx` 规则（请参见第 30.12.2 节“离散本地配置文件执行模式 (cx)”）或命名配置文件转换（请参见第 30.12.7 节“命名配置文件转换”）。

30.2.4 帽子

AppArmor “帽子”属于本地配置文件，它们存在一些额外的限制，以及允许使用 `change_hat` 转换到这些配置文件的隐式规则。有关详细说明，请参见第 34 章“使用 ChangeHat 构建 Web 应用程序的配置文件”。

30.2.5 更改规则

AppArmor 提供了 `change_hat` 和 `change_profile` 规则，用于控制域转换。`change_hat` 通过在配置文件中定义帽子来指定，而 `change_profile` 规则会引用另一个配置文件，并以关键字 `change_profile` 开头：

```
change_profile -> /usr/bin/foobar,
```

`change_hat` 和 `change_profile` 都提供应用程序导向的配置文件转换，而无需启动单独的应用程序。`change_profile` 在装载的任何配置文件之间均提供通用的单向转换。`change_hat` 提供可回转的父子转换，其中，应用程序可从父配置文件切换到帽子配置文件，如果它提供正确的机密密钥，则稍后可恢复为父配置文件。

`change_profile` 最适合用于应用程序需要经历可信设置阶段，随后可以降低其特权级别的情况。在启动阶段映射或打开的任何资源在配置文件发生更改后仍可访问，但新配置文件将限制新资源的打开，甚至会限制在转变之前打开的某些资源。具体而言，在可以限制功能和文件资源（前提是它们未经过内存映射）的情况下，内存资源仍然可用。

`change_hat` 最适合用于应用程序需运行不提供应用程序资源（例如 Apache 的 `mod_php`）直接访问途径的虚拟机或解释器的情况。由于 `change_hat` 将返回机密密钥储存在应用程序的内存中，因此在特权降级阶段，不应具有直接访问内存的权限。正确分隔文件访问权限也很重要，因为帽子可以限制对文件句柄的访问，但不会关闭文件句柄。如果应用程序在进行缓冲并通过缓冲提供对所打开文件的访问，内核可能看不到对这些文件的访问，因此新配置文件不会限制此类访问。



警告：域转换的安全性

`change_hat` 和 `change_profile` 域转换不如通过执行完成的域转换安全，因为它们不会影响进程的内存映射，也不会关闭已打开的资源。

30.3 Include 语句

Include 语句是可提取其他 AppArmor 配置文件的组件以简化配置文件的指令。Include 文件会检索程序的访问权限。通过使用 `include`，您可以向程序赋予访问其它程序也需要的目录路径和文件的权限。使用 `include` 可减小配置文件的大小。

Include 语句通常以井号 (`#`) 开头。这会造成混淆，因为配置文件中的注释也使用井号。因此，仅当不存在前置 `#` (`##include` 是注释) 并且 `#` 与 `include` 之间不存在空格 (`# include` 是注释) 时，才将 `#include` 视为 `include`。

您也可以使用不带前导 `#` 的 `include`。

```
include "/etc/apparmor.d/abstractions/foo"
```

等同于使用

```
#include "/etc/apparmor.d/abstractions/foo"
```



注意：无尾随 “,”

请注意，由于 `include` 遵循 C 预处理器语法，因此不含尾随的 “,”，这与大多数 AppArmor 规则一样。

您可以通过在语法中进行细微的更改来修改 `include` 的行为。如果在包含路径两侧使用 `"`，则会指示解析器执行绝对或相对路径查找。

```
include "/etc/apparmor.d/abstractions/foo"    # absolute path
include "abstractions/foo"    # relative path to the directory of current file
```

请注意，在使用相对路径 `include` 时，如果包含了文件，则会将此文件视为其 `include` 的当前新文件。例如，假设您在 `/etc/apparmor.d/bar` 文件中操作，那么

```
include "abstractions/foo"
```

会包含文件 `/etc/apparmor.d/abstractions/foo`。如果

```
include "example"
```

存在于 `/etc/apparmor.d/abstractions/foo` 文件中，则其会包含 `/etc/apparmor.d/abstractions/example`。

使用 `<>` 会指定按顺序尝试 `include` 路径（由 `-I` 指定，默认为 `/etc/apparmor.d` 目录）。假设 `include` 路径为

```
-I /etc/apparmor.d/ -I /usr/share/apparmor/
```

则 `include` 语句

```
include <abstractions/foo>
```

将尝试 `/etc/apparmor.d/abstractions/foo`，如果该文件不存在，则下一次会尝试 `/usr/share/apparmor/abstractions/foo`。



提示

可以手动覆盖默认的 `include` 路径，方法是将 `-I` 传递给 `apparmor_parser`，或者在 `/etc/apparmor/parser.conf` 中设置 `include` 路径：

```
Include /usr/share/apparmor/
Include /etc/apparmor.d/
```

允许多个项，其提取顺序与在 `apparmor_parser` 命令行中使用 `-I` 或 `--Include` 时的顺序相同。

如果 include 以 “/” 结尾，则会将它视为目录 include，并会包含该目录中的所有文件。为帮助您构建应用程序的配置文件，AppArmor 提供了三类 include：抽象、程序块和 tunable。

30.3.1 抽象

抽象是按常见应用程序任务分组的 include。这些任务包括访问身份验证机制、访问名称服务例程、一般的图形要求以及系统统计。这些抽象中列出的文件特定于命名任务。需要其中某个文件的程序通常也需要抽象文件中列出的其他文件（取决于程序的本地配置和具体要求）。[/etc/apparmor.d/abstractions](#) 中提供了抽象。

30.3.2 程序块

program-chunks 目录 ([/etc/apparmor.d/program-chunks](#)) 中包含一些专用于程序套件的配置文件块，这些块在套件外部一般没有作用，因此，配置文件向导 (**aa-logprof** 和 **aa-genprof**) 从不建议在配置文件中使用的这些配置文件块。目前，程序块仅适用于 postfix 程序套件。

30.3.3 Tunables

tunables 目录 ([/etc/apparmor.d/tunables](#)) 包含全局变量定义。在配置文件中使用时，这些变量将扩展为一个无需更改整个配置文件即可更改的值。请将应该可供每个配置文件使用的所有 tunable 定义添加到 [/etc/apparmor.d/tunables/global](#)。

30.4 功能项 (POSIX.1e)

功能规则很简单，就是 `capability` 一词后接 POSIX.1e 功能名称（如 [capabilities\(7\)](#) 手册页中所定义）。您可以在单条规则中列出多个功能，或者仅使用关键字 `capability` 授予所有已实现的功能。

```
capability dac_override sys_admin,    # multiple capabilities
```

```
capability, # grant all capabilities
```

30.5 网络访问控制

AppArmor 允许基于地址类型和地址族调解网络访问。下面将说明网络访问规则语法：

```
network [[<domain> ❶][<type ❷>][<protocol ❸>]]
```

❶ 支持的

域：inet、ax25、ipx、appletalk、netrom、bridge、x25、inet6、rose、netbeui、secu

❷ 支持的类型：stream、dgram、seqpacket、rdm、raw、packet

❸ 支持的协议：tcp、udp、icmp

AppArmor 工具仅支持族和类型规范。AppArmor 模块在“ACCESS DENIED”消息中仅会发出 network DOMAIN TYPE。配置文件生成工具（YaST 和命令行）仅输出这些内容。

下面的示例说明可在 AppArmor 配置文件中使用的可能网络相关规则。请注意，AppArmor 工具目前不支持最后两条规则的语法。

```
network ❶,  
network inet ❷,  
network inet6 ❸,  
network inet stream ❹,  
network inet tcp ❺,  
network tcp ❻,
```

❶ 允许所有网络。不应用与域、类型或协议相关的限制。

❷ 允许 IPv4 网络的一般用法。

❸ 允许 IPv6 网络的一般用法。

❹ 允许使用 IPv4 TCP 网络。

❺ 允许使用 IPv4 TCP 网络（上一条规则的释义）。

❻ 允许使用 IPv4 和 IPv6 TCP 网络。

30.6 配置文件名称、标志、路径和通配

通常，通过指定程序可执行文件的完整路径来将配置文件关联到相应程序。例如，对于标准配置文件（请参见第 30.2.1 节 “标准配置文件”），通过以下方式来定义配置文件

```
/usr/bin/foo { ... }
```

下列各节介绍在命名配置文件、将某个配置文件放入其他现有配置文件的环境中，或者指定文件路径时可以运用的若干有用技巧。

AppArmor 显式区分目录路径名和文件路径名。对需要显式区分的任何目录路径使用尾随 /：

/some/random/example/* r
允许对 /some/random/example 目录中的文件进行读取访问。

/some/random/example/ r
仅允许对该目录进行读取访问。

/some/**/ r
授予对 /some 下的任何目录（但不包括 /some/ 本身）的读取访问权限。

/some/random/example/** r
授予对 /some/random/example 下的文件和目录（但不包括 /some/random/example/ 本身）的读取访问权限。

/some/random/example/**[^/] r
授予对 /some/random/example 下的文件的读取访问权限。显式排除目录 ([^/])。

通配（亦称为常规表达式匹配）是您在修改目录路径时使用通配符将一组文件或子目录包含在内的情况。使用通配语法可以指定文件资源，类似于常用的外壳（如 csh、bash 和 zsh）使用的通配语法。

<u>*</u>	代替任意数目的任何字符， <u>/</u> 除外。 示例：任意数目的文件路径元素。
<u>**</u>	代替任意数目的字符，包括 <u>/</u> 。

	示例：任意数目的路径元素，包括整个目录。
<u>?</u>	代替任意单独字符， <u>/</u> 除外。
<u>[abc]</u>	代替一个字符 <u>a</u> 、 <u>b</u> 或 <u>c</u> 。 示例：匹配 <u>/home[01]/*/.plan</u> 的规则允许程序访问 <u>/home0</u> 和 <u>/home1</u> 中的用户的 <u>.plan</u> 文件。
<u>[a-c]</u>	代替一个字符 <u>a</u> 、 <u>b</u> 或 <u>c</u> 。
<u>{ab,cd}</u>	扩展为一条匹配 <u>ab</u> 的规则，以及一条匹配 <u>cd</u> 的规则。 示例：匹配 <u>/usr, www}/pages/**</u> 的规则授予对 <u>/usr/pages</u> 和 <u>/www/pages</u> 中的网页的访问权限。
<u>[^a]</u>	代替任何字符， <u>a</u> 除外。

30.6.1 配置文件标志

配置文件标志控制相关配置文件的行为。您可以通过手动编辑配置文件定义来将配置文件标志添加到其中。请参见以下语法：

```
/path/to/profiled/binary flags=(list_of_flags) {
    [...]
}
```

可以使用以逗号 “,” 或空格 “ ” 分隔的多个标志。配置文件标志有三种基本类型：模式、相对和附加标志。

模式标志为 complain（允许并记录非法访问）。如果省略该标志，则配置文件处于强制模式（强制执行策略）。



提示

将整个配置文件设置为控诉模式的更灵活的方式是在 `/etc/apparmor.d/force-complain/` 目录中基于该配置文件创建一个符号链接。

```
ln -s /etc/apparmor.d/bin.ping /etc/apparmor.d/force-complain/bin.ping
```

相对标志为 `chroot_relative`（指出配置文件相对于 `chroot` 而不是名称空间）或 `namespace_relative`（默认值，表示路径相对于 `chroot` 外部）。这两个标志是互斥的。

附加标志由两对互斥的标志构成：`attach_disconnected` 或 `no_attach_disconnected`（确定解析为名称空间外部的路径名是否附加到根目录，即，它们的开头是否包含 “/” 字符），`chroot_attach` 或 `chroot_no_attach`（在 `chroot` 环境中访问位于 `chroot` 外部但在名称空间内部的文件时，控制路径名生成）。

30.6.2 在配置文件中使用的变量

AppArmor 允许在配置文件中使用的变量来包含路径。使用全局变量可使配置文件具有可移植性，使用局部变量可以创建路径的快捷方式。

举个典型的示例，在用户主目录装入到不同位置的网络方案中，全局变量就很方便。您无需在所有受影响的配置文件中修改主目录的路径，而只需更改变量的值。全局变量在 `/etc/apparmor.d/tunables` 下定义，需要通过 `include` 语句来使用。`/etc/apparmor.d/tunables/home` 文件中提供了此用例的变量定义（`@{HOME}` 和 `@{HOMEDIRS}`）。

局部变量在配置文件的头部定义。这样便于提供 `chroot` 路径的基础，例如：

```
@{CHROOT_BASE}=/tmp/foo
/sbin/rsyslogd {
...
# chrooted applications
@{CHROOT_BASE}/var/lib/*/dev/log w,
@{CHROOT_BASE}/var/log/** w,
...
}
```

在下面的示例中，@{HOMEDIRS} 会列出所有用户主目录的储存位置，@{HOME} 是主目录的空格分隔列表。接下来，@{HOMEDIRS} 会按用于储存用户主目录的两个新特定位置进行扩展。

```
@{HOMEDIRS}=/home/  
@{HOME}=@{HOMEDIRS}/*/ /root/  
[...]  
@{HOMEDIRS}+=/srv/nfs/home/ /mnt/home/
```



注意

在当前的 AppArmor 工具中，只能在手动编辑和维护配置文件时使用变量。

30.6.3 模式匹配

配置文件名称可以包含通配表达式，这样配置文件便可匹配多个二进制文件。

下面的示例适用于 **foo** 二进制文件驻留在 /usr/bin 或 /bin 中的系统。

```
/usr/,bin/foo { ... }
```

在下面的示例中，对可执行文件 /bin/foo 进行匹配时，/bin/foo 配置文件是完全匹配项，因此已将其选中。对于可执行文件 /bin/fat，配置文件 /bin/foo 不匹配，并且由于 /bin/f* 配置文件比 /bin/** 更具体（较不宽泛），因此选择了 /bin/f* 配置文件。

```
/bin/foo { ... }  
  
/bin/f* { ... }  
  
/bin/** { ... }
```

有关配置文件名称通配示例的详细信息，请参见 AppArmor 的手册页 [man 5 apparmor.d](#) 以及“[通配](#)”一节。

30.6.4 名称空间

名称空间用于提供不同的配置文件集。例如，一个配置文件集用于系统，另一个配置文件集用于 chroot 环境或容器。名称空间是分层的 — 名称空间可以看到其子项，但子项看不到其父项。名称空间的名称以冒号 `:` 开头，后接一个字母数字字符串、一个尾随冒号 `:` 和一个可选的双斜线 `//`，例如

```
:childNameSpace://
```

装载到子名称空间的配置文件以其名称空间名称为前缀（从父项的角度看）：

```
:childNameSpace://apache
```

可以通过 `change_profile` API 或命名配置文件转换进入名称空间：

```
/path/to/executable px -> :childNameSpace://apache
```

30.6.5 配置文件命名和附件规范

配置文件可以有一个名称和一个附件规范。这样，您便可为配置文件指定一个比包含模式匹配（请参见第 30.6.3 节“模式匹配”）的名称更有意义且符合逻辑的名称，便于用户/管理员识别。例如，默认配置文件

```
/** { ... }
```

可命名为

```
profile default /** { ... }
```

另外，可为包含模式匹配的配置文件命名。例如：

```
/usr/lib64/firefox*/firefox-*bin { ... }
```

可命名为

```
profile firefox /usr/lib64/firefox*/firefox-*bin { ... }
```

30.6.6 别名规则

别名规则提供了另一种操作站点特定布局的配置文件路径映射的方式。它们是另一种修改路径的方式（一种方式是使用变量），在解析变量后执行。别名规则告知要将具有相同源前缀的规则看作是规则位于目标前缀处。

```
alias /home/ -> /usr/home/
```

前缀与 /home/ 匹配的所有规则都将允许访问 /usr/home/。例如，

```
/home/username/** r,
```

也允许访问

```
/usr/home/username/** r,
```

利用别名，您无需重新编写规则即可快速重新映射它们。它们可确保源路径仍可供访问 — 在本示例中，别名规则确保 /home/ 下的路径仍可供访问。

使用别名规则可以同时指向多个目标。

```
alias /home/ -> /usr/home/  
alias /home/ -> /mnt/home/
```



注意

在当前的 AppArmor 工具中，只能在手动编辑和维护配置文件时使用别名规则。



提示

请在文件 /etc/apparmor.d/tunables/alias 中插入全局别名定义。

30.7 文件权限访问模式

文件权限访问模式包括以下模式的组合：

<u>r</u>	读取模式
<u>w</u>	写入模式（与 <u>a</u> 互斥）
<u>a</u>	追加模式（与 <u>w</u> 互斥）
<u>k</u>	文件锁定模式
<u>l</u>	链接模式
<u>link</u> FILE -> TARGET	链接对规则（不能与其他访问模式结合使用）

30.7.1 读取模式 (r)

允许程序拥有读取资源的权限。必需对外壳脚本和其他解释内容授予读取访问权限，该权限确定正在执行的进程是否可以进行核心转储。

30.7.2 写入模式 (w)

允许程序拥有写入资源的权限。将被取消链接（删除）的文件必须拥有此权限。

30.7.3 追加模式 (a)

允许程序写入到文件的末尾。与 w 模式相反，追加模式不包含重写数据、重命名或删除文件的功能。追加权限通常用于需要能够写入日志文件，但不应该能够操作日志文件中任何现有数据的应用程序。由于追加权限是与写入模式关联的权限的子集，w 和 a 权限标志不能结合使用，它们是互斥的。

30.7.4 文件锁定模式 (k)

应用程序可以采用文件锁。在以前的 AppArmor 版本中，如果应用程序有权访问文件，AppArmor 便允许锁定文件。通过使用独立的文件锁定模式，AppArmor 可确保仅对需要锁定的文件进行锁定，如此可增强安全性，因为在多种拒绝服务攻击场景中都可以使用锁定。

30.7.5 链接模式 (l)

链接模式调解对硬链接的访问。创建链接后，目标文件的访问权限必须与所创建的链接相同（但目标不需要链接访问权限）。

30.7.6 链接对

链接模式授予链接到任意文件的权限，前提是该链接具有目标所授予的权限的子集（子集权限测试）。

```
/srv/www/htdocs/index.html rl,
```

通过指定源和目标，链接对规则可让您更好地控制创建硬链接的方式。默认情况下，链接对规则不会强制执行链接子集权限测试，而标准规则链接权限则需要进行此测试。

```
link /srv/www/htdocs/index.html -> /var/www/index.html
```

要强制让规则要求进行该测试，可使用 subset 关键字。以下规则是等效的：

```
/var/www/index.html l,  
link subset /var/www/index.html -> /**,
```



注意

YaST 和命令行工具目前不支持链接对规则。要使用这些规则，请手动编辑配置文件。使用工具更新此类配置文件是安全的操作，因为这种方式不会改动链接对项。

30.7.7 可选的允许规则和文件规则

allow 前缀是可选的，如果未指定并且不使用 deny（参见第 30.7.9 节“拒绝规则”）关键字，则按惯常隐式应用该前缀。

```
allow file /example r,  
allow /example r,
```

```
allow network,
```

您还可以使用可选的 `file` 关键字。如果您省略该关键字并且不存在其他以某个关键字（例如 `network` 或 `mount`）开头的规则类型，则自动隐式应用该前缀。

```
file /example/rule r,
```

等效于

```
/example/rule r,
```

下面的规则授予对所有文件的访问权限：

```
file,
```

等效于

```
/** rwmlk,
```

文件规则可以使用前导或尾随权限。不应将权限指定为尾随权限，而应在规则的开头使用。这一点非常重要，因为这样可使文件规则的行为类似于任何其他规则类型。

```
/path rw,          # old style
rw /path,          # leading permission
file rw /path,      # with explicit 'file' keyword
allow file rw /path, # optional 'allow' keyword added
```

30.7.8 拥有者条件规则

可以扩展文件规则，使其可以按条件应用于文件的拥有者用户（`fsuid` 需与文件的 `uid` 相匹配）。要实现此目的，需在规则的前面添加 `owner` 关键字。拥有者条件规则像普通的文件规则一样不断累积。

```
owner /home/*/** rw
```

将文件所有权条件与链接规则结合使用时，将针对目标文件执行所有权测试，因此，用户必须拥有该文件才能链接到该文件。



注意：普通文件规则的优先级

拥有者条件规则被视为普通文件规则的子集。如果某个普通文件规则与某个拥有者条件文件规则重叠，这两条规则将会合并。参见以下示例。

```
/foo r,  
owner /foo rw, # or w,
```

这些规则会合并 — 结果是每个人均具有 r 权限，只有拥有者具有 w 权限。



提示

要指定除文件拥有者以外的每个人，请使用关键字 other。

```
owner /foo rw,  
other /foo r,
```

30.7.9 拒绝规则

拒绝规则可用于批注已知拒绝或使其静止。配置文件生成工具不会询问有关拒绝规则所处理的已知拒绝的信息。发生拒绝后，此类拒绝也不会显示在审计日志中，以使日志文件保持精简。如果不需要此行为，请在拒绝项的前面添加关键字 audit。

此外，还可以将拒绝规则与允许规则结合使用。这样，您便可以先指定一条宽泛的允许规则，然后再去掉几个不应允许的已知文件。拒绝规则还可与拥有者规则结合使用来拒绝用户拥有的文件。下面的示例允许对 users 目录中的任何内容进行读写访问，但不允许对 .ssh/ 文件进行写入访问：

```
deny /home/*/ .ssh/** w,  
owner /home/*/* rw,
```

一般不建议大量使用拒绝规则，因为这会大大增加理解配置文件的作用的难度。不过，审慎使用拒绝规则可以简化配置文件。因此，工具只会生成拒绝特定文件的配置文件，而不会在拒绝规则中使用通配。要添加使用通配的拒绝规则，请手动编辑配置文件。使用工具更新此类配置文件是安全的操作，因为这种方式不会改动拒绝项。

30.8 挂载规则

AppArmor 可以限制装入和卸载操作，包括文件系统类型和装入标志。规则语法基于 `mount` 命令语法，以 `mount`、`remount` 或 `umount` 关键字开头。条件是可选项，如果不指定条件，则认为要匹配所有项。例如，不指定文件系统表示要匹配所有文件系统。

可以使用 `options=` 或 `options in` 指定条件。

`options=` 指定必须完全符合的条件。规则

```
mount options=ro /dev/foo -E /mnt/,
```

匹配

```
root # mount -o ro /dev/foo /mnt
```

但不匹配

```
root # mount -o ro,atime /dev/foo /mnt
root # mount -o rw /dev/foo /mnt
```

`options in` 要求至少使用一个所列的装入选项。规则

```
mount options in (ro,atime) /dev/foo -> /mnt/,
```

匹配

```
root # mount -o ro /dev/foo /mnt
root # mount -o ro,atime /dev/foo /mnt
root # mount -o atime /dev/foo /mnt
```

但不匹配

```
root # mount -o ro,sync /dev/foo /mnt
root # mount -o ro,atime,sync /dev/foo /mnt
root # mount -o rw /dev/foo /mnt
root # mount -o rw,noatime /dev/foo /mnt
root # mount /dev/foo /mnt
```

如果使用多个条件，规则将为每组选项授予权限。规则

```
mount options=ro options=atime
```

匹配

```
root # mount -o ro /dev/foo /mnt
root # mount -o atime /dev/foo /mnt
```

但不匹配

```
root # mount -o ro,atime /dev/foo /mnt
```

单独的装入规则是不同的，选项不会累积。规则

```
mount options=ro,
mount options=atime,
```

与下列规则不等效

```
mount options=(ro,atime),
mount options in (ro,atime),
```

下面的规则允许在 /mnt/ 上装入只读的 /dev/foo 并使用 inode 访问时间，或者允许使用 “nodev” 和 “user” 的某种组合在 /mnt/ 上装入 /dev/foo。

```
mount options=(ro, atime) options in (nodev, user) /dev/foo -> /mnt/,
```

允许

```
root # mount -o ro,atime /dev/foo /mnt
root # mount -o nodev /dev/foo /mnt
root # mount -o user /dev/foo /mnt
root # mount -o nodev,user /dev/foo /mnt
```

30.9 Pivot Root 规则

AppArmor 可以限制对根文件系统的更改。语法为

```
pivot_root [oldroot=OLD_ROOT] NEW_ROOT
```

在 “pivot_root” 规则中指定的路径必须以 “/” 结尾，因为它们是目录。

```
# Allow any pivot
pivot_root,

# Allow pivoting to any new root directory and putting the old root
# directory at /mnt/root/old/
pivot_root oldroot=/mnt/root/old/,

# Allow pivoting the root directory to /mnt/root/
pivot_root /mnt/root/,

# Allow pivoting to /mnt/root/ and putting the old root directory at
# /mnt/root/old/
pivot_root oldroot=/mnt/root/old/ /mnt/root/,

# Allow pivoting to /mnt/root/, putting the old root directory at
# /mnt/root/old/ and transition to the /mnt/root/sbin/init profile
pivot_root oldroot=/mnt/root/old/ /mnt/root/ -> /mnt/root/sbin/init,
```

30.10 PTrace 规则

AppArmor 支持限制 ptrace 系统调用。ptrace 规则将会累积，因此，授予的 ptrace 权限是全部所列 ptrace 规则权限的并集。如果某条规则未指定访问列表，则会隐式授予权限。

trace 和 tracedby 权限控制 ptrace(2)；read 和 readby 控制 proc(5) 文件系统访问、kcmp(2)、futexes (get_robust_list(2)) 和 perf 跟踪事件。

要允许 ptrace 操作，跟踪进程和被跟踪进程都需要有正确的权限。也就是说，跟踪进程需要有 trace 权限，被跟踪进程需要有 tracedby 权限。

示例 AppArmor PTrace 规则：

```
# Allow all PTrace access
ptrace,

# Explicitly allow all PTrace access,
ptrace (read, readby, trace, tracedby),

# Explicitly deny use of ptrace(2)
```

```
deny ptrace (trace),

# Allow unconfined processes (eg, a debugger) to ptrace us
ptrace (readby, tracedby) peer=unconfined,

# Allow ptrace of a process running under the /usr/bin/foo profile
ptrace (trace) peer=/usr/bin/foo,
```

30.11 信号规则

AppArmor 支持限制进程间的信号。AppArmor 信号规则会累积，因此，授予的信号权限是全部所列信号规则权限的并集。如果规则未显式指明访问列表，则隐式应用 AppArmor 信号权限。

发送方进程和接收方进程都必须拥有正确的权限。

示例信号规则：

```
# Allow all signal access
signal,

# Explicitly deny sending the HUP and INT signals
deny signal (send) set=(hup, int),

# Allow unconfined processes to send us signals
signal (receive) peer=unconfined,

# Allow sending of signals to a process running under the /usr/bin/foo
# profile
signal (send) peer=/usr/bin/foo,

# Allow checking for PID existence
signal (receive, send) set=("exists"),

# Allow us to signal ourselves using the built-in @{profile_name} variable
signal peer=@{profile_name},

# Allow two real-time signals
```

```
signal set=(rtmin+0 rtmin+32),
```

30.12 执行模式

执行模式（也称为配置文件转换）包括以下模式：

<u>Px</u>	离散配置文件执行模式
<u>Cx</u>	离散本地配置文件执行模式
<u>Ux</u>	未受限执行模式
<u>ix</u>	继承执行模式
<u>m</u>	允许使用 <code>mmap(2)</code> 调用执行 <code>PROT_EXEC</code>

30.12.1 离散配置文件执行模式 (px)

此模式要求为在 AppArmor 域转换时执行的资源定义一个离散安全配置文件。如果未定义配置文件，则会拒绝访问。

与 Ux、ux、px 和 ix 不兼容。

30.12.2 离散本地配置文件执行模式 (cx)

类似于 Px，但 Cx 不搜索全局配置文件集，而只搜索当前配置文件的本地配置文件。应用程序可以通过这种配置文件转换获得助手应用程序的备用配置文件。



注意：离散本地配置文件执行模式 (cx) 的限制

目前，Cx 转换仅对顶层配置文件适用，不能在帽子和子配置文件中使用。将来会去除这项限制。

与 Ux、ux、Px、px、cx 和 ix 不兼容。

30.12.3 未受限执行模式 (ux)

允许程序执行资源，不对被执行的资源应用任何 AppArmor 配置文件。此模式可用于使被限制的程序能够执行需要特权的操作，如重新引导计算机等。通过在其他可执行文件中添加具有特权的部分并授予未受限的执行权限，您可以避开对全部受限制进程强制施加的限制。允许根进程不受限制意味着它可以更改 AppArmor 策略本身。有关限制内容的详细信息，请参见 [apparmor\(7\)](#) 手册页。

此模式与 [ux](#)、[px](#)、[Px](#) 和 [ix](#) 不兼容。

30.12.4 不安全的执行模式

仅在非常特殊的情况下才使用小写形式的执行模式 — [px](#)、[cx](#)、[ux](#)。这些模式不会整理 [LD_PRELOAD](#) 等变量的环境。因此，调用域可能会对被调用资源产生过度影响。仅当绝对**必须**以非受限方式运行子项并且必须使用 [LD_PRELOAD](#) 时，才使用这些模式。任何使用此类模式的配置文件几乎都不会提供任何安全性。使用这些模式需自负后果。

30.12.5 继承执行模式 (ix)

当构建了配置文件的程序执行命名程序时，[ix](#) 会阻止 [execve\(2\)](#) 上的常规 AppArmor 域转换。相反，被执行的资源继承当前配置文件。

当被限制的程序需要调用其它被限制的程序时此模式非常实用，无须获得目标程序配置文件的权限或失去当前配置文件的权限。没有任何版本会整理环境，因为 [ix](#) 执行不会更改特权。

与 [cx](#)、[ux](#) 和 [px](#) 不兼容。隐式应用 [m](#)。

30.12.6 允许可执行映射 (m)

此模式允许使用 [mmap\(2\)](#) 的 [PROT_EXEC](#) 标志将文件映射到内存中。此标志将页面标示为可执行。某些体系结构使用此标志来提供不可执行的数据页面，这可以增加恶意利用的企图困难度。AppArmor 使用此模式来限制可由行为正常的程序（或者强制实施不可执行内存访问控制的体系结构上的所有程序）用作库的文件，以及限制为 [ld\(1\)](#) 指定的无效 [-L](#) 标志以及为 [ld.so\(8\)](#) 指定的 [LD_PRELOAD](#) 和 [LD_LIBRARY_PATH](#) 所产生的影响。

30.12.7 命名配置文件转换

默认情况下，`px` 和 `cx`（也包括其清洁执行变体）将转换到名称与可执行文件名称匹配的配置文件。使用命名配置文件转换，您可以指定要转换到的配置文件。如果多个二进制文件需要共享单个配置文件，或者这些二进制文件需要使用的配置文件不同于其名称指定的配置文件，此方法将非常有用。可以将命名配置文件转换与 `cx`、`Cx`、`px` 和 `Px` 结合使用。目前，每个配置文件仅可具有 12 个命名配置文件转换。

命名配置文件转换使用 `->` 来指示需要转换到的配置文件的名称：

```
/usr/bin/foo
{
  /bin/** px -> shared_profile,
  ...
  /usr/*bash cx -> local_profile,
  ...
  profile local_profile
  {
    ...
  }
}
```



注意：常规转换与命名转换之间的差别

与通配结合使用时，常规转换提供“一对多”关系 — `/bin/** px` 将转换到 `/bin/ping`、`/bin/cat` 等，具体取决于所运行的程序。

命名转换提供“多对一”关系 — 所有程序不管其名称是什么，只要与规则匹配，都将转换到指定的配置文件。

命名配置文件转换具有模式 `Nx`，因此会显示在日志中。要转换到的配置文件的名称列于 `name2` 字段中。

30.12.8 配置文件转换的回退模式

px 和 cx 转换会指定硬依赖项 — 如果指定的配置文件不存在，则执行将会失败。使用继承回退时，执行将会成功，但会继承当前配置文件。要指定继承回退，请将 ix 与 cx、Cx、px 和 Px 结合使用，构成模式 cix、Cix、pix 和 Pix。

```
/path Cix -> profile_name,
```

或

```
Cix /path -> profile_name,
```

其中 -> profile_name 是可选项。

如果您添加未受限 ux 模式（这样最终模式为 cux、CUx、puX 和 PUx），上面所述同样适用。如果未找到指定的配置文件，这些模式允许回退到“未受限”。

```
/path PUx -> profile_name,
```

或

```
PUx /path -> profile_name,
```

其中 -> profile_name 是可选项。

您也可以对命名配置文件转换使用回退模式。

30.12.9 执行模式中的变量设置

选择 Px、Cx 或 Ux 执行模式之一时，请注意在子进程继承这些模式之前，以下环境变量会从环境中去除。因此，如果向依赖于以下任何变量的应用程序或进程应用的配置文件带有 Px、Cx 或 Ux 标志，这些应用程序或进程将不再可正常运行：

- GCONV_PATH
- GETCONF_DIR
- HOSTALIASES
- LD_AUDIT

- LD_DEBUG
- LD_DEBUG_OUTPUT
- LD_DYNAMIC_WEAK
- LD_LIBRARY_PATH
- LD_ORIGIN_PATH
- LD_PRELOAD
- LD_PROFILE
- LD_SHOW_AUXV
- LD_USE_LOAD_BIAS
- LOCALDOMAIN
- LOCPATH
- MALLOC_TRACE
- NLSPATH
- RESOLV_HOST_CONF
- RES_OPTIONS
- TMPDIR
- TZDIR

30.12.10 **safe** 和 **unsafe** 关键字

您可以对规则使用 **safe** 和 **unsafe** 关键字来取代执行模式的大小写修饰符。例如，

```
/example_rule Px,
```

等同于下列任何一项

```
safe /example_rule px,
```

```
safe /example_rule Px,  
safe px /example_rule,  
safe Px /example_rule,
```

规则

```
/example_rule px,
```

等同于下列任何一项

```
unsafe /example_rule px,  
unsafe /example_rule Px,  
unsafe px /example_rule,  
unsafe Px /example_rule,
```

safe/unsafe 关键字是互斥的，可在文件规则中的 owner 关键字后面使用，因此，规则关键字的顺序如下

```
[audit] [deny] [owner] [safe|unsafe] file_rule
```

30.13 资源限制控制

AppArmor 可以设置和控制应用程序的资源限制（rlimit，也称为 ulimit）。默认情况下，AppArmor 不会控制应用程序的 rlimit，而只控制限制配置文件中指定的这些限制。有关资源限制的详细信息，请参见 [setrlimit\(2\)](#)、[ulimit\(1\)](#) 或 [ulimit\(3\)](#) 手册页。

AppArmor 会利用系统的 rlimit，因此不会另外提供审计（正常情况下会发生审计）。此外，它不能提高系统设置的 rlimit，AppArmor rlimit 只能降低应用程序的当前资源限制。

进程的子项会继承这些值，即使转换到了新配置文件或者应用程序变得不受限制，这些值也会保留。因此，当应用程序转换到新配置文件时，该配置文件可以进一步降低应用程序的 rlimit。

AppArmor 的 rlimit 规则还可以调解应用程序硬限制的设置（如果应用程序尝试提高这些限制）。应用程序不能将硬限制提高到超过配置文件中指定的限制的水平。提高的硬限制不会像设置的值那样会被继承，因此，当应用程序转换到新配置文件时，它可以任意提高其限制（只要不超过配置文件中指定的值）。

AppArmor 的 rlimit 控制除了会确保应用程序的软限制小于或等于应用程序的硬限制外，不会在其他方面影响软限制。

AppArmor 硬限制规则一般采用如下格式：

```
set rlimit RESOURCE <= value,
```

其中 RESOURCE 和 VALUE 将替换为以下值：

cpu

CPU 时间限制，以秒为单位。

fsize、data、stack、core、rss、as、memlock、msgqueue

以字节为单位的数字，或者带后缀的数字，例如，该后缀可以是 K/KB（千字节）、M/MB（兆字节）、G/GB（千兆字节）

```
rlimit data <= 100M,
```

fsize、nofile、locks、sigpending、nproc^{*}、rtprio

大于或等于 0 的数字

nice

-20 到 19 的值

^{*}nproc rlimit 的处理方式不同于所有其他 rlimit。它不指示标准进程 rlimit，而是控制在任意时间可基于配置文件运行的最大进程数。如果超过限制，基于配置文件创建新进程将会失败，直到当前正在运行的进程数减少。



注意

目前无法使用工具将 rlimit 规则添加到配置文件中。可将 rlimit 控制添加到配置文件的唯一方法是使用文本编辑器手动编辑配置文件。工具仍会处理包含 rlimit 规则的配置文件，并且不会去除这些规则，因此，使用工具更新包含这些规则的配置文件是安全的操作。

30.14 审计规则

AppArmor 提供用于审计给定规则的功能，如此，当匹配这些规则时，审计日志中会显示审计消息。要对给定的规则启用审计消息，可在规则的前面添加 audit 关键字：

```
audit /etc/foo/*      rw,
```

如果只希望审计给定的权限，可将该规则分割为两条规则。在下面的示例中，打开文件向其写入数据时会生成审计消息，但打开文件读取数据时，则不会生成消息：

```
audit /etc/foo/*  w,  
/etc/foo/*      r,
```



注意

并非每次对文件执行读取或写入操作时都会生成审计消息，只有在打开文件进行读取或写入操作时才生成消息。

可将审计控制与 owner/other 条件文件规则结合使用，以便在用户访问他们拥有/不拥有的文件时提供审计：

```
audit owner /home/*/.ssh/**      rw,  
audit other /home/*/.ssh/**      r,
```

31 AppArmor 配置文件储存库

AppArmor 随附了一组默认已启用的配置文件。这些配置文件由 AppArmor 开发人员创建，储存在 /etc/apparmor.d 中。除了这些配置文件以外，SUSE Linux Enterprise Server 还为各应用程序及相关应用程序随附了配置文件。这些配置文件默认未启用，储存在与标准 AppArmor 配置文件不同的另一个目录中：/usr/share/apparmor/extra-profiles。

AppArmor 工具（YaST、**aa-genprof** 和 **aa-logprof**）支持使用本地储存库。每当您从头开始创建新配置文件时，您的本地储存库中已有一个非活动的配置文件，系统会询问您是要使用 /usr/share/apparmor/extra-profiles 中现有的非活动配置文件，还是基于该配置文件创建新配置文件。如果您决定使用此配置文件，系统会将它复制到默认已启用的配置文件所在的目录 (/etc/apparmor.d)，并且 AppArmor 每次启动时都会装载此配置文件。以后所进行的任何调整针对的都是 /etc/apparmor.d 下的活动配置文件。

32 使用 YaST 构建和管理配置文件

YaST 提供了用于构建配置文件以及管理 AppArmor® 配置文件的基本途径。它提供了两个界面：一个图形界面和一个基于文本的界面。基于文本的界面消耗的资源 and 带宽更少，因此，需要进行远程管理或者当本地图形环境不够方便时，基于文本的界面是更好的选择。尽管这两个界面的外观不同，但它们以类似的方式提供相同的功能。另一种方法是使用 AppArmor 命令，这些命令可以从终端窗口或通过远程连接控制 AppArmor。第 33 章 “从命令行构建配置文件” 中介绍了命令行工具。

从主菜单中启动 YaST，并在出现提示时输入 `root` 口令。或者，通过打开终端窗口，以 `root` 身份登录，然后输入 `yast2`（表示图形模式）或 `yast`（表示基于文本的模式）来启动 YaST。安全和用户部分显示了一个 AppArmor 配置图标。单击该图标可启动 AppArmor YaST 模块。

