系统管理指南：安全性服务

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

文件号码 819–7061–10
2006 年 9 月

060910@15490

目录

前言 23

第 1 部分　安全性概述 27

1　安全性服务（概述） 29
系统安全 29
Solaris 加密服务 30
验证服务 31
使用加密的验证 31
Solaris 审计 32
安全策略 32

第 2 部分　系统、文件和设备安全性 33

2　管理计算机安全性（概述） 35
Solaris 10 发行版中计算机安全性的增强 35
控制对计算机系统的访问 36
　维护物理安全性 36
　维护登录控制 36
控制对设备的访问 41
　设备策略（概述） 42
　设备分配（概述） 42
控制对计算机资源的访问 43
　限制和监视超级用户 43
　配置基于角色的访问控制以替换超级用户 43
　防止无意中误用计算机资源 43
　限制 `setuid` 可执行文件 45

使用自动安全性增强工具 45
使用 Solaris 安全工具包 45
使用 Solaris 资源管理功能 45
使用 Solaris Zones 46
监视计算机资源的使用情况 46
监视文件完整性 46
控制对文件的访问 46
通过加密保护文件 46
使用访问控制列表 47
跨计算机共享文件 47
限制对共享文件的 root 访问 47
控制网络访问 48
网络安全性机制 48
远程访问的验证和授权 49
防火墙系统 50
加密和防火墙系统 50
报告安全问题 51

3 控制对系统的访问（任务） 53
控制系统访问（任务列表） 53
保证登录和口令的安全（任务列表） 54
保证登录和口令的安全 54
▼ 如何显示用户的登录状态 54
▼ 如何显示没有口令的用户 55
▼ 如何临时禁止用户登录 56
▼ 如何监视失败的登录尝试 57
▼ 如何监视所有失败的登录尝试 58
▼ 如何创建拨号口令 59
▼ 如何临时禁用拨号登录 61
更改口令算法（任务列表） 61
更改口令加密的缺省算法 62
▼ 如何指定口令加密算法 62
▼ 如何为 NIS 域指定新的口令算法 63
▼ 如何为 NIS+ 域指定新的口令算法 63
▼ 如何为 LDAP 域指定新的口令算法 64
▼ 如何安装第三方的口令加密模块 64

监视和限制超级用户（任务列表） 66
监视和限制超级用户 66
▼ 如何监视正在使用 su 命令的用户 66
▼ 如何限制和监视超级用户登录 67
SPARC: 控制对系统硬件的访问（任务列表） 68
控制对系统硬件的访问 68
▼ 如何要求硬件访问口令 68
▼ 如何禁用系统的中止序列 69

4 控制对设备的访问（任务） 71
配置设备（任务列表） 71
配置设备策略（任务列表） 71
配置设备策略 72
▼ 如何查看设备策略 72
▼ 如何更改现有设备上的设备策略 73
▼ 如何审计设备策略中的更改 74
▼ 如何从 /dev/* 设备检索 IP MIB-II 信息 75
管理设备分配（任务列表） 76
管理设备分配 76
▼ 如何使设备可分配 76
▼ 如何授权用户来分配设备 77
▼ 如何查看有关设备的分配信息 78
▼ 强制分配设备 78
▼ 强制解除设备分配 79
▼ 如何更改可以分配的设备 79
▼ 如何审计设备分配 81
分配设备（任务列表） 81
分配设备 82
▼ 如何分配设备 82
▼ 如何挂载已分配的设备 83
▼ 如何解除设备分配 85
设备保护（参考） 87
设备策略命令 87
设备分配 87

5 使用基本审计报告工具（任务） 93
基本审计报告工具（概述） 93
BART 功能 93
BART 组件 94
使用 BART（任务列表） 95
使用 BART（任务） 96
BART 安全注意事项 96
▼ 如何创建清单 96
▼ 如何自定义清单 99
▼ 如何比较同一系统在一段时间内的清单 104
▼ 如何比较不同系统的清单与控制系统的清单 107
▼ 如何通过指定文件属性自定义 BART 报告 111
▼ 如何通过使用 Rules 文件自定义 BART 报告 111
BART 清单、Rules 文件和报告（参考） 113
BART 清单文件格式 113
BART Rules 文件格式 114
BART 报告 116

6 控制对文件的访问（任务） 119
使用 UNIX 权限保护文件 119
用于查看和保证文件安全的命令 119
文件和目录的拥有权 120
UNIX 文件权限 120
特殊文件权限（`setuid`、`setgid` 和 Sticky 位） 121
缺省 `umask` 值 122
文件权限模式 123
使用访问控制列表保护文件 125
文件的 ACL 项 125
目录的 ACL 项 126
用于管理 ACL 的命令 126
防止可执行文件危及安全 127
保护文件（任务列表） 127
使用 UNIX 权限保护文件（任务列表） 128
▼ 如何显示文件信息 128
▼ 如何更改文件的属主 129
▼ 如何更改文件的组拥有权 130

▼ 如何在符号模式下更改文件权限 131
▼ 如何在绝对模式下更改文件权限 132
▼ 如何在绝对模式下更改特殊文件权限 133
使用 ACL 保护文件（任务列表） 134
▼ 如何检查文件是否具有 ACL 134
▼ 如何将 ACL 项添加到文件 135
▼ 如何复制 ACL 137
▼ 如何更改文件的 ACL 项 137
▼ 如何删除文件的 ACL 项 138
▼ 如何显示文件的 ACL 项 138
防止程序受到安全风险（任务列表） 140
▼ 如何使用特殊文件权限查找文件 140
▼ 如何禁止程序使用可执行栈 142

7 使用自动安全性增强工具（任务） 143
自动安全性增强工具 (Automated Security Enhancement Tool, ASET) 143
ASET 安全级别 144
ASET 任务列表 144
ASET 执行日志 147
ASET 报告 148
ASET 主文件 151
ASET 环境文件 (`asetenv`) 151
配置 ASET 152
恢复 ASET 修改的系统文件 154
使用 NFS 系统进行网络操作 154
ASET 环境变量 155
ASET 文件示例 158
运行 ASET（任务列表） 159
▼ 如何交互运行 ASET 160
▼ 如何定期运行 ASET 161
▼ 如何停止定期运行 ASET 162
▼ 如何在服务器上收集 ASET 报告 162
解决 ASET 问题 164
ASET 错误消息 164

第 3 部分　角色、权限配置文件和权限 167

8　使用角色和权限（概述） 169
基于角色的访问控制（概述） 169
RBAC：超级用户模型的替代项 169
Solaris RBAC 元素和基本概念 171
RBAC 授权 174
授权和权限 174
特权应用程序和 RBAC 174
RBAC 权限配置文件 175
RBAC 角色 175
RBAC 中的配置文件 Shell 176
名称服务范围和 RBAC 176
直接指定安全属性时的安全注意事项 177
权限（概述） 177
权限保护内核进程 177
权限说明 178
具有权限的系统的管理差别 179
如何实现权限 180
进程如何获取权限 181
指定权限 181
权限和设备 182
权限和调试 183

9　使用基于角色的访问控制（任务） 185
使用 RBAC（任务列表） 185
配置 RBAC（任务列表） 186
配置 RBAC 186
▼ 如何规划 RBAC 实现 187
▼ 如何使用 GUI 创建和指定角色 188
▼ 如何通过命令行创建角色 191
▼ 如何将角色指定给本地用户 194
▼ 如何审计角色 195
▼ 如何使 root 用户成为角色 196
使用角色（任务列表） 198
使用角色 198

▼ 如何在终端窗口中承担角色 199
▼ 如何在 Solaris Management Console 中承担角色 201
管理 RBAC（任务列表） 202
管理 RBAC 203
▼ 如何更改角色的属性 203
▼ 如何创建或更改权限配置文件 205
▼ 如何更改用户的 RBAC 属性 207
▼ 如何为传统应用程序添加 RBAC 属性 209

10 基于角色的访问控制（参考） 211
权限配置文件的内容 211
主管理员权限配置文件 212
系统管理员权限配置文件 212
操作员权限配置文件 213
打印机管理权限配置文件 213
基本 Solaris 用户权限配置文件 214
所有权限配置文件 214
权限配置文件的顺序 215
查看权限配置文件的内容 215
授权命名和委托 215
授权命名约定 215
授权粒度示例 216
授权中的授权委托 216
支持 RBAC 的数据库 216
RBAC 数据库关系 217
RBAC 数据库和名称服务 218
`user_attr` 数据库 218
`auth_attr` 数据库 219
`prof_attr` 数据库 220
`exec_attr` 数据库 221
`policy.conf` 文件 222
RBAC 命令 223
管理 RBAC 的命令 223
要求授权的命令 224

11 权限（任务） 227
管理和使用权限（任务列表） 227
管理权限（任务列表） 227
管理权限 228
▼ 如何确定进程的权限 228
▼ 如何确定程序所需的权限 230
▼ 如何为命令添加权限 232
▼ 如何将权限指定给用户或角色 233
▼ 如何限制用户或角色的权限 234
▼ 如何运行具有特权命令的 Shell 脚本 235
确定权限（任务列表） 236
确定已指定的权限 236
▼ 如何确定已直接指定给您的权限 236
▼ 如何确定可以运行的特权命令 238
▼ 如何确定角色可以运行的特权命令 240

12 权限（参考） 245
用于处理权限的管理命令 245
包含权限信息的文件 246
权限和审计 247
防止权限升级 247
传统应用程序和权限模型 248

第 4 部分 Solaris 加密服务 249

13 Solaris 加密框架（概述） 251
Solaris 加密框架的新增功能 251
Solaris 加密框架 252
Solaris 加密框架术语 252
Solaris 加密框架的范围 253
Solaris 加密框架中的管理命令 254
Solaris 加密框架中的用户级命令 254
第三方软件的二进制文件签名 254
Solaris 加密框架插件 255
加密服务和区域 255

14 Solaris 加密框架（任务） 257
使用加密框架（任务列表） 257
使用 Solaris 加密框架保护文件（任务列表） 257
使用 Solaris 加密框架保护文件 258
▼ 如何生成对称密钥 258
▼ 如何计算文件摘要 260
▼ 如何计算文件的 MAC 261
▼ 如何加密和解密文件 263
管理加密框架（任务列表） 266
管理加密框架 266
▼ 如何列出可用提供器 266
▼ 如何添加软件提供器 270
▼ 如何禁止使用用户级机制 272
▼ 如何禁止使用内核软件提供器 275
▼ 如何列出硬件提供器 278
▼ 如何禁用硬件提供器机制和功能 279
▼ 如何刷新或重新启动所有加密服务 281

第 5 部分 验证服务和安全通信 283

15 使用验证服务（任务） 285
安全 RPC 概述 285
NFS 服务和安全 RPC 285
使用安全 NFS 的 DES 加密 286
Kerberos 验证 286
Diffie-Hellman 验证 286
管理安全 RPC（任务列表） 289
使用安全 RPC 管理验证 289
▼ 如何重新启动安全 RPC Keyserver 289
▼ 如何为 NIS+ 主机设置 Diffie-Hellman 密钥 290
▼ 如何为 NIS+ 用户设置 Diffie-Hellman 密钥 291
▼ 如何为 NIS 主机设置 Diffie-Hellman 密钥 292
▼ 如何为 NIS 用户设置 Diffie-Hellman 密钥 293
▼ 如何通过 Diffie-Hellman 验证共享 NFS 文件 294

16 使用 PAM 297
PAM（概述） 297
使用 PAM 的益处 297
PAM 组件 298
Solaris 10 发行版对 PAM 所做的更改 298
PAM（任务） 299
PAM（任务列表） 299
规划 PAM 实现 300
▼ 如何添加 PAM 模块 300
▼ 如何使用 PAM 防止从远程系统进行 Rhost 样式的访问 301
▼ 如何记录 PAM 错误报告 301
PAM 配置文件（参考） 302
PAM 配置文件语法 302
PAM 的服务名称 302
PAM 模块类型 302
PAM 控制标志 303
PAM 模块 304
通用 `pam.conf` 文件的示例 304

17 使用 SASL 307
SASL（概述） 307
SASL（参考） 307
SASL 插件 308
SASL 环境变量 308
SASL 选项 308

18 使用 Solaris 安全 Shell（任务） 311
Solaris 安全 Shell（概述） 311
Solaris 安全 Shell 验证 312
企业中的 Solaris 安全 Shell 313
Solaris 10 发行版中 Solaris 安全 Shell 的增强功能 314
Solaris 安全 Shell（任务列表） 314
配置 Solaris 安全 Shell（任务列表） 315
配置 Solaris 安全 Shell 315
▼ 如何为 Solaris 安全 Shell 设置基于主机的验证 315
▼ 如何启用 Solaris 安全 Shell v1 318