32.1 手动添加配置文件

AppArmor 允许您通过手动向配置文件添加项的方式来创建 AppArmor 配置文件。选择要为其创建配置文件的应用程序，然后添加项。

1. 启动 YaST，选择 AppArmor 配置，然后在主窗口中单击手动添加配置文件。
2. 浏览系统以找到要创建配置文件的应用程序。
3. 找到应用程序后，选择它并单击打开。AppArmor 配置文件对话框窗口中将出现一个空的基本配置文件。
4. 在 AppArmor 配置文件对话框中，通过单击对应的按钮并参考第 32.2.1 节 “添加项”、第 32.2.2 节 “编辑项” 或第 32.2.3 节 “删除项” 来添加、编辑或删除 AppArmor 配置文件项。
5. 完成后，单击完成。

32.2 编辑配置文件



提示

YaST 提供针对 AppArmor 配置文件的基本操作，例如创建或编辑配置文件。不过，最直接的编辑 AppArmor 配置文件的方式是使用 **vi** 这样的文本编辑器：

```
tux > sudo vi /etc/apparmor.d/usr.sbin.httpd2-prefork
```

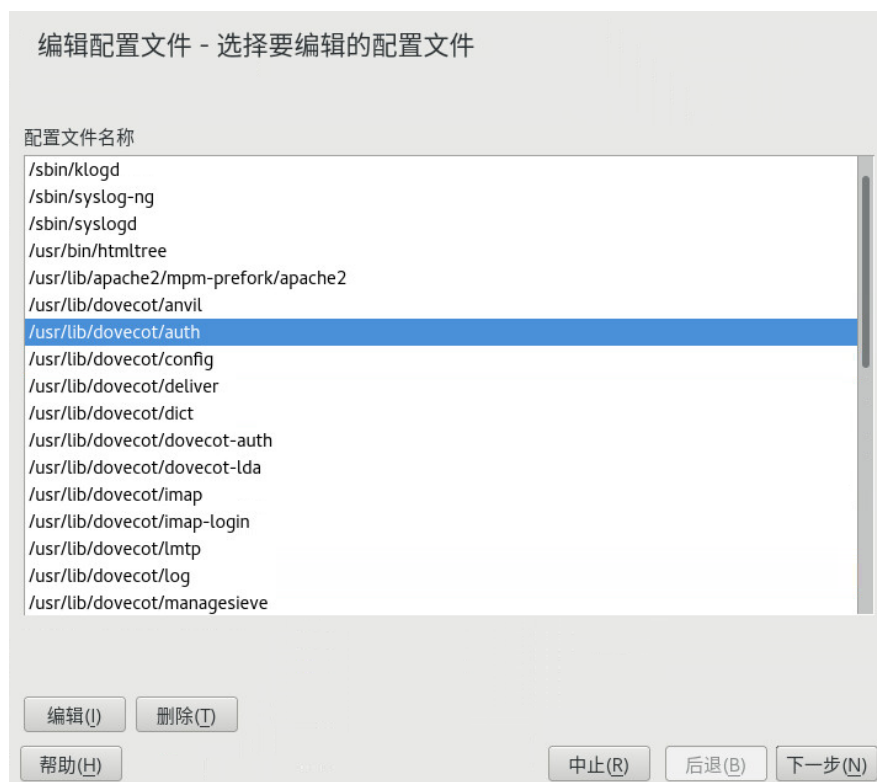


提示

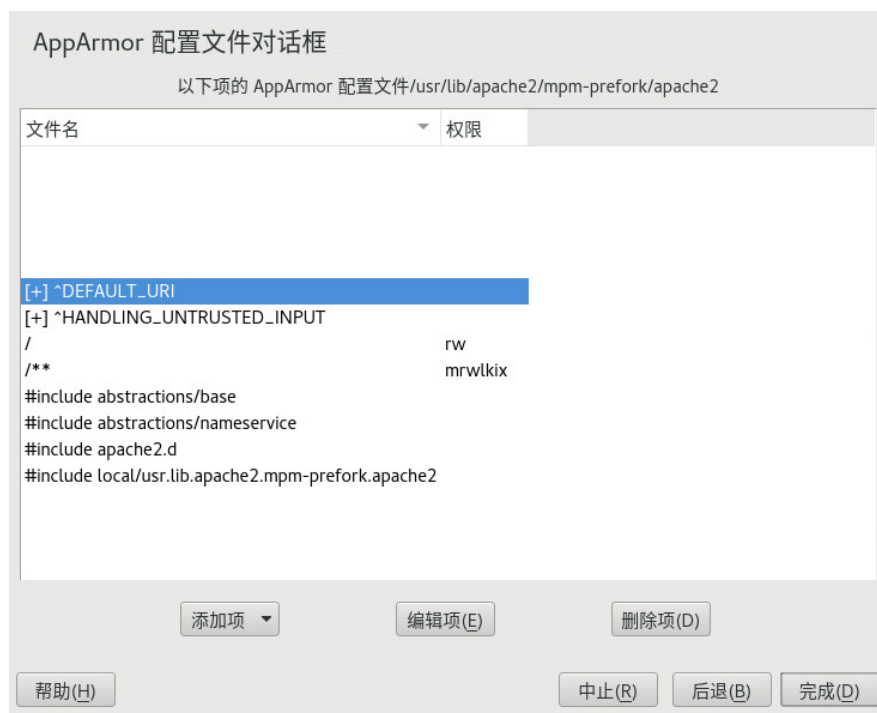
vi 编辑器还包含语法（错误）高亮显示和语法错误高亮显示功能，当编辑后的 AppArmor 配置文件存在语法错误时，它会以视觉方式发出警告。

AppArmor 允许您通过添加、编辑或删除项来手动编辑 AppArmor 配置文件。要编辑配置文件，请执行以下操作：

1. 启动 YaST，选择 AppArmor 配置，然后在主窗口中单击管理现有配置文件。



2. 在已构建配置文件的应用程序列表中，选择要编辑的配置文件。
3. 单击编辑。此时 AppArmor 配置文件对话框窗口会显示配置文件。



4. 在 AppArmor 配置文件对话框窗口中，通过单击对应的按钮并参考第 32.2.1 节“添加项”、第 32.2.2 节“编辑项”或第 32.2.3 节“删除项”来添加、编辑或删除 AppArmor 配置文件项。
5. 完成后，单击完成。
6. 在出现的弹出窗口中，单击是(Y)以确认对配置文件所做的更改，然后重新装载 AppArmor 配置文件集。

提示：AppArmor 中的语法检查

AppArmor 包含语法检查功能，如果您尝试使用 YaST AppArmor 工具处理的配置文件中存在任何语法错误，它会发出通知。如果发生错误，请以 `root` 身份手动编辑配置文件，然后使用 `systemctl reload apparmor` 重新装载配置文件集。

32.2.1 添加项

AppArmor 配置文件窗口中的添加项按钮会列出您可以添加到 AppArmor 配置文件的项类型。在列表中选择以下选项之一：

文件

在弹出窗口中，指定文件的绝对路径，包括允许的访问类型。完成后，单击确定。

必要时，您可以使用通配。有关通配的详细信息，请参见第 30.6 节“配置文件名称、标志、路径和通配”。有关文件访问权限的详细信息，请参见第 30.7 节“文件权限访问模式”。



目录

在弹出窗口中，指定目录的绝对路径，包括允许的访问类型。必要时，您可以使用通配。完成后，单击确定。

有关通配的详细信息，请参见第 30.6 节“配置文件名称、标志、路径和通配”。有关文件访问权限的详细信息，请参见第 30.7 节“文件权限访问模式”。



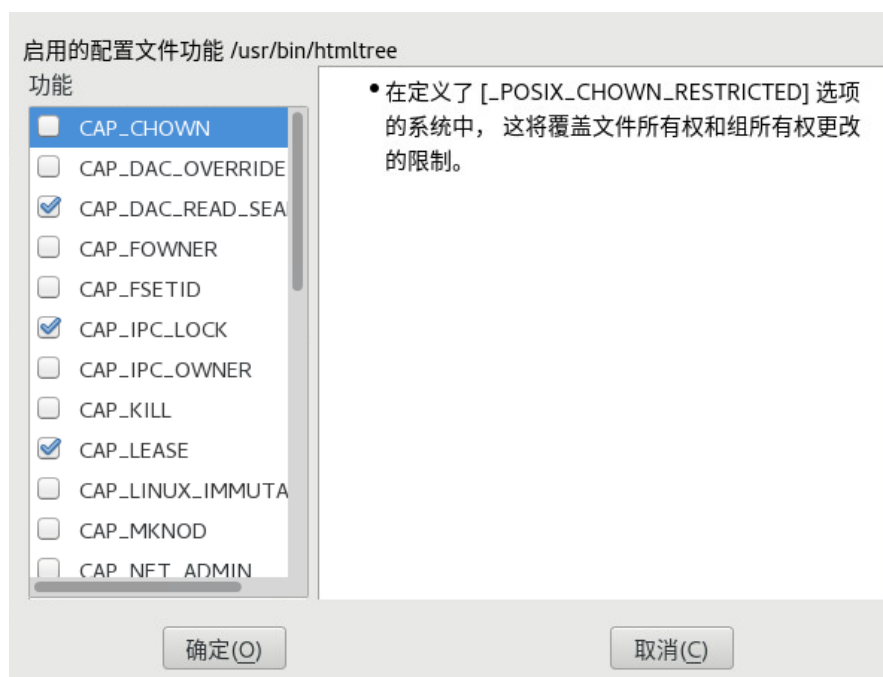
网络规则

在弹出窗口中，选择适当的网络系列和套接字类型。有关更多信息，请参考第 30.5 节“网络访问控制”。



功能

在对话窗口中，选择适当的功能。这些语句用于启用 32 个 POSIX.1e 功能。有关功能的详细信息，请参见第 30.4 节“功能项 (POSIX.1e)”。完成选择后，单击确定。



Include 文件

在弹出窗口中，浏览到要用作 include 的文件。Include 是可提取其他 AppArmor 配置文件的组件以简化配置文件的指令。有关更多信息，请参考第 30.3 节“Include 语句”。

帽子

在弹出窗口中，指定要添加到当前配置文件的子配置文件（帽子）的名称，然后单击创建 Hat。有关更多信息，请参考第 34 章 “使用 ChangeHat 构建 Web 应用程序的配置文件”。

请输入想添加到
配置文件的 Hat 名称
/usr/bin/htmltree.

要添加的 Hat 名称(H)

创建 Hat(C) 中止(R)

32.2.2 编辑项

选择编辑项时，会打开一个弹出窗口。请在此窗口中编辑选定的项。

在弹出窗口中，编辑需要修改的项。必要时，您可以使用通配。完成后，单击确定。

有关通配的详细信息，请参见第 30.6 节 “配置文件名称、标志、路径和通配”。有关访问权限的信息，请参见第 30.7 节 “文件权限访问模式”。

32.2.3 删除项

要删除给定配置文件中的项，请选择删除项。AppArmor 将去除选定的配置文件项。

32.3 删除配置文件

AppArmor 允许您手动删除 AppArmor 配置文件。只需选择要删除其配置文件的应用程序，然后按如下所示删除：

1. 启动 YaST，选择 AppArmor 配置，然后在主窗口中单击管理现有配置文件。
2. 选择要删除的配置文件。
3. 单击删除。
4. 在打开的弹出窗口中，单击是以删除该配置文件，然后重新装载 AppArmor 配置文件集。

32.4 管理 AppArmor

可以通过启用或禁用 AppArmor 来更改其状态。启用 AppArmor 可保护您的系统抵御潜在的程序漏洞攻击。禁用 AppArmor 将撤销对系统的保护，即使已设置了配置文件。要更改 AppArmor 的状态，请启动 YaST，选择 AppArmor 配置，然后在主窗口中单击设置。



要更改 AppArmor 的状态，请按照第 32.4.1 节“更改 AppArmor 状态”中所述继续操作。要更改单个配置文件的模式，请按照第 32.4.2 节“更改单个配置文件的模式”中所述继续操作。

32.4.1 更改 AppArmor 状态

更改 AppArmor 的状态时，请将其设置为已启用或已禁用。启用 AppArmor 后，系统将安装并运行 AppArmor，并强制执行其安全策略。

1. 启动 YaST，选择 AppArmor 配置，然后在主窗口中单击设置。
2. 选中启用 AppArmor 以启用 AppArmor，或取消选中该选项以禁用 AppArmor。
3. 单击 AppArmor 配置窗口中的完成。



提示

对于正在运行的程序，一律需要将其重新启动才能应用配置文件。

32.4.2 更改单个配置文件的模式

AppArmor 可在两种不同的模式下应用配置文件。在**投诉**模式下，会检测对 AppArmor 配置文件规则的违规，例如构建了配置文件的程序访问配置文件不允许的文件。冲突是允许的，但也会被记录。此模式有利于开发配置文件，AppArmor 工具用其来生成配置文件。在**强制**模式下装载配置文件会强制执行该配置文件中定义的策略，并在 `rsyslogd`（或者 `auditd` 或 `journalctl`，具体取决于系统配置）中报告策略违规尝试。

您可在配置文件模式配置对话框中查看和编辑当前装载的 AppArmor 配置文件的模式。在开发配置文件期间，可以使用此功能来确定系统的状态。在系统性配置文件构建（请参见第 33.7.2 节“**系统性配置文件构建**”）期间，您可以使用此工具来调整和监视您正在探测其行为的配置文件的范围。

要编辑应用程序的配置文件模式，请执行以下操作：

1. 启动 YaST，选择 AppArmor 配置，然后在主窗口中单击设置。
2. 在配置配置文件模式部分，选择配置。
3. 选择要更改其模式的配置文件。
4. 选择转换模式以将此配置文件设置为**投诉**模式或**强制**模式。
5. 应用您的设置，然后单击完成退出 YaST。

要更改所有配置文件的模式，请使用全部设置为强制或全部设置为控诉。



提示：列出可用的配置文件

默认只会列出活动的配置文件（系统上安装了其匹配应用程序的配置文件）。要在安装相关应用程序之前设置配置文件，请单击显示所有配置文件，然后从显示的列表中选择要配置的配置文件。

33 从命令行构建配置文件

AppArmor® 可让用户使用命令行界面而不是图形界面来管理和配置系统安全性。可以使用 AppArmor 命令行工具来跟踪 AppArmor 的状态，以及创建、删除或修改 AppArmor。



提示：背景信息

在开始使用 AppArmor 命令行工具管理配置文件之前，请查看第 29 章 “使程序免疫” 和第 30 章 “配置文件组件和语法” 中提供的 AppArmor 一般简介。

33.1 检查 AppArmor 状态

AppArmor 可能会处于以下三种状态中的任何一种：

已卸载

AppArmor 未在内核中激活。

正在运行

AppArmor 已在内核中激活，并在强制执行 AppArmor 程序策略。

已停止

AppArmor 已在内核中激活，但未强制执行策略。

通过检查 `/sys/kernel/security/apparmor/profiles` 来检测 AppArmor 的状态。如果 `cat/sys/kernel/security/apparmor/profiles` 报告了一系列配置文件，则 AppArmor 正在运行。如果该文件为空且未返回任何消息，则表示 AppArmor 已停止。如果该文件不存在，则表示 AppArmor 已卸载。

使用 `systemctl` 来管理 AppArmor。您可以使用此工具执行以下操作：

`sudo systemctl start apparmor`

行为取决于 AppArmor 的状态。如果 AppArmor 未激活，`start` 会将其激活并启动，同时将其置于运行状态。如果 AppArmor 已停止，`start` 会导致重新扫描 AppArmor 配置文件（通常位于 `/etc/apparmor.d` 中）并将其置于运行状态。如果 AppArmor 已在运行，`start` 会发出警告，而不执行任何操作。



注意：已在运行的进程

要对已在运行的进程应用 AppArmor 配置文件，需要将其重新启动。

`sudo systemctl stop apparmor`

通过去除内核内存中的所有配置文件来停止正在运行的 AppArmor，这会有效禁用所有访问控制，并将 AppArmor 置于停止状态。如果 AppArmor 已停止，`stop` 会尝试再次卸载配置文件，但不会有任何结果。

`sudo systemctl reload apparmor`

导致 AppArmor 模块重新扫描 `/etc/apparmor.d` 中的配置文件，但不取消限制正在运行的进程。强制执行新创建的配置文件，并去除 `/etc/apparmor.d` 目录中最近删除的配置文件。

33.2 构建 AppArmor 配置文件

AppArmor 模块配置文件定义以纯文本文件的形式储存在 `/etc/apparmor.d` 目录中。有关这些文件的详细语法说明，请参见第 30 章“配置文件组件和语法”。

`/etc/apparmor.d` 目录中的所有文件被解释为配置文件并装载为配置文件。要防止装载配置文件，对此目录中的文件进行重命名不是一种有效的方式。您必须去除此目录中的配置文件，以有效防止读取和评估这些配置文件；或者对配置文件调用 `aa-disable`，这会在 `/etc/apparmor.d/disabled/` 中创建一个符号链接。

您可以使用 `vi` 等文本编辑器来访问和更改这些配置文件。以下各节包含构建配置文件的详细步骤：

添加或创建 AppArmor 配置文件

有关详细信息，请参见第 33.3 节“添加或创建 AppArmor 配置文件”

编辑 AppArmor 配置文件

有关详细信息，请参见第 33.4 节“编辑 AppArmor 配置文件”

删除 AppArmor 配置文件

有关详细信息，请参见第 33.6 节“删除 AppArmor 配置文件”

33.3 添加或创建 AppArmor 配置文件

要为应用程序添加或创建 AppArmor 配置文件，可以使用系统的或独立的配置文件构建方法，视您的需要而定。第 33.7 节 “构建配置文件的两种方式” 中详细介绍了这两种方法。

33.4 编辑 AppArmor 配置文件

以下步骤说明编辑 AppArmor 配置文件的过程：

1. 如果您当前不是以 `root` 身份登录的，请在终端窗口中输入 `su`。
2. 出现提示时输入 `root` 口令。
3. 使用 `cd /etc/apparmor.d/` 转到配置文件目录。
4. 输入 `ls` 以查看所有当前安装的配置文件。
5. 在 `vim` 等文本编辑器中打开配置文件以进行编辑。
6. 完成必要的修改后保存配置文件。
7. 在终端窗口中输入 `systemctl reload apparmor` 以重新启动 AppArmor。

33.5 卸载未知的 AppArmor 配置文件



警告：卸载所需配置文件的风险

`aa-remove-unknown` 会卸载未储存在 `/etc/apparmor.d` 中的所有配置文件，例如，自动生成的 LXD 配置文件。这可能会损害系统的安全性。使用 `-n` 参数可列出所有将会卸载的配置文件。

要卸载不再位于 `/etc/apparmor.d/` 中的所有 AppArmor 配置文件，请运行：

```
tux > sudo aa-remove-unknown
```

您可以列显要去除的配置文件列表：

```
tux > sudo aa-remove-unknown -n
```

33.6 删除 AppArmor 配置文件

以下步骤说明删除 AppArmor 配置文件的过程。

1. 从内核中去除 AppArmor 定义：

```
tux > sudo apparmor_parser -R /etc/apparmor.d/PROFILE
```

2. 去除定义文件：

```
tux > sudo rm /etc/apparmor.d/PROFILE  
tux > sudo rm /var/lib/apparmor/cache/PROFILE
```

33.7 构建配置文件的两种方式

掌握第 30 章“配置文件组件和语法”中介绍的 AppArmor 配置文件语法后，您无需借助工具也能创建配置文件，但需要的工作量比较大。要避免这种情况，请使用 AppArmor 工具来自动创建和优化配置文件。

可采用两种方法创建 AppArmor 配置文件。这两种方法都有可用的工具。

独立式配置文件构建

此方法适用于运行时间有限的小型应用程序，如邮件客户端等用户客户端应用程序。有关更多信息，请参考第 33.7.1 节“独立式配置文件构建”。

系统性配置文件构建

此方法适用于一次性为许多程序构建配置文件，还适用于为运行时间长达数日、数周或在重引导后连续运行的应用程序（如 Web 服务器、邮件服务器等网络服务器应用程序）构建配置文件。有关更多信息，请参考第 33.7.2 节“系统性配置文件构建”。

使用 AppArmor 工具可使自动开发配置文件的过程变得更易于管理。

1. 确定符合您的要求的配置文件构建方式。
2. 执行一次静态分析。根据所选的配置文件构建方法运行 **aa-genprof** 或 **aa-autodep**。

3. 启用动态学习。为所有构建了配置文件的程序启动学习模式。

33.7.1 独立式配置文件构建

独立式配置文件的生成和改进由名为 **aa-genprof** 的程序来管理。此方法很简单，因为 **aa-genprof** 会负责处理所有事宜，但在程序测试运行的整个过程中都需要运行 **aa-genprof**（在开发配置文件过程中不能重引导计算机），因此适合的场景比较有限。

要使用 **aa-genprof** 以独立式方法构建配置文件，请参见第 33.7.3.8 节 “**aa-genprof** — 生成配置文件”。

33.7.2 系统性配置文件构建

此方法之所以称为**系统性配置文件构建**，是因为它会一次性更新系统中所有的配置文件，而不像 **aa-genprof** 或独立式配置文件构建那样只针对一个或少数几个配置文件。采用系统性配置文件构建方法时，构建和改进配置文件的自动化程度会有所下降，但更灵活。此方法适用于为长时间运行且其行为会在重引导后持续的应用程序构建配置文件，或者一次性为许多程序构建配置文件。

按如下方式为一组应用程序构建 AppArmor 配置文件：

1. 为构成应用程序的各个程序创建配置文件。

尽管此方法是系统性的，但 AppArmor 只监视具有配置文件及其子配置文件的程序。要让 AppArmor 将某个程序考虑在内，您至少应通过 **aa-autodep** 为此程序创建一个大概的配置文件。要创建此大概的配置文件，请参见第 33.7.3.1 节 “**aa-autodep** — 创建大概的配置文件”。

2. 使相关的配置文件进入学习或提示模式。

在终端窗口中输入以下命令

```
tux > sudo aa-complain /etc/apparmor.d/*
```

（以 **root** 身份登录后），为所有构建了配置文件的程序激活学习或控诉模式。也可以通过第 32.4.2 节 “更改单个配置文件的模式” 中所述的 “YaST 配置文件模式” 模块使用此功能。

处于学习模式时，访问请求不会被阻止，即使配置文件指示应阻止时也是如此。这样您就可以完整运行若干测试（如[步骤 3](#)所示）并了解程序正常运行时的访问需要。通过此信息，您可以确定配置文件的防护程度。

有关使用学习模式和提示模式的具体说明，请参见[第 33.7.3.2 节 “aa-complain — 进入控诉或学习模式”](#)。

3. 演习应用程序。

运行应用程序并行使其功能。演习程序的多少功能由您决定，但您必须让程序访问代表其访问要求的每个文件。由于执行不受 **aa-genprof** 的监督，因此此步骤可以持续数天或数周时间，而且在所有系统重引导后仍会持续。

4. 分析日志。

进行系统性配置文件构建时，请直接运行 **aa-logprof**，而不要让 **aa-genprof** 来运行它（进行独立式配置文件构建时会如此操作）。**aa-logprof** 的一般格式如下：

```
tux > sudo aa-logprof [ -d /path/to/profiles ] [ -f /path/to/logfile ]
```

有关使用 **aa-logprof** 的详细信息，请参见[第 33.7.3.9 节 “aa-logprof — 扫描系统日志”](#)。

5. 重复[步骤 3](#)和[步骤 4](#)。

这样会生成最佳的配置文件。此重复操作方式会捕捉可经过训练并重新载入策略引擎的较小数据集。后续重复操作将生成较少的消息，而且运行速度较快。

6. 编辑配置文件。

您应该查看已生成的配置文件。可以使用文本编辑器打开和编辑 `/etc/apparmor.d/` 中的配置文件。

7. 返回到强制模式。

此时系统会重新强制执行配置文件的规则，而不仅仅是记录信息。此操作可以通过从配置文件中去除 `flags=(complain)` 文本手动完成，也可以使用 **aa-enforce** 命令自动完成，该命令的工作方式与 **aa-complain** 命令完全相同，只不过会将配置文件设置为强制模式。也可以通过[第 32.4.2 节 “更改单个配置文件的模式”](#)中所述的“YaST 配置文件模式”模块使用此功能。

要确保所有配置文件都已解除控诉模式并已置于强制模式，请输入 **aa-enforce /etc/apparmor.d/***。

8. 重新扫描所有配置文件。

要让 AppArmor 重新扫描所有配置文件并在内核中更改该强制模式，请输入 `systemctl reload apparmor`。

33.7.3 构建配置文件的工具汇总

`apparmor-utils` RPM 软件包提供了用于构建 AppArmor 配置文件的所有实用程序（储存在 `/usr/sbin` 中）。每个工具都有不同的用途。

33.7.3.1 `aa-autodep` — 创建大概的配置文件

此工具可为所选的程序或应用程序创建大概的配置文件。您可以为二进制可执行文件和已解释的脚本程序生成大概的配置文件。生成的配置文件之所以称为“大概的配置文件”，是因为它不一定包含 AppArmor 正确限制程序所需的所有配置文件项。最小的 `aa-autodep` 大概配置文件至少具备基础的 `include` 指令，它包含大多数程序都需要的基本配置文件项。对于某些类型的程序，`aa-autodep` 会生成内容更丰富的配置文件。配置文件的生成方式是在命令行上列出的可执行文件上以递归方式调用 `ldd(1)`。

要生成大概的配置文件，请使用 `aa-autodep` 程序。程序参数可以是程序的简单名称（`aa-autodep` 通过搜索外壳的路径变量找到此名称），也可以是完全限定的路径。程序本身可以是任意类型（ELF 二进制文件、外壳脚本、Perl 脚本，等等）。`aa-autodep` 会生成一个大概的配置文件，后续的动态配置文件构建过程会对其进行改进。

生成的大概配置文件将写入到 `/etc/apparmor.d` 目录，并使用 AppArmor 配置文件命名约定，即在程序绝对路径后面命名配置文件，同时将路径中的正斜线 (`/`) 字符替换为句点 (`.`) 字符。`aa-autodep` 的一般语法是在终端窗口中输入以下命令：

```
tux > sudo aa-autodep [ -d /PATH/TO/PROFILES ] [PROGRAM1 PROGRAM2...]
```

如果不输入程序名称，则计算机会提示您输入它（它们）。如果您将配置文件保存在非默认位置，`/path/to/profiles` 会覆盖 `/etc/apparmor.d` 的默认位置。

开始构建配置文件前，您必须为作为应用程序一部分的各个主可执行服务创建配置文件（它们启动后不会从属于其它已具备配置文件的程序）。此类程序的查找取决于相关的应用程序。以下是查找此类程序的一些策略：

目录

如果需要构建配置文件的所有程序都位于同一个目录内，而且此目录中没有其他程序，只需使用 **aa-autodep** `/path/to/your/programs/*` 命令就可以为此目录中的所有程序创建基本配置文件。

pstree -p

您可以运行应用程序并使用标准的 Linux **pstree** 命令找到所有运行中的进程。然后手动搜寻这些程序的位置，并针对每个程序运行 **aa-autodep**。如果程序在您的路径中，则 **aa-autodep** 会为您找到它们。如果程序不在您的路径中，则标准的 Linux 命令 **find** 可能有助于您查找程序。执行 **find / -name 'MY_APPLICATION' -print** 以确定应用程序的路径（MY_APPLICATION 为示例应用程序）。如果情况适合，您可以使用通配符。

33.7.3.2 aa-complain — 进入控诉或学习模式

控诉和学习模式工具 (**aa-complain**) 会检测对 AppArmor 配置文件规则的违规，例如构建了配置文件的程序访问配置文件不允许的其他文件。冲突是允许的，但也会被记录。要改进配置文件，请开启控诉模式，对程序运行一套测试以生成反映程序访问需求的日志事件，然后使用 AppArmor 工具对日志进行后处理，以将日志事件转换为改进的配置文件。

手动激活控诉模式（使用命令行）会在配置文件的顶部添加一个标志，因此 `/bin/foo` 将变成 `/bin/foo flags=(complain)`。要使用控诉模式，请打开一个终端窗口，然后以 root 身份输入以下命令：

- 如果示例程序 (PROGRAM1) 在您的路径中，请使用：

```
tux > sudo aa-complain [PROGRAM1 PROGRAM2 ...]
```

- 如果程序不在您的路径中，请指定整个路径，如下所示：

```
tux > sudo aa-complain /sbin/PROGRAM1
```

- 如果配置文件不在 `/etc/apparmor.d` 中，请输入以下命令以取代默认位置：

```
tux > sudo aa-complain /path/to/profiles/PROGRAM1
```

- 按如下所示指定 `/sbin/program1` 的配置文件：

```
tux > sudo aa-complain /etc/apparmor.d/sbin.PROGRAM1
```

上述每条命令都会激活所列配置文件或程序的控诉模式。如果程序名不包含其整个路径，则 **aa-complain** 命令会搜索该程序的 `$PATH`。例如，**aa-complain /usr/sbin/*** 会查找与 `/usr/sbin` 中的所有程序关联的配置文件，并将其置于控诉模式。**aa-complain /etc/apparmor.d/*** 会将 `/etc/apparmor.d` 中的所有配置文件都置于控诉模式。



提示：使用 YaST 切换配置文件模式

YaST 提供了一个用于切换控诉模式和强制模式的图形前端。有关信息，请参见第 32.4.2 节“更改单个配置文件的模式”。

33.7.3.3 aa-decode — 解码 AppArmor 日志文件中的十六进制编码字符串

aa-decode 将解码 AppArmor 日志输出中的十六进制编码字符串。它还可以处理有关标准输入的审计日志，转换任何十六进制编码的 AppArmor 日志项，并在标准输出中显示这些项。

33.7.3.4 aa-disable — 禁用 AppArmor 安全配置文件

使用 **aa-disable** 可以禁用一个或多个 AppArmor 配置文件的强制模式。此命令将从内核中卸载配置文件，并防止在 AppArmor 启动时装载该配置文件。使用 **aa-enforce** 或 **aa-complain** 实用程序可更改此行为。

33.7.3.5 aa-easyprof — 轻松生成配置文件

aa-easyprof 提供了便于使用的界面来生成 AppArmor 配置文件。**aa-easyprof** 支持使用模板和配置文件组来快速构建应用程序的配置文件。尽管 **aa-easyprof** 有助于生成配置文件，但其实用程序依赖于所用模板、配置文件组和抽象的质量。此外，此工具在创建配置文件方面的限制比手动或使用 **aa-genprof** 和 **aa-logprof** 创建配置文件要少一些。

有关详细信息，请参见 **aa-easyprof** (8) 的手册页。

33.7.3.6 aa-enforce — 进入强制模式

强制模式会检测对 AppArmor 配置文件规则的违规，例如构建了配置文件的程序访问配置文件不允许的文件。违规会被记录下来，并且不会允许此类事件。默认会启用强制模式。如要仅记录违规但仍允许此类事件，请使用控诉模式。

手动激活强制模式（使用命令行）会去除配置文件顶部的 `complain` 标志，使 `/bin/foo flags=(complain)` 变成 `/bin/foo`。要使用强制模式，请打开一个终端窗口，然后输入以下命令。

- 如果示例程序 (`PROGRAM1`) 在您的路径中，请使用：

```
tux > sudo aa-enforce [PROGRAM1 PROGRAM2 ...]
```

- 如果程序不在您的路径中，请指定整个路径，如下所示：

```
tux > sudo aa-enforce /sbin/PROGRAM1
```

- 如果配置文件不在 `/etc/apparmor.d` 中，请输入以下命令以取代默认位置：

```
tux > sudo aa-enforce -d /path/to/profiles/ program1
```

- 按如下所示指定 `/sbin/program1` 的配置文件：

```
tux > sudo aa-enforce /etc/apparmor.d/sbin.PROGRAM1
```

上述命令会激活所列配置文件和程序的强制模式。

如果不输入程序或配置文件的名称，则计算机会提示您输入名称。`/path/to/profiles` 会取代默认位置 `/etc/apparmor.d`。

参数可以是一个程序列表或一个配置文件列表。如果程序名不包含其整个路径，则 **aa-enforce** 命令会搜索程序的 `$PATH`。



提示：使用 YaST 切换配置文件模式

YaST 提供了一个用于切换控诉模式和强制模式的图形前端。有关信息，请参见第 32.4.2 节“更改单个配置文件的模式”。

33.7.3.7 aa-exec — 使用指定的配置文件限制程序

使用 **aa-exec** 可以启动指定的配置文件和/或配置文件名称空间所限制的程序。如果同时指定了配置文件和名称空间，程序将由新名称空间中的配置文件限制。如果仅指定了配置文件名称空间，将使用当前限制的配置文件名称。如果配置文件和名称空间均未指定，将使用标准的配置文件附件运行命令 — 如同未使用 **aa-exec** 命令一样。

有关该命令的选项的详细信息，请参见该命令的手册页 **man 8 aa-exec**。

33.7.3.8 aa-genprof — 生成配置文件

aa-genprof 是用于生成 AppArmor 配置文件的实用程序。它可以对指定的程序运行 **aa-autodep** 以创建大概配置文件（如果此程序尚不存在配置文件）、将其设置为控诉模式、将其重新装载到 AppArmor、标记日志、提示用户执行程序以及运用其功能。其语法如下所示：

```
tux > sudo aa-genprof [ -d /path/to/profiles ] PROGRAM
```

要为 Apache Web 服务器程序 **httpd2-prefork** 创建配置文件，请以 **root** 身份执行以下操作：

1. 输入 **systemctl stop apache2**。
2. 接下来，输入 **aa-genprof httpd2-prefork**。

现在，**aa-genprof** 会执行以下操作：

1. 使用外壳的路径变量解析 **httpd2-prefork** 的完整路径。您也可以指定完整路径。
在 SUSE Linux Enterprise Server 上，默认的完整路径为 **/usr/sbin/httpd2-prefork**。
2. 检查 **httpd2-prefork** 是否已存在配置文件。如果已有，则会被更新。如果不存在，则会使用第 33.7.3 节 “构建配置文件的工具汇总” 中所述的 **aa-autodep** 创建一个配置文件。
3. 将此程序的配置文件置于学习或控诉模式，这样会记录配置文件违规，但允许此类违规以便继续操作。日志事件如下所示（请参见 **/var/log/audit/audit.log**）：

```
type=APPARMOR_ALLOWED msg=audit(1189682639.184:20816): \
```

```
apparmor="DENIED" operation="file_mmap" parent=2692 \  
profile="/usr/sbin/httpd2-prefork//HANDLING_UNTRUSTED_INPUT" \  
name="/var/log/apache2/access_log-20140116" pid=28730 comm="httpd2-  
prefork" \  
requested_mask="::r" denied_mask="::r" fsuid=30 ouid=0
```

如果您未运行审计守护程序，AppArmor 事件将直接记录到 `systemd` 日记（请参见《管理指南》，第 17 章 “**journalctl**: 查询 `systemd` 日志”）：

```
Sep 13 13:20:30 K23 kernel: audit(1189682430.672:20810): \  
apparmor="DENIED" operation="file_mmap" parent=2692 \  
profile="/usr/sbin/httpd2-prefork//HANDLING_UNTRUSTED_INPUT" \  
name="/var/log/apache2/access_log-20140116" pid=28730 comm="httpd2-  
prefork" \  
requested_mask="::r" denied_mask="::r" fsuid=30 ouid=0
```

您也可以使用 **dmesg** 命令来查看这些事件：

```
audit(1189682430.672:20810): apparmor="DENIED" \  
operation="file_mmap" parent=2692 \  
profile="/usr/sbin/httpd2-prefork//HANDLING_UNTRUSTED_INPUT" \  
name="/var/log/apache2/access_log-20140116" pid=28730 comm="httpd2-  
prefork" \  
requested_mask="::r" denied_mask="::r" fsuid=30 ouid=0
```

4. 使用日志事件的起始标记来标记要考虑的日志。例如：

```
Sep 13 17:48:52 figwit root: GenProf: e2ff78636296f16d0b5301209a04430d
```

3. 看到工具的提示时，在另一个终端窗口中运行应用程序，并执行尽可能多的应用程序功能。如此，学习模式便可以记录程序在正常运行时需要访问的文件和目录。例如，在新终端窗口中输入 **systemctl start apache2**。

4. 执行程序功能后，在 **aa-genprof** 终端窗口中选择以下可用选项：

- **S** 会在 **aa-genprof** 启动并重新装载配置文件后从标记位置开始对系统日志运行 **aa-genprof**。如果日志中存在系统事件，AppArmor 会分析学习模式日志文件。这会生成一系列问题，您必须回答这些问题以引导 **aa-genprof** 生成安全配置文件。
- **F** 会退出工具。



注意

如果出现添加帽子的请求，请进入第 34 章 “使用 ChangeHat 构建 Web 应用程序的配置文件”。

5. 回答两类问题：

- 构建配置文件的程序访问了配置文件中没有的资源（请参见例 33.1 “学习模式例外：控制对特定资源的访问”）。
- 构建配置文件的程序执行了一个程序，而安全域转换尚未定义（请参见例 33.2 “学习模式例外：定义项的权限”）。

这两种类别都会生成一系列问题，您必须回答这些问题，以将资源或程序添加到配置文件。例 33.1 “学习模式例外：控制对特定资源的访问”和例 33.2 “学习模式例外：定义项的权限”提供了每种类别的示例。后续步骤将说明回答这些问题时的选项。

- 处理执行权限非常复杂。您必须决定如何继续处理此项，指定向此项授予哪种执行权限类型：

例 33.1：学习模式例外：控制对特定资源的访问

```
Reading log entries from /var/log/audit/audit.log.
Updating AppArmor profiles in /etc/apparmor.d.

Profile: /usr/sbin/cupsd
Program: cupsd
Execute: /usr/lib/cups/daemon/cups-lpd
Severity: unknown
```

```
(I)nherit / (P)rofile / (C)hild / (N)ame / (U)nconfined / (X)ix /  
(D)eny / Abo(r)t / (F)inish
```

继承 (ix)

子程序继承父程序的配置文件，以与父程序相同的访问控制运行。当被限制的
程序需要调用其它被限制的程序时此模式非常实用，无须获得目标程序配置文
件的权限或失去当前配置文件的权限。当子程序是**助手应用程序**（例如，使用
less 作为分页器的 **/usr/bin/mail** 客户端）时，通常使用此模式。

配置文件 (px/px)

子程序以自己的配置文件运行，此配置文件必须载入内核。如果不存在配置文
件，则尝试执行子程序时会因访问被拒而失败。这最适合父程序在调用全局服
务的情形，如进行 DNS 查找或通过系统的 MTA 发送邮件时。

选择含清洁执行的配置文件 (Px) 选项可以整理在传递给子进程时可能会修改执
行行为的环境变量的环境。

子项 (cx/cx)

设置目标为子配置文件的转换。它与 px/Px 转换类似，只不过是转换为子配置
文件。

选择含清洁执行的配置文件 (Cx) 选项可以整理在传递给子进程时可能会修改执
行行为的环境变量的环境。

未受限 (ux/ux)

对执行的资源不应用任何 AppArmor 配置文件，子项在完全没有限制的情况下
运行。

选择含清洁执行的未受限 (Ux) 选项可以整理在传递给子进程时可能会修改执
行行为的环境变量的环境。请注意，运行无限制的配置文件会造成安全漏洞，
攻击者可能会利用此漏洞来避开 AppArmor。请仅在万不得已的情况下才这样
做。

mmap (m)