▼ 如何在 Solaris 安全 Shell 中配置端口转发 318
使用 Solaris 安全 Shell（任务列表） 319
使用 Solaris 安全 Shell 320
▼ 如何生成用于 Solaris 安全 Shell 的公钥/私钥对 320
▼ 如何更改 Solaris 安全 Shell 私钥的口令短语 323
▼ 如何使用 Solaris 安全 Shell 登录到远程主机 323
▼ 如何减少 Solaris 安全 Shell 中的口令提示 324
▼ 如何将 ssh-agent 命令设置为自动运行 326
▼ 如何在 Solaris 安全 Shell 中使用端口转发 327
▼ 如何使用 Solaris 安全 Shell 复制文件 328
▼ 如何设置到防火墙外部主机的缺省连接 329

19 Solaris 安全 Shell（参考） 331
典型的 Solaris 安全 Shell 会话 331
Solaris 安全 Shell 中会话的特征 331
Solaris 安全 Shell 中的验证和密钥交换 332
Solaris 安全 Shell 中的命令执行和数据转发 333
Solaris 安全 Shell 中的客户机和服务器配置 333
Solaris 安全 Shell 中的客户机配置 333
Solaris 安全 Shell 中的服务器配置 333
Solaris 安全 Shell 中的关键字 334
Solaris 安全 Shell 中的主机特定参数 337
Solaris 安全 Shell 和登录环境变量 337
维护 Solaris 安全 Shell 中的已知主机 338
Solaris 安全 Shell 软件包和初始化 339
Solaris 安全 Shell 文件 339
Solaris 安全 Shell 命令 341

第 6 部分 Kerberos 服务 343

20 Kerberos 服务介绍 345
什么是 Kerberos 服务? 345
Kerberos 服务的工作方式 346
初始验证：票证授予票证 346
后续 Kerberos 验证 348
Kerberos 远程应用程序 349

Kerberos 主体 349
Kerberos 领域 350
Kerberos 安全服务 351
各种 Kerberos 发行版的组件 351
Kerberos 组件 352
Solaris 10 发行版中的 Kerberos 增强功能 353
Solaris 10 6/06 发行版的 Kerberos 新增功能 355
Solaris 9 发行版中的 Kerberos 组件 355
SEAM 1.0.2 组件 355
Solaris 8 发行版中的 Kerberos 组件 355
SEAM 1.0.1 组件 355
SEAM 1.0 组件 356

21 规划 Kerberos 服务 357
为什么要规划 Kerberos 部署? 357
Kerberos 领域 358
领域名称 358
领域数 358
领域分层结构 358
将主机名映射到领域 359
客户机名称和服务主体名称 359
KDC 端口和管理服务端口 359
从 KDC 数 360
将 GSS 凭证映射到 UNIX 凭证 360
自动将用户迁移到 Kerberos 领域 360
要使用的数据库传播系统 361
领域内的时钟同步 361
客户机安装选项 361
Kerberos 加密类型 361
SEAM Administration Tool 中的联机帮助 URL 362

22 配置 Kerberos 服务（任务） 363
配置 Kerberos 服务（任务列表） 363
配置其他 Kerberos 服务（任务列表） 364
配置 KDC 服务器 364
▼ 如何配置主 KDC 365

▼ 如何配置从 KDC 372
配置跨领域验证 377
▼ 如何建立分层跨领域验证 377
▼ 如何建立直接跨领域验证 378
配置 Kerberos 网络应用程序服务器 380
▼ 如何配置 Kerberos 网络应用程序服务器 380
配置 Kerberos NFS 服务器 382
▼ 如何配置 Kerberos NFS 服务器 382
▼ 如何创建凭证表 384
▼ 如何向凭证表中添加单个项 384
▼ 如何提供各领域之间的凭证映射 385
▼ 如何使用多种 Kerberos 安全模式设置安全的 NFS 环境 386
配置 Kerberos 客户机 388
配置 Kerberos 客户机（任务列表） 388
▼ 如何创建 Kerberos 客户机安装配置文件 388
▼ 如何自动配置 Kerberos 客户机 389
▼ 如何交互配置 Kerberos 客户机 391
▼ 如何手动配置 Kerberos 客户机 393
▼ 如何以 root 用户身份访问受 Kerberos 保护的 NFS 文件系统 399
▼ 在 Kerberos 领域中配置用户自动迁移 401
同步 KDC 和 Kerberos 客户机的时钟 404
交换主 KDC 和从 KDC 405
▼ 如何配置可交换的从 KDC 405
▼ 如何交换主 KDC 和从 KDC 406
管理 Kerberos 数据库 412
备份和传播 Kerberos 数据库 412
▼ 如何备份 Kerberos 数据库 413
▼ 如何恢复 Kerberos 数据库 414
▼ 如何重新装入 Kerberos 数据库 415
▼ 如何重新配置主 KDC 以使用增量传播 415
▼ 如何重新配置从 KDC 以使用增量传播 418
▼ 如何配置从 KDC 以使用完全传播 420
▼ 如何验证 KDC 服务器是否已同步 423
▼ 如何手动将 Kerberos 数据库传播到从 KDC 425
设置并行传播 426
设置并行传播的配置步骤 426
管理存储文件 427

▼ 如何删除存储文件 427
增强 Kerberos 服务器的安全性 428
▼ 如何仅启用基于 Kerberos 的应用程序 428
▼ 如何限制对 KDC 服务器的访问 428

23 Kerberos 错误消息和疑难解答 431
Kerberos 错误消息 431
SEAM Administration Tool 错误消息 431
常见的 Kerberos 错误消息 (A-M) 432
常见的 Kerberos 错误消息 (N-Z) 438
Kerberos 疑难解答 441
`krb5.conf` 文件的格式存在问题 441
传播 Kerberos 数据库时出现问题 442
挂载基于 Kerberos 的 NFS 文件系统时出现问题 442
以 `root` 身份进行验证时出现问题 443
观察从 GSS 凭证到 UNIX 凭证的映射 443

24 管理 Kerberos 主体和策略（任务） 445
管理 Kerberos 主体和策略的方法 445
SEAM Administration Tool 446
SEAM Tool 的等效命令行 446
SEAM Tool 修改的唯一文件 447
SEAM Tool 的打印和联机帮助功能 447
在 SEAM Tool 中处理大型列表 447
▼ 如何启动 SEAM Tool 448
管理 Kerberos 主体 450
管理 Kerberos 主体（任务列表） 451
自动创建新的 Kerberos 主体 451
▼ 如何查看 Kerberos 主体列表 452
▼ 如何查看 Kerberos 主体属性 454
▼ 如何创建新的 Kerberos 主体 456
▼ 如何复制 Kerberos 主体 459
▼ 如何修改 Kerberos 主体 459
▼ 如何删除 Kerberos 主体 460
▼ 如何设置缺省值以创建新的 Kerberos 主体 461
▼ 如何修改 Kerberos 管理权限 462

管理 Kerberos 策略 463
管理 Kerberos 策略（任务列表） 464
▼ 如何查看 Kerberos 策略列表 464
▼ 如何查看 Kerberos 策略属性 466
▼ 如何创建新的 Kerberos 策略 468
▼ 如何复制 Kerberos 策略 470
▼ 如何修改 Kerberos 策略 470
▼ 如何删除 Kerberos 策略 471
SEAM Tool 参考 472
SEAM Tool 面板说明 472
以受限 Kerberos 管理权限使用 SEAM Tool 474
管理密钥表文件 476
管理密钥表文件（任务列表） 476
▼ 如何将 Kerberos 服务主体添加至密钥表文件 477
▼ 如何从密钥表文件中删除服务主体 479
▼ 如何显示密钥表文件中的密钥列表（主体） 480
▼ 如何临时禁用对主机上的服务的验证 481

25 使用 Kerberos 应用程序（任务） 485
Kerberos 票证管理 485
是否需要担心票证? 485
创建 Kerberos 票证 486
查看 Kerberos 票证 487
销毁 Kerberos 票证 489
Kerberos 口令管理 489
口令选择建议 489
更改口令 490
授予对帐户的访问权限 493
Kerberos 用户命令 494
基于 Kerberos 的命令概述 495
转发 Kerberos 票证 497
示例－使用基于 Kerberos 的命令 498

26 Kerberos 服务（参考） 501
Kerberos 文件 501
Kerberos 命令 502

Kerberos 守护进程 503
Kerberos 术语 504
特定于 Kerberos 的术语 504
特定于验证的术语 504
票证类型 505
Kerberos 验证系统的工作方式 508
使用 Kerberos 获取服务访问权限 509
获取用于票证授予服务的凭证 509
获取用于服务器的凭证 510
获取对特定服务的访问权限 511
使用 Kerberos 加密类型 512
使用 gsscred 表 513
Solaris Kerberos 和 MIT Kerberos 之间的显著差异 513

第 7 部分 Solaris 审计 515

27 Solaris 审计（概述） 517
什么是审计? 517
审计如何工作? 518
审计如何与安全相关? 519
审计术语和概念 519
审计事件 520
审计类和预选 521
审计记录和审计标记 521
审计文件 522
审计存储 523
检查审计跟踪 524
Solaris 10 发行版中的 Solaris 审计增强功能 524

28 规划 Solaris 审计 527
规划 Solaris 审计（任务列表） 527
规划 Solaris 审计（任务） 527
▼ 如何在区域中规划审计 528
▼ 如何规划审计记录的存储 529
▼ 如何规划要审计的对象及内容 529
确定审计策略 531

控制审计成本 533
延长审计数据的处理时间带来的成本 533
分析审计数据的成本 533
存储审计数据的成本 533
有效审计 534

29 管理 Solaris 审计（任务） 535
Solaris 审计（任务列表） 535
配置审计文件（任务列表） 536
配置审计文件 536
▼ 如何修改 audit_control 文件 536
▼ 如何配置 syslog 审计日志 539
▼ 如何更改用户审计特征 541
▼ 如何添加审计类 543
▼ 如何更改审计事件的类成员关系 544
配置和启用审计服务（任务列表） 546
配置和启用审计服务 546
▼ 如何创建审计文件的分区 547
▼ 如何配置 audit_warn 电子邮件别名 550
▼ 如何配置审计策略 550
▼ 如何启用审计 553
▼ 如何禁用审计 555
▼ 如何更新审计服务 556
管理审计记录（任务列表） 557
管理审计记录 557
▼ 如何显示审计记录格式 558
▼ 如何合并审计跟踪中的审计文件 561
▼ 如何从审计跟踪中选择审计事件 563
▼ 如何查看二进制审计文件的内容 565
▼ 如何清除 not_terminated 审计文件 567
▼ 如何防止审计跟踪溢出 568

30 Solaris 审计（参考） 569
审计命令 569
auditd 守护进程 569
audit 命令 570

bsmrecord 命令 570
auditreduce 命令 571
praudit 命令 572
auditconfig 命令 574
在审计服务中使用的文件 574
system 文件 574
syslog.conf 文件 575
audit_class 文件 575
audit_control 文件 575
audit_event 文件 576
audit_startup 脚本 576
audit_user 数据库 577
audit_warn 脚本 577
bsmconv 脚本 578
用于管理审计的权限配置文件 579
审计和 Solaris Zones 579
审计类 580
审计类的定义 580
审计类语法 581
审计策略 582
进程审计特征 582
审计跟踪 583
二进制审计文件名称约定 583
二进制审计文件名称 583
二进制审计文件时间标记 584
审计记录结构 584
审计记录分析 585
审计标记格式 585
acl 标记 587
arbitrary 标记（已过时） 587
arg 标记 588
attribute 标记 588
cmd 标记 589
exec_args 标记 589
exec_env 标记 589
exit 标记（已过时） 590
file 标记 590

group 标记（已过时） 591
groups 标记 591
header 标记 591
in_addr 标记 592
ip 标记（已过时） 592
ipc 标记 592
ipc_perm 标记 593
iport 标记 593
opaque 标记（已过时） 594
path 标记 594
path_attr 标记 594
privilege 标记 595
process 标记 595
return 标记 597
sequence 标记 597
socket 标记 597
subject 标记 598
text 标记 600
trailer 标记 600
uauth 标记 600
zonename 标记 601