此权限与 **PROT_EXEC** 标志结合表示基于配置文件运行的程序可以使用 mmap
系统调用来访问资源。这意味着可以执行其中映射的数据。如果在配置文件构
建过程运行期间要求此权限，系统会提示您包含此权限。

拒绝

在配置文件中添加一条拒绝规则，以永久阻止程序访问指定的目录路径项。AppArmor 随后会继续处理下一个事件。

中止

中止 **aa-logprof**，到目前为止输入的所有规则更改将会丢失，所有配置文件都将保持不变。

完成

关闭 **aa-logprof**，同时保存到目前为止输入的所有规则更改并修改所有配置文件。

- 例 33.2 “学习模式例外：定义项的权限”演示了 AppArmor 建议允许使用通配模式 /var/run/nscd/* 进行读取，然后使用一个抽象来涵盖常用的 Apache 相关访问规则。

例 33.2：学习模式例外：定义项的权限

```
Profile: /usr/sbin/httpd2-prefork
Path:    /var/run/nscd/dbSz9CTr
Mode:    r
Severity: 3

1 - /var/run/nscd/dbSz9CTr
[2 - /var/run/nscd/*]

(A)llow / [(D)eny] / (G)lob / Glob w/(E)xt / (N)ew / Abo(r)t /
(F)inish / (O)pts
Adding /var/run/nscd/* r to profile.

Profile: /usr/sbin/httpd2-prefork
Path:    /proc/11769/attr/current
Mode:    w
Severity: 9

[1 - #include <abstractions/apache2-common>]
2 - /proc/11769/attr/current
3 - /proc/*/attr/current
```

```
(A)llow / [(D)eny] / (G)lob / Glob w/(E)xt / (N)ew / Abo(r)t /  
(F)inish / (O)pts  
Adding #include <abstractions/apache2-common> to profile.
```

AppArmor 提供了一个或多个路径或 include。通过输入选项编号选择所需的选项，然后继续执行下一步。



注意

AppArmor 菜单中并不会始终显示所有这些选项。

#include

AppArmor 配置文件的此部分代表一个 include 文件，它会获取程序的访问权限。通过使用 include，您可以向程序赋予访问其它程序也需要的目录路径和文件的权限。使用 include 可减小配置文件的大小。选择建议的 include 是不错的做法。

通配形式

按下一步所述选择通配即可访问该部分。有关通配语法的更多信息，请参见第 30.6 节“配置文件名称、标志、路径和通配”。

实际路径

这是程序要正常运行需要访问的实际路径。

选择路径或 include 后，通过选择允许或拒绝来处理它，将它以项的形式加入到 AppArmor 配置文件中。如果您对显示的目录路径项不满意，也可以使用通配对其进行处理。

以下选项用于处理学习模式项和构建配置文件：

选择 **Enter**

允许访问选定的目录路径。

允许

允许访问指定的目录路径项。AppArmor 会给出文件访问权限建议。有关更多信息，请参考第 30.7 节“文件权限访问模式”。

拒绝

阻止程序访问指定目录路径项的权限。AppArmor 随后会继续处理下一个事件。

新建

提示您输入针对此事件的自己的规则，允许指定正则表达式。如果该表达式实际上不满足最初提示问题的事件，AppArmor 会要求您确认并允许您重新输入表达式。

通配

选择特定的路径，或使用通配符创建与更多路径集匹配的一般规则。要选择提供的任何路径，请输入该路径前面列显的编号，然后决定如何继续处理所选项。

有关通配语法的详细信息，请参见第 30.6 节“配置文件名称、标志、路径和通配”。

保留扩展名的通配

此选项会修改原始目录路径，不过会保留文件扩展名。例如，`/etc/apache2/file.ext` 将变成 `/etc/apache2/*.ext`，文件名被通配符（星号）替换。这样程序就可以访问建议目录下以 `.ext` 为扩展名的所有文件。

中止

中止 **aa-logprof**，到目前为止输入的所有规则更改将会丢失，所有配置文件都将保持不变。

完成

关闭 **aa-logprof**，同时保存到目前为止输入的所有规则更改并修改所有配置文件。

6. 要使用 **vi** 查看并编辑配置文件，请在终端窗口中输入 **vi /etc/apparmor.d/PROFILENAME**。在 vim 中编辑 AppArmor 配置文件时，如要启用语法高亮显示，请使用 **:syntax on** 命令，然后使用 **:set syntax=apparmor** 命令。有关 vim 和语法高亮显示的详细信息，请参见第 33.7.3.14 节“**apparmor.vim**”。
7. 重新启动 AppArmor，然后使用 **systemctl reload apparmor** 命令重新装载配置文件集，包括新建的配置文件。

与用于构建 AppArmor 配置文件的图形前端一样，YaST 添加配置文件向导 **aa-genprof** 也支持使用 `/usr/share/apparmor/extra-profiles` 下的本地配置文件储存库。

要使用本地储存库中的配置文件，请按如下所示继续操作：

1. 按上文所述启动 **aa-genprof**。

如果 **aa-genprof** 找到了非活动的本地配置文件，则终端窗口中会显示以下几行：

```
Profile: /usr/bin/opera

[1 - Inactive local profile for /usr/bin/opera]

[(V)iew Profile] / (U)se Profile / (C)reate New Profile / Abo(r)t /
(F)inish
```

2. 要使用此配置文件，请按 **U**（使用配置文件）并执行上文所述的配置文件生成过程。
要在激活配置文件之前对其进行检查，请按 **V**（查看配置文件）。
要忽略现有的配置文件，请按 **C**（创建新配置文件），并执行上文所述的配置文件生成过程从头开始创建配置文件。
3. 完成后，按 **F**（完成）退出 **aa-genprof** 并保存更改。

33.7.3.9 aa-logprof — 扫描系统日志

aa-logprof 是一个交互式工具，用于查看 `/var/log/audit/audit.log` 内的日志项中的控诉和强制模式事件，或直接查看 `systemd` 日记中的此类事件（请参见《管理指南》，第 17 章“**journalctl**：查询 `systemd` 日志”），以及在 AppArmor 安全配置文件中生成新的项。

运行 **aa-logprof** 后，它会开始扫描在控诉和强制模式下生成的日志文件，如果存在现有配置文件集未涵盖的新安全事件，它会给出修改配置文件的建议。**aa-logprof** 使用此信息来观察程序行为。

如果某个受限制的程序派生并执行另一个程序，**aa-logprof** 会注意到这种情况，并会询问用户在启动子进程时应使用哪种执行模式。执行模式 **ix**、**px**、**Px**、**ux**、**Ux**、**cx**、**Cx** 以及命名的配置文件均为用于启动子进程的选项。如果子进程具有单独的配置文件，则默认选择为 **Px**。如果不存在单独的配置文件，则默认为 **ix**。系统会对有单独配置文件的子进程运行 **aa-autodep** 并将其装载到 AppArmor 中（如果它正在运行）。

aa-logprof 退出时，将保存所做的更改以更新配置文件。如果 AppArmor 处于活动状态，将重新装载更新的配置文件；如有任何生成安全事件的进程仍在 `null-XXXX` 配置文件（在控诉模式下临时创建的独有配置文件）中运行，这些进程将设置为基于其适当配置文件运行。

要运行 **aa-logprof**，请以 `root` 身份登录并在终端窗口中输入 **aa-logprof**。以下选项可用于 **aa-logprof**：

aa-logprof -d/path/to/profile/directory/

如果配置文件不位于标准目录 `/etc/apparmor.d/` 下，指定配置文件所处位置的完整路径。

aa-logprof -f/path/to/logfile/

如果日志文件不在默认目录或 `/var/log/audit/audit.log` 中，请指定日志文件所在位置的完整路径。

aa-logprof -m "string marker in logfile"

标记 **aa-logprof** 要在系统日志中查看的起点。**aa-logprof** 会忽略系统日志中位于指定标记前面的所有事件。如果标记包含空格，必须将标记括在引号中才能正常工作。例如：

```
root # aa-logprof -m "17:04:21"
```

或

```
root # aa-logprof -m e2ff78636296f16d0b5301209a04430d
```

aa-logprof 将扫描日志，询问您如何处理记录的每个事件。每个问题都会显示一个带有编号的 AppArmor 规则列表，按列表中项目的编号可以添加相应规则。

默认情况下，**aa-logprof** 会在 `/etc/apparmor.d/` 中查找配置文件。以 `root` 身份运行 **aa-logprof** 往往便足以更新配置文件。不过，您有时可能需要搜索存档的日志文件，例如当程序的执行时间段超过日志轮换期时（日志文件已存档，新的日志文件已开始时）。在这种情况下，您可以输入 **zcat -f `ls -ltr /path/to/logfile\*` | aa-logprof -f -**。

33.7.3.10 aa-logprof 示例 1

下面的示例说明 **aa-logprof** 如何处理访问 `/etc/group` 文件的 `httpd2-prefork`。[] 表示默认选项。

在本示例中，对 `/etc/group` 的访问是 `httpd2-prefork` 访问名称服务的一部分。相应的响应是 1，它包括预定义的 AppArmor 规则集。选择 1 `#include`，名称服务软件包会解析与 DNS 查找相关的所有问题，同时使配置文件更不容易损坏，这样对 DNS 配置和关联名称服务配置文件软件包的更改可一次完成，而无需修改许多配置文件。

```
Profile: /usr/sbin/httpd2-prefork
Path:    /etc/group
New Mode: r

[1 - #include <abstractions/nameservice>]
 2 - /etc/group
[(A)llow] / (D)eny / (N)ew / (G)lob / Glob w/(E)xt / Abo(r)t / (F)inish
```

请选择以下响应之一：

选择 **Enter**

触发默认操作，在本示例中为允许访问指定的目录路径项。

允许

允许访问指定的目录路径项。AppArmor 会给出文件访问权限建议。有关更多信息，请参见第 30.7 节“文件权限访问模式”。

拒绝

永久阻止程序访问指定的目录路径项。AppArmor 随后会继续处理下一个事件。

新建

提示您输入您自己对此事件的规则，允许您指定任意形式的常规表达式。如果输入的表达式实际上不满足最初提示问题的事件，AppArmor 会要求您确认并允许您重新输入表达式。

通配

选择特定的路径，或使用通配符创建与更多路径集匹配的一般规则。要选择提供的任何路径，请输入路径前面列显的编号，然后决定如何继续处理所选项。

有关通配语法的详细信息，请参见第 30.6 节“配置文件名称、标志、路径和通配”。

保留扩展名的通配

此选项会修改原始目录路径，不过会保留文件扩展名。例如，`/etc/apache2/file.ext` 将变成 `/etc/apache2/*.ext`，文件名被通配符（星号）替换。这样程序就可以访问建议目录下以 `.ext` 为扩展名的所有文件。

中止

中止 **aa-logprof**，到目前为止输入的所有规则更改将会丢失，所有配置文件都将保持不变。

完成

关闭 **aa-logprof**，同时保存到目前为止输入的所有规则更改并修改所有配置文件。

33.7.3.11 aa-logprof 示例 2

例如，在为 vsftpd 构建配置文件时，会看到以下问题：

```
Profile: /usr/sbin/vsftpd
Path:    /y2k.jpg

New Mode: r

[1 - /y2k.jpg]

(A)llow / [(D)eny] / (N)ew / (G)lob / Glob w/(E)xt / Abo(r)t / (F)inish
```

此问题中出现了几个要注意的项目。首先，我们注意到 vsftpd 正在询问树顶的路径项，即便默认情况下 SUSE Linux Enterprise Server 上的 vsftpd 是从 `/srv/ftp` 提供 FTP 文件。这是因为 vsftpd 使用的是 chroot，对于 chroot jail 内部的代码部分，AppArmor 看到的是对 chroot 环境（而非全局绝对路径）的文件访问。

第二个要注意的事项是，您应该授予对该目录中所有 JPEG 文件的 FTP 读取权限，以便可以使用保留扩展名的通配并使用建议的路径 `/*.jpg`。这样做会破坏前面授予对单个 `.jpg` 文件的访问权限的所有规则，并会阻止此后有关访问 `.jpg` 文件的所有问题。

最后，您应该授予对 FTP 文件的更广泛的访问权限。如果在最后一项中选择通配，**aa-logprof** 会将建议的路径 `/y2k.jpg` 替换为 `/*`。此外，您应该授予对整个目录树的更多访问权限，在这种情况下，可以使用新建路径选项并输入 `/**/*.jpg`（这会授予对整个目录树中所有 `.jpg` 文件的访问权限）或 `/**`（这会授予对目录树中所有文件的访问权限）。

这些项会处理读取访问权限。写权限与此类似，不同的是您在使用写权限的常规表达式时最好更加保守。处理执行权限较为复杂。例 33.1 “学习模式例外：控制对特定资源的访问”中提供了相应示例。

下面的示例将构建 `/usr/bin/mail` 邮件客户端的配置文件，**aa-logprof** 已发现 `/usr/bin/mail` 将 `/usr/bin/less` 作为助手应用程序执行，用于将长邮件“分页”。结果会显示以下提示：

```
/usr/bin/nail -> /usr/bin/less
(I)nherit / (P)rofile / (C)hild / (N)ame / (U)nconfined / (X)ix / (D)eny
```



注意

`/usr/bin/mail` 的实际可执行文件为 `/usr/bin/nail`，这不是印刷错误。

`/usr/bin/less` 程序可以简单地在长度大于一个屏幕的文本之间滚动，这其实就是 `/usr/bin/mail` 使用它的原因。但 **less** 实际上是一个功能强大的大型程序，使用了许多其他助手应用程序，如 **tar** 和 **rpm**。



提示

对 **tar** 文件或 **RPM** 文件运行 **less**，它即会显示这些容器的库存。

您不希望阅读邮件时自动运行 **rpm**（这会直接导致 Microsoft* Outlook 样式的病毒攻击，因为 **RPM** 具有安装和修改系统程序的能力），因此这种情况下的最佳选择为使用继承。这会使在本文中执行的 **less** 程序运行于 `/usr/bin/mail` 的配置文件之下。这会产生两种结果：

- 您需要将 `/usr/bin/less` 的所有基本文件权限添加到 `/usr/bin/mail` 的配置文件中。
- 您可以避免将 `tar` 和 `rpm` 等助手应用程序添加到 `/usr/bin/` 配置文件，这样，当 `/usr/bin/mail` 在此环境中运行 `/usr/bin/mail/less` 时，`less` 程序要比不受 AppArmor 保护时安全得多。另一个选择是使用 `Cx` 执行模式。有关执行模式的详细信息，请参见第 30.12 节“执行模式”。

在其它情况下，您可能想要使用配置文件选项。这会对 `aa-logprof` 产生以下影响：

- 写入配置文件的规则使用 `px/Px`，这会强制转换到子项自己的配置文件。
- `aa-logprof` 会为子项构造一个配置文件，然后将子进程的事件指派到子项的配置文件并向 `aa-logprof` 用户提出问题，开始以构建父配置文件的相同方式构建此配置文件。如果您将子项作为独立程序运行，也会应用该配置文件。

如果某个受限制的程序派生并执行另一个程序，`aa-logprof` 会注意到这种情况，并会询问用户在启动子进程时应使用哪种执行模式。执行模式“继承”、“配置文件”、“未受限”、“子项”、“命名配置文件”，或用于拒绝执行的选项将会显示。

如果子进程具有单独的配置文件，则默认选择为“配置文件”。如果不存在配置文件，则默认选择为“继承”。第 30.7 节“文件权限访问模式”中介绍了继承选项 (`ix`)。

该配置文件选项指示子程序应在其自己的配置文件中运行。一个从属问题会询问是否要清理子程序从父项继承的环境。如果您选择清理环境，AppArmor 配置文件中将会添加执行修饰符 `Px`。如果您选择不清理，配置文件中将会添加 `px`，这样就不会进行环境清理。如果您选择配置文件执行模式，默认的执行模式为 `Px`。

不建议使用未受限执行模式，仅当没有任何其他选项能够可靠地生成程序的配置文件时，才应使用该模式。选择未受限模式会打开一个警告对话框，要求您确认该选择。如果您确认并选择是，另一个对话框将会打开，询问是否要清理环境。要在配置文件中执行模式 `Ux`，请选择是。要在配置文件中改用执行模式 `ux`，请选择否。默认选择的值为 `Ux`（表示未受限执行模式）。

重要：未受限运行

选择 `ux` 或 `Ux` 会造成很大的风险，它不会对子程序的最终执行行为强制执行策略（从安全角度而言）。

33.7.3.12 aa-unconfined — 识别不受保护的进程

aa-unconfined 命令会检查系统上的开放网络端口，将其与系统上装载的配置文件集进行比较，并报告不具备 AppArmor 配置文件的网络服务。它需要未被 AppArmor 配置文件限制的 root 特权。

要从 /proc 文件系统中检索进程可执行链接，就必须以 root 身份运行 **aa-unconfined**。此程序易受以下竞态条件的影响：

- 未链接的可执行文件被误处理
- 进程在 **netstat(8)** 之间终止，而且进一步的检查被误处理



注意

此程序仅列出使用 TCP 和 UDP 的进程。简而言之，此程序不适用于取证，仅在实验室中辅助构建所有可访问网络的进程的配置文件。

33.7.3.13 aa-notify

aa-notify 是一个便利的实用程序，它可在桌面环境中显示 AppArmor 通知。如果您不想检查 AppArmor 日志文件，而仅希望桌面告知您违反策略的事件，此实用程序会非常便捷。要启用 AppArmor 桌面通知，请运行 **aa-notify**：

```
tux > sudo aa-notify -p -u USERNAME --display DISPLAY_NUMBER
```

其中，USERNAME 是您登录时使用的用户名，DISPLAY_NUMBER 是您当前使用的 X Window 显示编号，例如 :0。该进程在后台运行，每当发生拒绝事件时，就会显示通知。



提示

活动的 X Window 显示编号保存在 \$DISPLAY 变量中，因此，您可以使用 --display \$DISPLAY 来避免查找当前显示编号。

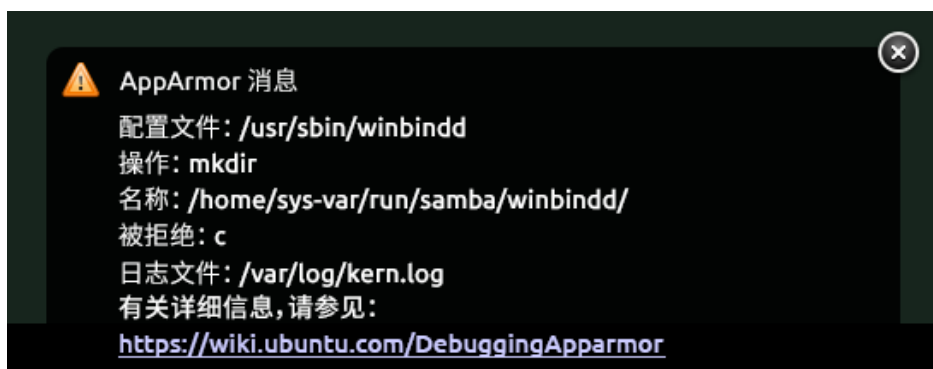


图 33.1： **aa-notify** GNOME 中的消息

您还可以使用 `-s DAYS` 选项配置 **aa-notify**，以显示指定的过去几天的通知摘要。有关 **aa-notify** 的详细信息，请参见其手册页 `man 8 aa-notify`。

33.7.3.14 **apparmor.vim**

vim 文本编辑器的语法高亮显示文件会用颜色高亮显示 AppArmor 配置文件的各种功能。利用 vim 和 vim 的 AppArmor 语法模式，您可以看到用颜色高亮显示的配置文件的语义含意。要使用 vim 来查看和编辑配置文件，请在终端窗口中输入 vim。

在 vim 中编辑 AppArmor 配置文件时，如要启用语法颜色标记，请使用 `:syntax on` 命令，然后使用 `:set syntax=apparmor` 命令。为确保 vim 将编辑的文件类型正确识别为 AppArmor 配置文件，请将

```
# vim:ft=apparmor
```

添加到配置文件的末尾。



提示

vim 对 `/etc/apparmor.d/` 中的文件自动启用了 AppArmor 高亮显示。

启用此功能时，vim 会对配置文件的行上色：

蓝色

评论

白色

普通读权限的行

棕色

功能语句和“提示”标记

黄色

授予写权限的行

环保

授予执行权限的行 (ix 或 px)

红色

授予无限制权限的行 (ux)

红色背景

不会正确装载到 AppArmor 模块的语法错误

要获取有关语法突出显示的更多 vim 帮助，请使用 [apparmor.vim](#) 和 [vim](#) 手册页以及 vim 编辑器内部的 :help 语法。AppArmor 语法储存在 [/usr/share/vim/current/syntax/apparmor.vim](#) 中。

33.8 重要的文件名和目录

下面的列表包含 AppArmor 框架使用的最重要的文件和目录。如果您打算手动管理配置文件以及对其查错，请确保您了解这些文件和目录：

[/sys/kernel/security/apparmor/profiles](#)

表示当前装载的配置文件集的虚拟化文件。

[/etc/apparmor/](#)

AppArmor 配置文件的位置。

[/usr/share/apparmor/extra-profiles](#)

AppArmor 随附但默认未启用的本地配置文件储存库。

/etc/apparmor.d/

配置文件的位置，采用将路径中的 `/` 替换为 `.`（不替换根目录 `/`）的约定命名，这样可以更方便地管理配置文件。例如，程序 `/usr/sbin/smbd` 的配置文件命名为 `usr.sbin.smbd`。

/etc/apparmor.d/abstractions/

抽象的位置。

/etc/apparmor.d/program-chunks/

程序块的位置。

/proc/*/attr/current

检查此文件可以查看进程的限制状态以及用于限制该进程的配置文件。**`ps auxZ`** 命令可自动检索这些信息。

34 使用 ChangeHat 构建 Web 应用程序的配置文件

AppArmor® 配置文件表示单个程序实例或进程的安全策略。它适用于一个可执行程序，但是，如果该程序的一部分需要与其它部分不同的访问权限，该程序可以“变换帽子”以使用有别于主程序访问权限的安全环境。这称作**帽子**或**子配置文件**。

ChangeHat 使程序能够在 AppArmor 配置文件内变成**帽子**或从帽子变成程序。这样您就可以定义比进程更细级别的安全性。此功能要求您将各应用程序配置为“感知 ChangeHat”，也就是说，将其修改为可以在应用程序执行期间的特定时间向 AppArmor 模块发出转换安全域的请求。Apache Web 服务器就是一款 ChangeHat 感知型应用程序。

一个配置文件可以有任意数目的子配置文件，但总共只能有两个级别：子配置文件不能有其他子配置文件。子配置文件是作为单独的配置文件编写的。其名称由包含配置文件的名称后接子配置文件名称构成，两者之间以 `_` 分隔。

子配置文件可储存在父配置文件所在的同一个文件中，也可储存在不同的文件中。在包含许多帽子的站点上，建议采用后一种储存方式 — 它使策略缓存能够在帽子级别处理更改。如果所有帽子都位于父配置文件所在的同一文件中，则必须重新编译父配置文件和所有帽子。

要用作帽子的外部子配置文件必须以单词 `hat` 或字符 `^` 开头。

下面两个子配置文件**不能**用作帽子：

```
/foo//bar { }
```

或

```
profile /foo//bar { }
```

而下面两个子配置文件将被视为帽子：

```
^/foo//bar { }
```

或

```
hat /foo//bar { } # this syntax is not highlighted in vim
```

请注意，帽子的安全性比完整配置文件的安全性要弱得多。攻击者有可能能够通过利用程序中特定类型的 bug 从帽子中逃脱并进入包含配置文件。这是因为，帽子的安全性由包含进程处理的某个机密密钥所决定，而帽子中运行的代码对该密钥不得拥有访问权限。因此，`change_hat` 在应用程序服务器中的作用最大。在这些服务器中，某个语言解释器（例如 PERL、PHP 或 Java）会隔离代码片段，以便防止这些代码直接访问包含进程的内存。

本章的其余内容将介绍如何在 Apache 中使用 `change_hat` 来包含通过 `mod_perl` 和 `mod_php` 运行的 Web 服务器组件。通过提供与下文第 34.1.2 节 “位置和目录指令” 中将介绍的 `mod_apparmor` 类似的应用程序模块，可以对任意应用程序服务器使用类似的方法。



提示：更多信息

有关详细信息，请参见 `change_hat` 手册页。

34.1 配置 Apache 以使用 `mod_apparmor`

AppArmor 会为 Apache 程序提供 `mod_apparmor` 模块（`apache2-mod-apparmor` 软件包）。此模块使 Apache Web 服务器能够感知 ChangeHat。请连同 Apache 一起安装此模块。当 Apache 可感知 ChangeHat 后，便会检查以下自定义的 AppArmor 安全配置文件，检查的顺序为向其收到的各 URI 请求指定的顺序。

- URI 特定的帽子。例如 `^www_app_name/templates/classic/images/bar_left.gif`
- `DEFAULT_URI`
- `HANDLING_UNTRUSTED_INPUT`



注意：Apache 配置

如果您安装 `apache2-mod-apparmor`，请确保启用该模块，然后执行以下命令重启 Apache：

```
tux > a2enmod apparmor && sudo systemctl reload apache2
```

Apache 的配置方式是在纯文本配置文件中放置指令。主配置文件通常为 `/etc/apache2/httpd.conf`。编译 Apache 时，您可以指明此文件的位置。您可以将指令放置在这些配置文件的任一个中以改变 Apache 的行为方式。对主配置文件进行更改后，需使用 `sudo systemctl reload apache2` 重新装载 Apache，以便识别更改。

34.1.1 虚拟主机指令

`<VirtualHost>` 和 `</VirtualHost>` 指令用于封装一组仅应用于特定虚拟主机的指令。有关 Apache 虚拟主机指令的详细信息，请参见 <http://httpd.apache.org/docs/2.4/en/mod/core.html#virtualhost>。

特定于 `ChangeHat` 的配置关键字为 `AADefaultHatName`。此关键字的用法类似于 `AAHatName`，例如 `AADefaultHatName My_Funky_Default_Hat`。

您可以使用此关键字来指定用于虚拟主机和其他 Apache 服务器指令的默认帽子，这样便可对不同的虚拟主机使用不同的默认值。`AAHatName` 指令可以覆盖此关键字，仅当不存在匹配的 `AAHatName` 或者不存在 URI 所命名的帽子时，才检查此关键字。如果 `AADefaultHatName` 帽子不存在，它将回退到 `DEFAULT_URI` 帽子（如果存在）。

如果不存在匹配的帽子，则返回到“父” Apache 帽子。

34.1.2 位置和目录指令

位置和目录指令会在程序配置文件中指定帽子名称，以便 Apache 可调用与其安全性相关的帽子。对于 Apache，您可以在 <http://httpd.apache.org/docs/2.4/en/sections.html> 找到有关位置和目录指令的文档。

下面的位置指令示例针对给定的位置指定 `mod-apparmor` 应使用特定的帽子：

```
<Location /foo/>
  AAHatName MY_HAT_NAME
</Location>
```

此指令会尝试对所有以 `/foo/`（`/foo/`、`/foo/bar`、`/foo/cgi/path/blah_blah/blah` 等）开头的 URI 使用 `MY_HAT_NAME`。

目录指令的工作方式与位置指令相似，不同的是它代表的是文件系统中的路径，示例如下：

```
<Directory "/srv/www/www.example.org/docs">
  # Note lack of trailing slash
  AAHatName example.org
</Directory>
```

34.2 管理 ChangeHat 感知型应用程序

在上一节中，您已了解了 `mod_apparmor`，以及它如何帮助您保护特定的 Web 应用程序。本节通过一个真实的示例来逐步讲解如何为某个 Web 应用程序创建帽子，并使用 AppArmor 的 `change_hat` 功能来保护该应用程序。请注意，由于 YaST 的 AppArmor 模块功能有限，本章主要使用 AppArmor 的命令行工具。

34.2.1 使用 AppArmor 的命令行工具

为便于演示，我们选择名为 **Adminer** (<http://www.adminer.org/en/>) 的 Web 应用程序。它是一个以 PHP 编写的全功能 SQL 数据库管理工具，不过只包括一个 PHP 文件。要正常运行 Adminer，您需要设置一个 Apache Web 服务器、PHP 及其 Apache 模块，以及一个适用于 PHP 的数据库驱动程序 — 本示例使用 MariaDB。您可使用以下命令安装所需的软件包

```
zypper in apache2 apache2-mod_apparmor apache2-mod_php5 php5 php5-mysql
```

要设置用于运行 Adminer 的 Web 环境，请执行以下步骤：

过程 34.1：设置 WEB 服务器环境

1. 确保对 Apache 启用了 `apparmor` 和 `php5` 模块。要在任何情况下均启用这些模块，请使用：

```
tux > a2enmod apparmor php5
```

然后使用以下命令重新启动 Apache

```
tux > sudo systemctl restart apache2
```

2. 确保 MariaDB 正在运行。如果不确定，请使用以下命令将其重新启动

```
tux > sudo systemctl restart mariadb
```

3. 从 <http://www.adminer.org> 下载 Adminer，将其复制到 `/srv/www/htdocs/adminer/`，然后将其重命名为 `adminer.php`，使其完整路径为 `/srv/www/htdocs/adminer/adminer.php`。

4. 在网页浏览器的 Adminer URI 地址字段中输入 `http://localhost/adminer/adminer.php` 以测试 Adminer。如果您将 Adminer 安装到了远程服务器上，请将 `localhost` 替换为该服务器的实际主机名。

Adminer 3.7.1

[\(MySQL\) root](#)

登录

系统	MySQL ▾
服务器	localhost
用户名	
口令	
数据库	

登录 ☐ 永久登录

图 34.1：ADMINER 登录页



提示

如果您在查看 Adminer 登录页时遇到问题，请尝试检查 Apache 错误日志 `/var/log/apache2/error.log` 以寻求帮助。无法访问网页的另一个原因可能是，您的 Apache 已受 AppArmor 的控制，并且其 AppArmor 配置文件过于严格，不允许查看 Adminer。请使用 **aa-status** 检查该配置文件，如果需要，可使用以下命令暂时将 Apache 设置为控诉模式

```
root # sudo aa-complain usr.sbin.httpd2-prefork
```

Adminer 的 Web 环境就绪后，您需要配置 Apache 的 `mod_apparmor`，使 AppArmor 能够检测对 Adminer 的访问以及对特定“帽子”进行的更改。

过程 34.2：配置 mod_apparmor

1. Apache 在 `/etc/apache2/` 和 `/etc/apache2/conf.d/` 下有多个配置文件。请选择所需的配置文件并在文本编辑器中打开。本示例使用 `vim` 编辑器创建新的配置文件 `/etc/apache2/conf.d/apparmor.conf`。

```
tux > sudo vim /etc/apache2/conf.d/apparmor.conf
```

2. 将以下代码段复制到编辑后的文件中。

```
<Directory /srv/www/htdocs/adminer>
    AAHatName adminer
</Directory>
```

当 Web 用户访问 Apache 文档根目录中的 `/adminer` 目录（以及该目录中的任何文件/目录）时，此代码段可让 Apache 告知 AppArmor 发生了 `change_hat` 事件。请记住，`adminer.php` 应用程序就放置在该位置。

3. 保存文件，关闭编辑器，然后使用以下命令重新启动 Apache

```
tux > sudo systemctl restart apache2
```

现在，Apache 就能识别 Adminer 并知道“帽子”发生的更改了。接下来我们在 AppArmor 配置中创建 Adminer 的相关帽子。如果您目前还没有 AppArmor 配置文件，请先创建一个，然后再继续。请记住，如果 Apache 的主二进制文件是 `/usr/sbin/httpd2-prefork`，则相关的配置文件将命名为 `/etc/apparmor.d/usr.sbin.httpd2-prefork`。

过程 34.3：创建 ADMINER 的帽子

1. 在文本编辑器中打开文件 `/etc/apparmor.d/usr.sbin.httpd2-prefork`（如果该文件不存在，请创建一个）。其内容应如下所示：

```
#include <tunables/global>

/usr/sbin/httpd2-prefork {
    #include <abstractions/apache2-common>
    #include <abstractions/base>
    #include <abstractions/php5>
```

```

capability kill,
capability setgid,
capability setuid,

/etc/apache2/** r,
/run/httpd.pid rw,
/usr/lib{,32,64}/apache2*/** mr,
/var/log/apache2/** rw,

^DEFAULT_URI {
    #include <abstractions/apache2-common>
    /var/log/apache2/** rw,
}

^HANDLING_UNTRUSTED_INPUT {
    #include <abstractions/apache2-common>
    /var/log/apache2/** w,
}
}

```

2. 在最后一个右花括号 (}) 前面插入以下部分：

```

^adminer flags=(complain) {
}

```

请注意帽子名称后面添加的 `(complain)` — 此部分告知 AppArmor 将 `adminer` 帽子保持控诉模式。这是因为，我们稍后需要通过访问 Adminer 来了解帽子配置文件。

3. 保存文件，然后依次重新启动 AppArmor 和 Apache。

```
tux > sudo systemctl reload apparmor apache2
```

4. 检查 `adminer` 帽子是否确实处于控诉模式。

```

tux > sudo aa-status
apparmor module is loaded.
39 profiles are loaded.
37 profiles are in enforce mode.

```

```
[...]
/usr/sbin/httpd2-prefork
/usr/sbin/httpd2-prefork//DEFAULT_URI
/usr/sbin/httpd2-prefork//HANDLING_UNTRUSTED_INPUT
[...]
2 profiles are in complain mode.
/usr/bin/getopt
/usr/sbin/httpd2-prefork//adminer
[...]
```

我们可以看到，`httpd2-prefork//adminer` 装载为控诉模式。

最后一个任务是找出 `adminer` 帽子的正确规则集。这就是我们将 `adminer` 帽子设置为控诉模式的原因 — 当我们通过网页浏览器使用 `adminer.php` 时，日志记录工具将收集有关其访问要求的有用信息。然后，**`aa-logprof`** 将帮助我们创建该帽子的配置文件。

过程 34.4：生成 `adminer` 帽子的规则

1. 在网页浏览器中打开 Adminer。如果您将 Adminer 安装在本地，则 URI 为 `http://localhost/adminer/adminer.php`。
2. 选择要使用的数据库引擎（在本例中为 MariaDB），并使用现有的数据库用户名和口令登录 Adminer。您此时不需要指定数据库名称，可以在登录后再指定。使用 Adminer 执行所需的任意操作 — 创建新数据库、创建数据库的新表、设置用户特权，等等。
3. 简单测试 Adminer 的用户界面后，切换回控制台并检查日志中收集的数据。

```
tux > sudo aa-logprof
Reading log entries from /var/log/audit/audit.log.
Updating AppArmor profiles in /etc/apparmor.d.
Complain-mode changes:

Profile: /usr/sbin/httpd2-prefork^adminer
Path:    /dev/urandom
Mode:    r
Severity: 3

1 - #include <abstractions/apache2-common>
[...]
```

```
[8 - /dev/urandom]

[(A)llow] / (D)eny / (G)lob / Glob w/(E)xt / (N)ew / Abo(r)t / (F)inish /
(0)pts
```

通过 **aa-logprof** 消息，我们可以确定系统已正确检测到这个新的 **adminer** 帽子：

```
Profile: /usr/sbin/httpd2-prefork^adminer
```

aa-logprof 命令会要求您选取每个已发现的 AppArmor 事件的正确规则。指定要使用的规则，并使用 **Allow** 确认。有关使用 **aa-genprof** 和 **aa-logprof** 接口的详细信息，请参见第 33.7.3.8 节 “**aa-genprof** — 生成配置文件”。



提示

aa-logprof 通常会针对所检查的事件提供多个有效规则。有些规则属于**抽象** — 影响特定的常用目标组的预定义规则集。包含这样的抽象（而非直接的 URI 规则）有时会很有用：

```
1 - #include <abstractions/php5>
2 - /var/lib/php5/sess_3jdmii9cacj1e3jnahbtopajl7p064ai242]
```

在上面的示例中，建议点击 1 并使用 **A** 确认，以允许抽象。

4. 完成最后一项更改后，系统会要求您保存更改的配置文件。

```
The following local profiles were changed. Would you like to save them?
[1 - /usr/sbin/httpd2-prefork]

(S)ave Changes / [(V)iew Changes] / Abo(r)t
```

点击 **S** 保存更改。

5. 使用 **aa-enforce** 将配置文件设置为强制模式

```
tux > sudo aa-enforce usr/sbin/httpd2-prefork
```

然后使用 **aa-status** 检查其状态

```
tux > sudo aa-status
apparmor module is loaded.
39 profiles are loaded.
38 profiles are in enforce mode.
[...]
/usr/sbin/httpd2-prefork
/usr/sbin/httpd2-prefork//DEFAULT_URI
/usr/sbin/httpd2-prefork//HANDLING_UNTRUSTED_INPUT
/usr/sbin/httpd2-prefork//adminer
[...]
```

我们可以看到，//adminer 帽子已从**控诉**模式转变到**强制**模式。

6. 尝试在网页浏览器中运行 Adminer，如果在运行时遇到问题，请将它切换到控诉模式，重复前面出现问题的步骤，并使用 aa-logprof 更新配置文件，直到您对应用程序的功能感到满意为止。

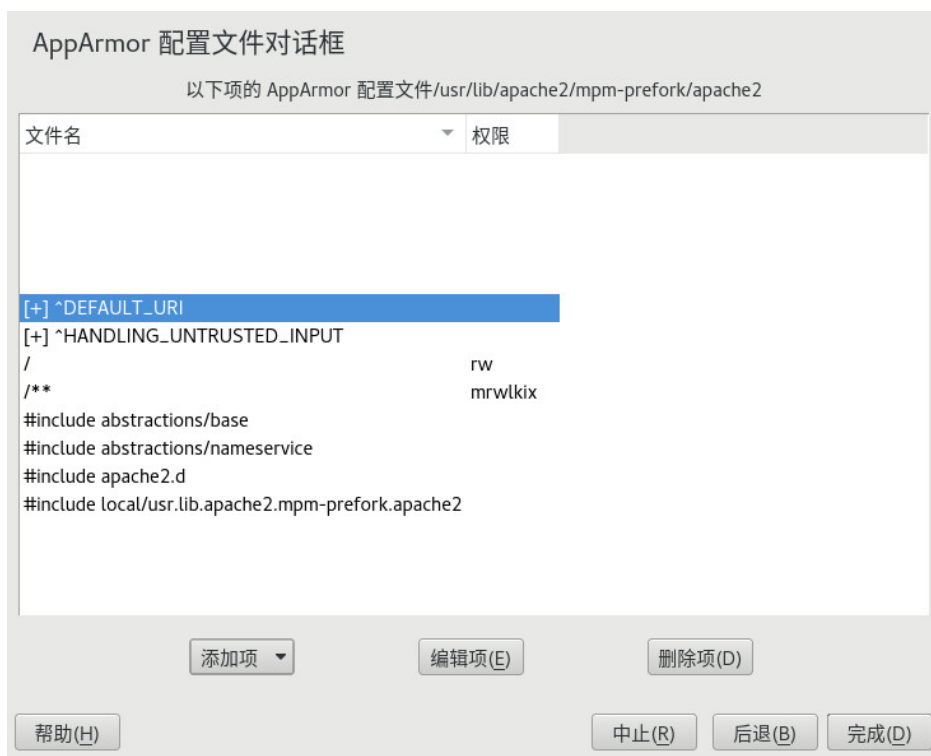


注意：帽子与父配置文件的关系

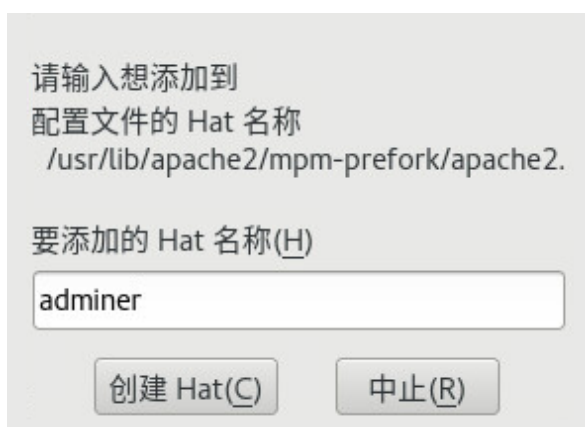
配置文件 ^adminer 仅在基于父配置文件 usr.sbin.httpd2-prefork 运行的进程的环境中可用。

34.2.2 在 YaST 中向帽子添加帽子和项

使用编辑配置文件对话框时（有关使用方法，请参见第 32.2 节“编辑配置文件”），或者使用手动添加 配置文件添加新的配置文件时（有关使用方法，请参见第 32.1 节“手动添加配置文件”），您具有将帽子（子配置文件）添加到 AppArmor 配置文件的选项。如下所示通过 AppArmor 配置文件对话框窗口添加 ChangeHat 子配置文件。



1. 在 AppArmor 配置文件对话框窗口中单击添加项，然后选择帽子。输入帽子名称对话框即会打开：



2. 输入要添加到 AppArmor 配置文件的帽子的名称。此名称为 URI，此 URI 被访问时将接收到帽子中设定的权限。
3. 单击创建帽子。返回到 AppArmor 配置文件对话框屏幕。
4. 添加新帽子后，单击完成。

35 使用 pam_apparmor 限制用户

AppArmor 配置文件将应用于可执行程序；如果该程序的某个部分所需的访问权限不同于其他部分，该程序可以通过 `change_hat` 将帽子更改为另一角色（也称为子配置文件）。`pam_apparmor` PAM 模块允许应用程序基于组名、用户名或默认配置文件将已通过身份验证的用户限制到子配置文件。要实现此目的，需将 `pam_apparmor` 注册为 PAM 会话模块。默认不会安装 `pam_apparmor` 软件包，您可以使用 YaST 或 **zypper** 来安装。安装该软件包后，可以在 `/usr/share/doc/packages/pam_apparmor/README` 中找到有关如何设置和配置 `pam_apparmor` 的细节。有关 PAM 的细节，请参见第 3 章“通过 PAM 进行身份验证”。

36 管理已构建配置文件的应用程序

创建配置文件并使应用程序免疫后，只要您保持对 AppArmor® 配置文件的维护（包括分析日志文件、优化配置文件、备份配置文件集并使其保持最新），SUSE® Linux Enterprise Server 的效率就会提升并得到更好的保护。您可以通过设置事件电子邮件通知、运行 `aa-logprof` 工具以根据系统日志项更新配置文件，以及处理维护问题，在这些问题造成影响之前对其妥善处理。

36.1 对安全事件拒绝做出反应

收到安全事件拒绝时，请检查访问违规情况并确定此事件表示的是威胁还是正常应用程序行为的一部分。要进行判断，您必须具备特定于该应用程序的知识。如果拒绝的操作是正常应用程序行为的一部分，请在命令行中运行 `aa-logprof`。

如果拒绝的操作不是正常应用程序行为的一部分，则应将此访问视为一次可能的入侵企图（已被阻止），并将此通知发送给贵组织中负责安全性的人员。

36.2 维护安全配置文件

在生产环境中，您应计划如何维护所有部署的应用程序的配置文件。安全策略是部署过程中不可分割的一部分。您应计划采取措施来备份和恢复安全策略文件，计划软件的更改，还应允许根据您的环境要求对安全策略进行任何必要的修改。

36.2.1 备份安全配置文件

通过备份配置文件，当磁盘崩溃后，您可能就不必重新构建所有程序的配置文件。另外，如果配置文件被更改，您可以使用备份的文件轻松地恢复之前设置。

通过将配置文件复制到指定目录而备份配置文件。

1. 首先，您应将这些文件存档到一个文件中。为此，请打开终端窗口并以 `root` 身份输入以下命令：

```
tux > sudo tar zclpf profiles.tgz /etc/apparmor.d
```

要确保系统定期备份您的安全策略文件，最简单的方法是在备份系统存档的目录列表中包含 `/etc/apparmor.d` 目录。

2. 您也可以使用 `scp` 或 Nautilus 等文件管理器将文件储存到某种储存媒体、网络或另一台计算机中。

36.2.2 更改安全配置文件

如果您确定系统对其应用程序要求更高或更低的安全性，维护安全配置文件会包含对它们的更改。要在 AppArmor 中更改配置文件，请参见第 32.2 节“编辑配置文件”。

36.2.3 将新软件引入您的环境

将新的应用程序版本或补丁添加到系统时，请务必更新配置文件以满足您的需要。根据贵公司的软件部署策略，您有若干选择。您可以将您的补丁和升级程序部署到测试或生产环境中。以下介绍每种方式的操作方法。

如果您打算在测试环境中部署补丁或进行升级，更新配置文件的最佳方法是在终端中以 `root` 身份运行 `aa-logprof`。有关详细说明，请参见第 33.7.3.9 节“aa-logprof — 扫描系统日志”。

如果您打算直接在生产环境中部署补丁或进行升级，更新配置文件的最佳方法是经常监视系统，以确定是否需要将任何新的拒绝添加到配置文件，并根据需要使用 `aa-logprof` 进行更新。有关详细说明，请参见第 33.7.3.9 节“aa-logprof — 扫描系统日志”。

37 支持

本章简要介绍维护方面的任务。您将了解到如何更新 AppArmor®，还会获得一个可用手册页的列表，这些手册页提供有关如何使用 AppArmor 所提供的命令行工具的基本帮助。查错一节提供了使用 AppArmor 时会遇到的一些常见问题及其解决方法。请遵循本章中的指导报告 AppArmor 的缺陷或提出增强请求。

37.1 联机更新 AppArmor

我们采用与 SUSE Linux Enterprise Server 的任何其他更新相同的方式提供 AppArmor 软件包更新。您可以像检索和应用 SUSE Linux Enterprise Server 随附的任何其他软件包一样来检索和应用这些更新。

37.2 使用手册页

您可以使用手册页。在终端中输入 **man apparmor** 打开 AppArmor 手册页。手册页分布在编号为 1 到 8 的部分中。每个部分针对一种类别的文档：

表 37.1：手册页：部分和类别

部分	类别
1	用户命令
2	系统调用
3	库函数
4	设备驱动程序信息
5	配置文件格式
6	游戏
7	高级概念

部分	类别
8	管理员命令

区号用于对各个手册页进行区分。例如，[exit\(2\)](#) 描述退出系统调用，而 [exit\(3\)](#) 描述退出 C 库函数。

AppArmor 手册页包括：

- [aa-audit\(8\)](#)
- [aa-autodep\(8\)](#)
- [aa-complain\(8\)](#)
- [aa-decode\(8\)](#)
- [aa-disable\(8\)](#)
- [aa-easyprof\(8\)](#)
- [aa-enforce\(8\)](#)
- [aa-enxec\(8\)](#)
- [aa-genprof\(8\)](#)
- [aa-logprof\(8\)](#)
- [aa-notify\(8\)](#)
- [aa-status\(8\)](#)
- [aa-unconfined\(8\)](#)
- [aa_change_hat\(8\)](#)
- [logprof.conf\(5\)](#)
- [apparmor.d\(5\)](#)

- [apparmor.vim\(5\)](#)
- [apparmor\(7\)](#)
- [apparmor_parser\(8\)](#)
- [apparmor_status\(8\)](#)

37.3 更多信息

<http://wiki.apparmor.net> 上提供了有关 AppArmor 产品的详细信息。所安装系统的 [/usr/share/doc/manual](#) 中提供了 AppArmor 的产品文档。

我们提供了 AppArmor 邮件列表，用户可向其发送邮件或加入其中，以与开发人员沟通。有关详细信息，请参见<https://lists.ubuntu.com/mailman/listinfo/apparmor>。

37.4 查错

本节列出了使用 AppArmor 时可能出现的最常见问题和错误消息。

37.4.1 如何应对应用程序行为异常？

如果您注意到应用程序行为异常或其他类型的应用程序问题，应当先检查日志文件中的拒绝消息，以确定 AppArmor 对应用程序的限制是否过于严格。如果检测到有拒绝消息指出 AppArmor 对应用程序或服务的限制过于严格，请更新您的配置文件，以正确处理应用程序的用例。请使用 **aa-logprof** 执行此操作（第 33.7.3.9 节“aa-logprof — 扫描系统日志”）。

如果您决定在不受 AppArmor 保护的情况下运行应用程序或服务，请从 [/etc/apparmor.d](#) 中去除应用程序的配置文件，或将其移到其他位置。

37.4.2 我的配置文件似乎不再正常工作...

如果您一直在使用旧版的 AppArmor，后来更新了系统（但保留了旧的配置文件集），您可能会发现，在更新之前运行很正常的某些应用程序现在却出现奇怪的行为，或者无法运行。

此版本的 AppArmor 为配置文件语法和 AppArmor 工具引入了一组新功能，这可能会给旧版 AppArmor 配置文件造成问题。这些功能包括：

- 文件锁定
- 网络访问控制
- SYS_PTRACE 功能
- 目录路径访问

当前版本的 AppArmor 可调解文件锁定，并为此引入了新的权限模式 (k)。如果请求文件锁定权限的应用程序受到旧版配置文件的限制，而这些配置文件并未显式包含用于锁定文件的权限，则这些应用程序可能会行为异常或完全失败。如果您怀疑存在这种情况，请在 /var/log/audit/audit.log 下的日志文件中检查如下所示的项：

```
type=AVC msg=audit(1389862802.727:13939): apparmor="DENIED" \
operation="file_lock" parent=2692 profile="/usr/bin/opera" \
name="/home/tux/.qt/.qtrc.lock" pid=28730 comm="httpd2-prefork" \
requested_mask="::k" denied_mask="::k" fsuid=30 ouid=0
```

按如下所述使用 **aa-logprof** 命令更新配置文件。

第 30.5 节 “网络访问控制” 中所述的基于网络系列和类型规范的新网络访问控制语法可能导致应用程序行为异常甚至无法运行。如果您发现网络相关的应用程序出现奇怪的行为，请在 /var/log/audit/audit.log 下的日志文件中检查如下所示的项：

```
type=AVC msg=audit(1389864332.233:13947): apparmor="DENIED" \
operation="socket_create" family="inet" parent=29985 profile="/bin/ping" \
sock_type="raw" pid=30251 comm="ping"
```

此日志项表示我们的示例应用程序（在本例中为 /bin/ping）无法获取用于打开网络连接的 AppArmor 权限。为确保应用程序能够进行网络访问，需要显式指明此权限。要将配置文件更新为使用新语法，请按如下所述使用 **aa-logprof** 命令。

如果进程尝试访问 /proc/PID/fd/* 中的文件，当前内核需要 SYS_PTRACE 功能。新配置文件需要该文件项和该功能项，而旧配置文件只需要该文件项。例如，旧语法中的以下语句

```
/proc/*/fd/** rw,
```

将转换为新语法中的以下规则：

```
capability SYS_PTRACE,  
/proc/*/fd/** rw,
```

要将配置文件更新为使用新语法，请按如下所述使用 YaST 更新配置文件向导或 **aa-logprof** 命令。

在此版本的 AppArmor 中，对配置文件规则语法进行了几项更改，以便更好地将目录访问与文件访问区分开来。因此，在旧版本中与文件路径和目录路径匹配的某些规则现在可能只与文件路径匹配。这可能导致 AppArmor 无法访问某个关键目录，从而触发应用程序的行为异常和各种日志消息。以下示例重点指出了路径语法最重要的更改。

使用旧语法时，下面的规则将允许访问 /proc/net 中的文件和目录。它只允许目录访问操作读取该目录中的项，而不授予对该目录下的文件或目录的访问权限。例如，星号 (*) 将会匹配 /proc/net/dir/foo，但由于 foo 是 dir 下的一个文件或目录，因此无法访问它。

```
/proc/net/* r,
```

要使用新语法获得相同的行为，需要使用两条规则，而不是一条。第一条规则允许访问 /proc/net 下的文件，第二条规则允许访问 /proc/net 下的目录。目录访问只可用于列出内容，而不能实际访问该目录下的文件或目录。

```
/proc/net/* r,  
/proc/net/*/ r,
```

下面的规则在新旧语法中的工作方式类似，允许访问 /proc/net 下的文件和目录（但不允许访问 /proc/net/ 的目录列表本身）：

```
/proc/net/** r,
```

要在新语法中使用上述表达式区分文件访问和目录访问，需使用以下两条规则：第一条规则仅允许递归访问 /proc/net 下的目录，而第二条规则仅显式允许进行递归文件访问。

```
/proc/net/**/ r,  
/proc/net/**[^/] r,
```

下面的规则在新旧语法中的工作方式类似，允许访问 `/proc/net` 下以 `foo` 开头的文件和目录：

```
/proc/net/foo** r,
```

要在新语法中区分文件访问和目录访问并使用 `**` 通配模式，需使用以下两条规则：第一条规则在旧语法中会同时匹配文件和目录，而在新语法中只会匹配文件，因为没有尾随斜线。第二条规则在旧语法中既不匹配文件也不匹配目录，而在新语法中只会匹配目录：

```
/proc/net/**foo r,  
/proc/net/**foo/ r,
```

以下规则说明了 `?` 通配模式的用法变化。在旧语法中，第一条规则会同时匹配文件和目录（四个字符，最后一个字符可以是除斜线以外的任何字符）。在新语法中，它只匹配文件（没有尾随斜线）。第二条规则在旧配置文件语法中不匹配任何内容，而在新语法中只会匹配目录。最后一条规则显式匹配名为 `bar`（`/proc/net/foo?` 下）的文件。使用旧语法时，此规则将应用于文件和目录：

```
/proc/net/foo? r,  
/proc/net/foo?/ r,  
/proc/net/foo?/bar r,
```

要查找并解决语法更改相关的问题，请在更新后花些时间检查您要保留的配置文件，并按如下所述继续处理您保留了其配置文件的每个应用程序：

1. 将应用程序的配置文件置于控诉模式：

```
tux > sudo aa-complain /path/to/application
```

系统会对违当前配置文件的所有操作生成日志项，但不会强制执行该配置文件，并且应用程序的行为不受限制。

2. 运行涵盖您需要其能够执行的所有任务的应用程序。
3. 根据运行应用程序时生成的日志项更新配置文件：

```
tux > sudo aa-logprof /path/to/application
```

4. 将生成的配置文件重新置于强制模式：

```
tux > sudo aa-enforce /path/to/application
```

37.4.3 使用 Apache 解决问题

安装其他 Apache 模块（例如 `apache2-mod_apparmor`）或者对 Apache 做出配置更改后，再次构建 Apache 的配置文件，以确定是否需要向配置文件添加额外的规则。如果您不再次构建 Apache 的配置文件，Apache 可能无法正常启动，或者无法为网页提供服务。

37.4.4 如何从使用的配置文件列表中排除特定的配置文件？

运行 `aa-disable PROGRAMNAME` 可以禁用 `PROGRAMNAME` 的配置文件。此命令会创建指向 `/etc/apparmor.d/disable/` 中的配置文件的符号链接。要重新激活配置文件，请删除该链接，并运行 `systemctl reload apparmor`。

37.4.5 我是否可以管理未安装在我系统上的应用程序的配置文件？

要使用 AppArmor 管理配置文件，您需要有权访问运行应用程序的系统的日志。因此，只要您有权访问运行应用程序的计算机，就无需在配置文件构建主机上运行该应用程序。您可以在一个系统上运行应用程序，将日志（`/var/log/audit.log`，如果未安装 `audit`，则为 `journalctl | grep -i apparmor > path_to_logfile`）传输到配置文件构建主机，然后运行 `aa-logprof -f PATH_TO_LOGFILE`。

37.4.6 如何找出和修复 AppArmor 语法错误

手动编辑 AppArmor 配置文件可能会产生语法错误。如果您的配置文件中存在语法错误，尝试启动或重新启动 AppArmor 时，会显示类似下面的错误结果。此示例显示了整个分析程序错误的语法。

```
root # systemctl start apparmor.service
Loading AppArmor profiles AppArmor parser error in /etc/apparmor.d/
usr.sbin.squid \
  at line 410: syntax error, unexpected TOK_ID, expecting TOK_MODE
Profile /etc/apparmor.d/usr.sbin.squid failed to load
```

使用 AppArmor YaST 工具时，会有一条图形错误消息指出哪个配置文件包含错误，并要求您予以修复。



要修复语法错误，请以 `root` 身份在终端窗口中登录，打开配置文件，然后更正语法。使用 `systemctl reload apparmor` 重新装载配置文件集。



提示：vi 中的 AppArmor 语法高亮显示

SUSE Linux Enterprise Server 上的编辑器 `vi` 支持 AppArmor 配置文件的语法高亮显示功能。包含语法错误的行的背景将会显示为红色。

37.5 报告 AppArmor 的 Bug

AppArmor 的开发人员热切希望能够提供最高品质的产品。您的反馈和 bug 报告有助于我们保持较高的品质。当您在 AppArmor 中遇到 bug 时，请提交此产品的 bug 报告：

1. 在网页浏览器中转到 <http://bugzilla.suse.com/> 并单击登录。
2. 输入您的 SUSE 帐户数据并单击登录。如果您没有 SUSE 帐户，请单击创建帐户并提供所需的数据。
3. 如果您的问题已有报告，请查看此错误报告，必要时在报告中添加其它信息。
4. 如果您的问题尚无报告，请在顶部导航栏中选择新建进入输入错误页面。
5. 选择要提交错误的产品。对于您而言，这应该是您的产品的发行版。单击“提交”。
6. 选择产品版本、组件（在本例中为 AppArmor）、硬件平台和严重性。
7. 输入一个用于描述问题的简短标题，然后在下面添加更详尽的说明，包括日志文件。您可以在 bug 报告中创建附件，以提供屏幕截图、日志文件或测试案例。
8. 输入所有详细信息后，单击提交以将报告发送到开发人员。

38 AppArmor 术语表

抽象

请参见下面的**配置文件基础类**。

Apache

Apache 是一个基于 Unix 的免费 Web 服务器。目前它是因特网上使用最为广泛的 Web 服务器。Apache 网站 <http://www.apache.org> 上提供了有关 Apache 的详细信息。

应用程序防火墙

AppArmor 会限制应用程序以及允许它们执行的操作。它使用特权限制来防止攻击者在受保护的服务器上使用恶意程序，甚至可防止以非预期的方式使用可信赖的应用程序。

攻击签名

系统或网络活动中警示可能存在病毒或黑客攻击的模式。入侵检测系统可使用攻击签名来区分合法的活动和可能存在恶意的活动。

AppArmor 不依赖于攻击特征，可提供“前瞻性”而非“反应性”的攻击防御。这种方式比较优越，因为该方式可杜绝必须为 AppArmor 定义攻击特征期间存在易受攻击的风险，而使用攻击特征来提供保护的产品就存在此风险。

GUI

图形用户界面.代表一种软件前端，在计算机用户和应用程序之间提供一个友好且易于使用的界面。其元素包括窗口、图标、按钮、光标和滚动条。

通配

文件名替代。您可以不指定明确的文件名路径，而是使用帮助字符 `*`（替代任意数目的字符，但 `/` 或 `?` 这样的特殊字符除外）和 `?`（仅替代一个字符）来一次性找到多个文件/目录。`**` 是特殊替代方式，它会匹配当前目录下的任意文件或目录。

HIP

主机入侵防御。与操作系统内核相协作以阻止异常的应用程序行为，异常的行为被视为未知攻击。在网络级阻止主机上的恶意包，使它们不能“伤害”它们所针对的应用程序。

强制访问控制

限制对象访问权限的方式，基于分配给用户、文件和其它对象的固定安全属性。控制是强制性的，也就是说用户和程序不能修改它们。

配置文件

AppArmor 配置文件全面定义单个应用程序可以访问哪些系统资源以及拥有哪些特权。

配置文件基础类

常用的应用程序活动所需的配置文件组建模块，如 DNS 查询和用户身份验证。

RPM

RPM 包管理器任何人都可使用的开放打包系统。它可在 Red Hat Linux、SUSE Linux Enterprise Server 及其他 Linux 和 Unix 系统上运行。他可以安装、卸装、验证、查询和更新计算机软件包。有关更多信息，请参见<http://www.rpm.org/>。

SSH

安全 Shell。一项服务，允许您从远程计算机访问服务器并通过安全连接发出文本命令。

优化的访问控制

AppArmor 指定各个程序可以读取、写入和执行哪些文件，从而为网络服务提供优化的访问控制。这确保了每个程序只会执行意料之中的操作，而不会执行其它操作。

URI

通用资源标识符。指向 Web 上的对象的所有类型的名称和地址的通称。URL 是一种 URI。

URL

Uniform Resource Locator，统一资源定位器。Web 上的文档和其他资源的全球地址。

该地址的第一部分表示要使用的协议，第二部分指定资源所在的 IP 地址或域名。

例如，当您访问 <http://www.suse.com> 时，使用的就是 HTTP 协议，如 URL 的开头部分所示。

漏洞

系统或网络不能防御攻击的部分。计算机系统的特性使个人能够进行不正确的操作，或使未经授权的用户能够获取系统的控制权。设计、管理或实施上的不足，或硬件、固件或软件上的缺陷。如果受到攻击，漏洞可能会导致无法接受的影响，包括对信息的未经授权访问或对关键处理的破坏。

V SELinux 对比

39 配置 SELinux 338

39 配置 SELinux

在本章中，您将了解如何在 SUSE Linux Enterprise Server 上设置和管理 SELinux。其中包括以下主题：

- 为何要使用 SELinux?
- 了解 SELinux
- 设置 SELinux
- 管理 SELinux

39.1 为何要使用 SELinux?

SELinux 是作为附加的 Linux 安全解决方案开发的，它运用了 Linux 内核中的安全框架。其目的是除了默认的现有权限“读取”、“写入”和“执行”所能提供的安全策略，以及指派对 Linux 上提供的不同功能的权限之外，能够支持更精细的安全策略。为实现此目的，SELinux 会捕获到达内核的所有系统调用，并默认拒绝这些调用。这意味着，在启用了 SELinux 但未配置任何其他设置的系统上，一切都无法正常运行。要使系统能够执行任何操作，管理员需要编写规则并将其放入策略中。

下面的示例解释了为何需要 SELinux 这样的解决方案（或其类似产品 AppArmor）：

“有一天早上，我发现我的服务器被黑客入侵了。这台服务器运行的是安装了全套补丁的 SUSE Linux Enterprise Server。服务器上配置了防火墙，并且不提供任何不需要的服务。进一步的分析表明，黑客是通过一个有漏洞的 PHP 脚本入侵的，这个脚本是这台服务器上运行的某个 Apache 虚拟主机的一部分。入侵者设法通过 Apache Web 服务器所用的 `wwwrun` 帐户获得了外壳访问权限。入侵者以这个 `wwwrun` 用户的身份在 `/var/tmp` 和 `/tmp` 目录中创建了多个脚本，这些脚本组成了针对多台服务器发起分布式拒绝服务攻击的僵尸网络。”

对于这次攻击，有意思的地方在于，它是在一台实际上并无任何错误的服务器上发生的。所有权限的设置都是正确的，但入侵者还是设法进入了系统。此示例清楚地证明，在某些情况下，还需要进一步提高安全性 — 超越 SELinux 所能提供的安全性。可以使用 AppArmor 作为替代，其复杂性相对较低但功能较不完整。

AppArmor 可以限制特定进程读取/写入和执行文件的能力（及其他能力）。其主要理念是进程内部发生的操作都不能脱离控制。

而 SELinux 则是使用附加到对象（例如文件、二进制文件、网络套接字）的标签，并通过这些标签确定特权边界，从而建立一个可以跨越多个进程甚至整个系统的限制级别。

SELinux 由美国国家安全局 (NSA) 开发，Red Hat 从一开始就大力参与了它的开发。SELinux 的第一个版本在 Red Hat Enterprise Linux 4™ 时代（大约为 2006 年）推出。最初它仅提供对最关键的的服务的支持，多年以来，它已发展成为一套能够提供许多规则的系统，这些规则可收集到策略中，以便为各种各样的服务提供保护。

SELinux 是根据通用准则和 FIPS 140 等一些认证标准开发的。由于某些客户特意要求解决方案符合这些标准，SELinux 很快变得相对流行。

Novell 于 2005 年收购的公司 Immunix 开发了用于替代 SELinux 的 AppArmor。AppArmor 立足的安全原理与 SELinux 相同，但采用了完全不同的方法，它可以通过简单易用的向导驱动式过程，将服务限制为仅可执行需要它们执行的操作。然而，AppArmor 从未达到过与 SELinux 相同的高度，尽管有些不错的论据指出，使用 AppArmor 保护服务器的效果比使用 SELinux 的效果更好。

由于许多组织请求在他们所用的 Linux 发行套件中包含 SELinux，SUSE 在 SUSE Linux Enterprise Server 中提供了 SELinux 框架支持。这并不意味着，在不久的将来，默认的 SUSE Linux Enterprise Server 安装将从 AppArmor 改为 SELinux。

39.1.1 支持状态

SUSE Linux Enterprise Server 支持 SELinux 框架。也就是说，SLES 提供了您能够在服务器上使用 SELinux 所需的所有二进制文件和库。

SUSE Linux Enterprise Server 中的 SELinux 支持处于一个相当早期的阶段，意味着它可能会出现意外的行为。为了尽可能减少这种风险，最好仅使用 SUSE Linux Enterprise Server 上默认提供的二进制文件。

39.1.2 了解 SELinux 组件

在开始配置 SELinux 之前，您应该对 SELinux 的组织方式有个大致的了解。重要组件包括以下三个：

- Linux 内核中的安全框架
- SELinux 库和二进制文件
- SELinux 策略

SUSE Linux Enterprise Server 的默认内核支持 SELinux 以及管理 SELinux 所需的工具。关于 SELinux，管理员最重要的工作是管理策略。



警告：不包含默认策略

SUSE Linux Enterprise Server 中不提供任何默认策略或参考策略。SELinux 在没有策略的情况下无法运行，因此您必须构建并安装一个策略。SELinux 参考策略项目 (<https://github.com/SELinuxProject/refpolicy/wiki>) 提供了有关如何创建您自己的策略的示例和详细信息，应该会对您有所帮助。本章也提供了有关如何管理 SELinux 策略的引导。

在 SELinux 策略中，安全性标签将应用到 Linux 服务器上的不同对象。这些对象通常是用户、端口、进程和文件。使用这些安全性标签可以创建规则，用于定义在服务器上允许和不允许执行哪些操作。请记住，SELinux 默认会拒绝任何操作，而创建适当的规则可以允许进行确实有必要的访问。因此，您要在系统上使用的所有程序应该都有相应的规则。或者，您应将系统的某些组成部分配置为以非受限模式运行，这意味着特定的端口、程序、用户、文件和目录将不受 SELinux 的保护。如果您只想使用 SELinux 来保护某些必不可少的服务，而不特别担心其他服务，此模式将十分有用。要想使系统真正安全，应该避免这种做法。

要确保为系统提供适当的保护，您需要有 SELinux 策略。这必须是一个定制的策略，其中的所有文件都带有标签，并且所有服务和用户也都带有安全性标签，以指明哪些文件和目录可由哪个用户访问以及在服务器上处理。制定此类策略涉及到巨大的工作量。

SELinux 的复杂性也是它受到诟病的主要原因之一。由于典型的 Linux 系统太过复杂，用户很容易忽视某些问题，留下漏洞被入侵者滥用并进入其系统。即使已将 SELinux 设置为完全按照预期方式运行，管理员也仍然很难做到兼顾 SELinux 的方方面面。在复杂性方面，AppArmor 采用完全不同的方法，通过自动化过程使管理员能够设置 AppArmor 保护并确切了解发生的情况。

请注意，免费提供的 SELinux 策略在您的服务器上也许可以正常运行，但不太可能会提供与自定义策略同等程度的保护。SUSE 不支持第三方策略。

39.2 策略

策略是 SELinux 中的关键组件。请注意，SUSE Linux Enterprise Server 不包含默认策略或参考策略，您必须先构建并安装一个根据您的需求自定义的策略，然后再继续操作。（请参见 [警告：不包含默认策略](#)。）

您的 SELinux 策略需定义用于指定哪些对象可以访问系统上的哪些文件、目录、端口和进程的规则。为此，需为所有这些对象定义一个安全环境。在已应用策略来标记文件系统的 SELinux 系统上，您可对任何目录使用 `ls -Z` 命令来查找该目录中各文件的安全环境。[例 39.1：“使用 `ls -Z` 查找安全环境设置”](#) 显示了采用 SELinux 标记文件系统的 SUSE Linux Enterprise Server 系统上 / 目录中各目录的安全环境设置。

例 39.1：使用 `ls -Z` 查找安全环境设置

```
ls -Z
system_u:object_r:bin_t bin
system_u:object_r:boot_t boot
system_u:object_r:device_t dev
system_u:object_r:etc_t etc
system_u:object_r:home_root_t home
system_u:object_r:lib_t lib
system_u:object_r:lib_t lib64
system_u:object_r:lost_found_t lost+found
system_u:object_r:mnt_t media
system_u:object_r:mnt_t mnt
system_u:object_r:usr_t opt
system_u:object_r:proc_t proc
system_u:object_r:default_t root
system_u:object_r:bin_t sbin
system_u:object_r:security_t selinux
system_u:object_r:var_t srv
system_u:object_r:sysfs_t sys
system_u:object_r:tmp_t tmp
system_u:object_r:usr_t usr
system_u:object_r:var_t var
```

安全环境中最重要的一行是环境类型。它是安全环境中以 `_t` 结尾的部分。它告知 SELinux 允许对象进行哪种访问。策略中指定了规则，用于定义哪种类型的用户或角色有权访问哪种类型的环境。例如，使用如下所示的规则可以进行这样的定义：

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

此示例规则指出，允许环境类型为 `user_t` 的用户（此用户称为源对象）使用 `read`、`execute` 和 `gettattr` 权限访问环境类型为 `bin_t` 的“file”类别对象（目标）。

您要使用的标准策略包含海量的规则。为使其更易于管理，通常会将策略分割成多个模块。这样，管理员便可为不同的系统组件打开或关闭保护。

为系统编译策略时，您可以选择使用模块化策略或单体式策略，如果选择后者，将使用一个巨型策略来保护系统上的一切组件。强烈建议使用模块化策略，而不要使用单体式策略。模块化策略要容易管理得多。

39.3 安装 SELinux 软件包并修改 GRUB 2

确保安装所有 SELinux 组件的最简单的方法是使用 YaST。下述过程说明了要在安装的 SUSE Linux Enterprise Server 上执行哪些操作：

1. 启动 YaST。
2. 选择软件 > 软件管理。
3. 选择视图 > 搜索，确保搜索范围名称、关键字和摘要处于选中状态。现在输入关键字 `selinux` 并单击搜索。您现在即会看到软件包列表。选择安装所有软件包，单击接受，在安装完成后单击继续。
4. 返回选择视图 > 搜索，搜索 `packagehub`。选择安装 `PackageHub-release`，然后单击接受进行安装。安装该软件包后，单击完成关闭软件管理模块。
5. 重新打开软件管理模块。转到选择视图 > 模式，选择安装整个 C/C++ 开发类别，然后进行安装。（由 PackageHub 提供 C/C++ 开发模式。）

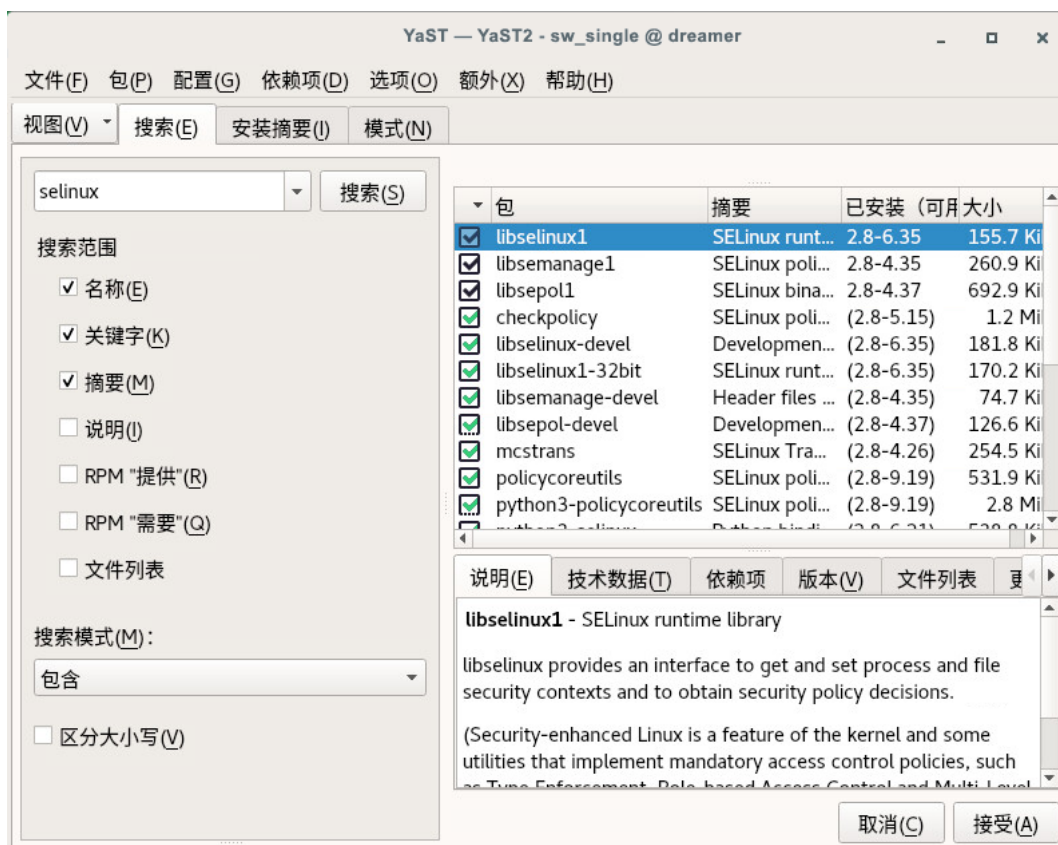


图 39.1：在 YAST 中选择所有 SELINUX 软件包

安装 SELinux 软件包后，需要修改 GRUB 2 引导加载程序。请在 YaST 中选择系统 > 引导加载程序 > 内核参数来完成此操作。现在，将以下参数添加到可选内核命令行参数：

```
security=selinux selinux=1 enforcing=0
```

这些选项用于以下目的：

security=selinux

此选项告知内核要使用 SELinux 而非 AppArmor

selinux=1

此选项用于启用 SELinux

`enforcing=0`

此选项将 SELinux 置于宽松模式。在此模式下，SELinux 可完全正常运行，但不会强制执行策略中的任何安全性设置。请使用此模式来配置您的系统。要启用 SELinux 保护，请在系统完全正常运行后，将该选项更改为 `enforcing=1`，并在 `/etc/selinux/config` 中添加 `SELINUX=enforcing`。

安装 SELinux 软件包并启用 SELinux GRUB 2 引导参数后，重引导您的服务器以激活配置。

39.4 SELinux 策略

策略是 SELinux 不可或缺的组件。SUSE Linux Enterprise Server 15 SP3 不包含默认策略或参考策略，您必须针对您的安装构建一个自定义策略。要进行测试和学习，请参见 <https://github.com/SELinuxProject/refpolicy/wiki> 上的 SELinux 参考策略项目。您必须有一个策略，否则 SELinux 不会正常运行。

安装策略后，便可以开始标记文件系统了。运行

```
tux > sudo restorecon -Rp /
```

以启动 `/sbin/setfiles` 命令来标记系统上的所有文件。系统将使用 `/etc/selinux/minimum/contexts/files/file_contexts` 输入文件。`file_contexts` 文件需尽可能与您的实际文件系统相匹配，否则可能导致系统完全无法引导。如果发生这种情况，请使用 `semanage fcontext` 命令修改 `file_contexts` 中的记录，使之与服务器使用的文件系统的实际结构相匹配。例如，

```
tux > sudo semanage fcontext -a -t samba_share_t /etc/example_file
```

将文件类型从默认的 `etc_t` 更改为 `samba_share_t`，并将以下记录添加到相关的 `file_contexts.local` 文件中：

```
/etc/example_file    unconfined_u:object_r:samba_share_t:s0
```

然后运行

```
tux > sudo restorecon -v /etc/example_file
```

以使类型更改生效。

在执行此操作之前，请务必阅读本章的其余内容，以便全面了解环境类型如何应用到文件和目录。在开始之前，请不要忘记备份 `file_contexts` 文件。



注意：用户 nobody

使用 **semanage** 时，您可能会收到一条消息，指出 `nobody` 的主目录有问题。在这种情况下，请将用户 `nobody` 的登录外壳更改为 `/sbin/nologin`。这样 `nobody` 的设置便与当前策略设置相匹配。

再次重引导后，SELinux 应当会正常运行。要进行校验，请使用 **sestatus -v** 命令。此命令应该会返回类似于例 39.2：“校验 SELinux 是否正常运行”中的输出。

例 39.2：校验 SELINUX 是否正常运行

```
tux > sudo sestatus -v
SELinux status:                enabled
SELinuxfs mount:              /selinux
Current mode:                  permissive
Mode from config file:        permissive
Policy version:                26
Policy from config file:      minimum

Process contexts:
Current context:               root:staff_r:staff_t
Init context:                  system_u:system_r:init_t
/sbin/mingetty                 system_u:system_r:sysadm_t
/usr/sbin/sshd                 system_u:system_r:sshd_t

File contexts:
Controlling term:             root:object_r:user_devpts_t
/etc/passwd                    system_u:object_r:etc_t
/etc/shadow                    system_u:object_r:shadow_t
/bin/bash                      system_u:object_r:shell_exec_t
/bin/login                     system_u:object_r:login_exec_t
/bin/sh                        system_u:object_r:bin_t ->
system_u:object_r:shell_exec_t
/sbin/agetty                   system_u:object_r:getty_exec_t
/sbin/init                     system_u:object_r:init_exec_t
```

```

/sbin/mingetty          system_u:object_r:getty_exec_t
/usr/sbin/sshd          system_u:object_r:sshd_exec_t
/lib/libc.so.6          system_u:object_r:lib_t ->
                        system_u:object_r:lib_t
/lib/ld-linux.so.2      system_u:object_r:lib_t ->
                        system_u:object_r:ld_so_t