词汇表 603

索引 613

前言

《系统管理指南：安全性服务》是多卷集的一部分，该多卷集包含 Solaris™ 操作系统管理信息的重要部分。本书假定您已经安装了 SunOS™ 5.10 操作系统，并且设置了计划使用的任何网络软件。SunOS 5.10 操作系统属于 Solaris 10 产品系列，它还包含 Solaris 公用桌面环境 (Common Desktop Environment, CDE) 等许多功能。

注 – 此 Solaris 发行版支持使用以下 SPARC® 和 x86 系列处理器体系结构的系统：UltraSPARC®、SPARC64、AMD64、Pentium 和 Xeon EM64T。支持的系统可以在 `http://www.sun.com/bigadmin/hcl` 上的《Solaris 10 Hardware Compatibility List》中找到。本文档列举了在不同类型的平台上进行实现时的所有差别。

在本文档中，这些与 x86 相关的术语表示以下含义：

- “x86” 泛指 64 位和 32 位的 x86 兼容产品系列。
- “x64” 指出了有关 AMD64 或 EM64T 系统的特定 64 位信息。
- “32 位 x86” 指出了有关基于 x86 的系统的特定 32 位信息。

若想了解本发行版支持哪些系统，请参见《Solaris 10 Hardware Compatibility List》。

目标读者

本书适用于所有负责管理一个或多个运行 Solaris 10 发行版的系统的人员。要使用本书，您应该有两年以上管理 UNIX® 系统的经验。参加 UNIX 系统管理培训课程可能会对您有所帮助。

系统管理卷的结构

以下是系统管理指南卷所包含主题的列表。

书名	主题
《System Administration Guide: Basic Administration》	用户帐户和组、服务器和客户机支持、关闭和引导系统、管理服务以及管理软件（软件包和修补程序）

书名	主题
《System Administration Guide: Advanced Administration》	打印服务、终端和调制解调器、系统资源（磁盘配额、记帐和 crontab）、系统进程以及 Solaris 软件问题疑难解答
《System Administration Guide: Devices and File Systems》	可移除介质、磁盘和设备、文件系统以及备份和恢复数据
《System Administration Guide: IP Services》	TCP/IP 网络管理、IPv4 和 IPv6 地址管理、DHCP（动态主机配置协议）、IPsec（Internet 协议安全）、IKE（Internet 密钥交换）、Solaris IP 过滤器、移动 IP、IP 网络多路径 (IP network multipathing, IPMP) 和 IPQoS（IP 服务质量）
《System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)》	DNS、NIS 和 LDAP 名称和目录服务，包括从 NIS 到 LDAP 的转换和从 NIS+ 到 LDAP 的转换
《System Administration Guide: Naming and Directory Services (NIS+)》	NIS+ 名称和目录服务
《System Administration Guide: Network Services》	Web 高速缓存服务器、与时间相关的服务、网络文件系统（NFS 和 Autofs）、邮件、SLP 和 PPP
《System Administration Guide: Security Services》	审计、设备管理、文件安全、BART（基本审计报告工具）、Kerberos 服务、PAM（可插拔验证模块）、Solaris 加密框架、权限、RBAC（基于角色的访问控制）、SASL（简单身份验证和安全层）和 Solaris 安全 Shell
《System Administration Guide: Solaris Containers-Resource Management and Solaris Zones》	资源管理主题项目和任务、扩展记帐、资源控制、公平共享调度程序 (fair share scheduler, FSS)、使用资源上限设置守护进程 (resource capping daemon, rcapd) 的物理内存控制以及动态资源池；使用 Solaris Zones 软件分区技术的虚拟化

相关的第三方 Web 站点引用

本文档参考了第三方 URL，这些 URL 提供了额外的相关信息。

Sun 对本文档中提到的第三方 Web 站点的可用性不承担任何责任。对于此类站点或资源中的（或通过它们获得的）任何内容、广告、产品或其他资料，Sun 并不表示认可，也不承担任何责任。对于因使用或依靠此类站点或资源中的（或通过它们获得的）任何内容、产品或服务而造成的或连带产生的实际或名义损坏或损失，Sun 概不负责，也不承担任何责任。

文档、支持和培训

Sun Web 站点提供有关以下附加资源的信息：

- 文档 (http://www.sun.com/documentation/)
- 支持 (http://www.sun.com/support/)
- 培训 (http://www.sun.com/training/)

印刷约定

下表介绍了本书中的印刷约定。

表 P–1 印刷约定

字体	含义	示例
`AaBbCc123`	命令、文件和目录的名称；计算机屏幕输出	编辑 `.login` 文件。 使用 `ls -a` 列出所有文件。 `machine_name% you have mail.`
`AaBbCc123`	用户键入的内容，与计算机屏幕输出的显示不同	`machine_name%` **`su`** `Password:`
aabbcc123	要使用实名或值替换的命令行占位符	删除文件的命令为 `rm` *filename*。
AaBbCc123	保留未译的新词或术语以及要强调的词	这些称为 *Class* 选项。 **注意**：有些强调的项目在联机时以粗体显示。
新词术语强调	新词或术语以及要强调的词	**高速缓存**是存储在本地的副本。 请**勿**保存文件。
《书名》	书名	阅读《用户指南》的第 6 章。

命令中的 shell 提示符示例

下表列出了 C shell、Bourne shell 和 Korn shell 的缺省 UNIX 系统提示符和超级用户提示符。

表 P–2 Shell 提示符

Shell	提示符
C shell	`machine_name%`
C shell 超级用户	`machine_name#`
Bourne shell 和 Korn shell	`$`
Bourne shell 和 Korn shell 超级用户	`#`

第 1 部 分

安全性概述

本书重点介绍可增强 Solaris 操作系统安全性的功能。本书适用于系统管理员和使用这些安全性功能的用户。在概述一章中介绍了本书的主题。

◆ ◆ ◆ 第 1 章

1 安全性服务（概述）

为了维护 Solaris 操作系统 (Solaris OS) 的安全，Solaris 软件提供了以下功能：

- 第 29 页中的 “系统安全” －此功能可防止侵入、保护计算机资源和设备不被误用、使文件免遭用户或入侵者的恶意修改或无意修改

  有关系统安全的一般讨论，请参见第 2 章。
- 第 30 页中的 “Solaris 加密服务” －此功能可加密数据，以便只有发送者和指定的接收者才能阅读内容，以及管理加密提供器
- 第 31 页中的 “验证服务” －此功能可安全地标识用户，需要提供用户名和某种形式的证明（通常为口令）
- 第 31 页中的 “使用加密的验证” －此功能可确保已验证方可以不受窃听、修改或电子欺骗地通信
- 第 32 页中的 “Solaris 审计” －此功能可确定系统安全更改的原因，包括文件访问、与安全相关的系统调用和验证失败
- 第 32 页中的 “安全策略” －为计算机或计算机网络设计和实现安全原则

系统安全

系统安全可确保正确地使用系统资源。访问控制可以限制允许访问系统资源的用户。可用于系统安全和访问控制的 Solaris OS 功能如下：

- **登录管理工具** －用于监视和控制用户能否登录的命令。请参见第 54 页中的 “保证登录和口令的安全（任务列表）” 。
- **硬件访问** －用于限制对 PROM 的访问，以及可引导系统的用户的命令。请参见第 68 页中的 “SPARC: 控制对系统硬件的访问（任务列表）” 。
- **资源访问** －用于尽可能合理利用计算机资源并尽可能减少误用这些资源的工具和策略。请参见第 43 页中的 “控制对计算机资源的访问” 。
- **基于角色的访问控制 (Role-based access control, RBAC)** －一种体系结构，用于创建允许执行特定管理任务的特殊受限用户帐户。请参见第 169 页中的 “基于角色的访问控制（概述）” 。

- **特权** – 进程执行操作的独立权利。这些进程权限在内核中执行。请参见第 177 页中的 “权限（概述）”。
- **设备管理** – 为已通过 UNIX 权限保护的设备提供额外保护的设备**策略**。设备**分配**控制对外围设备（如麦克风或 CD-ROM 驱动器）的访问。取消分配后，设备清除脚本就可以删除来自该设备的任何数据。请参见第 41 页中的 “控制对设备的访问”。
- **基本审计和报告工具 (Basic Audit Reporting Tool, BART)** – 系统中文件的文件属性快照，称为**清单**。通过比较系统间的清单或一个系统在一段时间内的清单，可以监视对文件所做的更改，以降低安全风险。请参见第 5 章。
- **文件权限** – 文件或目录的属性。权限对允许读取、写入、执行文件或搜索目录的用户和组进行限制。请参见第 6 章。
- **安全性增强功能脚本** – 通过使用这些脚本，可以调整许多系统文件和参数以降低安全风险。请参见第 7 章。

Solaris 加密服务

密码学是一门对数据进行加密和解密的学科。密码学用于保证完整性、保密性和真实性。完整性指数据未被更改。保密性指其他用户无法读取数据。数据的真实性指传送内容与发送内容一致。用户验证指用户提供一个或多个身份证明。验证机制以数学方式检验数据源或身份证明。加密机制对数据进行加密，以使临时观察者无法读取数据。加密服务为应用程序和用户提供验证和加密机制。

加密算法使用散列、链接和其他数学方法来创建难以破解的密码。验证机制要求发送者和接收者根据数据计算出的数字完全相同。加密机制依赖于发送者和接收者对有关加密方法的信息的共享。只有接收者和发送者才可以使用此信息解密消息。Solaris OS 提供集中式的加密框架，以及与特定应用程序关联的加密机制。

- **Solaris™ 加密框架** – 一种用于内核级和用户级使用者的加密服务中心框架。其用途包括口令、IPsec 和第三方应用程序。该加密框架包含许多软件加密模块。通过该框架，可以指定应用程序能够使用的软件加密模块或硬件加密源。该框架基于 PKCS #11 v2 库构建。此库是按照 RSA Security Inc. 推出的 PKCS #11 加密令牌接口 (Cryptographic Token Interface, Cryptoki) 标准实现的。此库可为第三方开发者提供 API，以插入应用程序加密要求。请参见第 13 章。
- **每个应用程序的加密机制** –
  - 有关如何在安全 RPC 中使用 DES，请参见第 285 页中的 “安全 RPC 概述”。
  - 有关如何在 Kerberos 服务中使用 DES、3DES、AES 和 ARCFOUR，请参见第 20 章。
  - 有关如何在 Solaris 安全 Shell 中使用 RSA、DSA 和密码（如 Blowfish），请参见第 18 章。
  - 有关如何在口令中使用加密算法，请参见第 61 页中的 “更改口令算法（任务列表）”。

验证服务

验证是一种根据预定义的条件识别用户或服务的机制。验证服务的范围包括从简单的名称-口令对到更详细的质询-响应系统（如智能卡和生物识别技术）。强验证机制依赖于用户提供的只有自己了解的信息，以及可检验的个人项目。用户名便是用户了解的信息。可检验的个人项目则包括智能卡和指纹等。Solaris 的验证功能如下：

- **安全 RPC** – 一种验证机制，该机制使用 Diffie-Hellman protocol（Diffie-Hellman 协议）保护 NFS 挂载和名称服务（如 NIS 或 NIS+）。请参见第 285 页中的 “安全 RPC 概述”。
- **可插拔验证模块 (Pluggable Authentication Module, PAM)** – 一种框架，用于在系统登录服务中插入各种验证技术，而无需重新编译服务。某些系统登录服务包括 `login` 和 `ftp`。请参见第 16 章。
- **简单身份验证和安全层 (Simple Authentication and Security Layer, SASL)** – 一种为网络协议提供验证和安全性服务的框架。请参见第 17 章。
- **Solaris 安全 Shell** – 一种安全远程登录和传输协议，用于加密不安全网络上的通信。请参见第 18 章。
- **Kerberos 服务** – 一种客户机/服务器体系结构，用于为加密提供验证。请参见第 20 章。
- **Solaris 智能卡** – 一种带有微处理器和内存的塑料卡，可在读卡器上使用以访问系统。请参见《Solaris Smartcard Administration Guide》。

使用加密的验证

使用加密的验证是安全通信的基础。验证有助于确保源和目标是预定的双方。加密在源方对通信进行编码，在目标方对通信进行解码。加密可防止入侵者读取其可能会设法拦截的传输数据。Solaris 用于安全通信的功能有：

- **Solaris 安全 Shell** – 一种协议，用于保护数据传送和交互式用户网络会话，以防止窃听、会话劫持和 "man-in-the-middle"攻击。通过公钥密码学提供了强验证。X Windows 服务和其他网络服务可通过安全 Shell 连接建立安全通道，以获得其他保护。请参见第 18 章。
- **Kerberos 服务** – 一种提供使用加密的验证的客户机/服务器体系结构。请参见第 20 章。
- **Internet 协议安全体系结构 (Internet Protocol Security Architecture, IPsec)** – 一种提供 IP 数据报保护的体系结构。这些保护包括保密性、数据高完整性、数据验证和部分序列完整性。请参见《System Administration Guide: IP Services》中的第 19 章，“IP Security Architecture (Overview)”。

Solaris 审计

审计是系统安全和可维护性的一个基本概念。审计是一种进程，用于检查系统中的操作和事件的历史记录，以确定发生的情况。历史记录保存在日志中，其中记录了完成的操作、完成时间、操作者和受影响的对象。请参见第 27 章。

安全策略

本书中短语“安全策略”或“policy（策略）”指组织的安全原则。站点的安全策略是规则集，可用于定义要处理的信息的敏感度并防止信息受到未经授权的访问。安全技术（如 Solaris 安全 Shell、验证、RBAC、授权、特权和资源控制）可提供保护信息的措施。

描述安全技术实现的特定方面时，某些安全技术也使用策略一词。例如，Solaris 审计使用审计策略选项来配置审计策略的某些方面。下表指向有关使用策略一词描述其实现特定方面的功能的词汇表、手册页和信息：

表 1–1 Solaris OS 中使用的策略

词汇表定义	选择的手册页	详细信息
audit policy（审计策略）	`audit_control`(4)、`audit_user`(4)、`auditconfig`(1M)	第 27 章
policy in the cryptographic framework（加密框架中的策略）	`cryptoadm`(1M)	第 13 章
device policy（设备策略）	`getdevpolicy`(1M)	第 41 页中的 “控制对设备的访问”
Kerberos policy（Kerberos 策略）	`krb5.conf`(4)	第 24 章
network policies（网络策略）	`ipfilter`(5)、`ifconfig`(1M)、`ike.config`(4)、`ipsecconf`(1M)、`routeadm`(1M)	《System Administration Guide: IP Services》中的第 IV 部分，“IP Security”
password policy（口令策略）	`passwd`(1)、`nsswitch.conf`(4)、`crypt.conf`(4)、`policy.conf`(4)	第 34 页中的 “维护登录控制”
RBAC policy（RBAC 策略）	`rbac`(5)	第 221 页中的 “`exec_attr` 数据库”

第 2 部分

系统、文件和设备安全性

本部分介绍可以在非联网系统上配置的安全性。以下各章讨论了如何规划、监视和控制对磁盘、文件及外围设备的访问。

◆ ◆ ◆ 第 2 章

管理计算机安全性（概述）

保持计算机的信息安全是一项重要的系统管理任务。本章概述了有关管理计算机安全性的信息。

以下是本章中概述信息的列表：

- 第 35 页中的 “Solaris 10 发行版中计算机安全性的增强”
- 第 36 页中的 “控制对计算机系统的访问”
- 第 41 页中的 “控制对设备的访问”
- 第 43 页中的 “控制对计算机资源的访问”
- 第 46 页中的 “控制对文件的访问”
- 第 48 页中的 “控制网络访问”
- 第 51 页中的 “报告安全问题”

Solaris 10 发行版中计算机安全性的增强

从 Solaris 9 发行版开始，已经引入了以下功能来增强系统安全：

- 强口令加密可用并可配置。有关更多信息，请参见第 38 页中的 “口令加密”。
- 设备策略通过权限进行了增强。有关更多信息，请参见第 42 页中的 “设备策略（概述）”。

  对于设备分配，将来的 Solaris OS 发行版可能不支持 `/etc/security/dev` 目录。
- 基本审计报告工具 (Basic Audit Reporting Tool, BART) 可以监视系统中文件的真实性。有关更多信息，请参见第 5 章。
- 文件可以通过强加密进行保护。有关更多信息，请参见第 46 页中的 “通过加密保护文件”。
- 权限在内核级别增强进程权利。有关更多信息，请参见第 177 页中的 “权限（概述）”。
- Solaris 加密框架集中了提供者和使用者的加密服务。有关更多信息，请参见第 13 章。
- PAM 框架为许多程序（如 Solaris 安全 Shell）提供功能。有关更多信息，请参见第 298 页中的 “Solaris 10 发行版对 PAM 所做的更改”。

- Solaris Zones 和资源管理控制对计算机资源的访问。有关更多信息，请参见《System Administration Guide: Solaris Containers-Resource Management and Solaris Zones》。

控制对计算机系统的访问

在工作场所中，可以将所有与服务器连接的计算机视为一个大型的多功能系统。您要负责此大型系统的安全性。您需要保护网络，防止其受到尝试访问此网络的外来者的破坏，还需要确保网络中各计算机上的数据的完整性。

在文件级别，Solaris OS 提供了可用于保护文件、目录和设备的标准安全功能。在系统和网络级别，安全问题基本相同。第一道安全防线是控制对系统的访问。您可以执行以下操作来控制和监视系统访问：

- 第 36 页中的 “维护物理安全性”
- 第 36 页中的 “维护登录控制”
- 第 41 页中的 “控制对设备的访问”
- 第 43 页中的 “控制对计算机资源的访问”
- 第 46 页中的 “控制对文件的访问”
- 第 48 页中的 “控制网络访问”
- 第 51 页中的 “报告安全问题”

维护物理安全性

要控制对系统的访问，必须维护计算环境的物理安全性。例如，处于登录状态并且无人值守的系统容易受到未经授权的访问攻击。入侵者可以获取访问操作系统和网络的权限。应该物理保护计算机环境和计算机硬件，防止其受到未经授权的访问攻击。

您可以保护 SPARC 系统，防止硬件设置受到未经授权的访问。使用 `eeprom` 命令要求在访问 PROM 时提供口令。有关更多信息，请参见第 68 页中的 “如何要求硬件访问口令”。

维护登录控制

您还必须防止对系统或网络进行未经授权的登录，这可以通过指定口令和控制登录来实现。系统上的所有帐户都应该具有口令。口令是一种简单的验证机制。如果帐户没有设置口令，则猜中用户名的入侵者可以访问整个网络。强口令算法可防止强力攻击。

用户登录系统时，`login` 命令会根据 `/etc/nsswitch.conf` 文件中列出的信息检查相应的名称服务或目录服务数据库。此文件包括以下项：

- `files`－表示本地系统上的 `/etc` 文件
- `ldap`－表示 LDAP 服务器上的 LDAP 目录服务
- `nis`－表示 NIS 主服务器上的 NIS 数据库
- `nisplus`－表示 NIS+ 根服务器上的 NIS+ 数据库

有关 nsswitch.conf 文件的说明，请参见 nsswitch.conf(4) 手册页。有关名称服务和目录服务的信息，请参见《System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)》或《System Administration Guide: Naming and Directory Services (NIS+)》。

login 命令会检验用户提供的用户名和口令。如果用户名不在口令文件中，则 login 命令会拒绝对系统进行访问。如果口令对于指定的用户名而言不正确，则 login 命令会拒绝对系统进行访问。当用户提供有效的用户名及其对应口令时，系统会授予此用户访问系统的权限。

成功登录到系统之后，可使用 PAM 模块简化应用程序登录。有关更多信息，请参见第 16 章。

Solaris 系统提供了完善的验证和授权机制。有关网络级别的验证和授权机制的介绍，请参见第 49 页中的 “远程访问的验证和授权”。

管理口令信息

用户登录系统时，必须提供用户名和口令。尽管登录名是公开的，但是口令必须保密。口令应该只有每个用户才知道。应该要求用户谨慎选择其口令。用户应该经常更改其口令。

口令最初是在设置用户帐户时创建的。要维护用户帐户的安全性，可以设置口令生命期，以便强制用户定期更改其口令。还可以通过锁定口令来禁用用户帐户。有关管理口令的详细信息，请参见《System Administration Guide: Basic Administration》中的第 4 章，“Managing User Accounts and Groups (Overview)”以及 passwd(1) 手册页。

本地口令

如果网络使用本地文件来验证用户，则口令信息保存在系统的 /etc/passwd 和 /etc/shadow 文件中。用户名和其他信息保存在口令文件 /etc/passwd 中。加密的口令本身保存在单独的**阴影**文件 /etc/shadow 中。这种安全措施可防止用户获取访问加密口令的权限。尽管任何可以登录到系统的用户都能使用 /etc/passwd 文件，但是仅有超级用户或等效角色才能读取 /etc/shadow 文件。可以使用 passwd 命令来更改本地系统上的用户口令。

NIS 和 NIS+ 口令

如果网络使用 NIS 来验证用户，则口令信息保存在 NIS 口令列表中。NIS 不支持口令生命期。可以使用命令 passwd -r nis 来更改存储在 NIS 口令列表中的用户口令。

如果网络使用 NIS+ 来验证用户，则口令信息保存在 NIS+ 数据库中。可以通过仅允许授权用户进行访问来保护 NIS+ 数据库中的信息。可以使用 passwd -r nisplus 命令来更改存储在 NIS+ 数据库中的用户口令。

LDAP 口令

Solaris LDAP 名称服务将口令信息和阴影信息存储在 LDAP 目录树的 ou=people 容器中。在 Solaris LDAP 名称服务客户机上，可以使用 passwd -r ldap 命令来更改用户口令。LDAP 名称服务将口令存储在 LDAP 系统信息库中。

在 Solaris 10 发行版中，口令策略是在 Sun Java™ System Directory Server 上强制执行的。具体而言，客户机的 pam_ldap 模块遵循在 Sun Java System Directory Server 上强制执行的口令策略控制。有关更多信息，请参见《System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)》中的“LDAP Naming Services Security Model”。

口令加密

强口令加密可提供较早的攻击防线。Solaris 软件提供了四种口令加密算法。与 UNIX 算法相比，其中的两种 MD5 算法和 Blowfish 算法可提供更为强大的口令加密。

口令算法标识符

可以在 /etc/security/policy.conf 文件中指定站点的算法配置。在 policy.conf 文件中，算法以其标识符命名，如下表所示。

表 2–1 口令加密算法

标识符	说明	算法手册页
1	与 BSD 和 Linux 系统上的 MD5 算法兼容的 MD5 算法。	crypt_bsdmd5(5)
2a	与 BSD 系统上的 Blowfish 算法兼容的 Blowfish 算法。	crypt_bsdbf(5)
md5	Sun MD5 算法，此算法被视为比 BSD 和 Linux 版本的 MD5 更为强大。	crypt_sunmd5(5)
__unix__	传统的 UNIX 加密算法。此算法是 policy.conf 文件中的缺省模块。	crypt_unix(5)

policy.conf 文件中的算法配置

以下说明了 policy.conf 文件中的缺省算法配置：

```
#

...

# crypt(3c) Algorithms Configuration

#

# CRYPT_ALGORITHMS_ALLOW specifies the algorithms that are allowed to

# be used for new passwords.  This is enforced only in crypt_gensalt(3c).

#

CRYPT_ALGORITHMS_ALLOW=1,2a,md5
```

```
# To deprecate use of the traditional unix algorithm, uncomment below

# and change CRYPT_DEFAULT= to another algorithm.  For example,

# CRYPT_DEFAULT=1 for BSD/Linux MD5.

#

#CRYPT_ALGORITHMS_DEPRECATE=__unix__



# The Solaris default is the traditional UNIX algorithm.  This is not

# listed in crypt.conf(4) since it is internal to libc.  The reserved

# name __unix__ is used to refer to it.

#

CRYPT_DEFAULT=__unix__

...
```

更改 CRYPT_DEFAULT 的值之后，将使用与新值关联的算法对新用户的口令进行加密。当前用户更改其口令之后，其旧口令的加密方式会影响加密新口令所用的算法。

例如，假设 CRYPT_ALGORITHMS_ALLOW=1,2a,md5，CRYPT_DEFAULT=1。下表说明了将使用何种算法来生成加密口令。

标识符=口令算法		
初始口令	更改后的口令	说明
1 = crypt_bsdmd5	使用同一算法	1 标识符也是 CRYPT_DEFAULT 的值。继续使用 crypt_bsdmd5 算法对用户口令进行加密。
2a = crypt_bsdbf	使用同一算法	2a 标识符位于 CRYPT_ALGORITHMS_ALLOW 列表中。因此，使用 crypt_bsdbf 算法对新口令进行加密。
md5 = crypt_md5	使用同一算法	md5 标识符位于 CRYPT_ALGORITHMS_ALLOW 列表中。因此，使用 crypt_md5 算法对新口令进行加密。
__unix__ = crypt_unix	使用 crypt_bsdmd5 算法	__unix__ 标识符不在 CRYPT_ALGORITHMS_ALLOW 列表中。因此，不能使用 crypt_unix 算法。将使用 CRYPT_DEFAULT 算法对新口令进行加密。