```

39.5 配置 SELinux

现在，您有了一个完全正常运行的 SELinux 系统，接下来我们来进一步配置这个系统。在当前状态下，SELinux 可正常运行，但并未处于强制模式。这意味着，它不会限制您执行任何操作，但会记录如果处于强制模式下它将会执行的所有操作。此模式很有用，因为您可以根据日志文件来确定它将阻止您执行什么操作。作为第一项测试，请将 SELinux 置于强制模式，并在采取以下做法之后确定您是否仍可使用服务器：检查在 `/etc/selinux/config` 中设置了 `SELINUX=enforcing` 的同时，是否在 GRUB 2 配置文件中设置了 `enforcing=1` 选项。重引导服务器，然后检查它是否仍可按预期方式正常启动。如果是，请保留当前设置，然后开始修改服务器，并使一切按预期方式工作。不过，您有可能甚至无法正常引导服务器。在这种情况下，请切换回 SELinux 的非强制模式，然后开始调整服务器。

在开始调整服务器之前，请校验 SELinux 安装。您已使用 `sestatus -v` 命令查看当前模式、进程和文件环境。接下来，运行

```
tux > sudo semanage boolean -l
```

以列出所有可用的布尔开关，同时校验您是否可以访问该策略。例 39.3 “获取布尔值列表并校验策略访问权限”显示了此命令的部分输出。

例 39.3：获取布尔值列表并校验策略访问权限

```

tux > sudo semanage boolean -l
SELinux boolean          Description
ftp_home_dir             -> off   ftp_home_dir
mozilla_read_content     -> off   mozilla_read_content
spamassassin_can_network -> off   spamassassin_can_network
httpd_can_network_relay  -> off   httpd_can_network_relay
openvpn_enable_homedirs  -> off   openvpn_enable_homedirs
gpg_agent_env_file       -> off   gpg_agent_env_file

```

```
allow_httpd_awstats_script_anon_write -> off
allow_httpd_awstats_script_anon_write
httpd_can_network_connect_db -> off httpd_can_network_connect_db
allow_ftp_full_access -> off allow_ftp_full_access
samba_domain_controller -> off samba_domain_controller
httpd_enable_cgi -> off httpd_enable_cgi
virt_use_nfs -> off virt_use_nfs
```

在此阶段，可输出有用信息的另一个命令是

```
tux > sudo semanage fcontext -l
```

此命令会显示策略提供的默认文件环境设置（有关此命令的部分输出，请参见例 39.4: “获取文件环境信息”）。

例 39.4：获取文件环境信息

```
tux > sudo semanage fcontext -l
/var/run/usb(/.*)? all files
system_u:object_r:hotplug_var_run_t
/var/run/utmp regular file
system_u:object_r:initrc_var_run_t
/var/run/vbe.* regular file
system_u:object_r:hald_var_run_t
/var/run/vmnet.* socket
system_u:object_r:vmware_var_run_t
/var/run/vmware.* all files
system_u:object_r:vmware_var_run_t
/var/run/watchdog\.*.pid regular file
system_u:object_r:watchdog_var_run_t
/var/run/winbindd(/.*)? all files
system_u:object_r:winbind_var_run_t
/var/run/wnn-unix(/.*) all files
system_u:object_r:canna_var_run_t
/var/run/wpa_supplicant(/.*)? all files
system_u:object_r:NetworkManager_var_run_t
/var/run/wpa_supplicant-global socket
system_u:object_r:NetworkManager_var_run_t
/var/run/xdmctl(/.*)? all files
system_u:object_r:xdm_var_run_t
```

```
/var/run/yiff-[0-9]+\..pid          regular file
system_u:object_r:soundd_var_run_t
```

39.6 管理 SELinux

基本 SELinux 配置现在可以正常运行，接下来可以配置 SELinux 来保护您的服务器。SELinux 中使用一组额外的规则来具体定义哪个进程或用户可以访问哪些文件、目录或端口。为此，SELinux 会将一个环境应用到每个文件、目录、进程和端口。此环境是一个安全性标签，定义应如何处理此文件、目录、进程或端口。这些环境标签由 SELinux 策略使用，该策略具体定义应该对环境标签执行什么操作。默认情况下，该策略会阻止所有非默认访问，这意味着，作为管理员，您需要在服务器上启用所有非默认的功能。

39.6.1 查看安全环境

如上文所述，您可以标记文件、目录和端口。每个标签中都使用了不同的环境。要能够执行日常管理工作，您最需要关注的就是类型环境。作为管理员，您基本上要处理的都是类型环境。许多命令允许您使用 `-Z` 选项列出当前环境设置。例 39.5：“根目录中各目录的默认环境”中显示了根目录中的目录采用了什么环境设置。

例 39.5：根目录中各目录的默认环境

```
tux > sudo ls -Z
dr-xr-xr-x. root root system_u:object_r:bin_t:s0      bin
dr-xr-xr-x. root root system_u:object_r:boot_t:s0     boot
drwxr-xr-x. root root system_u:object_r:cgroup_t:s0   cgroup
drwxr-xr-x+ root root unconfined_u:object_r:default_t:s0 data
drwxr-xr-x. root root system_u:object_r:device_t:s0   dev
drwxr-xr-x. root root system_u:object_r:etc_t:s0      etc
drwxr-xr-x. root root system_u:object_r:home_root_t:s0 home
dr-xr-xr-x. root root system_u:object_r:lib_t:s0      lib
dr-xr-xr-x. root root system_u:object_r:lib_t:s0      lib64
drwx----- . root root system_u:object_r:lost_found_t:s0 lost+found
drwxr-xr-x. root root system_u:object_r:mnt_t:s0      media
drwxr-xr-x. root root system_u:object_r:autofs_t:s0   misc
drwxr-xr-x. root root system_u:object_r:mnt_t:s0      mnt
```

```

drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 mnt2
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 mounts
drwxr-xr-x. root root system_u:object_r:autofs_t:s0 net
drwxr-xr-x. root root system_u:object_r:usr_t:s0 opt
dr-xr-xr-x. root root system_u:object_r:proc_t:s0 proc
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 repo
dr-xr-x--. root root system_u:object_r:admin_home_t:s0 root
dr-xr-xr-x. root root system_u:object_r:bin_t:s0 sbin
drwxr-xr-x. root root system_u:object_r:security_t:s0 selinux
drwxr-xr-x. root root system_u:object_r:var_t:s0 srv
-rw-r--r-. root root unconfined_u:object_r:swapfile_t:s0 swapfile
drwxr-xr-x. root root system_u:object_r:sysfs_t:s0 sys
drwxrwxrwt. root root system_u:object_r:tmp_t:s0 tmp
-rw-r--r-. root root unconfined_u:object_r:etc_runtime_t:s0 tmp2.tar
-rw-r--r-. root root unconfined_u:object_r:etc_runtime_t:s0 tmp.tar
drwxr-xr-x. root root system_u:object_r:usr_t:s0 usr
drwxr-xr-x. root root system_u:object_r:var_t:s0 var

```

在上面的列表中，您可以看到所有目录的完整环境。它由用户、角色和类型组成。s0 设置指示多级安全环境中的安全性级别。本文不讨论这些环境。在此类环境中，请确保设置了 s0。环境类型定义在目录中允许哪种活动。例如，我们来比较一下环境类型为 `admin_home_t` 的 `/root` 目录与环境类型为 `home_root_t` 的 `/home` 目录。SELinux 策略中会为这些环境类型定义不同种类的访问。

安全性标签不仅与文件相关联，还与端口和进程等其他项相关联。例如，在例 39.6: “使用 **ps Zaux** 显示进程的 SELinux 设置” 中，您可以看到服务器上进程的环境设置。

例 39.6：使用 **ps Zaux** 显示进程的 SELINUX 设置

```

tux > sudo ps Zaux

```

LABEL	USER	PID	%CPU	%MEM	VSZ	RSS	TTY
STAT START TIME COMMAND							
system_u:system_r:init_t	root	1	0.0	0.0	10640	808	?
Ss 05:31 0:00 init [5]							
system_u:system_r:kernel_t	root	2	0.0	0.0	0	0	?
05:31 0:00 [kthreadd]							
system_u:system_r:kernel_t	root	3	0.0	0.0	0	0	?
05:31 0:00 [ksoftirqd/0]							

```

system_u:system_r:kernel_t      root          6  0.0  0.0      0    0 ?      S
    05:31  0:00 [migration/0]
system_u:system_r:kernel_t      root          7  0.0  0.0      0    0 ?      S
    05:31  0:00 [watchdog/0]
system_u:system_r:sysadm_t      root        2344  0.0  0.0  27640   852 ?
    Ss   05:32  0:00 /usr/sbin/mcelog --daemon --config-file /etc/mcelog/
mcelog.conf
system_u:system_r:sshd_t        root        3245  0.0  0.0  69300  1492 ?
    Ss   05:32  0:00 /usr/sbin/sshd -o PidFile=/var/run/sshd.init.pid
system_u:system_r:cupsd_t       root        3265  0.0  0.0  68176  2852 ?
    Ss   05:32  0:00 /usr/sbin/cupsd
system_u:system_r:nsd_t         root        3267  0.0  0.0  772876 1380 ?
    Ssl  05:32  0:00 /usr/sbin/nsd
system_u:system_r:postfix_master_t root        3334  0.0  0.0  38320  2424 ?
    Ss   05:32  0:00 /usr/lib/postfix/master
system_u:system_r:postfix_qmgr_t postfix      3358  0.0  0.0  40216  2252 ?      S
    05:32  0:00 qmgr -l -t fifo -u
system_u:system_r:crond_t       root        3415  0.0  0.0  14900   800 ?
    Ss   05:32  0:00 /usr/sbin/cron
system_u:system_r:fsdaemon_t    root        3437  0.0  0.0  16468  1040 ?      S
    05:32  0:00 /usr/sbin/smartd
system_u:system_r:sysadm_t      root        3441  0.0  0.0  66916  2152 ?
    Ss   05:32  0:00 login -- root
system_u:system_r:sysadm_t      root        3442  0.0  0.0   4596   800 tty2
    Ss+  05:32  0:00 /sbin/mingetty tty2

```

39.6.2 选择 SELinux 模式

在 SELinux 中可以使用三种不同的模式：

强制：

这是默认模式。SELinux 根据策略中的规则保护服务器，并将其所有活动记录到审计日志中。

宽松：

此模式用于查错。如果设置为“宽松”，SELinux 将不保护您的服务器，但仍会将发生的所有情况记录到日志文件中。

已禁用：

在此模式下，SELinux 处于完全关闭状态，并且不会记录任何日志，但文件系统标签不会从文件系统中去除。

您已了解如何在引导时使用强制引导参数通过 GRUB 2 设置当前的 SELinux 模式。

39.6.3 修改 SELinux 环境类型

管理员的一项重要工作是设置文件的环境类型，以确保 SELinux 正常工作。

如果在特定的目录中创建了一个文件，该文件默认将继承父目录的环境类型。但是，如果将某个文件从一个位置移到另一个位置，它将保留在原来的位置中所用的环境类型。

要设置文件的环境类型，可以使用 **semanage fcontext** 命令。使用此命令可将新环境类型写入策略，但不会立即更改实际的环境类型！要应用策略中的环境类型，需要接着运行 **restorecon** 命令。

使用 **semanage fcontext** 时的难点在于找出您实际需要的环境。可使用

```
tux > sudo semanage fcontext -l
```

列出策略中的所有环境，但从该列表中找出所需的实际环境可能有点困难，因为该列表很长（请参见例 39.7：“查看默认文件环境”）。

例 39.7：查看默认文件环境

```
tux > sudo semanage fcontext -l | less
```

SELinux fcontext	type	Context
/	directory	
system_u:object_r:root_t:s0		
/*	all files	
system_u:object_r:default_t:s0		
/[^/]+	regular file	
system_u:object_r:etc_runtime_t:s0		
/\..autofsck	regular file	
system_u:object_r:etc_runtime_t:s0		
/\..autorelabel	regular file	
system_u:object_r:etc_runtime_t:s0		
/\..journal	all files	X:>>None>>

/\..suspended	regular file
system_u:object_r:etc_runtime_t:s0	
/a?quota\.(user group)	regular file
system_u:object_r:quota_db_t:s0	
/afs	directory
system_u:object_r:mnt_t:s0	
/bin	directory
system_u:object_r:bin_t:s0	
/bin/*.*	all files
system_u:object_r:bin_t:s0	

可以通过三种方式找出您的服务可用的环境设置：

- 安装服务并查看所用的默认环境设置：这是最简单的做法，也是建议的做法。
- 查阅特定服务的手册页。某些服务提供了以 `_selinux` 结尾的手册页，其中包含查找正确环境设置所需的所有信息。

找到正确的环境设置后，使用 **semanage fcontext** 应用该设置。此命令接受 `-t` 环境类型作为第一个参数，后接您要将环境设置应用到的目标目录或文件的名称。要将环境应用到目标目录（要在其中应用该环境）中已存在的所有对象，请将正则表达式 `(/.*)?` 添加到该目录的名称中。这表示：选择性匹配名称为一个斜线后接任何字符的对象。**semanage** 手册页的示例部分包含了 **semanage** 的一些实用的用法示例。有关正则表达式的详细信息，请参见 <http://www.regular-expressions.info/> 上的教程及其他资源。

- 显示系统上所有可用环境类型的列表：

```
tux > sudo seinfo -t
```

由于仅使用该命令会输出非常多的信息，应将它与 **grep** 或类似命令结合使用，以过滤结果。

39.6.4 应用文件环境

为帮助您正确应用 SELinux 环境，以下过程说明了如何使用 **semanage fcontext** 和 **restorecon** 设置环境。您会发现，在首次尝试时，具有非默认文档根的 Web 服务器不会工作。更改 SELinux 环境后，它就会正常工作：

1. 创建 /web 目录，然后对其进行更改：

```
tux > sudo mkdir /web && cd /web
```

2. 使用文本编辑器创建包含文本 “welcome to my Web site” 的 /web/index.html 文件。
3. 使用编辑器打开 /etc/apache2/default-server.conf 文件，将 DocumentRoot 一行更改为 DocumentRoot /web
4. 启动 Apache Web 服务器：

```
tux > sudo systemctl start apache2
```

5. 与您的本地 Web 服务器建立会话：

```
tux > w3m localhost
```

您会收到**连接被拒绝**消息。按 **Enter**，然后按 **q** 退出 w3m。

6. 找到默认 Apache DocumentRoot（即 /srv/www/htdocs）的当前环境类型。此类型应当设置为 httpd_sys_content_t：

```
tux > sudo ls -Z /srv/www
```

7. 在策略中设置新环境，然后按 **Enter**：

```
tux > sudo semanage fcontext -a -f "" -t httpd_sys_content_t '/web(/.*) ?'
```

8. 应用新环境类型：

```
tux > sudo restorecon /web
```

9. 显示目录 /web 中各文件的环境。您将看到，新环境类型已正确设置到 /web 目录，但未设置到其包含的内容。

```
tux > sudo ls -Z /web
```

10. 以递归方式将新环境应用到 /web 目录。现已正确设置类型环境。

```
tux > sudo restorecon -R /web
```

11. 重新启动 Web 服务器：

```
tux > sudo systemctl restart apache2
```

现在，您应该可以访问 /web 目录的内容了。

39.6.5 配置 SELinux 策略

更改策略行为的最简单的方法是使用布尔值。这些布尔值是一些开关，可用来更改策略中的设置。要找出可用的布尔值，请运行

```
tux > sudo semanage boolean -l
```

此命令会显示一个较长的布尔值列表，其中简短说明了每个布尔值的作用。找到想要设置的布尔值后，可以使用 **setsebool -P** 后接您要更改的布尔值的名称。使用 **setsebool** 时，务必始终使用 **-P** 选项。此选项会将设置写入到磁盘上的策略文件，这是确保系统重引导后自动应用布尔值的唯一方式。

下面的过程提供了更改布尔设置的示例

1. 列出与 FTP 服务器相关的布尔值。

```
tux > sudo semanage boolean -l | grep ftp
```

2. 关闭布尔值：

```
tux > sudo setsebool allow_ftp_anon_write off
```

请注意，写入更改不会花费太长时间。然后校验布尔值是否确实关闭：

```
tux > sudo semanage boolean -l|grep ftpd_anon
```

3. 重引导服务器。

4. 再次检查以确认 `allow_ftpd_anon_write` 布尔值是否仍处于开启状态。由于此布尔设置尚未写入策略，因此您会发现它处于关闭状态。
5. 切换布尔值并将设置写入策略：

```
tux > sudo setsebool -P allow_ftpd_anon_write
```

39.6.6 使用 SELinux 模块

默认情况下，SELinux 使用模块化策略。这意味着，实现 SELinux 功能的策略不是单个巨型策略，而是由许多较小模块构成。每个模块都涉及 SELinux 配置的特定部分。引入 SELinux 模块概念的目的在于方便第三方供应商使其服务与 SELinux 兼容。要获取各 SELinux 模块的概览，可以使用 `semodule -l` 命令。此命令会列出 SELinux 当前使用的所有模块及其版本号。

作为管理员，您可以开启或关闭模块。如果您只想禁用 SELinux 的某个部件（而不是所有部件），以便在不受 SELinux 保护的情况下运行特定的服务，此功能非常有用。尤其是在尚无完全受支持的 SELinux 策略的 SUSE Linux Enterprise Server 中，有效的做法可能是关闭您不需要的所有模块，以便将精力集中在确实需要 SELinux 保护的服务上。要关闭 SELinux 模块，请使用

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

要再次将它开启，可以使用

```
tux > sudo semodule -e modulename
```

要更改任何策略模块文件的内容，请在新策略模块文件中编辑更改。为此，请先安装 `selinux-policy-devel` 软件包。然后，在 `audit2allow` 创建的文件所在的目录中，运行：

```
tux > make -f /usr/share/selinux/devel/Makefile
```

完成 `make` 后，可以使用 `semodule -i` 将模块手动装载到系统中。

39.7 查错

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。这涉及到 auditd 服务是否正在运行。如果您发现 `/var/log/audit` 是空的，请使用以下命令启动 auditd 服务

```
tux > sudo systemctl start auditd
```

并使用以下命令在系统目标中启用它

```
tux > sudo systemctl enable auditd
```

例 39.8: “`/etc/audit/audit.log` 中的示例行” 中提供了 `/var/log/audit/audit.log` 的部分示例内容

例 39.8: `/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_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=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 sccontext=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 sccontext=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
```

乍看之下，`audit.log` 中的行读起来有点困难。但如果仔细分析，就会发现它们不是那么难以理解。每一行都可细分为不同的部分。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 type=AVC 开头

msg=audit(1348173901.309:300):

这是时戳，遗憾的是，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：

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

avc: denied { append }:

已被拒绝的特定操作。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

您可以采用另一种方法来取代在 `audit.log` 中自行截获事件的做法。您可以使用 **audit2allow** 命令，它会帮助分析 `/var/log/audit/audit.log` 中晦涩难懂的日志消息。audit2allow 查错会话一律由三个不同的命令组成。首先，您会使用 **audit2allow -w -a** 以更易于理解的方式显示审计信息。**audit2allow -w -a** 默认对 `audit.log` 文件运行。如果您要分析 `audit.log` 文件中的特定消息，请将此消息复制到临时文件，然后使用以下命令分析该临时文件：

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

例 39.9：分析审计消息

```
tux > 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。

要生成可装载的模块以允许这种访问，请运行

```
tux > 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` 文件的所有事件中的强制规则，或使用 `audit2allow -i FILENAME` 显示应用于您储存在特定文件中的消息的规则：

例 39.10：查看哪些行拒绝了访问

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

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

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

如果您要对已记录到 `audit.log` 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 39.11：创建允许先前被拒绝的操作的策略模块

```
tux > 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`）。

VI Linux 审计框架

- 40 了解 Linux 审计 **361**
- 41 设置 Linux 审计框架 **398**
- 42 审计规则集简介 **410**
- 43 有用资源 **421**

40 了解 Linux 审计

此版本的 SUSE Linux Enterprise Server 随附的 Linux 审计框架提供符合 CAPP（受控访问保护配置文件）规范的审计系统，该系统能够可靠地收集任何安全相关事件的信息。您可以通过检查审计记录来确定是否发生任何安全策略违规以及由谁造成。

提供审计框架是 CC-CAPP/EAL（通用准则受控访问保护配置文件/评估保障级别）认证的一项重要要求。信息技术安全信息通用准则 (CC) 是适用于独立安全评估的国际标准。通用准则可帮助客户评判他们想要部署在任务关键型设置中的任何 IT 产品的安全级别。

通用准则安全评估有两套评估要求：功能要求和保障要求。功能要求描述受评估产品的安全属性，汇总于受控访问保护配置文件 (CAPP) 中。保障要求汇总于评估保障级别 (EAL) 中。EAL 描述要使评估者确信安全属性存在、有效且得到实施所必须执行的任何活动。此类活动的示例包括记录开发人员寻找安全漏洞的活动、执行的修补过程和测试。

通过本指南，您可基本理解审计的工作原理以及设置方法。有关通用准则本身的详细信息，请参见[通用准则网站 \(https://www.commoncriteriaportal.org/\)](https://www.commoncriteriaportal.org/) .

Linux 审计为您提供了详细分析系统上发生的情况的方法，可帮助您提高系统的安全性。但是，它本身并不提供额外的安全性 — 它不能防范系统的代码出现故障或系统被以任何方式恶意利用。审计只可用于跟踪这些问题，并帮助您采取额外的安全措施（例如 AppArmor）来防止这些问题。

审计包括多个组件，每个组件都为总体框架提供着关键功能。审计内核模块会截获系统调用并记录相关事件。`auditd` 守护程序会将审计报告写入磁盘。各种命令行实用程序会处理审计追踪的显示、查询和存档。

审计可让您执行以下操作：

将用户与进程相关联

审计会将进程映射到启动它们的用户 ID。这样，管理员或安全员便可以确切地跟踪哪个用户拥有哪个进程，并判断该用户是否可能正在系统上执行恶意操作。

❗ 重要：重命名用户 ID

审计不会处理 UID 的重命名。因此，请避免重命名 UID（例如，将 tux 从 uid=1001 重命名为 uid=2000），而是将 UID 作废。否则，您将需要更改 auditctl 数据（审计规则），并且在正确检索旧数据时会遇到问题。

查看审计追踪

Linux 审计提供了用于将审计报告写入磁盘并将其转换成直观易懂的格式的工具。

查看特定的审计事件

审计提供了可供您过滤特定相关事件的审计报告的实用程序。您可以过滤：

- 用户
- 组
- 审计 ID
- 远程主机名
- 远程主机地址
- 系统调用
- 系统调用参数
- 文件
- 文件操作
- 成功或失败

应用选择性审计

审计提供了用于过滤相关事件的审计报告以及调整审计以仅记录选定事件的方法。您可以创建自己的规则集，让审计守护程序仅记录您想关注的事件。

保证报告数据的可用性

审计报告由 root 拥有，因此只能由 root 去除。未获授权的用户无法去除审计日志。

防止审计数据丢失

如果内核耗尽了内存，将会超出审计守护程序的积压或速率上限，在此情况下，审计可能会触发系统关闭，以防止事件脱离审计的控制。这种关闭是审计内核组件触发的系统立即暂停，不会将最新日志同步到磁盘。默认配置是在系统日志中记录一条警告，而不是暂停系统。

如果系统在记录日志时耗尽了磁盘空间，可将审计系统配置为执行正常关闭。默认配置会告知审计守护程序在耗尽磁盘空间时停止日志记录。

40.1 Linux 审计组件简介

下图说明各审计组件相互之间的交互方式：

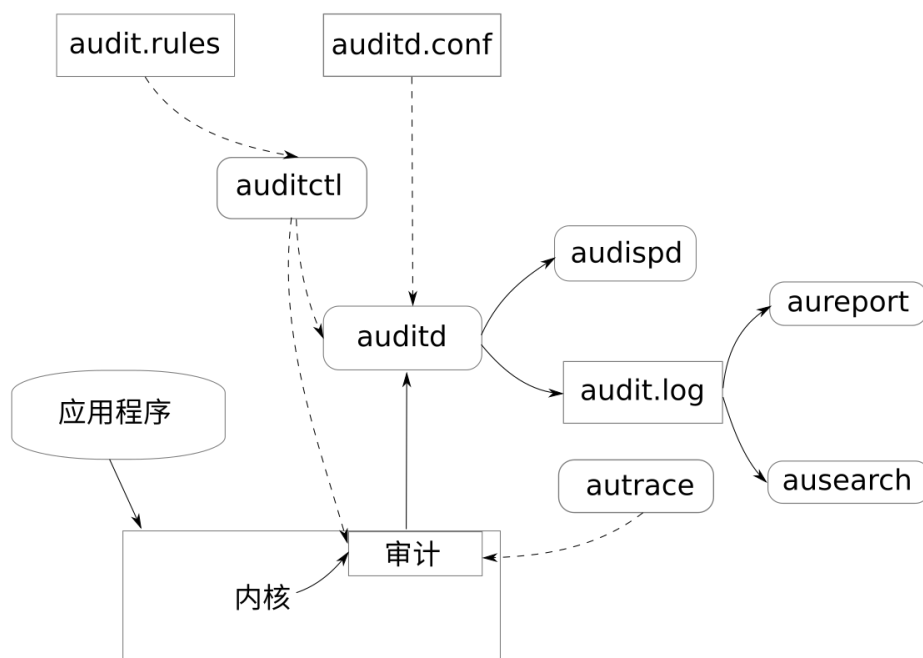


图 40.1：Linux 审计组件简介

实线箭头表示组件之间的数据流，虚线箭头表示组件之间的控制线。

auditd

审计守护程序负责将通过审计内核接口生成并由应用程序和系统活动触发的审计消息写入磁盘。审计守护程序的启动方式由 `systemd` 控制。审计系统功能（启动后）由 `/etc/audit/auditd.conf` 控制。有关 `auditd` 及其配置的详细信息，请参见第 40.2 节“配置审计守护程序”。

auditctl

auditctl 实用程序可控制审计系统。它可控制审计接口的日志生成参数和内核设置，以及用于确定要跟踪哪些事件的规则集。有关 **auditctl** 的详细信息，请参见第 40.3 节“使用 **auditctl** 控制审计系统”。

审计规则

`/etc/audit/audit.rules` 文件包含一系列 **auditctl** 命令，在系统引导时，启动审计守护程序后会紧接着装载这些命令。有关审计规则的详细信息，请参见第 40.4 节“将参数传递到审计系统”。

aureport

aureport 实用程序可让您基于审计事件日志创建自定义报告。您可以轻松编写生成报告的脚本，而各种其他应用程序可以使用脚本的输出来绘制这些结果的图表以及执行其他操作。有关 **aureport** 的详细信息，请参见第 40.5 节“了解审计日志和生成报告”。

ausearch

ausearch 实用程序可以使用所记录的格式的各种键或其他特征在审计日志文件中搜索特定的事件。有关 **ausearch** 的详细信息，请参见第 40.6 节“使用 **ausearch** 查询审计守护程序日志”。

audispd

审计调度程序守护程序 (**audispd**) 可用于将事件通知中继到其他应用程序，而不是将其写入磁盘上的审计日志中（或除了执行此操作之外）。有关 **audispd** 的详细信息，请参见第 40.9 节“中继审计事件通知”。

autrace

autrace 实用程序以类似于 **strace** 的方式跟踪单个进程。**autrace** 的输出将记录到审计日志。有关 **autrace** 的详细信息，请参见第 40.7 节“使用 **autrace** 分析进程”。

aulast

列显最后几个登录用户的列表，类似于 **last**。**aulast** 在整个审计日志（或给定的审计日志文件）中向后搜索，并基于审计日志中的时间范围显示所有登录和注销用户的列表。

aulastlog

以类似于 **lastlog** 的方式列显所有计算机用户的上次登录信息。将列显登录名、端口和上次登录时间。

40.2 配置审计守护程序

在您可以实际开始生成并处理审计日志之前，需先配置审计守护程序本身。`/etc/audit/auditd.conf` 配置文件确定审计系统在守护程序启动后的运行方式。对于大多数用例而言，SUSE Linux Enterprise Server 随附的默认设置应已足够。如果是 CAPP 环境，则需要调整其中的大部分参数。下面的列表简要介绍了可用参数：

```
log_file = /var/log/audit/audit.log
log_format = RAW
log_group = root
priority_boost = 4
flush = INCREMENTAL
freq = 20
num_logs = 5
disp_qos = lossy
dispatcher = /sbin/audispd
name_format = NONE
##name = mydomain
max_log_file = 6
max_log_file_action = ROTATE
space_left = 75
space_left_action = SYSLOG
action_mail_acct = root
admin_space_left = 50
admin_space_left_action = SUSPEND
disk_full_action = SUSPEND
disk_error_action = SUSPEND
##tcp_listen_port =
tcp_listen_queue = 5
tcp_max_per_addr = 1
##tcp_client_ports = 1024-65535
tcp_client_max_idle = 0
cp_client_max_idle = 0
```

根据您是否希望环境满足 CAPP 的要求，在配置审计守护程序时需要额外加强限制。在您需要使用特定的设置才能满足 CAPP 要求的位置，我们会提供“CAPP 环境”注释告知您如何调整配置。

log_file、log_format 和 log_group

log_file 指定审计日志应储存在什么位置。log_format 确定审计信息写入磁盘的方式，log_group 定义拥有日志文件的组。log_format 的可能的值为 raw（完全按照内核发送消息时的格式储存消息）或 nolog（丢弃消息，不将其写入磁盘）。如果您使用 nolog 模式，发送到审计调度程序的数据将不受影响。默认设置为 raw。如果您希望能够使用 **aureport** 和 **ausearch** 工具创建报告以及对审计日志进行查询，应保留该设置。可采用文本描述或使用组 ID 来指定 log_group 的值。



注意：CAPP 环境

在 CAPP 环境中，需让审计日志驻留在其自身的分区中。这样，您便可以确保审计守护程序的空间检测结果准确，并且可以避免其他进程消耗此空间。

priority_boost

确定审计守护程序应获得的优先级提升量。可能的值为 0 到 20。最终合理的值按如下方式计算：0 - priority_boost

flush 和 freq

指定是否、如何以及以什么频率将审计日志写入磁盘。flush 的有效值为 none、incremental、data 和 sync。none 告知审计守护程序将审计数据写入磁盘时无需进行特殊的操作。incremental 告知审计守护程序显式将数据刷写到磁盘。如果使用 incremental，则必须指定频率。freq 值 20 告知审计守护程序请求内核在每生成 20 条记录后将数据刷写到磁盘。data 选项始终将磁盘文件的数据部分保持同步，而 sync 选项会同时处理元数据和数据。



注意：CAPP 环境

在 CAPP 环境中，请确保审计追踪始终是完全最新且完整的。因此，请将 flush 参数与 sync 或 data 结合使用。

num_logs

指定当提供 rotate 作为 max_log_file_action 时要保留的日志文件数。可能的值为 0 到 99。小于 2 的值表示不轮换日志文件。如果增加要轮换的文件数，会增大审计守护程序所需的工作量。执行这种轮换时，auditd 无法始终快速地处理来自内核的新数据，这可能造成积压状况（触发 auditd 根据故障标志做出反应，如第 40.3 节“使用 **auditctl** 控制审计系统”中所述）。在这种情况下，建议提高积压上限。为此，可以更改 /etc/audit/audit.rules 文件中 -b 参数的值。

disp_qos 和 dispatcher

审计守护程序在启动期间会启动调度程序。审计守护程序会将审计消息中继到 dispatcher 中指定的应用程序。此应用程序必须高度受信任，因为它需要以 root 身份运行。disp_qos 会确定是否允许在审计守护程序与调度程序之间进行 lossy（有损）或 lossless（无损）通讯。

如果您选择 lossy，当消息队列已满时，审计守护程序可能会丢弃一些审计消息。如果将 log_format 设为 raw，这些事件仍会写入磁盘，但可能不会传递到调度程序。如果您选择 lossless，将阻止审计记录到磁盘，直到消息队列中出现空位。默认值为 lossy。

name_format 和 name

name_format 控制如何解析计算机名称。可能的值为 none（不使用名称）、hostname（**gethostname** 返回的值）、fqdn（通过 DNS 查找接收的完全限定主机名）、numeric（IP 地址）和 user。user 是一个自定义字符串，需要使用 name 参数来定义。

max_log_file 和 max_log_file_action

max_log_file 接受数字值，该值指定在触发可配置的操作之前，日志文件可以达到的最大文件大小（以兆字节为单位）。要执行的操作在 max_log_file_action 中指定。max_log_file_action 的可能的值为 ignore、syslog、suspend、rotate 和 keep_logs。ignore 告知审计守护程序在达到大小限制时不要执行任何操作，syslog 告知审计守护程序发出警告并将警告发送到系统日志，suspend 导致审计守护程序停止向磁盘写入日志，并使守护程序本身仍保持活动状态。rotate 使用 num_logs 设置触发日志轮换。keep_logs 也会触发日志轮换，但不使用 num_log 设置，因此始终保留所有日志。



注意：CAPP 环境

要在 CAPP 环境中保留完整的审计追踪，应使用 `keep_logs` 选项。如果使用单独的分区来保存审计日志，请调整 `max_log_file` 和 `num_logs` 以使用该分区上的全部可用空间。请注意，要轮换的文件越多，重新接收审计事件所需的时间就越长。

`space_left` 和 `space_left_action`

`space_left` 接受表示剩余磁盘空间的数字值（以兆字节为单位），用于触发审计守护程序执行的可配置操作。该操作在 `space_left_action` 中指定。此参数的可能的值为 `ignore`、`syslog`、`email`、`exec`、`suspend`、`single` 和 `halt`。`ignore` 告知审计守护程序忽略警告且不执行任何操作，`syslog` 指示审计守护程序向系统日志发送警告，`email` 向 `action_mail_acct` 下指定的帐户发送电子邮件。指定 `exec` 加上脚本路径会执行给定的脚本。请注意，您无法向脚本传递参数。`suspend` 告知审计守护程序停止写入磁盘但保持活动状态，而 `single` 会触发系统降级到单用户模式。`halt` 触发系统完全关闭。



注意：CAPP 环境

请确保将 `space_left` 设置为一个适当的值，使管理员有足够的时间对警报做出反应，并允许释放足够的磁盘空间以使审计守护程序能够继续工作。释放磁盘空间涉及到调用 `aureport -t`，并在单独的存档分区或资源上存档最旧的日志。`space_left` 的实际值取决于您的部署大小。将 `space_left_action` 设置为 `email`。

`action_mail_acct`

指定应将任何警报消息发送到的电子邮件地址或别名。默认设置为 `root`，但只要在系统上正确配置了电子邮件和网络，并且 `/usr/lib/sendmail` 存在，您就可以输入任何本地或远程帐户。

admin_space_left 和 admin_space_left_action

admin_space_left 接受表示剩余磁盘空间的数字值（以兆字节为单位）。达到此限制意味着系统已出现磁盘空间不足的情况，管理员需抓住这个最后的机会对此警报做出反应，释放磁盘空间来保存审计日志。admin_space_left 的值应小于 space_left 的值。admin_space_left_action 的可能的值与 space_left_action 的相同。



注意：CAPP 环境

请将 admin_space_left 设置为允许记录管理员操作的值。操作应设置为 single。

disk_full_action

指定当系统耗尽了用于保存审计日志的磁盘空间时需要执行的操作。有效值为 ignore、syslog、rotate、exec、suspend、single 和 halt。有关这些值的说明，请参见 space_left 和 space_left_action。



注意：CAPP 环境

由于当实在没有更多空间来保存任何审计日志时会触发 disk_full_action，您应该将系统降级到单用户模式 (single) 或将其彻底关闭 (halt)。

disk_error_action

指定当审计守护程序在将日志写入磁盘或轮换日志期间遇到任何类型的磁盘错误时需要执行的操作。可能的值与 space_left_action 的相同。



注意：CAPP 环境

根据有关处理任何类型的硬件故障的站点策略，使用 syslog、single 或 halt。

`tcp_listen_port`、`tcp_listen_queue`、`tcp_client_ports`、`tcp_client_max_idle` 和 `tcp_max_per_addr`

该审计守护程序可以接收来自其他审计守护程序的审计事件。TCP 参数可让您控制传入连接。使用 `tcp_listen_port` 指定一个介于 1 到 65535 之间的端口，`auditd` 将监听该端口。`tcp_listen_queue` 可让您配置等待中连接数的最大值。确保不要将值设置得太小，因为在某些情况下（例如在断电后），等待中的连接数可能会很高。`tcp_client_ports` 定义允许哪些客户端端口。请指定单个端口，或以短划线分隔的数字指定端口范围（例如，1-1023 表示所有特权端口）。

指定单个允许的客户端端口可能导致客户端难以重新启动其审计子系统，因为在连接关闭 `TIME_WAIT` 状态超时之前，客户端无法与同一主机地址和端口重新创建连接。如果客户端不再响应，`auditd` 将会控诉。请使用 `tcp_client_max_idle` 指定经过多少秒后将发生这种行为。请记住，此设置对所有客户端都有效，因此应高于任意单个客户端检测信号设置，最好是高两倍。`tcp_max_per_addr` 是表示允许从一个 IP 地址发出多少并发连接的数字值。



提示

我们建议为客户端和服务端使用特权端口，以防止非 root (CAP_NET_BIND_SERVICE) 程序绑定到这些端口。

完成 `/etc/audit/auditd.conf` 中的守护程序配置后，下一步是重点控制守护程序执行的审计量，并为守护程序指派足够其顺利运行的资源和限制。

40.3 使用 `auditctl` 控制审计系统

`auditctl` 负责控制审计守护程序的状态和某些基本系统参数。它控制对系统执行的审计量。`auditctl` 使用审计规则来控制系统的哪些组件需要接受审计及对其进行的审计范围。可在 `auditctl` 命令行中或者通过撰写规则集并指示审计守护程序处理此文件，将审计规则传递给审计守护程序。`auditd` 守护程序默认配置为检查 `/etc/audit/audit.rules` 下的审计规则。有关审计规则的更多细节，请参见第 40.4 节“将参数传递到审计系统”。

用于控制基本审计系统参数的主要 `auditctl` 命令包括：

- `auditctl -e`，用于启用或禁用审计
- `auditctl -f`，用于控制故障标志
- `auditctl -r`，用于控制审计消息的速率上限
- `auditctl -b`，用于控制积压上限
- `auditctl -s`，用于查询审计守护程序的当前状态
- `auditctl -S` 指定要审计的系统调用。在系统上运行 `auditctl -S` 之前，请添加 `-F arch=b64` 以防止出现体系结构不匹配警告。

您还可以在 `audit.rules` 文件中指定 `-e`、`-f`、`-r` 和 `-b` 选项，这样您就无需在审计守护程序每次启动时都重新输入这些选项。

每当您使用 `auditctl -s` 查询审计守护程序的状态，或使用 `auditctl -eFLAG` 更改状态标志时，都会列显一条状态消息（包括有关上述每个参数的信息）。下面的示例重点列出了典型的审计状态消息。

例 40.1： `auditctl -s` 的示例输出

```
enabled 1
failure 1
pid 790
rate_limit 0
backlog_limit 64
lost 0
backlog 0
backlog_wait_time 15000
loginuid_immutable 0 unlocked
```

表 40.1： 审计状态标志

标志	含义 [可能的值]	命令
<code>enabled</code>	设置启用标志。[0..2] 0=禁用，1=启用，2=启用并锁定配置。请注意，此标志只禁用日志记录系统调用，系统仍会记录其他事件。（请	<code>auditctl -e [0 1 2]</code>

标志	含义 [可能的值]	命令
	参见以下主题中的 man 3 audit_set_enabled: audit-devel 。)	
flag	设置故障标志。[0..2] 0=静默，1=printk，2=恐慌（立即暂停且不将等待中数据同步到磁盘）	auditctl -f [0 1 2]
pid	正在运行 auditd 的进程 ID。	—
rate_limit	设置每秒消息数上限。如果该值不为零且每秒消息数超过该上限，将触发故障标志中指定的操作。	auditctl -r RATE
backlog_limit	指定允许的未处理审计缓冲区的最大数目。如果所有缓冲区已满，将触发故障标志中指定的操作。	auditctl -b BACKLOG
lost	统计当前丢失的审计消息数。	—
backlog	统计当前未处理的审计缓冲区数。	—

40.4 将参数传递到审计系统

您可以在外壳中使用 **auditctl** 单独调用用于控制审计系统的命令，也可以使用 **auditctl** -R 从文件中批量读取此类命令。启动审计守护程序后，init 脚本会使用后一种方法从 `/etc/audit/audit.rules` 文件装载规则。规则按照从上到下的顺序执行。其中每条规则将扩展为单独的 **auditctl** 命令。规则文件中使用的语法与 **auditctl** 命令使用的语法相同。

通过在命令行上执行 **auditctl** 对运行中审计系统所做的更改在系统重新启动后不会保留。要持久保留更改，请将更改添加到 `/etc/audit/audit.rules` 文件；如果更改当前尚未装载到审计中，请使用 **systemctl restart auditd** 命令重新启动审计系统以装载修改后的规则集。

例 40.2：示例审计规则 — 审计系统参数

```
-b 1000 ❶  
-f 1 ❷  
-r 10 ❸  
-e 1 ❹
```

- ❶ 指定未处理审计缓冲区的最大数目。根据日志记录活动的级别，您可能需要调整缓冲区的数目，以免系统上的审计负载过于繁重。
- ❷ 指定要使用的故障标志。有关可能的值，请参见表 40.1 “审计状态标志”。
- ❸ 指定内核每秒可发出的最大消息数目。有关详细信息，请参见表 40.1 “审计状态标志”。
- ❹ 启用或禁用审计子系统。

使用审计，您可以跟踪以任何形式通过文件系统对重要文件、配置或资源进行的访问。您可以添加针对这些内容的监测项，并为每种监视项指派相应的键，以方便在日志中识别。

例 40.3：示例审计规则 — 文件系统审计

```
-w /etc/shadow ❶  
-w /etc -p rx ❷  
-w /etc/passwd -k fk_passwd -p rwx ❸
```

- ❶ **-w** 选项告知审计添加指定文件（在本例中为 `/etc/shadow`）的监测项。请求此文件访问权限的所有系统调用都将经过分析。

- ❷ 此规则添加对 `/etc` 目录的监测项，并对读取和执行此目录的访问操作应用权限过滤 (`-p rx`)。请求这两种权限中的任何一种权限的任何系统调用都将经过分析。系统仅将创建新文件和删除现有文件的操作记录为目录相关的事件。要获取此特定目录下各文件的更具体的事件，应该为每个文件单独添加一条规则。在添加包含文件监测项的规则之前，相应文件必须存在。不支持在创建文件时审计文件。
- ❸ 此规则向 `/etc/passwd` 添加一个文件监测项，并对读取、写入、执行和属性更改权限应用权限过滤。`-k` 选项可让您指定一个键，以便日后用来过滤此特定事件的审计日志（例如使用 `ausearch` 过滤）。您可对不同的规则使用相同的键，这样便能在搜索规则时将规则分组。还可以将多个键应用于一条规则。

系统调用审计甚至可让您以低于应用程序级别的级别来跟踪系统的行为。设计这些规则时，请考虑到审计大量系统调用可能会增加系统负载，并导致磁盘空间耗尽。请仔细考虑哪些事件需要跟踪，以及如何过滤事件才会使结果更具体。

例 40.4：示例审计规则 — 系统调用审计

```
-a exit,always -S mkdir ❶  
-a exit,always -S access -F a1=4 ❷  
-a exit,always -S ipc -F a0=2 ❸  
-a exit,always -S open -F success!=0 ❹  
-a task,always -F auid=0 ❺  
-a task,always -F uid=0 -F auid=501 -F gid=wheel ❻
```

- ❶ 此规则对 `mkdir` 系统调用激活审计。`-a` 选项会添加系统调用规则。每当输入 `mkdir` 系统调用 (`exit`、`always`) 时，此规则就会触发一个事件。`-S` 选项指定应对其应用此规则的系统调用。
- ❷ 此规则添加对 `access` 系统调用的审计，但仅当该系统调用的第二个参数 (`mode`) 为 `4` (`R_OK`) 时会进行审计。`exit,always` 告知审计在输入此系统调用时添加其审计环境，并在审计此系统调用后输出报告。
- ❸ 此规则添加 IPC 多路转换系统调用的审计环境。特定的 `ipc` 系统调用将作为第一个 `syscall` 参数传递，可使用 `-F a0=IPC_CALL_NUMBER` 选择该系统调用。
- ❹ 此规则审计失败的 `open` 调用尝试。

- ⑤ 此规则是任务规则（关键字 `task`）的示例。它与上述其他规则的不同之处在于，它会应用于派生或克隆的进程。要过滤此类事件，您只能使用派生时已知的字段，例如 UID、GID 和 AUID。此示例规则过滤带有审计 ID 0 的所有任务。
- ⑥ 最后这条规则使用了很多过滤器。所有过滤选项都与逻辑 AND 运算符相结合，表示此规则将应用于带有审计 ID 501、以 `root` 身份运行并使用 `wheel` 作为组的所有任务。系统会在用户登录时为某个进程分配审计 ID。然后，此 ID 将传给用户的初始进程所启动的任何子进程。即使用户更改其身份，审计 ID 也仍会保持不变，可用于跟踪原始用户的操作。



提示：过滤系统调用参数

有关过滤系统调用参数的更多细节，请参见第 42.6 节“过滤系统调用参数”。

您不仅可以将规则添加到审计系统，而且还可以去除规则。可通过不同的方法一次性删除整个规则集，或者删除系统调用规则或文件和目录监测项：

例 40.5：删除审计规则和事件

```
-D ①  
-d exit,always -S mkdir ②  
-W /etc ③
```

- ① 清除审计规则的队列并删除任何以前存在的规则。此规则用作 `/etc/audit/audit.rules` 文件中的第一条规则，可确保即将添加的规则不会与任何以前存在的规则相冲突。在执行 `autrace` 之前还需使用 `auditctl -D` 命令，以避免跟踪规则与 `audit.rules` 文件中存在的任何规则相冲突。
- ② 此规则删除某个系统调用规则。`-d` 选项必须位于需要从规则队列中删除的任何系统调用规则的前面，并且必须完全匹配。
- ③ 此规则告知审计从规则队列中丢弃包含 `/etc` 目录监测项的规则。此规则删除任何包含 `/etc` 目录监测项的规则，而不管使用了哪种权限过滤或键选项。

要了解审计设置中当前使用了哪些规则，请运行 `auditctl -l`。此命令显示所有规则，每行显示一条规则。

例 40.6：使用 `auditctl -l` 列出规则

```
exit,always watch=/etc perm=rx
```

```
exit,always watch=/etc/passwd perm=rwxa key=fk_passwd
exit,always watch=/etc/shadow perm=rwxa
exit,always syscall=mkdir
exit,always a1=4 (0x4) syscall=access
exit,always a0=2 (0x2) syscall=ipc
exit,always success!=0 syscall=open
```

注意：创建过滤规则

您可以使用各种过滤选项构建非常复杂的审计规则。有关可用于构建审计过滤规则的选项的详细信息以及审计规则的一般信息，请参见 [auditctl\(8\)](#) 手册页。

40.5 了解审计日志和生成报告

要了解 **aureport** 实用程序的作用，必须知道审计守护程序所生成的日志的构造方式，以及审计针对事件具体会记录哪些内容。只有在获知这些信息后，您才能确定哪些报告类型最适合您的需求。

40.5.1 了解审计日志

以下示例重点展示了审计所记录的两个典型事件，以及在审计日志中读取其追踪的方式。一个或多个（如果启用了日志轮换）审计日志储存在 `/var/log/audit` 目录中。第一个示例是个简单的 **less** 命令。第二个示例包含当用户尝试远程登录到运行审计的计算机时，日志中记录的大量 PAM 活动。

例 40.7：简单审计事件 — 查看审计日志

```
type=SYSCALL msg=audit(1234874638.599:5207): arch=c000003e syscall=2 success=yes
exit=4 a0=62fb60 a1=0 a2=31 a3=0 items=1 ppid=25400 pid
=25616 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts1
ses=1164 comm="less" exe="/usr/bin/less" key="doc_log"
type=CWD msg=audit(1234874638.599:5207): cwd="/root"
type=PATH msg=audit(1234874638.599:5207): item=0 name="/var/log/audit/audit.log"
inode=1219041 dev=08:06 mode=0100644 ouid=0 ogid=0 rdev=00:00
```

上述事件（一个简单的 `less /var/log/audit/audit.log`）在日志中写入了三条消息。所有消息密切相关，仅凭其中的一条消息将无法理解其他消息。第一条消息揭示了以下信息：

type

记录的事件类型。在本例中，为系统调用触发的事件指派了 `SYSCALL` 类型。记录 `CWD` 事件的目的是记录执行 `syscall` 时的当前工作目录。为传递给系统调用的每个路径生成了一个 `PATH` 事件。`open` 系统调用仅接受一个路径参数，因此仅生成了一个 `PATH` 事件。请务必注意，`PATH` 事件报告路径名字符串参数时并不经过任何进一步的解释，因此，相对路径需要与 `CWD` 事件报告的路径相结合才能确定访问的对象。

msg

括在括号中的消息 ID。该 ID 分为两个部分。`:` 前面的所有字符表示 Unix 纪元时戳。冒号后面的数字表示实际的事件 ID。所记录的来自一个应用程序系统调用的所有事件都具有相同的事件 ID。如果应用程序发出第二次系统调用，将会为其分配另一个事件 ID。

arch

引用系统调用的 CPU 体系结构。搜索日志时，请在您的任何 `ausearch` 命令中使用 `-i` 选项来解码此信息。

syscall

系统调用的类型，对此特定系统调用运行 `strace` 会列显此信息。此数据取自 `/usr/include/asm/unistd.h` 下的系统调用列表，可能会因体系结构而异。在本例中，`syscall=2` 表示 `less` 应用程序调用的 `open` 系统调用（参见 `man open(2)`）。

success

系统调用是成功还是失败。

exit

系统调用返回的退出值。对于本示例中使用的 `open` 系统调用，退出值为文件描述符编号。此值因系统调用而异。

a0 到 a3

系统调用的前四个参数，采用数字格式。这些参数的值与系统调用相关。本示例（`open` 系统调用）中使用了以下值：

```
a0=62fb60 a1=8000 a2=31 a3=0
```

a0 是传递的路径名的起始地址。a1 是标志。以十六进制表示的 8000 转换为以八进制表示的 100000，后者又转换为 0_LARGEFILE。a2 是模式，由于未指定 0_CREAT，因此未使用此参数。**open** 系统调用未传递 a3。请查看相关系统调用的手册页，了解可与该系统调用搭配使用的参数。

items

传递给应用程序的字符串数。

ppid

所分析进程的父进程的 ID。

pid

所分析进程的 ID。

auid

审计 ID。系统会在用户登录时为某个进程分配审计 ID。然后，此 ID 将传给用户的初始进程所启动的任何子进程。即使用户更改了其身份（例如，变成了 root），审计 ID 也会保持不变。因此，您始终可以跟踪原始登录用户的操作。

uid

启动该进程的用户的 ID。在本例中，该 ID 为 0（表示 root）。

gid

启动该进程的用户的组 ID。在本例中，该 ID 为 0（表示 root）。

euid、suid、fsuid

启动该进程的用户的有效用户 ID、设置的用户 ID 和文件系统用户 ID。

egid、sgid、fsgid

启动该进程的用户的有效组 ID、设置的组 ID 和文件系统组 ID。

tty

用于启动应用程序的终端。本示例在 SSH 会话中使用了一个伪终端。

ses

登录会话 ID。系统会在用户登录时设置此进程属性，它可以将任何进程关联到特定的用户登录操作。

comm

应用程序显示在任务列表中时所使用的名称。

exe

二进制程序的解析路径名。

subj

auditd 记录进程是否受到任何安全环境（例如 AppArmor）的约束。本例中所示的 unconstrained 表示进程不受 AppArmor 的限制。如果进程受到限制，将记录二进制文件路径名加上 AppArmor 配置文件模式。

key

如果您正在审计许多目录或文件，请向其中的每个监测项指派键字符串。将这些键与 ausearch 结合使用可以仅搜索此类型事件的日志。

示例 **less** 调用触发的第二条消息只揭示了执行 **less** 命令时的当前工作目录。

第三条消息揭示了以下信息（已引入 type 和 message 标志）：

item

在本示例中，item 引用了 a0 参数 — 与原始 SYSCALL 消息关联的路径。如果原始调用有多个路径参数（例如 **cp** 或 **mv** 命令），将会额外为第二个路径参数记录一个 PATH 事件。

name

表示作为参数传递给 open 系统调用的路径名。

inode

表示与 name 对应的 inode 编号。

dev

指定储存文件的设备。在本例中为 08:06，表示 /dev/sda1 或“第一个 IDE 设备上的第一个分区”。

mode

文件访问权限的数字表示形式。在本例中，root 拥有读取和写入权限，其组 (root) 拥有读取访问权限，而其余的所有用户和组无法访问该文件。

ouid 和 ogid

表示 inode 本身的 UID 和 GID。

rdev

不适用于此示例。rdev 项仅适用于块设备或字符设备，不适用于文件。

例 40.8 “高级审计事件 — 通过 SSH 登录” 重点展示了传入的 SSH 连接所触发的审计事件。大多数消息与 PAM 堆栈相关，反映 SSH PAM 进程的不同阶段。有几条审计消息带有嵌套的 PAM 消息，这些 PAM 消息表示已达到 PAM 进程的特定阶段。尽管审计会记录 PAM 消息，但它会为每个事件指派其自身的消息类型：

例 40.8：高级审计事件 — 通过 SSH 登录

```
type=USER_AUTH msg=audit(1234877011.791:7731): user pid=26127 uid=0 ❶
auid=4294967295 ses=4294967295 msg='op=PAM:authentication acct="root" exe="/usr/
sbin/sshd"
(hostname=jupiter.example.com, addr=192.168.2.100, terminal=ssh res=success)'
type=USER_ACCT msg=audit(1234877011.795:7732): user pid=26127 uid=0 ❷
auid=4294967295 ses=4294967295 msg='op=PAM:accounting acct="root" exe="/usr/
sbin/sshd"
(hostname=jupiter.example.com, addr=192.168.2.100, terminal=ssh res=success)'
type=CRED_ACQ msg=audit(1234877011.799:7733): user pid=26125 uid=0 ❸
auid=4294967295 ses=4294967295 msg='op=PAM:setcred acct="root" exe="/usr/sbin/
sshd"
(hostname=jupiter.example.com, addr=192.168.2.100, terminal=/dev/pts/0
res=success)'
type=LOGIN msg=audit(1234877011.799:7734): login pid=26125 uid=0
old auid=4294967295 new auid=0 old ses=4294967295 new ses=1172
type=USER_START msg=audit(1234877011.799:7735): user pid=26125 uid=0 ❹
auid=0 ses=1172 msg='op=PAM:session_open acct="root" exe="/usr/sbin/sshd"
(hostname=jupiter.example.com, addr=192.168.2.100, terminal=/dev/pts/0
res=success)'
type=USER_LOGIN msg=audit(1234877011.823:7736): user pid=26128 uid=0 ❺
auid=0 ses=1172 msg='uid=0: exe="/usr/sbin/sshd"
(hostname=jupiter.example.com, addr=192.168.2.100, terminal=/dev/pts/0
res=success)'
type=CRED_REFR msg=audit(1234877011.828:7737): user pid=26128 uid=0 ❻
auid=0 ses=1172 msg='op=PAM:setcred acct="root" exe="/usr/sbin/sshd"
(hostname=jupiter.example.com, addr=192.168.2.100, terminal=/dev/pts/0
res=success)'
```

- ❶ PAM 报告它已向远程主机 (jupiter.example.com, 192.168.2.100) 成功请求对 root 用户进行身份验证。发生此操作的终端为 ssh。
- ❷ PAM 报告它已成功确定是否已授权用户登录。
- ❸ PAM 报告已获取用于登录的适当身份凭证，并且终端已变为正常终端 (/dev/pts0)。
- ❹ PAM 报告它已成功为 root 打开会话。
- ❺ 用户已成功登录。此事件是 aureport -l 用来报告用户登录的事件。
- ❻ PAM 报告已成功重新获取身份凭证。

40.5.2 生成自定义审计报告

/var/log/audit 目录中储存的原始审计报告会逐渐变得很庞大且难以理解。要想更轻松地查找相关消息，请使用 aureport 实用程序并创建自定义报告。

以下用例重点展示了您可以使用 aureport 生成的几种可能的报告类型：

从另一文件读取审计日志

当审计日志移到另一台计算机后，或者当您想要在本地上分析多台计算机的日志，而又不想逐个连接其中每台计算机时，请将日志移到某个本地文件，然后在本地使用 aureport 分析这些日志：

```
tux > sudo aureport -if myfile

Summary Report
=====
Range of time in logs: 03/02/09 14:13:38.225 - 17/02/09 14:52:27.971
Selected time for report: 03/02/09 14:13:38 - 17/02/09 14:52:27.971
Number of changes in configuration: 13
Number of changes to accounts, groups, or roles: 0
Number of logins: 6
Number of failed logins: 13
Number of authentications: 7
Number of failed authentications: 573
Number of users: 1
Number of terminals: 9
Number of host names: 4
```

```
Number of executables: 17
Number of files: 279
Number of AVC's: 0
Number of MAC events: 0
Number of failed syscalls: 994
Number of anomaly events: 0
Number of responses to anomaly events: 0
Number of crypto events: 0
Number of keys: 2
Number of process IDs: 1211
Number of events: 5320
```

上述不带任何参数的 **aureport** 命令仅提供基于 myfile 中包含的日志生成的一般标准摘要报告。要创建更详细的报告，请将 -if 选项与下面的任何选项结合使用。例如，生成仅限特定时间范围的登录报告：

```
tux > sudo aureport -l -ts 14:00 -te 15:00 -if myfile

Login Report
=====
# date time auid host term exe success event
=====
1. 17/02/09 14:21:09 root: 192.168.2.100 sshd /usr/sbin/sshd no 7718
2. 17/02/09 14:21:15 0 jupiter /dev/pts/3 /usr/sbin/sshd yes 7724
```

将数字实体转换为文本

某些信息（例如用户 ID）将以数字形式列显。要将这些信息转换为直观易懂的文本格式，请在 **aureport** 命令中添加 -i 选项。

创建粗略的摘要报告

如果您要关注当前的审计统计（事件、登录、进程等），请运行不带任何其他选项的 **aureport**。

创建失败事件的摘要报告

如果您要将单纯的 **aureport** 命令所提供的总体统计细分为失败事件的统计，请使用 **aureport --failed**：

```
tux > sudo aureport --failed
```

```

Failed Summary Report
=====
Range of time in logs: 03/02/09 14:13:38.225 - 17/02/09 14:57:35.183
Selected time for report: 03/02/09 14:13:38 - 17/02/09 14:57:35.183
Number of changes in configuration: 0
Number of changes to accounts, groups, or roles: 0
Number of logins: 0
Number of failed logins: 13
Number of authentications: 0
Number of failed authentications: 574
Number of users: 1
Number of terminals: 5
Number of host names: 4
Number of executables: 11
Number of files: 77
Number of AVC's: 0
Number of MAC events: 0
Number of failed syscalls: 994
Number of anomaly events: 0
Number of responses to anomaly events: 0
Number of crypto events: 0
Number of keys: 2
Number of process IDs: 708
Number of events: 1583

```

创建成功事件的摘要报告

如果您要将单纯的 **aureport** 命令所提供的总体统计细分为成功事件的统计，请使用

aureport --success:

```

tux > sudo aureport --success

Success Summary Report
=====
Range of time in logs: 03/02/09 14:13:38.225 - 17/02/09 15:00:01.535
Selected time for report: 03/02/09 14:13:38 - 17/02/09 15:00:01.535
Number of changes in configuration: 13
Number of changes to accounts, groups, or roles: 0
Number of logins: 6
Number of failed logins: 0

```

```
Number of authentications: 7
Number of failed authentications: 0
Number of users: 1
Number of terminals: 7
Number of host names: 3
Number of executables: 16
Number of files: 215
Number of AVC's: 0
Number of MAC events: 0
Number of failed syscalls: 0
Number of anomaly events: 0
Number of responses to anomaly events: 0
Number of crypto events: 0
Number of keys: 2
Number of process IDs: 558
Number of events: 3739
```

创建摘要报告

除了专用摘要报告（主要事件摘要，以及失败和成功事件摘要），还可以将 `--summary` 选项与大多数其他选项结合使用，以仅创建特定关注方面的摘要报告。不过，并非所有报告都支持此选项。下面的示例创建了用户登录事件的摘要报告：

```
tux > sudo aureport -u -i --summary
```

```
User Summary Report
```

```
=====
```

```
total  auid
```

```
=====
```

```
5640  root
```

```
13   tux
```

```
3    wilber
```

创建事件报告

要了解审计记录的事件，请使用 `aureport -e` 命令。此命令会生成所有事件的带编号列表，其中包含日期、时间、事件编号、事件类型和审计 ID。

```
tux > sudo aureport -e -ts 14:00 -te 14:21
```

```
Event Report
```

```
=====
# date time event type auid success
=====
1. 17/02/09 14:20:27 7462 DAEMON_START 0 yes
2. 17/02/09 14:20:27 7715 CONFIG_CHANGE 0 yes
3. 17/02/09 14:20:57 7716 USER_END 0 yes
4. 17/02/09 14:20:57 7717 CRED_DISP 0 yes
5. 17/02/09 14:21:09 7718 USER_LOGIN -1 no
6. 17/02/09 14:21:15 7719 USER_AUTH -1 yes
7. 17/02/09 14:21:15 7720 USER_ACCT -1 yes
8. 17/02/09 14:21:15 7721 CRED_ACQ -1 yes
9. 17/02/09 14:21:15 7722 LOGIN 0 yes
10. 17/02/09 14:21:15 7723 USER_START 0 yes
11. 17/02/09 14:21:15 7724 USER_LOGIN 0 yes
12. 17/02/09 14:21:15 7725 CRED_REFR 0 yes
```

基于所有进程事件创建报告

要从进程的角度分析日志，请使用 **aureport -p** 命令。此命令会生成所有进程事件的带编号列表，其中包含日期、时间、进程 ID、可执行文件的名称、系统调用、审计 ID 和事件编号。

```
aureport -p
```

```
Process ID Report
```

```
=====
# date time pid exe syscall auid event
=====
1. 13/02/09 15:30:01 32742 /usr/sbin/cron 0 0 35
2. 13/02/09 15:30:01 32742 /usr/sbin/cron 0 0 36
3. 13/02/09 15:38:34 32734 /usr/lib/gdm/gdm-session-worker 0 -1 37
```

基于所有系统调用事件创建报告

要从系统调用的角度分析审计日志，请使用 **aureport -s** 命令。此命令会生成所有系统调用事件的带编号列表，其中包含日期、时间、系统调用的编号、进程 ID、使用此调用的命令的名称、审计 ID 和事件编号。

```
tux > sudo aureport -s
```

```

Syscall Report
=====
# date time syscall pid comm auid event
=====
1. 16/02/09 17:45:01 2 20343 cron -1 2279
2. 16/02/09 17:45:02 83 20350 mktemp 0 2284
3. 16/02/09 17:45:02 83 20351 mkdir 0 2285

```

基于所有可执行文件事件创建报告

要从可执行文件的角度分析审计日志，请使用 **aureport -x** 命令。此命令会生成所有可执行文件事件的带编号列表，其中包含日期、时间、可执行文件的名称、运行可执行文件的终端、执行可执行文件的主机、审计 ID 和事件编号。

```

aureport -x

Executable Report
=====
# date time exe term host auid event
=====
1. 13/02/09 15:08:26 /usr/sbin/sshd sshd 192.168.2.100 -1 12
2. 13/02/09 15:08:28 /usr/lib/gdm/gdm-session-worker :0 ? -1 13
3. 13/02/09 15:08:28 /usr/sbin/sshd ssh 192.168.2.100 -1 14

```

创建有关文件的报告

要基于审计日志生成侧重于文件访问的报告，请使用 **aureport -f** 命令。此命令会生成所有文件相关事件的带编号列表，其中包含日期、时间、所访问的文件的名称、访问文件的系统调用的编号、命令的成功或失败结果、访问文件的可执行文件、审计 ID 和事件编号。

```

tux > sudo aureport -f

File Report
=====
# date time file syscall success exe auid event
=====
1. 16/02/09 17:45:01 /etc/shadow 2 yes /usr/sbin/cron -1 2279
2. 16/02/09 17:45:02 /tmp/ 83 yes /bin/mktemp 0 2284
3. 16/02/09 17:45:02 /var 83 no /bin/mkdir 0 2285

```

创建有关用户的报告

要基于审计日志生成用于说明哪些用户正在您的系统上运行哪些可执行文件的报告，请使用 **aureport -u** 命令。此命令会生成所有用户相关事件的带编号列表，其中包含日期、时间、审计 ID、使用的终端、主机、可执行文件的名称和事件 ID。

```
aureport -u
```

```
User ID Report
```

```
=====
```

```
# date time auid term host exe event
```

```
=====
```

```
1. 13/02/09 15:08:26 -1 sshd 192.168.2.100 /usr/sbin/sshd 12
2. 13/02/09 15:08:28 -1 :0 ? /usr/lib/gdm/gdm-session-worker 13
3. 14/02/09 08:25:39 -1 ssh 192.168.2.101 /usr/sbin/sshd 14
```

创建有关登录的报告

要创建重点统计登录您计算机的尝试的报告，请运行 **aureport -l** 命令。此命令会生成所有登录相关事件的带编号列表，其中包含日期、时间、审计 ID、使用的主机和终端、可执行文件的名称、尝试的成功或失败结果，以及事件 ID。

```
tux > sudo aureport -l -i
```

```
Login Report
```

```
=====
```

```
# date time auid host term exe success event
```

```
=====
```

```
1. 13/02/09 15:08:31 tux: 192.168.2.100 sshd /usr/sbin/sshd no 19
2. 16/02/09 12:39:05 root: 192.168.2.101 sshd /usr/sbin/sshd no 2108
3. 17/02/09 15:29:07 geeko: ? tty3 /bin/login yes 7809
```

将报告范围限制在特定的时间范围

要分析特定时间范围的日志（例如，仅分析 2009 年 2 月 16 日工作时间的日志），请先运行 **aureport -t** 来确定这些数据是否包含在当前的 **audit.log** 中，或者日志是否已进行了轮换：

```
aureport -t
```

```
Log Time Range Report
```

```
=====
/var/log/audit/audit.log: 03/02/09 14:13:38.225 - 17/02/09 15:30:01.636
```

当前的 `audit.log` 包含所有所需的数据。如果情况并非如此，请使用 `-if` 选项将 **aureport** 命令指向包含所需数据的日志文件。

然后指定所需时间范围的开始与结束日期和时间，并将其与所需的报告选项结合使用。本示例重点统计登录尝试：

```
tux > sudo aureport -ts 02/16/09 8:00 -te 02/16/09 18:00 -l

Login Report
=====
# date time auid host term exe success event
=====
1. 16/02/09 12:39:05 root: 192.168.2.100 sshd /usr/sbin/sshd no 2108
2. 16/02/09 12:39:12 0 192.168.2.100 /dev/pts/1 /usr/sbin/sshd yes 2114
3. 16/02/09 13:09:28 root: 192.168.2.100 sshd /usr/sbin/sshd no 2131
4. 16/02/09 13:09:32 root: 192.168.2.100 sshd /usr/sbin/sshd no 2133
5. 16/02/09 13:09:37 0 192.168.2.100 /dev/pts/2 /usr/sbin/sshd yes 2139
```

开始日期和时间是使用 `-ts` 选项指定的。时戳等于或晚于给定开始时间的任何事件都会显示在报告中。如果省略日期，**aureport** 将假设您指的是**今天**。如果省略时间，它会假设开始时间是指定日期的午夜。

使用 `-te` 选项指定结束日期和时间。时戳等于或早于给定事件时间的任何事件都会显示在报告中。如果省略日期，**aureport** 将假设您指的是今天。如果省略时间，它会假设结束时间是现在。请使用与 `-ts` 相同的日期和时间格式。

除摘要报告以外的所有报告将以列格式列显并发送到 STDOUT，这意味着，这些数据可以十分轻松地写入到其他命令。[第 40.8 节 “可视化审计数据”](#) 中介绍的视觉化脚本是演示如何进一步处理审计所生成的数据的示例。

40.6 使用 **ausearch** 查询审计守护程序日志

aureport 工具可帮助您创建有关系统上发生的情况的总体摘要，但如果您要了解特定事件的细节，可以使用 **ausearch** 工具。

ausearch 可让您使用特殊的键和搜索短语搜索审计日志，这些键和短语与 `/var/log/audit/audit.log` 中的事件消息内显示的大多数标志相关。并非所有记录类型都包含相同的搜索短语。例如，`PATH` 记录中就不包含 `hostname` 或 `uid` 项。

搜索时，请确保选择适当的搜索准则来捕获所需的所有记录。另一方面，您在搜索特定类型的记录时，可能会随其一并获取相关的其他各种记录。之所以会这样，是因为内核的不同组件会提供与所要查找的记录相关的其他事件记录。例如，对于 `open` 系统调用，您在获取 `SYSCALL` 记录的同时始终会获取一条 `PATH` 记录。



提示：使用多个搜索选项

您可将任何命令行选项与 AND 逻辑运算符相结合，以缩小搜索范围。

从另一文件读取审计日志

将审计日志移到另一台计算机后，或者当您想要在本地上分析多台计算机的日志，而又不想逐个连接其中每台计算机时，请将日志移到某个本地文件，然后在本地使用 **ausearch** 搜索这些日志：

```
tux > sudo ausearch - option -if myfile
```

将数字结果转换为文本

某些信息（例如用户 ID）将以数字形式列显。要将这些信息转换为直观易懂的文本格式，请在 **ausearch** 命令中添加 `-i` 选项。

按审计事件 ID 搜索

如果您先前运行了审计报告或执行了 **autrace**，则应分析日志中特定事件的追踪。第 40.5 节“了解审计日志和生成报告”中所述的大多数报告类型都会在其输出中包含审计事件 ID。审计事件 ID 是审计消息 ID 的第二部分，后者由 Unix 纪元时戳和审计事件 ID 构成（以冒号分隔）。所记录的来自一个应用程序系统调用的所有事件都具有相同的事件 ID。在 **ausearch** 中使用此事件 ID 可以从日志中检索此事件的追踪。

使用如下所示的命令：

```
tux > sudo ausearch -a 5207
----
time->Tue Feb 17 13:43:58 2009
```

```
type=PATH msg=audit(1234874638.599:5207): item=0 name="/var/log/audit/audit.log" inode=1219041 dev=08:06 mode=0100644 ouid=0 ogid=0 rdev=00:00
type=CWD msg=audit(1234874638.599:5207): cwd="/root"
type=SYSCALL msg=audit(1234874638.599:5207): arch=c000003e syscall=2
success=yes exit=4 a0=62fb60 a1=0 a2=31 a3=0 items=1 ppid=25400 pid=25616
auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts1
ses=1164 comm="less" exe="/usr/bin/less" key="doc_log"
```

`ausearch -a` 命令会抓取并显示日志中与所提供审计事件 ID 相关的所有记录。此选项可与任何其他选项结合使用。

按消息类型搜索

要搜索特定消息类型的审计记录，请使用 **`ausearch -m MESSAGE_TYPE`** 命令。有效消息类型的示例包括 `PATH`、`SYSCALL` 和 `USER_LOGIN`。运行不带消息类型的 **`ausearch -m`** 会显示所有消息类型的列表。

按登录 ID 搜索

要查看与特定登录用户 ID 关联的记录，请使用 **`ausearch -ul`** 命令。此命令显示与指定的用户登录 ID 相关的所有记录，前提是该用户过去能够成功登录。

按用户 ID 搜索

使用 **`ausearch -ua`** 查看与任何用户 ID（用户 ID 和有效用户 ID）相关的记录。使用 **`ausearch -ui UID`** 查看与特定用户 ID 相关的报告。要搜索与特定有效用户 ID 相关的记录，请使用 **`ausearch -ue EUID`**。搜索用户 ID 是指搜索创建进程的用户的 ID。搜索有效用户 ID 是指搜索该用户 ID 以及运行此进程所需的特权。

按组 ID 搜索

使用 **`ausearch -ga`** 命令查看与任何组 ID（组 ID 和有效组 ID）相关的记录。使用 **`ausearch -gi GID`** 查看与特定用户 ID 相关的报告。要搜索与特定有效组 ID 相关的记录，请使用 **`ausearch -ge EGID`**。

按命令行名称搜索

使用 **`ausearch -c COMM_NAME`** 命令查看与特定命令相关的记录，例如，使用 **`ausearch -c less`** 可查看与 `less` 命令相关的所有记录。

按可执行文件名搜索

使用 `ausearch -x EXE` 命令查看与特定可执行文件相关的记录，例如，使用 `ausearch -x /usr/bin/less` 可查看与 `/usr/bin/less` 可执行文件相关的所有记录。

按系统调用名称搜索

使用 `ausearch -sc SYSCALL` 命令查看与特定系统调用相关的记录，例如，使用 `ausearch -sc open` 可查看与 `open` 系统调用相关的所有记录。

按进程 ID 搜索

使用 `ausearch -p PID` 命令查看与特定进程 ID 相关的记录，例如，使用 `ausearch -p 13368` 可查看与此进程 ID 相关的所有记录。

按事件或系统调用成功值搜索

使用 `ausearch -sv SUCCESS_VALUE` 查看包含特定系统调用成功值的记录，例如，使用 `ausearch -sv yes` 可查看所有成功的系统调用。

按文件名搜索

使用 `ausearch -f FILE_NAME` 查看包含特定文件名的记录，例如，使用 `ausearch -f /foo/bar` 可查看与 `/foo/bar` 文件相关的所有记录。您也可以仅使用文件名，但不能使用相对路径。

按终端搜索

使用 `ausearch -tm TERM` 查看仅与特定终端相关的记录，例如，使用 `ausearch -tm ssh` 可查看与 SSH 终端上的事件相关的所有记录，使用 `ausearch -tm tty` 可查看与该控制台相关的所有事件。

按主机名搜索

使用 `ausearch -hn HOSTNAME` 查看与特定远程主机名相关的记录，例如，使用 `ausearch -hn jupiter.example.com` 可查看与该主机名相关的所有记录。可以使用主机名、完全限定的域名或数字格式的网络地址。

按键字段搜索

查看包含审计规则集中指派的特定键（用于识别特定类型的事件）的记录。相关命令为 `ausearch -k KEY_FIELD`。例如，使用 `ausearch -k CFG_etc` 可显示包含 `CFG_etc` 键的所有记录。

按字词搜索

查看包含审计规则集中指派的特定字符串（用于识别特定类型的事件）的记录。整个字符串将与文件名、主机名和终端进行匹配。相关命令为 **`ausearch -w WORD`**。

将搜索范围限制在特定的时间范围

使用 **`-ts`** 和 **`-te`** 可将搜索范围限制在特定的时间范围。**`-ts`** 选项用于指定开始日期和时间，**`-te`** 选项用于指定结束日期和时间。这些选项可与上面所述的任何选项结合使用。这些选项的用法与在 **`aureport`** 中的用法类似。

40.7 使用 **autrace** 分析进程

除了使用设置的规则监视系统以外，您还可以使用 **autrace** 命令对各个进程执行专门的审计。**autrace** 的工作方式类似于 **strace**，但它收集的信息略有不同。**autrace** 的输出将写入到 `/var/log/audit/audit.log`，看上去与标准审计日志项并无任何不同。

对进程执行 **autrace** 时，请确保从队列中清除所有审计规则，以免这些规则与 **autrace** 本身添加的规则相冲突。使用 **`auditctl -D`** 命令删除审计规则。这会停止所有一般审计。

```
tux > sudo auditctl -D

No rules

autrace /usr/bin/less

Waiting to execute: /usr/bin/less
Cleaning up...
No rules
Trace complete. You can locate the records with 'ausearch -i -p 7642'
```

请一律使用要通过 **autrace** 跟踪的可执行文件的完整路径。完成跟踪后，**autrace** 会提供跟踪的事件 ID，因此您可以使用 **ausearch** 分析整个数据追踪。要将审计系统恢复为重新使用审计规则集，请使用 **`systemctl restart auditd`** 重新启动审计守护程序。

40.8 可视化审计数据

`/var/log/audit/audit.log` 中的数据追踪以及 **aureport** 生成的不同报告类型

（如第 40.5.2 节 “生成自定义审计报告” 中所述）都不会向用户提供直观的阅读体验。**aureport** 输出采用列格式，因此可轻松地用在用户可能连接到审计框架的任何 sed、Perl 或 awk 脚本中，以直观呈现审计数据。

可视化脚本（参见第 41.6 节 “配置日志可视化”）是展示如何使用 SUSE Linux Enterprise Server 或任何其他 Linux 发行套件提供的标准 Linux 工具创建易于阅读的审计输出的一个示例。以下示例可帮助您了解如何将纯文本审计报告转换为直观易懂的图形。

第一个示例说明程序与系统调用之间的关系。要了解此类数据，需要确定用于提供源数据（最终图形是在这些数据的基础上生成的）的相应 **aureport** 命令：

```
tux > sudo aureport -s -i
```

Syscall Report

```
=====
# date time syscall pid comm auid event
=====
1. 16/02/09 17:45:01 open 20343 cron unset 2279
2. 16/02/09 17:45:02 mkdir 20350 mktemp root 2284
3. 16/02/09 17:45:02 mkdir 20351 mkdir root 2285
...
```

可视化脚本需要对此报告执行的第一项操作是仅提取所需的列，在本例中为 `syscall` 和 `comm` 列。将输出排序并去除重复数据，然后将最终输出写入可视化程序自身中：

```
LC_ALL=C aureport -s -i | awk '/^[0-9]/ { print $6" "$4 }' | sort | uniq |
mkgraph
```

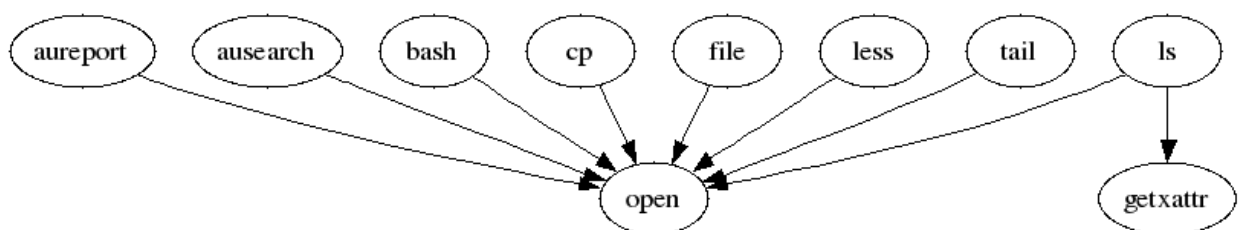


图 40.2：流程图 — 程序与系统调用之间的关系

第二个示例说明各种不同类型的事件以及已记录的每种类型的事件数量。用于提取此类信息的相应 **aureport** 命令是 **aureport -e**:

```
tux > sudo aureport -e -i --summary
```

```
Event Summary Report
=====
total  type
=====
2434  SYSCALL
816   USER_START
816   USER_ACCT
814   CRED_ACQ
810   LOGIN
806   CRED_DISP
779   USER_END
99    CONFIG_CHANGE
52    USER_LOGIN
```