有关配置算法选择的更多信息，请参见 policy.conf(4) 手册页。要指定口令加密算法，请参见第 61 页中的 “更改口令算法（任务列表）” 。

特殊的系统登录

访问系统的两种常用方法是使用普通用户登录或者使用 root 登录。此外，用户还可以使用许多特殊的**系统**登录来运行管理命令，而无需使用 root 帐户。作为系统管理员，您可以为这些登录帐户指定口令。

下表列出了一些系统登录帐户及其用法。这些系统登录可执行特殊的功能。每种登录都有各自的组标识 (group identification, GID) 号。每种登录都应该具有各自的口令，此口令应该只透露给需要知道的用户。

表 2–2 系统登录帐户及其用法

登录帐户	GID	用法
root	0	几乎没有任何限制。可覆盖所有其他登录、保护和权限。root 帐户具有访问整个系统的权限。应该非常谨慎地保护 root 登录口令。root 帐户（即超级用户）有权使用大多数 Solaris 命令。
daemon	1	控制后台处理。
bin	2	有权使用部分 Solaris 命令。
sys	3	有权访问许多系统文件。
adm	4	有权访问某些管理文件。
lp	71	有权访问打印机的对象数据文件和假脱机数据文件。
uucp	5	有权访问 UUCP（UNIX 对 UNIX 复制程序）的对象数据文件和假脱机数据文件。
nuucp	9	供远程系统用于登录到系统并启动文件传输。

远程登录

远程登录为入侵者提供了可乘之机。Solaris OS 提供了多个命令来监视、限制和禁用远程登录。有关过程，请参见第 54 页中的 “保证登录和口令的安全（任务列表）” 。

缺省情况下，远程登录无法获取控制或读取特定系统设备（如系统鼠标、键盘、帧缓冲区或音频设备）的权限。有关更多信息，请参见 logindevperm(4) 手册页。

拨号登录

如果能够通过调制解调器或拨号端口访问计算机，则可以添加额外的安全层。您可以要求通过调制解调器或拨号端口访问系统的用户提供**拨号口令**。拨号口令是附加口令，用户必须提供此口令之后才能被授予访问系统的权限。

仅有超级用户或具有等效功能的角色才能创建或更改拨号口令。要确保系统的完整性，应该大约每月更改一次口令。此功能最有效的功用就是要求提供拨号口令来获取访问网关系统的权限。有关如何设置拨号口令的信息，请参见第 59 页中的 “如何创建拨号口令”。

创建拨号口令涉及两个文件：/etc/dialups 和 /etc/d_passwd。dialups 文件包含需要拨号口令的端口的列表。d_passwd 文件包含需要加密口令作为附加拨号口令的 shell 程序的列表。这两个文件中的信息按如下方式处理：

- 如果 /etc/passwd 中的用户登录 shell 与 /etc/d_passwd 中的某项相匹配，则用户必须提供拨号口令。
- 如果在 /etc/d_passwd 中无法找到 /etc/passwd 中的用户登录 shell，则用户必须提供缺省口令。缺省口令是用于 /usr/bin/sh 的项。
- 如果 /etc/passwd 中的登录 shell 字段为空，则用户必须提供缺省口令。缺省口令是用于 /usr/bin/sh 的项。
- 如果 /etc/d_passwd 没有用于 /usr/bin/sh 的项，则对于其登录 shell 字段在 /etc/passwd 中为空或者不与 /etc/d_passwd 中的任何项相匹配的那些用户，系统不会提示输入拨号口令。
- 如果 /etc/d_passwd 仅包含 /usr/bin/sh:*: 项，则会禁用拨号登录。

控制对设备的访问

连接到计算机系统的外围设备会引起安全风险。麦克风可以获取会话内容并将其传输到远程系统。CD-ROM 可能会保存其信息供下一个 CD-ROM 设备用户读取。打印机可以从远程访问。系统不可或缺的设备也可能会出现安全问题。例如，网络接口（如 hme0）被视为不可或缺的设备。

Solaris 软件提供了两种方法来控制对设备的访问。**设备策略**可限制或防止对构成系统的设备进行访问。设备策略在内核中强制执行。**设备分配**可限制或防止对外围设备进行访问。设备分配在用户分配时强制执行。

设备策略在内核中使用 privilege（权限）来保护选定设备。例如，hme 之类的网络接口的设备策略要求具有所有权限才能进行读取或写入。

设备分配使用授权来保护外围设备，如打印机或麦克风。缺省情况下，不会启用设备分配。一旦启用，便可配置设备分配，以防止使用设备或者要求仅有获得授权才能访问该设备。当分配某个设备以供使用时，在当前用户解除分配此设备之前，其他用户将无法访问此设备。

可以在多个区域中配置 Solaris 系统以控制对设备的访问：

- **设置设备策略**－在 Solaris 10 发行版中，可以要求使用一个权限集来运行要访问特定设备的进程。没有这些权限的进程将无法使用设备。在系统引导时，Solaris 软件将配置设备策略。可以在安装期间为第三方驱动程序配置设备策略。安装之后，您可以作为系统管理员将设备策略添加到设备中。

- **使设备可分配**－启用设备分配之后，可以限制一台设备在某一时间只能由一个用户使用。可以进一步要求用户实现某些安全要求。例如，可以要求用户在获取授权后再使用设备。
- **禁止使用设备**－可以禁止计算机系统上的任何用户使用某一设备（如麦克风）。服务站系统可能是一个用于使特定的设备不可用的不错选择。
- **将设备限定到特殊区域**－可以指定在某个非全局区域中使用设备。有关更多信息，请参见《System Administration Guide: Solaris Containers-Resource Management and Solaris Zones》中的“ Device Use in Non-Global Zones”。有关设备和区域的更多基本介绍，请参见《System Administration Guide: Solaris Containers-Resource Management and Solaris Zones》中的“Configured Devices in Zones”。

设备策略（概述）

使用设备策略机制，可以指定打开设备的进程需要特定权限。仅有正在使用设备策略指定的权限运行的进程才能访问受设备策略保护的设备。Solaris OS 提供了缺省设备策略。例如，`hme0` 之类的网络接口要求对其进行访问的进程使用 `net_rawaccess` 权限运行。这一要求是在内核中强制执行的。有关权限的更多信息，请参见第 177 页中的 “权限（概述）”。

在早期的 Solaris OS 发行版中，设备节点仅受文件权限的保护。例如，组 `sys` 拥有的设备只能由 `sys` 组的成员打开。在 Solaris 10 发行版中，文件权限不会预测哪些用户可打开设备，而是使用文件权限**和**设备策略来保护设备。例如，`/dev/ip` 文件具有 `666` 项权限。但是，设备只能由具有相应权限的进程打开。

可以对设备策略的配置进行审计。`AUE_MODDEVPLCY` 审计事件可记录设备策略的更改。

有关设备策略的更多信息，请参见以下内容：

- 第 71 页中的 “配置设备策略（任务列表）”
- 第 87 页中的 “设备策略命令”
- 第 182 页中的 “权限和设备”

设备分配（概述）

使用设备分配机制，可以限制对外围设备（如 CD-ROM）的访问。可以在本地管理此机制。如果未启用设备分配，则外围设备仅受文件权限的保护。例如，外围设备在缺省情况下可供以下用户使用：

- 任何可以对软盘或 CD-ROM 进行读写操作的用户。
- 任何可以连接麦克风的用户。
- 任何可以访问已连接的打印机的用户。

设备分配可以限制设备仅由授权用户使用，还可以完全阻止对设备进行访问。分配了设备的用户对设备具有独占使用权，直到此用户解除分配此设备为止。解除分配设备时，设备清除脚本会删除任何剩余数据。可以编写设备清除脚本来清除没有脚本的设备中的信息。有关示例，请参见第 92 页中的 “编写新的设备清理脚本”。

可以对分配设备、解除分配设备以及列出可分配设备的尝试进行审计。审计事件是 ot 审计类的一部分。

有关设备分配的更多信息，请参见以下内容：

- 第 76 页中的 “管理设备分配（任务列表）”
- 第 87 页中的 “设备分配”
- 第 88 页中的 “设备分配命令”

控制对计算机资源的访问

作为系统管理员，您可以控制和监视系统活动。可以针对何人使用哪些资源设置限制。可以记录资源的使用，并可以监视正在使用资源的用户，还可以设置计算机以最大程度减少资源的不正确使用。

限制和监视超级用户

系统要求提供 root 口令才能进行超级用户访问。在缺省配置中，用户无法以 root 的身份远程登录系统。远程登录时，用户必须使用其用户名登录，然后使用 su 命令成为 root。可以监视使用 su 命令的用户，特别是那些尝试获取超级用户访问权限的用户。有关监视用户以及限制超级用户访问权限的过程，请参见第 66 页中的 “监视和限制超级用户（任务列表）”。

配置基于角色的访问控制以替换超级用户

基于角色的访问控制或 RBAC 旨在限制超级用户的功能。超级用户（即 root 用户）具有访问系统中所有资源的权限。借助 RBAC，可以使用一组具有单独权限的角色来替换 root。例如，可以设置一个角色来处理用户帐户创建，设置另一个角色来处理系统文件修改。创建了处理一种或一组功能的角色之后，即可从 root 的功能中删除这些功能。

每个角色都要求已知的用户使用自己的用户名和口令进行登录。登录之后，用户即可承担具有特定角色口令的角色。因此，知道 root 口令的用户只有有限的系统破坏力。有关 RBAC 的更多信息，请参见第 169 页中的 “基于角色的访问控制（概述）”。

防止无意中误用计算机资源

您可以通过以下方法来防止您自己和您的用户导致意外的错误：

- 可以通过正确设置 PATH 变量来阻止运行特洛伊木马程序。
- 可以为用户指定受限 shell。受限 shell 通过控制用户仅访问其工作所需的那些系统部分来防止出现用户错误。事实上，通过谨慎设置，可以确保用户仅访问有助于提高其工作效率的那些系统部分。

- 可以针对用户无需访问的文件设置限制性权限。

设置 PATH 变量

应该谨慎地正确设置 PATH 变量。否则，可能会无意中运行由其他用户引入的程序。入侵程序会破坏数据或损坏系统。这类对安全性构成威胁的程序称为**特洛伊木马程序**。例如，一个替代的 su 程序可能会放置在公共目录中，而您作为系统管理员可能会运行此替代程序。这类脚本看上去正好类似于常规的 su 命令。由于这类脚本会在执行之后自行删除，因此，几乎没有明确迹象表明您实际上已运行了特洛伊木马程序。

PATH 变量在登录时自动设置。路径通过以下启动文件设置：.login、.profile 和 .cshrc。当设置用户搜索路径以便将当前目录 (.) 置于最后时，系统会阻止运行这类特洛伊木马程序。供超级用户使用的 PATH 变量根本不应该包括当前目录。

自动安全性增强工具 (Automated Security Enhancement Tool, ASET) 会检查启动文件以确保 PATH 变量设置正确。ASET 还可确保 PATH 变量不包含点 (.) 项。

为用户指定受限 Shell

标准 shell 允许用户打开文件，执行命令等。受限 shell 可限制用户更改目录和执行命令。可使用 /usr/lib/rsh 命令调用受限 shell。请注意，受限 shell 并不是远程 shell，后者为 /usr/sbin/rsh。受限 shell 在以下方面不同于标准 shell：

- 受限用户只能使用其起始目录，因此无法使用 cd 命令来更改目录。因此，这类用户无法浏览系统文件。
- 受限用户无法更改 PATH 变量，因此这类用户只能使用系统管理员设置的路径中的命令。受限用户也无法使用完整路径名来执行命令或脚本。
- 用户不能使用 > 或 >> 来重定向输出。

使用受限 shell，可以限制用户对系统文件的访问能力。此 shell 将为需要执行特定任务的用户创建一种限制环境。但是，受限 shell 并非完全安全，它仅旨在防止不熟练的用户无意中造成破坏。

有关受限 shell 的信息，请使用 man -s1m rsh 命令查看 rsh(1M) 手册页。

受限 shell 的更安全的替代项是 Solaris 安全 Shell 中的 ssh 命令。使用 Solaris 安全 Shell，用户可以通过不安全的网络安全地访问远程主机。有关使用 Solaris 安全 Shell 的信息，请参见第 19 章。