由于此类报告已包含两列输出，因此只会馈送到可视化脚本并转换为条形图。

```
tux > sudo aureport -e -i --summary | mkbar events
```

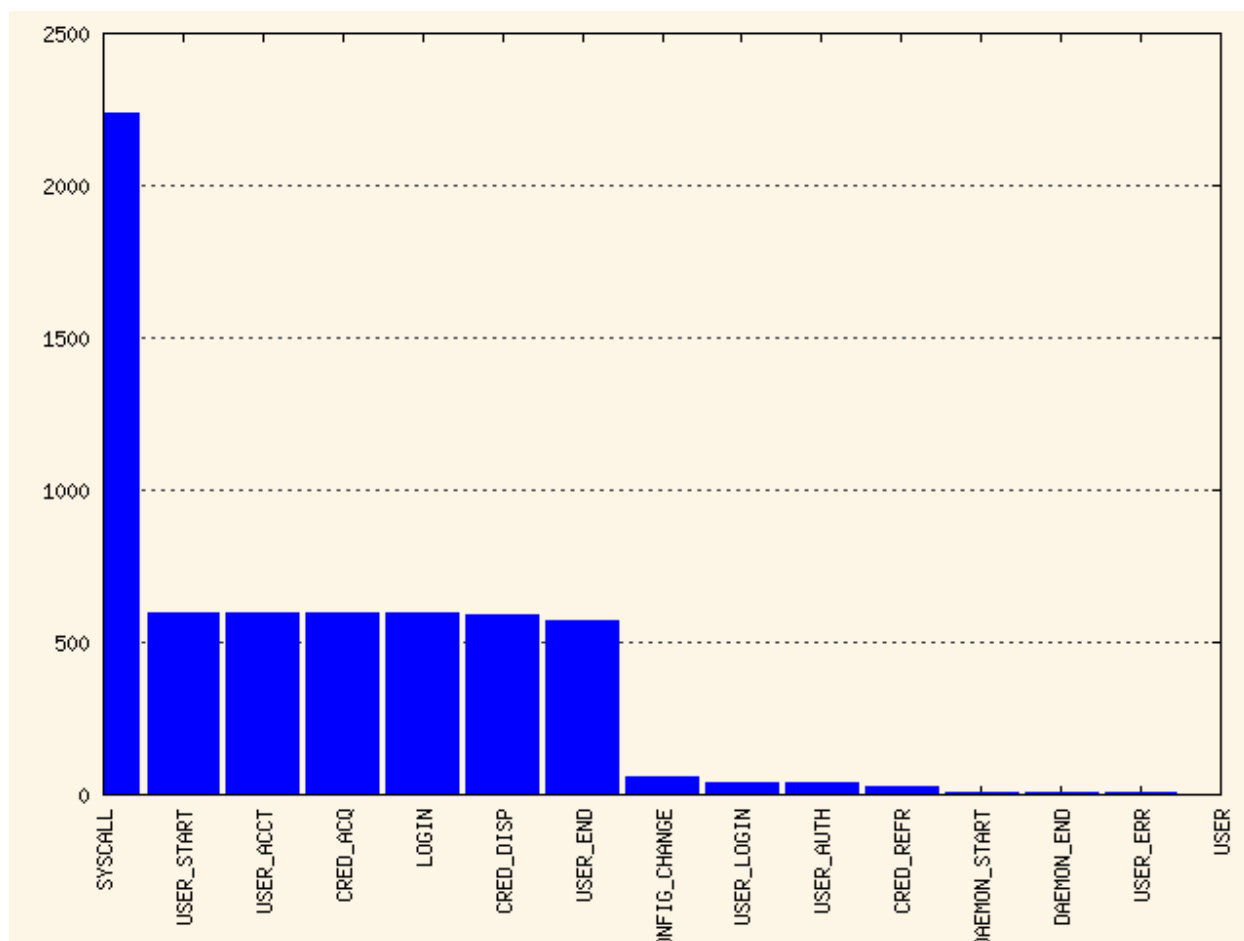


图 40.3：条形图 — 常见事件类型

有关审计数据可视化的背景信息，请参见审计项目的网站 <http://people.redhat.com/sgrubb/audit/visualize/index.html>。

40.9 中继审计事件通知

审计系统还允许外部应用程序实时访问和使用 `auditd` 守护程序。此功能由所谓的**审计调度程序**提供，举例而言，入侵检测系统可以通过此程序使用 `auditd` 来接收增强的检测信息。

`audispd` 是用于控制审计调度程序的守护程序。它通常由 `auditd` 启动。`audispd` 会提取审计事件并将其分发到想要对其进行实时分析的程序。`auditd` 的配置储存在 `/etc/audisp/audispd.conf` 中。该文件具有以下选项：

q_depth

指定事件调度程序内部队列的大小。如果系统日志指出审计事件开始被丢弃，请增大此值。默认值为 250。

overflow_action

指定审计守护程序对内部队列溢出的反应方式。可能的值为 ignore（不执行任何操作）、syslog（向系统日志发送警告）、suspend（`audispd` 将停止处理事件）、single（将计算机系统置于单用户模式）或 halt（关闭系统）。

priority_boost

指定审计事件调度程序的优先级（以及审计守护程序本身的优先级）。默认值为 4，即优先级无变化。

name_format

指定在审计事件中插入计算机节点名称的方式。可能的值为 none（不插入计算机名）、hostname（`gethostname` 系统调用返回的名称）、fqd（计算机的完全限定域名）、numeric（计算机的 IP 地址）或 user（name 选项中用户定义的字符串）。默认值为 none。

name

指定用于标识计算机的用户定义的字符串。name_format 选项必须设置为 user，否则会忽略此选项。

max_restarts

用于指定审计事件调度程序可以尝试重新启动崩溃插件的次数的非负数。默认值为 10。

例 40.9：示例 `/etc/audisp/audispd.conf`

```
q_depth = 250
overflow_action = SYSLOG
priority_boost = 4
name_format = HOSTNAME
#name = mydomain
```

插件程序将其配置文件安装在专用于 `audispd` 插件的特殊目录中。此目录默认为 `/etc/audisp/plugins.d`。插件配置文件具有以下选项：

active

指定程序是否使用 audispd。可能的值为 yes 或 no。

direction

指定插件预期会采用什么方式与审计通讯。它可向事件调度程序告知事件的流动方向。可能的值为 in 或 out。

path

指定插件可执行文件的绝对路径。对于内部插件，此选项会指定插件名称。

type

指定运行插件的方式。可能的值为 builtin 或 always。对内部插件（af_unix 和 syslog）使用 builtin，对大多数（如果不是所有）其他插件使用 always。默认值为 always。

args

指定传递给插件程序的参数。正常情况下，插件程序将从其配置文件读取其参数，不需要接收任何参数。限制为两个参数。

format

指定审计调度程序传递给插件程序的数据格式。有效选项为 binary 或 string。binary 以事件调度程序从审计守护程序接收数据时的原有格式传递数据。string 指示调度程序将事件更改为可由审计分析库分析的字符串。默认值为 string。

例 40.10：示例 /ETC/AUDISP/PLUGINS.D/SYSLOG.CONF

```
active = no
direction = out
path = builtin_syslog
type = builtin
args = LOG_INFO
format = string
```

41 设置 Linux 审计框架

本章介绍如何设置一个简单的审计方案，其中会详细说明配置和启用审计所涉及的每个步骤。在了解如何设置审计后，请考虑第 42 章“[审计规则集简介](#)”中的真实示例方案。

要在 SUSE Linux Enterprise Server 上设置审计，需要完成以下步骤：

过程 41.1：设置 LINUX 审计框架

1. 确保已安装所有必需的软件包：[audit](#) 和 [audit-libs](#)，可以选择安装 [audit-libs-python](#)。要根据第 41.6 节“[配置日志可视化](#)”中所述使用日志可视化功能，请安装 SUSE Linux Enterprise Server 媒体中的 [gnuplot](#) 和 [graphviz](#)。
2. 确定要审计的组件。有关详细信息，请参考第 41.1 节“[确定要审计的组件](#)”。
3. 检查或修改基本审计守护程序配置。有关详细信息，请参考第 41.2 节“[配置审计守护程序](#)”。
4. 对系统调用启用审计。有关详细信息，请参考第 41.3 节“[对系统调用启用审计](#)”。
5. 根据您的方案撰写审计规则。有关详细信息，请参考第 41.4 节“[设置审计规则](#)”。
6. 生成日志并配置定制报告。有关详细信息，请参考第 41.5 节“[配置审计报告](#)”。
7. 配置可选的日志可视化。有关详细信息，请参考第 41.6 节“[配置日志可视化](#)”。

！ 重要：控制审计守护程序

在配置审计系统的任何组件之前，请以 `root` 身份输入 **`systemctl status auditd`**，以确保审计守护程序未运行。SUSE Linux Enterprise Server 系统默认会在引导时启动审计，因此您需要输入 **`systemctl stop auditd`** 将其关闭。配置守护程序后，使用 **`systemctl start auditd`** 将其启动。

41.1 确定要审计的组件

在开始创建您自己的审计配置之前，请确定您希望使用审计所达到的程度。检查以下一般规则，确定哪种用例最适合您和您的要求：

- 如果您需要进行全面的安全审计以通过 CAPP/EAL 认证，请对系统调用启用完全审计，并配置各个配置文件和目录的监测项，类似于第 42 章 “审计规则集简介” 中所述的规则集。
- 如果您需要根据审计规则跟踪进程，请使用 [autrace](#)。
- 如果您需要通过文件和目录监测项来跟踪对重要数据或安全敏感数据的访问，请创建符合这些要求的规则集。根据第 41.3 节 “对系统调用启用审计” 中所述启用审计，然后继续第 41.4 节 “设置审计规则”。

41.2 配置审计守护程序

审计守护程序的基本设置是通过编辑 `/etc/audit/auditd.conf` 完成的。您也可以通过调用 YaST > 安全和用户 > Linux 审计框架 (LAF) 来使用 YaST 配置基本设置。使用日志文件和磁盘空间选项卡完成配置。

```
log_file = /var/log/audit/audit.log
log_format = RAW
log_group = root
priority_boost = 4
flush = INCREMENTAL
freq = 20
num_logs = 5
disp_qos = lossy
dispatcher = /sbin/audispd
name_format = NONE
##name = mydomain
max_log_file = 6
max_log_file_action = ROTATE
space_left = 75
space_left_action = SYSLOG
action_mail_acct = root
admin_space_left = 50
admin_space_left_action = SUSPEND
disk_full_action = SUSPEND
disk_error_action = SUSPEND
##tcp_listen_port =
```

```
tcp_listen_queue = 5
tcp_max_per_addr = 1
##tcp_client_ports = 1024-65535
tcp_client_max_idle = 0
cp_client_max_idle = 0
```

默认设置适用于许多设置。某些值（例如 `num_logs`、`max_log_file`、`space_left` 和 `admin_space_left`）取决于您的部署大小。如果磁盘空间有限，您应该减少要保留的日志文件数（如果日志是轮换的）；当磁盘空间即将耗尽时，您应该提前收到警告。要实现符合 CAPP 规范的设置，请根据第 40.2 节“配置审计守护程序”中所述调整 `log_file`、`flush`、`max_log_file`、`max_log_file_action`、`space_left`、`space_left_action` 和 `disk_error_action` 的值。符合 CAPP 规范的示例配置如下所示：

```
log_file = PATH_TO_SEPARATE_PARTITION/audit.log
log_format = RAW
priority_boost = 4
flush = SYNC                ### or DATA
freq = 20
num_logs = 4
dispatcher = /sbin/audispd
disp_qos = lossy
max_log_file = 5
max_log_file_action = KEEP_LOGS
space_left = 75
space_left_action = EMAIL
action_mail_acct = root
admin_space_left = 50
admin_space_left_action = SINGLE  ### or HALT
disk_full_action = SUSPEND        ### or HALT
disk_error_action = SUSPEND       ### or HALT
```

注释的前面带有 `###`，您可以根据注释从多个选项中进行选择。请不要在实际的配置文件中添加注释。



提示：更多信息

有关 `auditd.conf` 配置参数的详细背景信息，请参见第 40.2 节“配置审计守护程序”。

41.3 对系统调用启用审计

如果未安装审计框架，请安装 `audit` 软件包。标准的 SUSE Linux Enterprise Server 系统默认不会运行 `auditd`。使用以下命令启用 `auditd`：

```
tux > sudo systemctl enable auditd
```

可用的审计活动有以下几种级别：

基本日志记录

原有的（未经过任何进一步的配置）`auditd` 仅在 `/var/log/audit/audit.log` 中记录与其自身配置更改相关的事件。内核审计组件在 `auditctl` 发出请求之前不会生成任何事件（文件访问、系统调用等）。但是，其他内核组件和模块可能会记录不在 `auditctl` 控制范围内的审计事件，而这些事件将显示在审计日志中。默认情况下，唯一生成审计事件的模块是 AppArmor。

具有系统调用审计功能的高级日志记录

要审计系统调用并获取有意义的文件监测项，需要对系统调用启用审计环境。

由于即使在配置普通文件或目录监测项时也需要使用系统调用审计功能，因此您需要对系统调用启用审计环境。要想仅在当前会话期间启用审计环境，请以 `root` 身份执行 `auditctl -e 1`。要禁用此功能，请以 `root` 身份执行 `auditctl -e 0`。

系统默认会启用审计环境。要暂时关闭此功能，请使用 `auditctl -e 0`。

41.4 设置审计规则

使用审计规则确定审计应分析系统的哪些方面。一般情况下，这包括重要数据库以及安全相关的配置文件。如果需要对系统进行广泛分析，您也可以详细分析各种系统调用。第 42 章“[审计规则集简介](#)”中提供了一个非常详细的示例配置，其中包括 CAPP 合规环境中所需的大多数规则。

可在 **auditctl** 命令行中或者通过在 `/etc/audit/audit.rules` 中撰写规则集（每当审计守护程序启动就会处理该规则集），将审计规则传递给审计守护程序。要自定义 `/etc/audit/audit.rules`，请直接对其进行编辑，或使用 YaST：安全和用户 > Linux 审计框架 (LAF) > “auditctl” 的规则。在命令行中传递的规则不会持久保留，重新启动审计守护程序后需要重新输入。

用于对少数几个重要文件和目录进行非常基本的审计的简单规则集如下所示：

```
# basic audit system parameters
-D
-b 8192
-f 1
-e 1

# some file and directory watches with keys
-w /var/log/audit/ -k LOG_audit
-w /etc/audit/auditd.conf -k CFG_audit_conf -p rxwa
-w /etc/audit/audit.rules -k CFG_audit_rules -p rxwa

-w /etc/passwd -k CFG_passwd -p rwx
-w /etc/sysconfig/ -k CFG_sysconfig

# an example system call rule
-a entry,always -S umask

### add your own rules
```

配置基本审计系统参数（例如积压参数 `-b`）时，请使用所需的审计规则集对这些设置进行测试，以确定积压大小是否适合审计规则集导致的日志记录活动的级别。如果选择的积压大小太小，您的系统可能无法处理审计负载，并无法在超出积压上限时查询故障标志 (`-f`)。



重要：选择故障标志

选择故障标志时请注意，在超出审计系统的限制时，`-f 2` 会告知系统不将任何等待中数据刷写到磁盘便立即关闭。由于这种关闭并非正常关闭，请仅对最注重安全的环境使用 `-f 2`，对任何其他环境使用 `-f 1`（系统继续运行并发出警告，审计停止），以避免数据丢失或损坏。

目录监测项生成的输出不如这些目录下各文件的单独文件监测项那样详细。例如，要想生成 `/etc/sysconfig` 中的系统配置的详细日志，请添加每个文件的监测项。审计不支持通配，这意味着，您无法创建 `-w /etc/*` 这样的规则来监测 `/etc` 下的所有文件和目录。

为便于在日志文件中识别，每个文件和目录监测项中都添加了一个键。使用该键可以更轻松地梳理日志，找到与特定规则相关的事件。创建键时，请将适当的前缀与键结合使用，以区分单纯的日志文件监测项和配置文件监测项。在本例中，LOG 表示日志文件监测项，CFG 表示配置文件监测项。使用文件名作为键的一部分也有助于您更轻松地识别此类事件。

创建文件和目录监测项时，要注意的另一点是，审计无法处理创建规则时尚不存在的文件。审计不会监测在其运行后添加到系统中的任何文件，除非您将规则集扩展为监测此新文件。

有关创建自定义规则的详细信息，请参见第 40.4 节“将参数传递到审计系统”。

！ 重要：更改审计规则

更改审计规则后，请始终使用 `systemctl restart auditd` 重新启动审计守护程序，以重新读取更改的规则。

41.5 配置审计报告

为了避免必须挖掘原始审计日志才能大致了解系统当前发生的情况，请按特定间隔运行自定义审计报告：自定义审计报告可让您将重点放在关注的方面，并获取有关所监视事件的性质和频率的有意义统计。要详细分析单个事件，请使用 `ausearch` 工具。

在设置审计报告之前，请考虑以下问题：

- 您要通过生成定期报告来监视哪种类型的事件？根据第 40.5.2 节“生成自定义审计报告”中所述选择适当的 `aureport` 命令行。
- 您要将审计报告用于什么目的？确定是否要基于累积数据创建图表，或者是否要将这些数据传送到任何类型的电子表格或数据库中。如果您要直观呈现报告，请按第 41.6 节“配置日志可视化”中所示示例的相似方法设置 `aureport` 命令行，并进行进一步处理。
- 要何时以及按何间隔运行报告？使用 `cron` 设置适当的自动化报告。

本示例假设您想要找出对您的审计、PAM 和系统配置进行的任何访问尝试。执行以下操作，找出系统上的文件事件：

1. 生成所有事件的完整摘要报告，并检查摘要报告中的任何异常情况，例如查看“failed syscalls”记录，因为这些活动失败的原因可能是文件访问权限不足，或者文件不存在：

```
tux > sudo aureport
```

Summary Report

=====

Range of time in logs: 03/02/09 14:13:38.225 - 17/02/09 16:30:10.352

Selected time for report: 03/02/09 14:13:38 - 17/02/09 16:30:10.352

Number of changes in configuration: 24

Number of changes to accounts, groups, or roles: 0

Number of logins: 9

Number of failed logins: 15

Number of authentications: 19

Number of failed authentications: 578

Number of users: 3

Number of terminals: 15

Number of host names: 4

Number of executables: 20

Number of files: 279

Number of AVC's: 0

Number of MAC events: 0

Number of failed syscalls: 994

Number of anomaly events: 0

Number of responses to anomaly events: 0

Number of crypto events: 0

Number of keys: 2

Number of process IDs: 1238

Number of events: 5435

2. 运行失败事件的摘要报告，并在“files”记录中检查文件访问失败事件的数目：

```
tux > sudo aureport --failed
```

Failed Summary Report

=====

Range of time in logs: 03/02/09 14:13:38.225 - 17/02/09 16:30:10.352

Selected time for report: 03/02/09 14:13:38 - 17/02/09 16:30:10.352

Number of changes in configuration: 0

```
Number of changes to accounts, groups, or roles: 0
Number of logins: 0
Number of failed logins: 15
Number of authentications: 0
Number of failed authentications: 578
Number of users: 1
Number of terminals: 7
Number of host names: 4
Number of executables: 12
Number of files: 77
Number of AVC's: 0
Number of MAC events: 0
Number of failed syscalls: 994
Number of anomaly events: 0
Number of responses to anomaly events: 0
Number of crypto events: 0
Number of keys: 2
Number of process IDs: 713
Number of events: 1589
```

3. 要列出无法访问的文件列表，请运行失败文件事件的摘要报告：

```
tux > sudo aureport -f -i --failed --summary

Failed File Summary Report
=====
total  file
=====
80  /var
80  spool
80  cron
80  lastrun
46  /usr/lib/locale/en_GB.UTF-8/LC_CTYPE
45  /usr/lib/locale/locale-archive
38  /usr/lib/locale/en_GB.UTF-8/LC_IDENTIFICATION
38  /usr/lib/locale/en_GB.UTF-8/LC_MEASUREMENT
38  /usr/lib/locale/en_GB.UTF-8/LC_TELEPHONE
38  /usr/lib/locale/en_GB.UTF-8/LC_ADDRESS
38  /usr/lib/locale/en_GB.UTF-8/LC_NAME
```

```

38 /usr/lib/locale/en_GB.UTF-8/LC_PAPER
38 /usr/lib/locale/en_GB.UTF-8/LC_MESSAGES
38 /usr/lib/locale/en_GB.UTF-8/LC_MONETARY
38 /usr/lib/locale/en_GB.UTF-8/LC_COLLATE
38 /usr/lib/locale/en_GB.UTF-8/LC_TIME
38 /usr/lib/locale/en_GB.UTF-8/LC_NUMERIC
8 /etc/magic.mgc
...

```

要让此摘要报告仅重点统计几个关注的文件或目录（例如 `/etc/audit/auditd.conf`、`/etc/pam.d` 和 `/etc/sysconfig`），请使用如下所示的命令：

```

tux > sudo aureport -f -i --failed --summary |grep -e "/etc/audit/auditd.conf" -e "/etc/pam.d/" -e "/etc/sysconfig"

1 /etc/sysconfig/displaymanager

```

4. 然后在摘要报告中继续隔离日志中的这些关注项，并找出其事件 ID 以进行进一步分析：

```

tux > sudo aureport -f -i --failed |grep -e "/etc/audit/auditd.conf" -e "/etc/pam.d/" -e "/etc/sysconfig"

993. 17/02/09 16:47:34 /etc/sysconfig/displaymanager readlink no /bin/vim-normal root 7887
994. 17/02/09 16:48:23 /etc/sysconfig/displaymanager getxattr no /bin/vim-normal root 7889

```

5. 使用事件 ID 获取每个关注项的详细记录：

```

tux > sudo ausearch -a 7887 -i
----
time->Tue Feb 17 16:48:23 2009
type=PATH msg=audit(1234885703.090:7889): item=0 name="/etc/sysconfig/displaymanager" inode=369282 dev=08:06 mode=0100644 ouid=0 ogid=0 rdev=00:00
type=CWD msg=audit(1234885703.090:7889): cwd="/root"
type=SYSCALL msg=audit(1234885703.090:7889): arch=c000003e syscall=191 success=no exit=-61 a0=7e1e20 a1=7f90e4cf9187 a2=7ffffed5b57d0 a3=84 items=1 ppid=25548 pid=23045 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0

```

```
egid=0 sgid=0 fsgid=0 tty=pts2 ses=1166 comm="vim" exe="/bin/vim-normal"
key=(null)
```



提示：侧重于特定的时间范围

如果您要关注特定时间段的事件，请在 **aureport** 命令（**-ts** 和 **-te**）中使用开始与结束日期和时间来裁减报告。有关更多信息，请参考第 40.5.2 节“生成自定义审计报告”。

除最后一个步骤以外的所有步骤均可自动运行，您可以轻松编写其脚本并将其配置为 cron 作业。任何 **--failed --summary** 报告均可轻松转换为标绘文件与失败访问尝试的条形图。有关直观呈现审计报告数据的详细信息，请参见第 41.6 节“配置日志可视化”。

41.6 配置日志可视化

您可以使用脚本 **mkbar** 和 **mkgraph** 通过各种图形和图表来说明您的审计统计。与任何其他 **aureport** 命令一样，您可以编写绘图命令脚本，并轻松将其配置为以 cron 作业的形式运行。

mkbar 和 **mkgraph** 是由 Red Hat 的 Steve Grubb 创建的。所在网址为 <http://people.redhat.com/sgrubb/audit/visualize/>。由于 SUSE Linux Enterprise Server 中当前版本的审计未随附这些脚本，请执行以下操作以在您的系统上提供这些脚本：



警告：下载的内容有风险

使用 **mkbar** 和 **mkgraph** 需自负风险。从 Web 下载的任何内容均有可能给您的系统造成危害，以 **root** 特权运行时更是如此。

1. 将脚本下载到 **root** 的 **~/bin** 目录：

```
tux > sudo wget http://people.redhat.com/sgrubb/audit/visualize/mkbar -O ~/bin/mkbar
tux > sudo wget http://people.redhat.com/sgrubb/audit/visualize/mkgraph -O ~/bin/mkgraph
```

2. 调整 **root** 的文件读取、写入和执行权限：

```
tux > sudo chmod 744 ~/bin/mk{bar,graph}
```

要绘制摘要报告（例如第 41.5 节“配置审计报告”中所述的报告），请使用 **mkbar** 脚本。某些示例命令如下所示：

创建事件摘要

```
tux > sudo aureport -e -i --summary | mkbar events
```

创建文件事件摘要

```
tux > sudo aureport -f -i --summary | mkbar files
```

创建登录事件摘要

```
tux > sudo aureport -l -i --summary | mkbar login
```

创建用户事件摘要

```
tux > sudo aureport -u -i --summary | mkbar users
```

创建系统调用事件摘要

```
tux > sudo aureport -s -i --summary | mkbar syscalls
```

要创建上述任何事件类型的失败事件摘要图表，请向相关的 **aureport** 命令添加 **--failed** 选项。要仅涵盖特定的时间段，请在 **aureport** 中使用 **-ts** 和 **-te** 选项。对于上述任何命令，可以使用 **grep** 或 **egrep** 以及正则表达式缩小其范围，来进一步对其进行调整。有关示例，请查看 **mkbar** 脚本中的注释。上述所有命令均会生成一个 PNG 文件，其中包含所请求数据的条形图。要说明不同类型的审计对象（例如用户和系统调用）之间的关系，请使用 **mkgraph** 脚本。某些示例命令如下所示：

用户与可执行文件

```
tux > sudo LC_ALL=C aureport -u -i | awk '/^[0-9]/ { print $4 " "$7 }' |  
sort | uniq | mkgraph users_vs_exec
```

用户与文件

```
tux > sudo LC_ALL=C aureport -f -i | awk '/^[0-9]/ { print $8" "$4 }' |  
sort | uniq | mkgraph users_vs_files
```

系统调用与命令

```
tux > sudo LC_ALL=C aureport -s -i | awk '/^[0-9]/ { print $4" "$6 }' |  
sort | uniq | mkgraph syscall_vs_com
```

系统调用与文件

```
tux > sudo LC_ALL=C aureport -s -i | awk '/^[0-9]/ { print $5" "$4 }' |  
sort | uniq | mkgraph | syscall_vs_file
```

还可以结合图形来说明复杂的关系。有关更多信息和示例，请查看 **mkgraph** 脚本中的注释。此脚本生成的图形默认创建为 PostScript 格式，但您可以通过将脚本中的 EXT 变量从 ps 更改为 png 或 jpg，来更改输出格式。

42 审计规则集简介

下面的示例配置说明如何使用审计来监视系统。其中重点指出了要涵盖受控访问保护配置文件 (CAPP) 指定的可审计事件列表而需审计的最重要的事项。

示例规则集分为以下几个部分：

- 基本审计配置（参见第 42.1 节 “添加基本审计配置参数”）
- 审计日志文件和配置文件监测项（参见第 42.2 节 “添加审计日志文件和配置文件监测项”）
- 监视对文件系统对象的操作（参见第 42.3 节 “监视文件系统对象”）
- 监视安全数据库（参见第 42.4 节 “监视安全配置文件和数据库”）
- 监视其他系统调用（第 42.5 节 “监视其他系统调用”）
- 过滤系统调用参数（参见第 42.6 节 “过滤系统调用参数”）

要将此示例转换为配置文件以在您的在线环境中使用，请执行以下操作：

1. 根据您的环境选择相应的设置并进行调整。
2. 通过添加以下示例中的规则或修改现有规则来调整文件 `/etc/audit/audit.rules`。



注意：调整审计日志记录级别

如未根据您的需要调整，请勿将下面的示例复制到您的审计环境中。确定审计内容和审计范围。

整个 `audit.rules` 是 `auditctl` 命令的集合。此文件中的每一行都将扩展为完整的 `auditctl` 命令行。规则集中使用的语法与 `auditctl` 命令的语法相同。

42.1 添加基本审计配置参数

```
-D ①  
-b 8192 ②  
-f 2 ③
```

- ❶ 在开始定义新规则之前，请删除所有以前存在的规则。
- ❷ 设置用于容纳审计消息的缓冲区数目。根据系统上的审计日志记录级别增大或减小此数字。
- ❸ 设置当内核需要处理严重错误时要使用的故障标志。可能的值为 0（静默）、1（printk，列显故障消息）和 2（恐慌，暂停系统）。

使用 -D 选项清空规则队列可以确保审计只使用您通过此文件向其提供的规则集，而不使用任何其他规则集。要避免系统由于审计负载过高而发生故障，选择适当的缓冲区数目 (-b) 至关重要。选择恐慌故障标志 -f 2 可确保即使系统遇到严重错误也能保持审计记录的完整。审计会在出现严重错误时关闭系统，确保不会有任何进程脱离审计的控制，而如果选择级别 1 (printk)，则可能会发生脱离控制的情况。

❗ 重要：选择故障标志

在在线系统上使用审计规则集之前，请确保在测试系统上使用**最差状况的生产工作负载**全面评估设置。如果指定了 -f 2 标志，那么这种做法将更加重要，因为这会指示内核在超过任何阈值时进入恐慌状态（不将等待中数据刷写到磁盘即执行立即暂停）。请仅对最注重安全的环境考虑使用 -f 2 标志。

42.2 添加审计日志文件和配置文件监测项

添加审计配置文件和日志文件本身的监测项可确保您能够跟踪任何尝试篡改配置文件的操作，或检测任何尝试访问日志文件的操作。

📁 注意：创建目录和文件监测项

如果您需要有关文件访问的事件，创建目录监测项不一定足够实现此目的。仅当保存元数据更改以更新目录的 inode 时，才会触发有关目录访问的事件。要触发有关文件访问的事件，请添加要监视的每个文件的监测项。

```
-w /var/log/audit/ ❶  
-w /var/log/audit/audit.log
```

```
-w /var/log/audit/audit_log.1
-w /var/log/audit/audit_log.2
-w /var/log/audit/audit_log.3
-w /var/log/audit/audit_log.4

-w /etc/audit/auditd.conf -p wa ❷
-w /etc/audit/audit.rules -p wa
-w /etc/libaudit.conf -p wa
```

- ❶ 设置审计日志所在目录的监测项。对任何类型访问此目录的尝试均触发事件。如果您在使用日志轮换，请另外也添加所轮换日志的监测项。
- ❷ 设置审计配置文件的监测项。记录对此文件的所有写入和属性更改尝试。

42.3 监视文件系统对象

审计系统调用有助于您跟踪高于应用程序级别的系统活动。通过跟踪文件系统相关的系统调用，可大致了解应用程序是如何使用这些系统调用的，并确定这种用法是否适当。通过跟踪装入和卸载操作来跟踪外部资源（可移动媒体、远程文件系统等）的使用。

❗ 重要：审计系统调用

审计系统调用会产生高负载日志记录活动，而此活动又会给内核带来繁重的负载。如果内核的响应能力低于正常水平，可能会超出系统的积压和速率上限。请仔细评估要在审计规则集中包含哪些系统调用，并相应地调整日志设置。有关如何优化相关设置的细节，请参见第 40.2 节“配置审计守护程序”。

```
-a entry,always -S chmod -S fchmod -S chown -S chown32 -S fchown -S fchown32 -S lchown -S lchown32 ❶

-a entry,always -S creat -S open -S truncate -S truncate64 -S ftruncate -S ftruncate64 ❷

-a entry,always -S mkdir -S rmdir ❸

-a entry,always -S unlink -S rename -S link -S symlink ❹
```

```
-a entry,always -S setxattr ❸  
-a entry,always -S lsetxattr  
-a entry,always -S fsetxattr  
-a entry,always -S removexattr  
-a entry,always -S lremovexattr  
-a entry,always -S fremovexattr  
  
-a entry,always -S mknod ❹  
  
-a entry,always -S mount -S umount -S umount2 ❺
```

- ❶ 对更改文件所有权和权限相关的系统调用启用审计环境。根据系统的硬件体系结构启用或禁用 *32 规则。AMD64/Intel 64 等 64 位系统要求去除 *32 规则。
- ❷ 对文件内容修改相关的系统调用启用审计环境。根据系统的硬件体系结构启用或禁用 *64 规则。AMD64/Intel 64 等 64 位系统要求去除 *64 规则。
- ❸ 对任何目录操作（例如创建或去除目录）启用审计环境。
- ❹ 对任何链接操作（例如创建符号链接、创建链接、取消链接或重命名）启用审计环境。
- ❺ 对扩展文件系统属性相关的任何操作启用审计环境。
- ❻ 对用于创建特殊（设备）文件的 mknod 系统调用启用审计环境。
- ❼ 对任何装入或卸载操作启用审计环境。对于 x86 体系结构，请禁用 umount 规则。对于 Intel 64 体系结构，请禁用 umount2 规则。

42.4 监视安全配置文件和数据库

为确保您的系统不会出现意外的行为，请跟踪任何尝试更改 cron 和 at 配置或已安排作业列表的操作。跟踪任何对用户、组、口令和登录数据库的写入访问有助于识别操控系统用户数据库的尝试。

跟踪对系统配置（内核、服务、时间等）的更改有助于识别他人操控系统关键功能的任何尝试。还应监视对安全环境中的 PAM 配置的更改，因为身份验证堆栈中的更改只能由管理员做出，并且应该记录哪些应用程序正在使用 PAM 及其使用方式。上面所述同样适用于与安全身份验证和通讯相关的任何其他配置文件。

①

```
-w /var/spool/atspool
-w /etc/at.allow
-w /etc/at.deny

-w /etc/cron.allow -p wa
-w /etc/cron.deny -p wa
-w /etc/cron.d/ -p wa
-w /etc/cron.daily/ -p wa
-w /etc/cron.hourly/ -p wa
-w /etc/cron.monthly/ -p wa
-w /etc/cron.weekly/ -p wa
-w /etc/crontab -p wa
-w /var/spool/cron/root
```

②

```
-w /etc/group -p wa
-w /etc/passwd -p wa
-w /etc/shadow

-w /etc/login.defs -p wa
-w /etc/securetty
-w /var/log/lastlog
```

③

```
-w /etc/hosts -p wa
-w /etc/sysconfig/
w /etc/init.d/
w /etc/ld.so.conf -p wa
w /etc/localtime -p wa
w /etc/sysctl.conf -p wa
w /etc/modprobe.d/
w /etc/modprobe.conf.local -p wa
w /etc/modprobe.conf -p wa
```

④

```
w /etc/pam.d/
```

⑤

```
-w /etc/aliases -p wa
```

```
-w /etc/postfix/ -p wa
```

⑥

```
-w /etc/ssh/sshd_config
```

```
-w /etc/stunnel/stunnel.conf
```

```
-w /etc/stunnel/stunnel.pem
```

```
-w /etc/vsftpd.ftpusers
```

```
-w /etc/vsftpd.conf
```

⑦

```
-a exit,always -S sethostname
```

```
-w /etc/issue -p wa
```

```
-w /etc/issue.net -p wa
```

- ① 设置 at 和 cron 配置及已安排作业的监测项，并向这些事件指派标签。
- ② 设置用户、组、口令、登录数据库和日志的监测项，并设置标签以更好地识别任何登录相关的事件，例如失败的登录尝试。
- ③ 对 /etc/hosts 中的静态主机名配置设置监测项和标签。跟踪对系统配置目录 /etc/sysconfig 的更改。如果您要跟踪文件事件，请启用每个文件的监测项。对 /etc/init.d 目录中的引导配置发生的更改设置监测项和标签。如果您要跟踪文件事件，请启用每个文件的监测项。对 /etc/ld.so.conf 中的链接器配置发生的任何更改设置监测项和标签。对 /etc/localtime 设置监测项和标签。对内核配置文件 /etc/sysctl.conf、/etc/modprobe.d/、/etc/modprobe.conf.local 和 /etc/modprobe.conf 设置监测项和标签。
- ④ 设置 PAM 配置目录的监测项。如果您要跟踪目录级别下的特定文件，还需添加这些文件的显式监测项。
- ⑤ 设置 postfix 配置的监测项，以在日志中记录任何写入尝试或属性更改，并使用标签来更好地进行跟踪。
- ⑥ 对 SSH、**stunnel** 和 **vsftpd** 配置文件设置监测项和标签。
- ⑦ 执行对 sethostname 系统调用的审计，并对 /etc/issue 与 /etc/issue.net 中的系统标识配置设置监测项和标签。

42.5 监视其他系统调用

除了根据第 42.3 节“监视文件系统对象”中所述审计文件系统相关的系统调用以外，您还可以跟踪其他各种系统调用。跟踪任务创建有助于了解应用程序的行为。审计 `umask` 系统调用可以跟踪进程是如何修改创建掩码的。跟踪任何尝试更改系统时间的操作有助于识别尝试操控系统时间的任何人或进程。

```
❶  
-a entry,always -S clone -S fork -S vfork  
  
❷  
-a entry,always -S umask  
  
❸  
-a entry,always -S adjtimex -S settimeofday
```

- ❶ 跟踪任务创建。
- ❷ 添加 `umask` 系统调用的审计环境。
- ❸ 跟踪尝试更改系统时间的操作。可以使用 `adjtimex` 来调整时间。`settimeofday` 会设置绝对时间。

42.6 过滤系统调用参数

除了第 42.3 节“监视文件系统对象”和第 42.5 节“监视其他系统调用”中介绍的系统调用审计以外，您还可以更深入地跟踪应用程序行为。应用过滤器有助于将审计重点放在您主要关注的方面。本节介绍如何过滤非多路转换系统调用（例如 `access`）和多路转换系统调用（例如 `socketcall` 或 `ipc`）的系统调用参数。系统调用是否会进行多路转换取决于所用的硬件体系结构。`socketcall` 和 `ipc` 在 AMD64/Intel 64 等 64 位体系结构上均不会进行多路转。

！ 重要：审计系统调用

审计系统调用会产生高负载的日志记录活动，而后者又会给内核带来繁重的负载。如果内核的响应能力低于正常水平，可能会远远超出系统的积压和速率上限。请仔细评估要在审计规则集中包含哪些系统调用，并相应地调整日志设置。有关如何优化相关设置的细节，请参见第 40.2 节“配置审计守护程序”。

access 系统调用会检查是否允许某个进程读取、写入文件或文件系统对象或者测试该对象是否存在。请使用 `-F` 过滤标志以 `-F a1=ACCESS_MODE` 格式构建与特定 access 调用匹配的规则。在 `/usr/include/fcntl.h` 中检查 access 系统调用的可能参数列表。

```
-a entry,always -S access -F a1=4 ❶  
-a entry,always -S access -F a1=6 ❷  
-a entry,always -S access -F a1=7 ❸
```