限制对文件中数据的访问

由于 Solaris OS 是多用户环境，因此，文件系统的安全性是系统中最基本的安全风险。可以使用传统的 UNIX 文件保护措施来保护文件，还可以使用更安全的访问控制列表 (access control list, ACL)。

您可能需要允许某些用户读取某些文件，同时为另一些用户提供更改或删除某些文件的权限。您可能具有某些不希望其他任何用户查看的数据。第 6 章介绍了如何设置文件权限。

限制 setuid 可执行文件

可执行文件可能存在安全风险。许多可执行程序必须以 root 的身份（即超级用户身份）执行才能正常工作。运行这些 setuid 程序需要将用户 ID 设置为 0。运行这些程序的任何用户都使用 root ID 运行程序。如果在编写使用 root ID 运行的程序时未注重安全性，则此程序将引起潜在的安全问题。

除了 Sun 附带的 setuid 位设置为 root 的可执行程序之外，还应该禁止使用 setuid 程序。如果无法禁止使用 setuid 程序，则至少应该限制其使用。安全管理很少需要使用 setuid 程序。

有关更多信息，请参见第 127 页中的 “防止可执行文件危及安全”。有关过程，请参见第 140 页中的 “防止程序受到安全风险（任务列表）”。

使用自动安全性增强工具

ASET 安全软件包提供可用于控制和监视系统安全性的自动管理工具。ASET 提供了三种安全级别：低、中和高。可以指定一种 ASET 安全级别。级别越高，ASET 对文件的控制功能越强，从而可限制访问文件并提高系统的安全性。有关更多信息，请参见第 7 章。

使用 Solaris 安全工具包

虽然 ASET 可用于对系统进行少量安全更改，但 Solaris 安全工具包提供了一种灵活且可扩展的机制，用于最小化、加强和保护 Solaris 系统。Solaris 安全工具包（非正式名称为 JASS 工具包）是一种由用户用来对系统执行安全修改的工具。此工具可以提供有关系统安全状态的报告，还可以撤消工具以前运行的内容。JASS 工具包可以从 Sun 的 Web 站点 http://wwws.sun.com/security/jass 下载。此 Web 站点包含指向联机文档的链接。

Alex Noordergraaf 和 Glenn Brunette 合著的《Securing Systems with the Solaris Security Toolkit》（ISBN 0-13-141071-7，2003 年 6 月）中对此工具包进行了详细说明。本书是 Sun Microsystems Press 出版的 Sun BluePrints 系列的一部分。

使用 Solaris 资源管理功能

Solaris 软件提供了完善的资源管理功能。使用这些功能，可以按照服务器整合环境中的应用程序对资源使用进行分配、安排、监视以及设置上限操作。使用资源控制框架，可以对进程占用的系统资源设置约束。这类约束有助于防止尝试使系统资源发生泛洪的脚本引起的拒绝服务攻击。

使用 Solaris 资源管理功能，可以为特殊项目指定资源，还可以动态调整可用资源。有关更多信息，请参见《System Administration Guide: Solaris Containers-Resource Management and Solaris Zones》中的第一部分，“Resource Management”。

使用 Solaris Zones

Solaris Zones 提供了一种应用程序执行环境，其中单个 Solaris OS 实例中各进程与系统的其余部分隔离开来。这种隔离阻止了在一个区域中运行的进程监视或影响在其他区域中运行的进程。即使运行的进程具有超级用户功能，也不能查看或影响其他区域中的活动。

Solaris Zones 非常适用于将多个应用程序置于一台服务器中的环境。有关更多信息，请参见《System Administration Guide: Solaris Containers-Resource Management and Solaris Zones》中的第二部分，“Zones”。

监视计算机资源的使用情况

作为系统管理员，需要监视系统活动。您需要注意计算机的所有方面，包括以下各项：

- 什么是正常负载?
- 何人有权访问系统?
- 个人何时访问系统?
- 系统上通常运行什么程序?

了解这些知识之后，即可使用可用工具来审计系统使用情况并监视个别用户的活动。如果怀疑安全性受到破坏，则使用监视功能会非常有用。有关审计服务的更多信息，请参见第 27 章。

监视文件完整性

作为系统管理员，需要确保所管理的系统上安装的文件尚未被意外更改。在大型安装中，使用用于对每个系统上的软件栈进行比较和报告的工具可跟踪系统。使用基本审计报告工具 (Basic Audit Reporting Tool, BART)，可以通过在一段时间内对一个或多个系统执行文件级检查来全面验证系统。一个或多个系统的 BART **清单**在一段时间内的变化可以验证系统的完整性。BART 提供了清单创建、清单比较以及脚本报告规则。有关更多信息，请参见第 5 章。

控制对文件的访问

Solaris OS 是一种多用户环境。在多用户环境中，所有登录到系统的用户均可读取属于其他用户的文件。使用相应的文件权限，用户还可以使用属于其他用户的文件。有关更多介绍，请参见第 6 章。有关针对文件设置相应权限的逐步说明，请参见第 127 页中的 “保护文件（任务列表）”。

通过加密保护文件

通过使其他用户无法访问某个文件，可以保证此文件的安全。例如，具有权限 `600` 的文件无法由其属主和超级用户之外的用户读取。具有权限 `700` 的目录同样无法访问。但是，猜

中您的口令或者知道 root 口令的用户可以访问此文件。另外，每次将系统文件备份到脱机介质时，还可以在备份磁带上保留原本无法访问的文件。

Solaris 加密框架提供了 digest、 mac 和 encrypt 命令来保护文件。有关更多信息，请参见第 13 章。

使用访问控制列表

ACL（发音为“ackkl”）可以对文件权限进行更多控制。如果传统的 UNIX 文件保护无法提供足够的保护，则可添加 ACL。传统的 UNIX 文件保护可为三种用户类提供读取、写入和执行权限：属主、组和其他用户。ACL 可提供更为精确的安全性。可以使用 ACL 定义以下文件权限：

- 属主文件权限
- 属主组的文件权限
- 不属于属主组的其他用户的文件权限
- 特定用户的文件权限
- 特定组的文件权限
- 先前每个类别的缺省权限

有关使用 ACL 的更多信息，请参见第 125 页中的 “使用访问控制列表保护文件”。

跨计算机共享文件

网络文件服务器可以控制可共享的文件，还可以控制有权访问文件的客户机，以及允许这些客户机使用的访问类型。通常，文件服务器可以为所有客户机或特定客户机授予读写权限或只读权限。通过 share 命令使资源可用时，便会指定访问控制。

文件服务器上的 /etc/dfs/dfstab 文件列出了此服务器为网络中的客户机提供的文件系统。有关共享文件系统的更多信息，请参见《System Administration Guide: Network Services》中的“Automatic File-System Sharing”。

限制对共享文件的 root 访问

通常，不允许超级用户对通过网络共享的文件系统进行 root 访问。NFS 系统通过将请求程序的用户更改为具有用户 ID 60001 的用户 nobody，可以防止以 root 身份对已挂载的文件系统进行访问。用户 nobody 的访问权限与为公众提供的那些访问权限相同。用户 nobody 具有的访问权限与没有凭证的用户所具有的访问权限相同。例如，如果公众只对某个文件具有执行权限，则用户 nobody 只能执行此文件。

NFS 服务器可以按主机授予超级用户对共享文件系统的权限。要授予这些权限，请使用 share 命令的 root=*hostname* 选项。应该谨慎使用此选项。有关 NFS 的安全选项的介绍，请参见《System Administration Guide: Network Services》中的第 6 章，“Accessing Network File Systems (Reference)”。

控制网络访问

计算机常常是其他计算机配置的一部分。此配置称为**网络**。连接的计算机可通过网络交换信息。联网的计算机可以访问网络中其他计算机的数据和其他资源。计算机网络创建了一种强大且完善的计算环境。但是，网络同时也使计算机安全性变得更为复杂。

例如，在计算机网络中，独立的计算机都允许共享信息。未经授权的访问存在安全风险。由于许多用户都有权访问网络，因此更可能会出现未经授权的访问，尤其是由于用户错误进行的访问。口令使用不当也会导致未经授权的访问。

网络安全性机制

网络安全性通常基于限制或阻止来自远程系统的操作。下图介绍了可以对远程操作强制执行的安全限制。

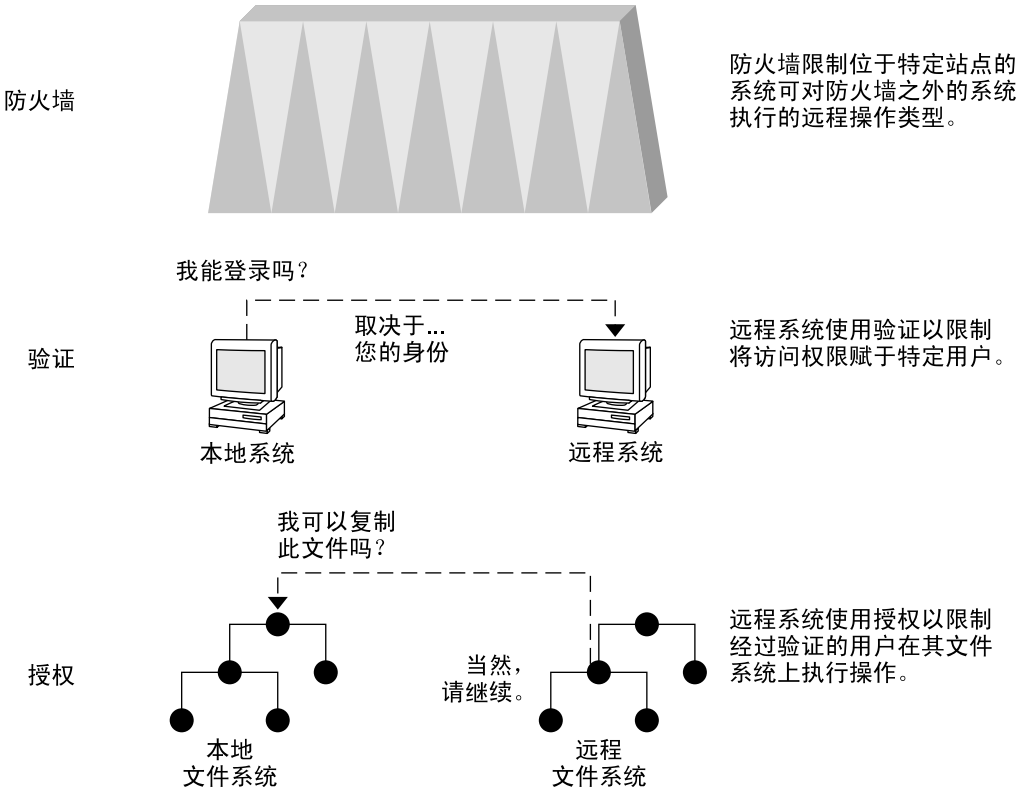


图 2–1 远程操作的安全限制

远程访问的验证和授权

验证是一种在特定用户访问远程系统时，限制这些用户的访问权限的方法。可以同时在系统级别和网络级别设置验证。**授权**是一种在授予用户访问远程系统的权限之后，限制此用户可执行的操作的方法。下表列出了可提供验证和授权的服务。

表 2–3 远程访问的验证和授权服务

服务	说明	更多信息
IPsec	IPsec 提供了基于主机和基于证书的验证以及网络通信流量加密。	《System Administration Guide: IP Services》中的第 19 章，“IP Security Architecture (Overview)”
Kerberos	Kerberos 使用加密功能对登录系统的用户进行验证和授权。	有关示例，请参见第 346 页中的 “Kerberos 服务的工作方式” 。
LDAP 和 NIS+	LDAP 目录服务和 NIS+ 名称服务可以提供网络级别的验证和授权。	《System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)》 和《System Administration Guide: Naming and Directory Services (NIS+)》
远程登录命令	使用远程登录命令，用户可以通过网络登录到远程系统并使用其资源。`rlogin`、`rcp` 和 `ftp` 是一些远程登录命令。如果是“受信任主机”，则会自动执行验证。否则，系统会要求进行自我验证。	《System Administration Guide: Network Services》中的第 29 章，“Accessing Remote Systems (Tasks)”
SASL	简单身份验证和安全层 (Simple Authentication and Security Layer, SASL) 是一种为网络协议提供验证和可选安全性服务的框架。可以使用插件来选择相应的验证协议。	第 307 页中的 “SASL（概述）”
安全 RPC	安全 RPC 通过对远程计算机发出请求的用户进行验证，可提高网络环境的安全性。可以使用 UNIX、DES 或 Kerberos 验证系统来实现安全 RPC。	第 285 页中的 “安全 RPC 概述”
	安全 RPC 还可用于在 NFS 环境中提供额外的安全性。具有安全 RPC 的 NFS 环境称为安全 NFS。安全 NFS 针对公钥使用 Diffie-Hellman 验证。	第 285 页中的 “NFS 服务和安全 RPC”
Solaris 安全 Shell	Solaris 安全 Shell 可以对不安全网络上的网络通信流量进行加密。Solaris 安全 Shell 通过单独使用口令、公钥或同时使用这两者来提供验证。Solaris 安全 Shell 针对公钥使用 RSA 和 DSA 验证。	第 311 页中的 “Solaris 安全 Shell（概述）”

安全 RPC 的可能替代项是 Solaris **特权端口**机制。为特权端口指定的端口号小于 `1024`。客户机系统验证客户机的凭证之后，此客户机便会使用特权端口与服务器建立连接。然后，服务器通过检查连接的端口号来检验客户机凭证。

未运行 Solaris 软件的客户机可能无法使用特权端口进行通信。如果客户机无法通过此端口进行通信，则会显示类似以下内容的错误消息：

“Weak Authentication

NFS request from unprivileged port”

防火墙系统

可以设置防火墙系统来防止外部对网络中的资源进行访问。**防火墙系统**是一台安全主机，可充当内部网络与外部网络之间的屏障。内部网络将所有其他网络均视为不可信对象。应该考虑将此设置作为内部网络和任何与其进行通信的外部网络（如 Internet）之间的强制性设置。

防火墙可充当网关和屏障，并可充当在网络之间传递数据的网关，还可充当阻止与网络来回自由传递数据的屏障。防火墙要求内部网络中的用户登录到防火墙系统之后才能访问远程网络中的主机。同样，外部网络中的用户必须先登录到防火墙系统，然后才会被授予访问内部网络中主机的权限。

防火墙还可用于某些内部网络之间。例如，可以设置防火墙或安全网关计算机来限制包的传送。如果网关计算机不是包的源地址或目标地址，则网关可以禁止两个网络之间的包交换。防火墙还应设置为仅转发特定协议的包。例如，可以允许包传送邮件，但不允许传送 `telnet` 或 `rlogin` 命令。ASET 以高安全级别运行时，会禁用 Internet 协议 (Internet Protocol, IP) 包的转发。

此外，从内部网络发送的所有电子邮件均会首先发送到防火墙系统。然后，防火墙将邮件传送给外部网络中的主机。防火墙系统还会接收所有传入的电子邮件，并将邮件分发给内部网络中的主机。

注意 – 防火墙可阻止未经授权的用户访问网络中的主机。应该严格维护对防火墙强制执行的安全性，但对网络中其他主机的安全性限制可以较为宽松。但是，突破防火墙的入侵者可以获取访问内部网络中所有其他主机的权限。

防火墙系统不应该包含任何受信任主机。**受信任主机**是指不要求用户提供口令即可从其中进行登录的主机。防火墙系统不应该共享其任何文件系统，也不应该挂载其他服务器的任何文件系统。

可以使用以下技术来通过防火墙加强系统的安全性：

- 通过 ASET 对防火墙系统强制执行高安全性，如第 7 章中所述。
- 通过 Solaris 安全工具包（非正式名称为 JASS 工具包）使用防火墙来加强 Solaris 系统的安全性。此工具包可以从 Sun 的 Web 站点 `http://wwws.sun.com/security/jass` 下载。
- 通过 IPsec 和 Solaris IP 过滤器提供防火墙保护。有关保护网络通信流量的更多信息，请参见《System Administration Guide: IP Services》中的第四部分，“IP Security”。

加密和防火墙系统

大多数局域网可以通过称为**包**的块在计算机之间传输数据。通过称为**包粉碎**的过程，网络之外的未经授权用户可能会破坏或损坏数据。

包粉碎涉及在包到达其目标之前捕获这些包。然后，入侵者会向内容中加入任意数据，并将这些包发送回其原始路线。在局域网中，不可能粉碎包，因为包会同时到达所有系统（包括服务器）。但是，可以在网关上粉碎包，因此请确保网络中的所有网关都受到保护。

大多数危险的攻击都会影响数据的完整性。这类攻击包括更改包的内容或者模拟用户。涉及窃听的攻击不会破坏数据的完整性。窃听者会记录会话内容以便稍后重放，但不会模拟用户。尽管窃听攻击不会攻击数据的完整性，但是会影响保密性。可以对通过网络的数据进行加密来保护敏感信息的保密性。

- 要对不安全网络上的远程操作进行加密，请参见第 18 章。
- 要对网络中的数据进行加密和验证，请参见第 20 章。
- 要对 IP 数据报进行加密，请参见《System Administration Guide: IP Services》中的第 19 章，“IP Security Architecture (Overview)”。

报告安全问题

如果您遇到可疑的安全性破坏，则可以与计算机应急响应组/协调中心 (Computer Emergency Response Team/Coordination Center, CERT/CC) 联系。CERT/CC 是国防部高级研究项目署 (Defense Advanced Research Projects Agency, DARPA) 资助的项目，设在卡耐基梅隆大学的软件工程学院。此项目署可以您帮助解决遇到的任何安全问题，还可以指示您与其他更适合特殊要求的计算机应急响应组联系。您可以通过全天候热线致电 CERT/CC：(412) 268-7090。或者，向 `cert@cert.sei.cmu.edu` 发送电子邮件与此响应组联系。

◆ ◆ ◆ 第 3 章

3 控制对系统的访问（任务）

本章介绍用于控制可以访问 Solaris 系统的用户的过程。以下是本章中信息的列表：

- 第 53 页中的 “控制系统访问（任务列表）”
- 第 54 页中的 “保证登录和口令的安全（任务列表）”
- 第 61 页中的 “更改口令算法（任务列表）”
- 第 66 页中的 “监视和限制超级用户（任务列表）”
- 第 68 页中的 “SPARC: 控制对系统硬件的访问（任务列表）”

有关系统安全的概述信息，请参见第 2 章。

控制系统访问（任务列表）

一台计算机的安全性取决于其最薄弱的登录点。以下任务列表说明了应监视和确保安全的若干方面：

任务	说明	参考
监视、允许和拒绝用户登录	监视异常登录活动。临时禁止登录。管理拨号登录。	第 54 页中的 “保证登录和口令的安全（任务列表）”
提供强口令加密	指定用于加密用户口令的算法。安装其他算法。	第 61 页中的 “更改口令算法（任务列表）”
监视和限制超级用户活动	定期监视超级用户活动。禁止 root 用户远程登录。	第 66 页中的 “监视和限制超级用户（任务列表）”
禁止访问硬件设置	禁止普通用户使用 PROM。	第 68 页中的 “SPARC: 控制对系统硬件的访问（任务列表）”

保证登录和口令的安全（任务列表）

以下任务列表说明监视用户登录和禁用用户登录的过程。

任务	说明	参考
显示用户的登录状态	列出有关用户登录帐户的详细信息，如全名和口令生命期信息。	第 54 页中的 “如何显示用户的登录状态”
查找没有口令的用户	仅查找其帐户无需口令的那些用户。	第 55 页中的 “如何显示没有口令的用户”
临时禁用登录	拒绝用户在系统关闭或例行维护时登录到计算机。	第 56 页中的 “如何临时禁止用户登录”
保存失败的登录尝试	为五次登录尝试后仍不能提供正确口令的用户创建日志。	第 57 页中的 “如何监视失败的登录尝试”
保存所有失败的登录尝试	为失败的登录尝试创建日志。	第 58 页中的 “如何监视所有失败的登录尝试”
创建拨号口令	要求通过调制解调器或拨号端口远程登录的用户提供其他口令。	第 59 页中的 “如何创建拨号口令”
临时禁用拨号登录	禁止用户通过调制解调器或端口远程拨入。	第 61 页中的 “如何临时禁用拨号登录”

保证登录和口令的安全

您可以限制远程登录和要求用户具有口令，也可以监视失败的访问尝试以及临时禁用登录。

▼ 如何显示用户的登录状态

1. **承担主管理员角色，或成为超级用户。**

   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2. **使用 `logins` 命令显示用户的登录状态。**

   ```
   # logins -x -l username
   ```

   `-x` 显示一组扩展的登录状态信息。

   `-l` *username* 显示指定用户的登录状态。变量 *username* 是用户的登录名称。必须以逗号分隔的列表形式指定多个登录名称。

logins 命令使用相应的口令数据库来获取用户的登录状态。该数据库可以是本地 /etc/passwd 文件，也可以是名称服务的口令数据库。有关更多信息，请参见 logins(1M) 手册页。

示例 3–1 显示用户的登录状态

在以下示例中，显示了用户 rimmer 的登录状态。

```
# logins -x -l rimmer

rimmer       500     staff           10   Annalee J. Rimmer

                     /export/home/rimmer

                     /bin/sh

                     PS 010103 10 7 -1
```

rimmer	标识用户的登录名称。
500	标识用户 ID (user ID, UID)。
staff	标识用户的主组。
10	标识组 ID (group ID, GID)。
Annalee J. Rimmer	标识注释。
/export/home/rimmer	标识用户的起始目录。
/bin/sh	标识登录 shell。
PS 010170 10 7 -1	指定口令生命期信息： ■ 上次更改口令的日期 ■ 更改之间要求的天数 ■ 在该天数后必须更改 ■ 警告期

▼ 如何显示没有口令的用户

1. **承担主管理员角色，或成为超级用户。**

   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2. **使用 logins 命令显示所有没有口令的用户。**

   ```
   # logins -p
   ```

-p 选项用于显示没有口令的用户列表。除非启用了名称服务，否则 logins 命令将使用本地系统的口令数据库。

示例 3–2 显示没有口令的用户

在以下示例中，用户 pmorph 没有口令。

```
# logins -p

pmorph          501     other           1       Polly Morph

#
```

▼ 如何临时禁止用户登录

在系统关闭或例行维护期间，可临时禁止用户登录。超级用户登录将不受影响。有关更多信息，请参见 nologin(4) 手册页。

1. **承担主管理员角色，或成为超级用户。**

   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2. **在文本编辑器中创建 /etc/nologin 文件。**

   ```
   # vi /etc/nologin
   ```

3. **添加有关系统可用性的消息。**

4. **关闭并保存该文件。**

示例 3–3 禁止用户登录

在此示例中，用户将收到系统不可用的通知。

```
# vi /etc/nologin

(Add system message here)



# cat /etc/nologin

***No logins permitted.***
```

```
***The system will be unavailable until 12 noon.***
```

您也可以将系统引导至运行级 0（单用户模式）以禁止登录。有关将系统引导至单用户模式的信息，请参见《System Administration Guide: Basic Administration》中的第 9 章，“Shutting Down a System (Tasks)”。

▼ 如何监视失败的登录尝试

此过程从终端窗口捕获失败的登录尝试。此过程不会从 CDE 或 GNOME 登录尝试中捕获失败的登录。

1. **承担主管理员角色，或成为超级用户。**

   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2. **在 `/var/adm` 目录中创建 `loginlog` 文件。**

   ```
   # touch /var/adm/loginlog
   ```

3. **在 `loginlog` 文件中，为 `root` 用户设置读写权限。**

   ```
   # chmod 600 /var/adm/loginlog
   ```

4. **在 `loginlog` 文件中，将组成员关系更改为 `sys`。**

   ```
   # chgrp sys /var/adm/loginlog
   ```

5. **检验日志是否正常工作。**

   例如，使用错误的口令五次登录系统。然后，显示 `/var/adm/loginlog` 文件。

   ```
   # more /var/adm/loginlog

   jdoe:/dev/pts/2:Tue Nov  4 10:21:10 2003

   jdoe:/dev/pts/2:Tue Nov  4 10:21:21 2003

   jdoe:/dev/pts/2:Tue Nov  4 10:21:30 2003

   jdoe:/dev/pts/2:Tue Nov  4 10:21:40 2003

   jdoe:/dev/pts/2:Tue Nov  4 10:21:49 2003

   #
   ```

在 loginlog 文件中，每次失败的尝试都对应一项。每一项都包含用户的登录名称、tty 设备以及登录尝试失败的时间。如果用户登录尝试失败的次数少于五次，则不会记录任何失败的登录尝试。

如果 loginlog 文件不断增大，则表明可能是存在侵入计算机系统的尝试。因此，应定期检查并清除该文件的内容。有关更多信息，请参见 loginlog(4) 手册页。