- ❶ 审计 access 系统调用，但仅当该系统调用的第二个参数 (mode) 为 4 (R_OK) 时会进行审计。此规则过滤所有用于测试对用户或进程所访问的文件或文件系统是否拥有足够读取权限的 access 调用。
- ❷ 审计 access 系统调用，但仅当该系统调用的第二个参数 (mode) 为 6 (表示 4 OR 2，相当于 R_OK OR W_OK) 时会进行审计。此规则过滤用于测试是否拥有足够读取和写入权限的 access 调用。
- ❸ 审计 access 系统调用，但仅当该系统调用的第二个参数 (mode) 为 7 (表示 4 OR 2 OR 1，相当于 R_OK OR W_OK OR X_OK) 时会进行审计。此规则过滤用于测试是否拥有足够读取、写入和执行权限的 access 调用。

socketcall 系统调用是多路转换系统调用。多路转换是指在所有可能的调用中只存在一个系统调用，并且 libc 会传递实际的系统调用作为第一个参数 (a0)。有关可能的系统调用，请查看 socketcall 的手册页；有关可能的参数值和系统调用名称的列表，请参见 `/usr/src/linux/include/linux/net.h`。审计支持使用 `-F a0=SYSCALL_NUMBER` 过滤特定的系统调用。

```
-a entry,always -S socketcall -F a0=1 -F a1=10 ❶  
## Use this line on x86_64, ia64 instead  
#-a entry,always -S socket -F a0=10  
  
-a entry,always -S socketcall -F a0=5 ❷
```

```
## Use this line on x86_64, ia64 instead
#-a entry, always -S accept
```

- ① 审计 socket(PF_INET6) 系统调用。-F a0=1 过滤器会匹配所有 socket 系统调用，-F a1=10 过滤器可将匹配范围缩小为传递 IPv6 协议系列域参数 (PF_INET6) 的 socket 系统调用。有关第一个参数 (a0)，请查看 [/usr/include/linux/net.h](#)；有关第二个参数 (a1)，请查看 [/usr/src/linux/include/linux/socket.h](#)。AMD64/Intel 64 等 64 位平台不会对 socketcall 系统调用使用多路转换。对于这些平台，请将规则注释掉，并添加对 PF_INET6 进行过滤的普通系统调用规则。
- ② 审计 socketcall 系统调用。过滤标志设置为过滤 a0=5（socketcall 的第一个参数），如果您检查 [/usr/include/linux/net.h](#)，会发现此设置转换为 accept 系统调用。AMD64/Intel 64 等 64 位平台不会对 socketcall 系统调用使用多路转换。对于这些平台，请将规则注释掉，并添加不含参数过滤的普通系统调用规则。

ipc 系统调用是多路转换系统调用的另一个示例。要调用的实际调用由传递给 ipc 系统调用的第一个参数决定。过滤这些参数有助于您将重点放在要关注的 IPC 调用上。在 [/usr/include/linux/ipc.h](#) 中查看可能的参数值。

```
①
## msgctl
-a entry,always -S ipc -F a0=14
## msgget
-a entry,always -S ipc -F a0=13
## Use these lines on x86_64, ia64 instead
#-a entry,always -S msgctl
#-a entry,always -S msgget

②
## semctl
-a entry,always -S ipc -F a0=3
## semget
-a entry,always -S ipc -F a0=2
## semop
-a entry,always -S ipc -F a0=1
## semtimedop
-a entry,always -S ipc -F a0=4
## Use these lines on x86_64, ia64 instead
```

```
#-a entry,always -S semctl
#-a entry,always -S semget
#-a entry,always -S semop
#-a entry,always -S semtimedop
```

③

```
## shmctl
-a entry,always -S ipc -F a0=24
## shmget
-a entry,always -S ipc -F a0=23
## Use these lines on x86_64, ia64 instead
#-a entry,always -S shmctl
#-a entry,always -S shmget
```

- ① 审计与 IPC SYSV 消息队列相关的系统调用。在本例中，a0 值指定要针对 `msgctl` 和 `msgget` 系统调用（14 和 13）添加审计。AMD64/Intel 64 等 64 位平台不会对 `ipc` 系统调用使用多路转换。对于这些平台，请将前两条规则注释掉，并添加不含参数过滤的普通系统调用规则。
- ② 审计与 IPC SYSV 消息信号相关的系统调用。在本例中，a0 值指定要针对 `semctl`、`semget`、`semop` 和 `semtimedop` 系统调用（3、2、1 和 4）添加审计。AMD64/Intel 64 等 64 位平台不会对 `ipc` 系统调用使用多路转换。对于这些平台，请将前四条规则注释掉，并添加不含参数过滤的普通系统调用规则。
- ③ 审计与 IPC SYSV 共享内存相关的系统调用。在本例中，a0 值指定要针对 `shmctl` 和 `shmget` 系统调用（24 和 23）添加审计。AMD64/Intel 64 等 64 位平台不会对 `ipc` 系统调用使用多路转换。对于这些平台，请将前两条规则注释掉，并添加不含参数过滤的普通系统调用规则。

42.7 使用键管理审计事件记录

配置了一些会生成事件的规则并填充日志后，您需要找到一种方法来辨别不同的事件。使用 **`ausearch`** 命令可以根据不同的准则过滤日志。通过使用 **`ausearch -m MESSAGE_TYPE`**，您至少可以过滤特定类型的事件。但是，要过滤与特定规则相关的事件，需要在 `/etc/audit/`

`audit.rules` 文件中将一个键添加到此规则。然后，每次该规则记录一个事件时，此键就会添加到相应事件记录。要检索这些日志项，只需运行 `ausearch -k YOUR_KEY` 获取与该规则相关且带有此特定键的记录列表。

例如，假设您已将下面的规则添加到规则文件：

```
-w /etc/audit/audit.rules -p wa
```

如果未向该规则指派键，您可能需要过滤 `SYSCALL` 或 `PATH` 事件，然后使用 `grep` 或类似工具来隔离与上述规则相关的所有事件。现在，使用 `-k` 选项将一个键添加到上述规则：

```
-w /etc/audit/audit.rules -p wa -k CFG_audit.rules
```

您可以指定任何文本字符串作为键。使用不同的键前缀（`CFG`、`LOG` 等）后接文件名来区分与不同文件类型（配置文件或日志文件）相关的各监测项。现在，可按如下所示查找与上述规则相关的所有记录：

```
ausearch -k CFG_audit.rules
```

```
----
```

```
time->Thu Feb 19 09:09:54 2009
```

```
type=PATH msg=audit(1235030994.032:8649): item=3 name="audit.rules~"
```

```
inode=370603 dev=08:06 mode=0100640 ouid=0 ogid=0 rdev=00:00
```

```
type=PATH msg=audit(1235030994.032:8649): item=2 name="audit.rules" inode=370603
```

```
dev=08:06 mode=0100640 ouid=0 ogid=0 rdev=00:00
```

```
type=PATH msg=audit(1235030994.032:8649): item=1 name="/etc/audit" inode=368599
```

```
dev=08:06 mode=040750 ouid=0 ogid=0 rdev=00:00
```

```
type=PATH msg=audit(1235030994.032:8649): item=0 name="/etc/audit" inode=368599
```

```
dev=08:06 mode=040750 ouid=0 ogid=0 rdev=00:00
```

```
type=CWD msg=audit(1235030994.032:8649): cwd="/etc/audit"
```

```
type=SYSCALL msg=audit(1235030994.032:8649): arch=c000003e syscall=82
```

```
success=yes exit=0 a0=7deeb0 a1=883b30 a2=2 a3=ffffffffffffffff items=4
```

```
ppid=25400 pid=32619 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0
```

```
sgid=0 fsgid=0 tty=pts1 ses=1164 comm="vim" exe="/bin/vim-normal"
```

```
key="CFG_audit.rules"
```

43 有用资源

我们提供的一些其他资源包含了有关 Linux 审计框架的有用信息：

审计手册页

随审计工具一并安装的多个手册页提供了非常详细的有用信息：

auditd(8)

Linux 审计守护程序

auditd.conf(5)

Linux 审计守护程序配置文件

auditctl(8)

用于帮助控制内核审计系统的实用程序

autrace(8)

与 **strace** 类似的程序

ausearch(8)

用于查询审计守护程序日志的工具

aureport(8)

用于生成审计守护程序日志摘要报告的工具

audispd.conf(5)

审计事件调度程序配置文件

audispd(8)

与插件程序通讯的审计事件调度程序守护程序。

<http://people.redhat.com/sgrubb/audit/index.html> 

Linux 审计项目的主页。此网站包含与 Linux 审计的不同方面相关的多个规范以及若干常见问题。

/usr/share/doc/packages/audit

审计包本身包含一个提供了基本设计信息的 README 文件以及一些适用于不同方案的示例 .rules 文件：

capp.rules: 受控访问保护配置文件 (CAPP)

lspp.rules: 标记安全保护配置文件 (LSPP)

nispom.rules: 国家工业安全计划操作手册第 8 章 (NISPOM)

stig.rules: 安全技术实施指南 (STIG)

<https://www.commoncriteriaportal.org/> 

通用准则项目的官方网站。全面了解通用准则安全认证计划，以及审计在此框架中所起的作用。

A 实现 PCI DSS 合规

为了保护客户和企业自身，处理信用卡付款的公司必须尽最大努力确保数据安全无虞。遵循支付卡行业数据安全标准有助于保护与付款流程相关的所有方面，以及实施安全相关的措施来确保数据和计算环境的安全。

本文档旨在帮助您基本了解如何配置，以符合支付卡行业数据安全标准。

请务必注意，保护系统的工作不仅涉及到配置，还要考虑到相关的整个环境和所有人员。

实施 PCI DSS 的一个重要组成部分是各种操作的组合：

1. 创建安全配置。
2. 跟踪并审查对配置进行的所有更改：谁在哪个时间点更改了什么配置。

A1 什么是 PCI DSS?

支付卡行业数据安全标准 (PCI DSS) 是指导商家保护持卡人数据的一套要求。该标准涵盖当前涉及 12 个要求主题的六个主要类别，规定如何实施、保护、维护和监视在信用卡持卡人数据处理中所涉及的系统。

PCI DSS 由 PCI 安全标准理事会 (SSC) 制定和维护，该理事会由 Visa、MasterCard、American Express、Discover 和 JCB 五大信用卡机构创立。2004 年 12 月发布了 PCI DSS 1.0，目的是解决日益猖獗的在线信用卡欺诈威胁。最新版本 PCI DSS 3.2 发布于 2016 年 4 月。

构建和维护安全网络与系统

1. 安装并维护防火墙配置以保护持卡人数据
2. 不要使用供应商为系统口令和其他安全参数提供的默认值

保护持卡人数据

3. 保护储存的持卡人数据
4. 对在开放的公共网络上传输的持卡人数据进行加密

维护漏洞管理程序

5. 保护所有系统免遭恶意软件的攻击，并定期更新防病毒软件或程序
6. 开发并维护安全系统和应用程序

实施严格的访问控制措施

7. 限制企业仅可访问其需要知道的持卡人数据
8. 识别并验证对系统组件的访问
9. 限制对持卡人数据的物理访问

定期监视和测试网络

10. 跟踪并监视对网络资源和持卡人数据的所有访问
11. 定期测试安全系统和流程

维护信息安全策略

12. 维护用于处理所有个人信息安全性的策略

PCI DSS 的大多数要求都是针对组织的指导原则，可帮助确保涉及持卡人数据的方方面面的安全性。对于技术方面，通常不会使用具体的措辞。

这意味着，需要由审计人员确定哪些安全性设置符合要求，哪些不符合要求。因此，本文档中的建议只能为实施 PCI DSS 提供一个入手点，需要进一步探讨。

A2 本文档的重点：与操作系统相关的方面

PCI DSS 中的许多方面都与持卡人数据相关。其中并非所有方面都与操作系统相关，本文档不会着重说明这些不相关的方面，而是重点介绍影响操作系统配置的方面，包括：

- 系统安全性
- 访问控制
- 旨在防范已知漏洞的系统维护

以下主题不在本文档的范畴内：数据处理应用程序、数据库设计，以及不属于操作系统范围的正式流程。具体而言，本文档不会详细讨论要求 9（限制物理访问）和要求 12（维护策略）。

A3 要求详细说明

下面的章节将按标准本身的顺序详细介绍 PCI DSS 的相关部分。

A3.1 要求 1：安装并维护防火墙配置以保护持卡人数据

本节列出的条款主要是设计、文档和正式流程方面的要求。对防火墙和路由器进行的所有更改均需经过审批、书面记录和校验，并且需要知会所有利益相关者。网络设计包括 DMZ 环境、互联网访问、受保护的数据库服务器网络、网段之间的流量过滤规则，等等。

除了专用的防火墙和路由器以外，还随附了基于 iptables 的主机防火墙。可以轻松将系统配置为仅允许特定入站端口上的连接通过。使用 YaST 防火墙模块还可以定义更复杂的规则。例如，禁止不是来自特定网络地址的连接。这样便可以将本地系统防火墙整合到一个能够最大限度提高网络安全性的总体防火墙设计中。

概括而言，要求 1 中的技术要点如下：

- 识别不安全的服务和协议。
- 限制出入系统的流量，以便直接避免不需要的和有害的流量。

1.1.6.b 识别所允许的不安全服务、协议和端口；校验是否书面记录了每项服务的安全功能

此任务嵌入在有关识别、记录系统上运行的所有服务和协议并证明其合理性的要求中。需要特殊关注可能导致安全风险的服务和协议。如果使用不安全的服务或协议，必须对其进行评估，以了解它的潜在安全影响。应该禁用或去除业务运营中不必要的服务或协议。

1.2.1.b 校验入站和出站流量是否限制为持卡人数据环境中所必需的流量

应该仅在具体指明的情形中允许出站流量。要允许特定的流量，需手动添加出站流量的规则。

在可能的情况下，将 SSH 守护程序限制为只能通过管理接口访问，而不能通过一般的网卡访问。定义服务允许的流量来源地址。

例如，要仅允许出站 DNS 请求通过接口 `eth0` 发往服务器 `10.0.0.1`，请使用：

```
root # firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 23 \
      -d 10.0.0.1/32 -o eth0 -p udp -m udp --dport 53 -j ACCEPT
root # firewall-cmd --reload
```

要阻止所有其他出站流量，请参见 [1.2.1.c 校验是否明确拒绝所有其他入站和出站流量](#)。

1.2.1.c 校验是否明确拒绝所有其他入站和出站流量

拒绝未根据上一节中所述定义其例外情况的所有出站和入站流量。转发通常已由某个内核参数完全禁用，且不应在端点服务器启用。

中的 `firewalld` 默认会阻止所有入站流量。

要阻止所有出站流量，请手动添加以下规则：

```
root # firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 0 -m
      conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
root # firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 0 -m
      conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
root # firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 99 -
      j DROP
root # firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 99 -
      j DROP
root # firewall-cmd --reload
```

此外，还可以通过 TCP 封装程序配置文件 `/etc/hosts.deny` 为特定的服务配置入站流量。

下面的大多数任务涉及到检查并校验定义的入站和出站规则是否真正将所有网段（例如 DMZ 和互联网）之间及其内部的流量限制为整个系统能够正常运行所需的最小必要程度。

1.3.3 实施防假冒措施来检测并阻止伪造的源 IP 地址进入网络

在中可通过两种方式实施防假冒措施：

- **使用 `iptables` 规则。** 这些规则仅允许来自指定接口上特定地址的输入：可以在系统设置中明确定义用于通讯的地址空间。使用违反这些定义的地址的任何行为可以记录到日志中并触发警报。
- **Linux 内核反向路径过滤：** 此功能会丢弃那些通过不同于初始包的接口的包答复。此功能在中默认已启用，可使用以下命令检查其启用状态：

```
tux > cat /proc/sys/net/ipv4/conf/all/rp_filter
```

如果已启用，此命令会返回 `1`。

1.3.5 仅允许“已建立的”连接进入网络

`firewalld` 通过 `iptables` 启用连接跟踪。默认会丢弃与标记为外部的接口所建立的连接。只允许与已建立的连接相关联的连接。

可以定义允许哪些服务连接到外部接口。但是，这种定义必须符合常规安全策略。

请记住，防御来自互联网的恶意连接的第一道防线应该是将会处理所有流量并充当把关者的专用防火墙系统。不需要的连接应该一律不能进入 DMZ 网络。不过，系统上的简单防火墙规则可帮助避免错误配置，充当另一道防线。

1.3.7 不要向未获授权方透露私用 IP 地址和路由信息

系统还可充当路由器，将来自一个接口的流量转发到另一个接口上的另一个网络。可以在外部接口上使用网络地址转换 (NAT)，这样便不会真正向外部公开内部 IP 地址。这种做法的目的是减少外部攻击者只需分析网络流量就能收集到的信息。还可以在通过特定接口连接到外部的虚拟化主机或基于容器的环境中使用 NAT。

A3.2 要求 2：不要使用供应商为系统口令和其他安全参数提供的默认值

在安装期间，管理员已设置一般系统口令。该设置还会使用口令检查器 (**cracklib**) 根据某个字典来识别输入的弱口令。这意味着，标准配置已包含客户为大多数服务定义的安全选项。

有关操作系统安全性的详细信息，请参见

2.1 在网络中安装系统之前始终先更改供应商提供的默认值，并去除或禁用不必要的默认帐户。

必须评估所有系统服务的配置是否符合所需的安全标准。这包括将所用的协议限制为仅允许当前安全的版本并禁用旧版实现，以及定义访问控制和身份验证。的默认设置已经能够提供良好的总体安全性，但还可以进一步优化。

例如，以下安全性设置可能与此相关：

- 默认情况下，SNMP 守护程序仅允许将传入请求发送到 `localhost`。但是，默认的社区字符串命名为 `public`，应在接受常规入站连接之前加以更改。
- 默认情况下，`sshd` 守护程序的某些不安全上游设置已在 `sshd` 配置文件 `/etc/ssh/sshd_config` 中列出并注释掉。例如，已禁用不安全的协议版本 1 和空口令 (`PermitEmptyPasswords no`)。

要进一步提高 SSH 安全性，请通过将 `PermitRootLogin` 设置为 `no` 来拒绝直接的 `root` 访问（如果适用）。

可以通过使用 AutoYaST 配置文件自动执行系统安装来自定义默认设置。这样便可以发布新的实例并自动启用已经过评估的配置。可以使用 SUSE Manager 实现此设置过程的自动化。有关详细信息，请参见 <https://documentation.suse.com/suma/> 上的 SUSE Manager 文档。

默认情况下，不会创建除 root 管理用户以外的其他帐户。`/etc/passwd` 中定义了一些系统帐户，但这些帐户未激活，因此不能直接使用。可以通过检查 `/etc/shadow` 文件中的内容来验证这一点。

在该文件中，第二列表示定义的口令：

- 星号 (*) 表示从未定义某个口令，因此已锁定相应帐户。
- 感叹号 (!) 表示已锁定的帐户，可能会独行显示，也可能显示在口令哈希的前面。

2.2 为所有系统组件制定配置标准。确保这些标准能够解决所有已知安全漏洞，并与行业接受的系统强化标准相一致。

如 PCI DSS 文档中所述，行业接受的安全强化标准的可能来源包括：

1. 互联网安全中心 (CIS)
2. 国际标准化组织 (ISO)
3. SysAdmin 审计网络安全 (SANS) 协会
4. 美国国家标准技术研究院 (NIST)

由于未明确规定 PCI DSS 要求，安全强化标准与具体要求之间没有直接的关系。不过，其他安全强化资源也可为符合这些规范提供帮助，其中包括

2.2.1 为每台服务器仅实施一项主要功能，以防止需要不同安全级别的功能在同一台服务器上共存。（例如，应在不同的服务器上实施 Web 服务器、数据库服务器和 DNS。）

要帮助隔离服务，可以使用 随附的各种虚拟化和容器化方法：KVM、Xen、LXC 和 Docker。

还可以在 VMware ESX 或 Microsoft Hyper-V 等第三方虚拟化服务器上运行 来实现服务隔离。

使用 内置的选项时，请参见：

- 有关虚拟化的信息，请参见《》。
- 有关容器化的信息，请参见《》（SUSE Linux Enterprise Server Docker 开源引擎指南）。

2.2.2 仅启用正常运行系统所必需的服务、协议、守护程序等。

此条款与要求 1 中的某个条款直接相关：仅允许真正需要且在使用安全协议和设置的服务（1.1.6.b 识别所允许的不安全服务、协议和端口；校验是否书面记录了每项服务的安全功能）。所有相关方必须清楚使用不安全通讯存在的风险。研究、明确书面记录并传达使用不安全协议和服务所带来的风险。

使用以下 **systemctl** 命令启用和禁用系统服务：

- `tux > systemctl status SERVICE`
- `tux > sudo systemctl enable SERVICE`
- `tux > sudo systemctl disable SERVICE`

要列出系统上安装的所有可用服务及其状态，请使用以下命令：

```
tux > systemctl list-unit-files --type=service
```

2.2.3.a 检查配置设置，以校验是否书面记录并实施了针对所有不安全服务、守护程序或协议的安全功能。

要对不安全的服务添加额外的安全层，请使用 VPN 隧道（例如 IPsec）。使用 VPN 隧道可以隔离此类服务的网络流量，并防范所有数据遭到内部和外部窃听。但请注意，VPN 隧道端点上的通讯仍不安全，隧道只能作为一种因应措施。

要在内部提高安全性，请使用 SELinux 或 AppArmor。不过，这些框架的设置不在本文档的范畴内。

- 有关 SELinux 的信息，请参见
- 有关 AppArmor 的信息，请参见

2.2.5.a 选择系统组件的样本并检查配置，以校验所有不需要的功能（例如脚本、驱动程序、功能、子系统、文件系统等）是否已去除。

Linux 内核是主要系统组件。它包括一个核心映像，根据硬件和系统设计装载的内核模块会对该映像进行扩展。例如：会根据系统的网卡自动装载网卡驱动程序。可以启用文件系统模块来扩展 Linux 内核的文件系统支持。

装载的内核模块的列表通常很长，其中包含了偶尔才使用的模块。内核模块框架允许将模块加入黑名单，以及限制要装载的功能。

要阻止装载模块，请通过 `/etc/modprobe.d` 目录配置这些模块。例如，只有配备软盘驱动器的系统才需要内核模块 `floppy`。在没有软盘驱动器的系统上，可以阻止装载该模块：创建包含以下内容的配置文件 `/etc/modprobe.d/00-disable-modules.conf`：

```
install floppy /bin/true
```

`floppy` 模块通常是在初始 RAM 磁盘执行期间装载的。因此，请使用 `mkinitrd` 脚本将此项配置更改传播到 `initrd` 文件。

```
tux > sudo mkinitrd
```

去除或限制应用程序功能会更困难，因为大多数情况下，功能已编译到应用程序或库本身之中。甚至通过删除文件也不一定能够干净地去除功能：如果该文件是从某个 RPM 软件包安装的，更新该软件包后就会重新安装该文件。

2.3 使用强加密来加密所有非控制台管理访问。使用 SSH、VPN 或 TLS 等技术进行基于 Web 的管理和其他非控制台管理访问。

加密所有管理网络访问：选择的手段应该是 SSH 以及符合安全理念的适当配置设置。

管理访问权限也可以通过网站授予。在这种情况下，必须对浏览器与服务器系统之间的完整连接链进行加密。可以通过 TLS 和 X.509 证书实现此目的。

A3.3 要求 3：保护存储的持卡人数据

本节说明如何安全处理持卡人和身份验证数据。以下定义适用：

- **持卡人数据**包括持卡人姓名和主帐号 (PAN) 等信息。
- **身份验证数据**包括个人识别号 (PIN) 和卡验证码 (CVC2)。

持卡人数据和身份验证数据之间的主要差别在于，绝对不允许储存身份验证数据。相比之下，PAN 等数据是可以储存的，但必须经过加密且不可读，以防攻击者访问这些储存的数据。用于储存持卡人数据的数据库设计不在本文档的范畴内。不过，您可通过不同的方式加密数据：

- DBMS 可以在数据库模式中使用列级加密。
- 或者，可以加密数据库文件。
- 支持全盘加密，因此始终会加密整个数据库储存区。不过，访问加密磁盘的方式与访问非加密磁盘的方式相同。要求 3.4.1 中对此进行了详细介绍。

3.4.1.a 如果使用磁盘加密，请检查配置并观察身份验证流程，以校验对加密文件系统的逻辑访问是否是通过一种与本机操作系统身份验证机制不同的机制（例如，不使用本地用户帐户数据库或一般的网络登录身份凭证）实现的。

PCI DSS 文档的指导说明对此项要求的规定如下：“全盘加密有助于在磁盘实物丢失时保护数据，因此可能适合对储存持卡人数据的便携式设备使用。”

从管理员的角度而言，使用 Linux 统一密钥设置 (LUKS)/dm-crypt 实现的块设备加密可提供一个抽象层，通过该抽象层可以像使用未加密磁盘那样使用加密磁盘。

因此，只能使用文件系统提供的一般 ACL 权限来限制访问控制。要符合此要求，所用的解密密钥不得与任何一般登录身份凭证或身份验证方法相关联。

使用 LUKS 通常可以满足此要求：引导、插入便携式设备或手动挂载磁盘时需要单独输入口令。

LUKS 已完全集成到 中，可以通过 YaST 使用它来创建新分区。

3.4.1.c 检查配置并观察流程，以校验可移动媒体上的持卡人数据是否无论储存在何处都会加密。

如 3.4.1.a 如果使用磁盘加密，请检查配置并观察身份验证流程，以校验对加密文件系统的逻辑访问是否是通过一种与本机操作系统身份验证机制不同的机制（例如，不使用本地用户帐户数据库或一般的网络登录身份凭证）实现的。中所述，LUKS/dm-crypt 提供的全盘加密可以满足此项要求。用户只能通过挂载磁盘时必须输入的解密口令来访问储存的数据。

A3.4 要求 4：对在开放的公共网络上传输的持卡人数据进行加密

在通过不安全的网络传输持卡人数据时必须对这些数据进行加密。理想情况下，应在外部和内部加密所有流量。这样，攻击者便很难获取内部信息以及对持卡人数据环境的特权访问权限。

4.1 通过开放的公共网络传输敏感的持卡人数据期间，使用强加密和安全协议（例如 TLS、IPSEC、SSH 等）保护这些数据，具体措施包括：(1) 仅接受可信的密钥和证书；(2) 使用的协议仅支持安全版本或配置；(3) 加密强度适合所用的加密方法。

必须防范传输敏感信息的连接遭到窃听和篡改。

对于传入的客户端请求，请结合使用 HTTPS 协议与安全的 TLS 连接。使用 X.509 公共证书进行身份验证，该证书在一定程度上能够证实服务器正是客户要访问的那个端点。

随附了一组可让 HTTPS 连接受到保护的服务和工具。例如，可以直接使用 Apache HTTP Server 或通过 **stunnel**（充当代理来提供 TLS 加密功能）提供这种保护。

可以使用 IPsec 或其他 VPN 技术来保护通过公共网络连接的网段之间的连接。还可以使用 X.509 公共证书保护此类连接。对于内部用途，可以使用私用证书颁发机构 (CA) 来为 X.509 证书签名以及跟踪可信密钥。

在中，可以直接通过 strongSwan（一个基于 IPsec 的 VPN 解决方案）或通过 OpenVPN（使用自定义安全协议）来确保做到这一点。

要管理操作系统，请使用 SSH。有关配置 SSH 以提供更高安全性的信息，请参见第 A3.1 节“要求 1：安装并维护防火墙配置以保护持卡人数据”和第 A3.2 节“要求 2：不要使用供应商为系统口令和其他安全参数提供的默认值”。

A3.5 要求 5：保护所有系统免遭恶意软件的攻击，并定期更新防病毒软件或程序

要符合 PCI DSS 规范，需要防御恶意软件的攻击。可以使用主流防病毒软件供应商提供的第三方防病毒软件，并将其集成到 Linux 环境中。随附了开源的防病毒引擎 ClamAV。

ClamAV 可提供一组有限的扫描功能，与第三方产品相比性能有限。因此，ClamAV 预期只能提供基本保护。

另一方面，由于 ClamAV 是随附的，在安装服务器期间可将其一并安装。这样就可以轻松满足此要求，但也需要清楚地知道它与第三方产品相比存在的缺点。

A3.6 要求 6：开发并维护安全系统和应用程序

此项要求的主要部分涉及到内部软件开发、文档和设计问题，超出了本文档的范畴。不过，提供了帮助确保系统安全的工具：

- 软件包管理器 Zypper 是的一个强大工具。除其他众多功能外，它还能解析软件包、产品、模式和补丁的依赖关系，具有一项锁定机制用于防止安装软件包，并提供了一个完整的更新堆栈，用于确保系统处于最新状态并使其能够防范已知安全问题。

zypper 是所有 安装的组成部分，当系统注册之后便可直接访问更新储存库。

有关 Zypper 的信息，请参见

- SUSE 提供了 SUSE Manager 用于进行系统管理，该程序提供了有效的方法确保系统处于最新状态。它提供 和 Red Hat Enterprise Linux 客户端系统的无缝管理。在大型系统环境中，当您需要检查每个系统的当前更新状态以及需要应对已知安全问题时，这一点极为有用。

有关 SUSE Manager 的信息，请参见 [SUSE Manager 文档页面 \(https://documentation.suse.com/suma/\)](https://documentation.suse.com/suma/)。

6.2.a 检查与安全补丁安装相关的策略和过程，以校验是否定义了以下操作的流程：(1) 在供应商发布适用关键安全补丁后的一个月内安装这些补丁；(2) 在适当时间范围内（例如三个月内）安装供应商提供的所有适用安全补丁。

要识别需要安装以保护系统安全的补丁，请执行以下操作：

首先刷新所有软件储存库，以获得最新信息：

```
tux > sudo zypper refresh
```

然后使用 Zypper 的补丁相关命令：

- 搜索尚未安装的重要安全修复程序：

```
tux > zypper list-patches --category security --severity important
```

- 也可以搜索 CVE 或 SUSE Bugzilla 编号。默认情况下，此命令只会列出必要的补丁。要同时显示已安装的补丁，请使用参数 --all：

```
tux > zypper list-patches --all --cve=CVE-2016-4957
```

- 要列出单个补丁的细节，请使用 patch-info 子命令：

```
tux > zypper patch-info SUSE-SLE-SERVER-12-SP1-2016-600
```

- 要仅安装重要的安全补丁，请使用 patch 子命令：

```
tux > sudo zypper patch --category security --severity important
```

要自动执行更新，可以使用所有 Zypper 子命令都支持的 `--non-interactive` 参数。
有关 Zypper 的详细信息，请参见

A3.7 要求 7：限制企业仅可访问其需要知道的持卡人数据

操作系统访问控制是个复杂的主题。同样，此项 PCI DSS 要求并无明确规定，也未具体指出需要实施哪种程度的限制。随附了用于限制对特定系统区域和组件的访问的所有常规 Linux 工具：

- 可以使用传统的 Unix 权限设置通过特定的用户和用户组来控制访问。
有关管理权限的信息，请参见
- 一种适用于文件系统的更灵活的机制是访问控制列表 (ACL)，它可提供更精细的控制方式。SELinux 能够实现最高的系统隔离性，并可防止进程获得超出允许范围的资源和访问权限。SELinux 和 AppArmor 不在本文档的范畴内，但应该采用它们来保护可能会被攻击者针对的关键系统。
 - 有关 SELinux 的信息，请参见
 - 有关 AppArmor 的信息，请参见

7.1.2 将特权用户 ID 的访问权限限制为履行职责所必需的最低特权。

标准 Unix 权限允许为用户和组 ID 设置 Read、Write 和 Execution 标志。名为 `others` 或 `world` 的常规组定义了不适合加入前两个组的用户的访问权限。这提供了一种简单直接的方法来授予或拒绝对文件系统资源的访问权限。

ACL 提供了额外的限制级别。使用它可为一个用户 ID 设置读写访问权限，并仅为另一个用户 ID 设置读取访问权限。对于组 ID 也可以采用这样的设置。

使用 `getfacl` 和 `setfacl` 命令（随附在的 `acl` 软件包中）可以直接修改文件系统资源。例如，要针对用户 `wilber` 检查和设置 `/tmp/test.txt` 文件的 ACL 限制，请执行以下命令：

```
tux > getfacl /tmp/test.txt  
# file: /tmp/test.txt
```

```
# owner: tux
# group: users
user::r--
group::r--
other::r--

tux > setfacl -m "u:wilber:rw" /tmp/test.txt

tux > getfacl /tmp/test.txt
# file: /tmp/test.txt
# owner: tux
# group: users
user::rw-
user:wilber:r--
group::r--
mask::r--
other::r--
```

标准 Unix 权限包括所谓的粘滞位。这允许使用比正在执行特定程序的用户更高的特权来执行这些程序。此功能最典型的示例是 **passwd** 工具，它需要修改 `/etc/shadow` 才能更改用户口令。

要以更循序渐进的方式显式允许对二进制文件的特定操作或行为，请使用扩展功能。一个默认使用扩展功能的命令的示例是 **ping**（包含在软件包 `iputils` 中）。

ping 通过网卡发送 ICMP IP 包。为此，它需要 `CAP_NET_RAW` 功能有效且受到允许 (+ep):

```
root # sudo getcap /usr/bin/ping
/usr/bin/ping = cap_net_raw+ep
```

可以使用可插入身份验证模块 (PAM) 来管理系统的登录访问控制。中提供了多个模块，允许记录登录时间、多个身份验证机制以及中心数据库（例如 NIS、LDAP 或 Active Directory）等设置。

有关管理权限的详细信息，请参见

A3.8 要求 8：识别并验证对系统组件的访问

理想情况下，应使用包含用户信息的中心数据库以及唯一标识符 (UID) 来授予或拒绝对特定系统组件的访问。这样就可以轻松为管理员授予对一组服务器的特殊访问权限，或者为数据库工程师授予对特定 DBMS 系统的权限。

在独立服务器上，唯一标识符通过标准 Linux 用户和组 ID 来管理。/etc/passwd 和 /etc/group 中列出了这些 ID。

8.1.4 去除/禁用在 90 天内处于非活动状态的用户帐户。

在此环境中，对用户帐户使用集中式基础结构（例如 NIS、LDAP 或 Active Directory）会带来许多优势：

- 可轻松识别并自动禁用非活动的帐户。
- 只需在一个位置禁用用户帐户。撤消用户的访问权限后，他们将无法使用任何依赖于集中式帐户基础结构的服务。

不过，如果您使用的是本地帐户，可以在用户登录时检查它们是否为非活动帐户。此模块会检查 /var/log/lastlog 中记录的上次登录时间，并计算该时间距离此时的天数。默认情况下，当非活动天数达到 90 天时，将会拒绝访问。

要列出本地帐户的上次登录时间，请使用 lastlog 命令。

8.1.6 通过在用户 ID 尝试访问最多六次后锁定该 ID 来限制重复的访问尝试。

如 [8.1.4 去除/禁用在 90 天内处于非活动状态的用户帐户](#)。中所述，集中式帐户基础结构可提供此功能。在系统上，可以使用 pam_tally2 PAM 模块检查和限制访问尝试。该模块将在登录时执行，会检查自上次成功登录以来所记录的失败尝试。要检查和重设置帐户状态，请使用 pam_tally2 工具。

8.1.7 将锁定持续时间设置为最少 30 分钟或直到管理员启用了该用户 ID。

[8.1.6 通过在用户 ID 尝试访问最多六次后锁定该 ID 来限制重复的访问尝试](#)。中所述的 PAM 模块 pam_tally2 可用于在登录尝试失败后，将帐户锁定一段指定的时间。必须在 PAM 配置中指定 unlock_time=1800 参数。默认情况下，只有管理员能够重新激活锁定的帐户。

8.3.1 针对进行管理访问的人员，将用于所有非控制台访问的多重身份验证纳入 CDE 中。

要使用多重验证机制验证进行管理访问的用户身份，请使用以下方法：

- 使用可插入身份验证模块 (PAM)：这可以提高将新方法添加到身份验证流程以及调整该流程时的灵活性。

第三方一次性口令 (OTP) 产品通常也有一个 Linux PAM 模块。

有关 PAM 的信息，请参见

- 要为 SSH 连接添加多重身份验证，除了使用口令外，还必须使用公共密钥。
要连接到某个系统，您必须证明自己拥有相应的私用密钥，然后在第二阶段输入口令。这意味着，攻击者需要先获取私用密钥才能尝试强行突破口令提示。

8.3.2 针对源自实体网络外部的所有远程网络访问（用户和管理员的访问，包括为了支持或维护目的而进行的第三方访问）纳入多重身份验证。

有关详细信息，请参见 [8.3.1 针对进行管理访问的人员，将用于所有非控制台访问的多重身份验证纳入 CDE 中。](#)

A3.9 要求 9：限制对持卡人数据的物理访问

对用于处理持卡人数据的系统的物理访问不属于一般操作系统安全性的范围。如要允许现场人员和访客直接访问系统，必须实施适当的设施进入管控措施。

A3.10 要求 10：跟踪并监控对网络资源和持卡人数据的所有访问

要跟踪用户活动，必须提供一个同步的时间参考。可以通过 NTP 协议实现此目的。NTP 允许服务器将其本地时间与中心系统保持同步。持卡人数据环境 (CDE) 内部的中心 NTP 服务器不应依赖于通过连接外部互联网来更新系统时间。作为替代方法，可以使用 DCF77 无线电传输或 GPS 接收器更新系统时间。

使用同步时间参考可以更轻松地关联所记录的日志文件中的事件。此参考可以包括中心系统日志服务器收集的一般系统日志项，或 `audit` 守护程序生成的内核审计消息。

有关审计的信息，请参见

可以通过定义集中储存的审计规则来满足本节中所述的所有审计要求。

A3.11 要求 11：定期测试安全系统和流程

测试所述安全机制也是 PCI DSS 的一项关键要求。评估配置并测试日志记录机制可以防范已知安全风险，并确保能够提供必要的信息来识别可能的安全违规。应在设计系统期间提前考虑测试功能。

随附了高级入侵检测环境 (AIDE)，用于跟踪系统完整性。AIDE 会创建所有相关操作系统文件的哈希值数据库。初始化后，可以使用它来校验以前保存的所有文件的完整性。要采用 AIDE，最好定期创建数据库快照并将其保存到一个中心系统，以便在此系统上评估可能发生的修改。

有关 AIDE 的详细信息，请参见

A3.12 要求 12：维护用于处理所有个人信息安全性的策略

需要处理宝贵信息的任何组织都应实施一项常规安全策略。策略应包括所有相关方面，使员工和利益相关者能够清楚地知道存在哪些可能的风险，以及如何避免这些风险。

此外，应定期评估所有安全策略并做出调整，以尽量保持最高的保护级别。

B GNU licenses

This appendix contains the GNU Free Documentation License version 1.2.

GNU Free Documentation License

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

0. PREAMBLE

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

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

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

1. APPLICABILITY AND DEFINITIONS

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

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

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

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

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

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

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

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

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

2. VERBATIM COPYING

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

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

3. COPYING IN QUANTITY

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

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

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

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

4. MODIFICATIONS

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

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

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

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

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

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

5. COMBINING DOCUMENTS

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

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

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

6. COLLECTIONS OF DOCUMENTS

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

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

7. AGGREGATION WITH INDEPENDENT WORKS

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

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

8. TRANSLATION

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

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

9. TERMINATION

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

10. FUTURE REVISIONS OF THIS LICENSE

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

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

ADDENDUM: How to use this License for your documents

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

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

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

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

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