▼ 如何监视所有失败的登录尝试

此过程捕获 syslog 文件中所有失败的登录尝试。

1. **承担主管理员角色，或成为超级用户。**

   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2. **使用所需的 SYSLOG 和 SYSLOG_FAILED_LOGINS 值设置 /etc/default/login 文件。**

   编辑 /etc/default/login 文件以更改相应项。请确保取消对 **SYSLOG=YES** 的注释。

   ```
   # grep SYSLOG /etc/default/login

   # SYSLOG determines whether the syslog(3) LOG_AUTH facility

   # should be used

   SYSLOG=YES

   ...

   SYSLOG_FAILED_LOGINS=0

   #
   ```

3. **使用正确的权限创建文件以保存日志信息。**

   a. **在 /var/adm 目录中创建 authlog 文件。**

      ```
      # touch /var/adm/authlog
      ```

   b. **在 authlog 文件中，为 root 用户设置读写权限。**

      ```
      # chmod 600 /var/adm/authlog
      ```

   c. **在 authlog 文件中，将组成员关系更改为 sys。**

      ```
      # chgrp sys /var/adm/authlog
      ```

4 **编辑 syslog.conf 文件以记录失败的口令尝试。**

这些失败应发送到 authlog 文件。

a. **在 syslog.conf 文件中键入以下项。**

使用制表符分隔 syslog.conf 的同一行中的字段。

```
auth.notice     <按 Tab 键>  /var/adm/authlog
```

b. **刷新 syslog 守护进程的配置信息。**

```
# svcadm refresh system/system-log
```

5 **检验日志是否正常工作。**

例如，使用错误的口令以普通用户的身份登录系统。然后，以主管理员角色或超级用户身份显示 /var/adm/authlog 文件。

```
# more /var/adm/authlog

Nov  4 14:46:11 example1 login: [ID 143248 auth.notice]

 Login failure on /dev/pts/8 from example2, stacey

#
```

6 **定期监视 /var/adm/authlog 文件。**

示例 3–4 在三次登录失败后记录访问尝试

按照上述过程进行操作，但在 /etc/default/login 文件中将 SYSLOG_FAILED_LOGINS 的值设置为 3。

示例 3–5 在三次登录失败后关闭连接

在 /etc/default/login 文件中取消对 RETRIES 项的注释，然后将 RETRIES 的值设置为 3。所做编辑将立即生效。在一个会话中重试登录三次后，系统便会关闭连接。

▼ 如何创建拨号口令

注意 – 首次建立拨号口令时，务必保持登录到至少一个端口。请在其他端口上测试该口令。如果注销以测试新口令，则可能无法重新登录。如果同时还登录到另一个端口，则可返回并修复相应错误。

1. **承担主管理员角色，或成为超级用户。**

   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2. **创建一个包含串行设备列表的 `/etc/dialups` 文件。**

   添加要使用拨号口令保护的所有端口。`/etc/dialups` 文件的内容应与以下信息类似：

   ```
   /dev/term/a

   /dev/term/b

   /dev/term/c
   ```

3. **对于要求拥有拨号口令的登录程序，创建一个 `/etc/d_passwd` 文件，并在文件中添加这些程序。**

   添加用户可在登录时运行的 shell 程序，例如 `uucico`、`sh`、`ksh` 和 `csh`。`/etc/d_passwd` 文件的显示应与以下信息类似：

   ```
   /usr/lib/uucp/uucico:encrypted-password:

   /usr/bin/csh:encrypted-password:

   /usr/bin/ksh:encrypted-password:

   /usr/bin/sh:encrypted-password:
   ```

   在此过程的后续部分，将为每个登录程序添加加密口令。

4. **在这两个文件中，将拥有权设置为 `root`。**

   ```
   # chown root /etc/dialups /etc/d_passwd
   ```

5. **在这两个文件中，将组拥有权设置为 `root`。**

   ```
   # chgrp root /etc/dialups /etc/d_passwd
   ```

6. **在这两个文件中，为 `root` 设置读写权限。**

   ```
   # chmod 600 /etc/dialups /etc/d_passwd
   ```

7. **创建加密口令。**

   a. **创建临时用户。**

      ```
      # useradd username
      ```

   b. **为临时用户创建口令。**

      ```
      # passwd username
      ```

```
New Password:       <键入口令>

Re-enter new Password:       <重新键入口令>

passwd: password successfully changed for username
```

c. 捕获加密口令。

```
# grep username /etc/shadow > username.temp
```

d. 编辑 *username*.temp 文件。

删除除加密口令以外的所有字段。第二个字段保存加密口令。

例如，在以下行中，加密口令为 `U9gp9SyA/JlSk`。

```
temp:U9gp9SyA/JlSk:7967:::::7988:
```

e. 删除临时用户。

```
# userdel username
```

8 将加密口令从 *username*.temp 文件复制到 /etc/d_passwd 文件中。

可为每个登录 shell 创建不同口令。或者，对每个登录 shell 使用相同口令。

9 将该口令通知拨号用户。

应确保在通知用户的过程中口令不会被篡改。

▼ 如何临时禁用拨号登录

1 承担主管理员角色，或成为超级用户。

主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2 将下面的项作为单独的一行放入 /etc/d_passwd 文件：

```
/usr/bin/sh:*:
```

更改口令算法（任务列表）

以下任务列表说明管理口令算法的过程：

任务	参考
提供强口令加密	第 62 页中的 “如何指定口令加密算法”
为名称服务提供强口令加密	第 63 页中的 “如何为 NIS 域指定新的口令算法”
	第 63 页中的 “如何为 NIS+ 域指定新的口令算法”
	第 64 页中的 “如何为 LDAP 域指定新的口令算法”
添加新的口令加密模块	第 64 页中的 “如何安装第三方的口令加密模块”

更改口令加密的缺省算法

缺省情况下，使用 `crypt_unix` 算法加密用户口令。通过更改缺省口令加密算法，可以使用更强大的加密算法，如 MD5 或 Blowfish。

▼ 如何指定口令加密算法

在此过程中，BSD-Linux 版本的 MD5 算法是用户更改其口令时使用的缺省加密算法。此算法适合由运行 Solaris、BSD 和 Linux 版本的 UNIX 的计算机构成的混合网络。有关口令加密算法和算法标识符的列表，请参见表 2–1。

1. **承担主管理员角色，或成为超级用户。**

   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2. **指定所选加密算法的标识符。**

   键入标识符作为 `/etc/security/policy.conf` 文件中的 `CRYPT_DEFAULT` 变量的值。

   可能需要在文件中添加注释以对选择进行说明。

```
# cat  /etc/security/policy.conf

...

CRYPT_ALGORITHMS_ALLOW=1,2a,md5

#

# Use the version of MD5 that works with Linux and BSD systems.

# Passwords previously encrypted with __unix__ will be encrypted with MD5

# when users change their passwords.
```

```
#
#
CRYPT_DEFAULT=__unix__
```

CRYPT_DEFAULT=1

在此示例中，算法配置可确保不会使用功能最弱的算法 crypt_unix 加密口令。使用 crypt_unix 模块加密口令的用户将在更改其口令时获得使用 crypt_bsdmd5 加密的口令。

有关配置算法选择的更多信息，请参见 policy.conf(4) 手册页。

示例 3–6 使用 Blowfish 算法加密口令

在此示例中，Blowfish 算法的标识符 2a 被指定为 policy.conf 文件中的 CRYPT_DEFAULT 变量的值：

```
CRYPT_ALGORITHMS_ALLOW=1,2a,md5
#CRYPT_ALGORITHMS_DEPRECATE=__unix__
CRYPT_DEFAULT=2a
```

此配置与使用 Blowfish 算法的 BSD 系统兼容。

▼ 如何为 NIS 域指定新的口令算法

当 NIS 域中的用户更改其口令时，NIS 客户机会查看 /etc/security/policy.conf 文件中的本地算法配置。NIS 客户机将加密口令。

1. **在 NIS 客户机的 /etc/security/policy.conf 文件中指定口令加密算法。**
2. **将修改过的 /etc/security/policy.conf 文件复制到 NIS 域中的每台客户机。**
3. **为尽可能减少混淆，将修改过的 /etc/security/policy.conf 文件复制到 NIS 根服务器和从服务器。**

▼ 如何为 NIS+ 域指定新的口令算法

当 NIS+ 域中的用户更改其口令时，NIS+ 名称服务会查看 NIS+ 主服务器的 /etc/security/policy.conf 文件中的算法配置。运行 rpc.nispasswd 守护进程的 NIS+ 主服务器将创建加密口令。

1. **在 NIS+ 主服务器的 /etc/security/policy.conf 文件中指定口令加密算法。**

2. **为尽可能减少混淆，将 NIS+ 主服务器的 /etc/security/policy.conf 文件复制到 NIS+ 域中的每台主机。**

▼ 如何为 LDAP 域指定新的口令算法

正确配置 LDAP 客户机后，LDAP 客户机便可以使用新的口令算法。LDAP 客户机的行为与 NIS 客户机的行为相同。

1. **在 LDAP 客户机的 /etc/security/policy.conf 文件中指定口令加密算法。**

2. **将修改过的 policy.conf 文件复制到 LDAP 域中的每台客户机。**

3. **确保客户机的 /etc/pam.conf 文件不使用 pam_ldap 模块。**

   确保注释符号 (#) 位于包含 pam_ldap.so.1 的项的前面。另外，请勿将新的 server_policy 选项与 pam_authtok_store.so.1 模块一起使用。

   客户机的 pam.conf 文件中的 PAM 项允许根据本地算法配置来加密口令。PAM 项还允许验证口令。

   当 LDAP 域中的用户更改其口令时，LDAP 客户机会查看 /etc/security/policy.conf 文件中的本地算法配置。LDAP 客户机将加密口令。然后，客户机将加密过的口令连同 {crypt} 标记一起发送到服务器。该标记告知服务器该口令已加密。该口令将按原样存储在服务器上。验证时，客户机先从服务器检索存储的口令。然后，将存储的口令与其从用户键入的口令生成的加密版本进行比较。

注 – 要利用 LDAP 服务器的口令策略控制，请将 server_policy 选项与 pam.conf 文件中的 pam_authtok_store 项一起使用。这样，将使用 Sun Java™ System Directory Server 的加密机制在服务器上加密口令。有关过程，请参见《System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)》中的第 11 章，“Setting Up Sun Java System Directory Server With LDAP Clients (Tasks)”。

▼ 如何安装第三方的口令加密模块

第三方口令加密算法通常在软件包中作为模块提供。运行 pkgadd 命令时，供应商的脚本应修改 /etc/security/crypt.conf 文件。然后，您可修改 /etc/security/policy.conf 文件以包含新模块及其标识符。

1. **使用 pkgadd 命令添加软件。**

   有关如何添加软件的详细说明，请参见《System Administration Guide: Basic Administration》中的“Adding or Removing a Software Package (pkgadd)”。

2. **确认是否已添加新模块及模块标识符。**

   读取 /etc/security/crypt.conf 文件中的加密算法列表。

例如，以下行说明已安装了用于实现 crypt_rot13 算法的模块：

```
# crypt.conf

#

md5 /usr/lib/security/$ISA/crypt_md5.so

rot13 /usr/lib/security/$ISA/crypt_rot13.so



# For *BSD - Linux compatibility

# 1 is MD5,  2a is Blowfish

1 /usr/lib/security/$ISA/crypt_bsdmd5.so

2a /usr/lib/security/$ISA/crypt_bsdbf.so
```

3 **将新安装算法的标识符添加到 /etc/security/policy.conf 文件。**

以下行显示了需要修改以添加 rot13 标识符的 policy.conf 文件的摘录：

```
# Copyright 1999-2002 Sun Microsystems, Inc.  All rights reserved.

# ...

#ident  "@(#)policy.conf        1.6     02/06/07 SMI"

# ...

# crypt(3c) Algorithms Configuration

CRYPT_ALGORITHMS_ALLOW=1,2a,md5,rot13

#CRYPT_ALGORITHMS_DEPRECATE=__unix__

CRYPT_DEFAULT=md5
```

在此示例中，如果当前口令是使用 crypt_rot13 算法加密的，则使用 rot13 算法。新用户口令使用 crypt_sunmd5 算法进行加密。此算法配置适用于仅 Solaris 网络。

监视和限制超级用户（任务列表）

以下任务列表说明如何监视和限制 root 用户登录：

任务	说明	参考
监视正在使用 su 命令的用户	定期扫描 sulog 文件。	第 66 页中的 “如何监视正在使用 su 命令的用户”
在控制台上显示超级用户活动	在超级用户尝试登录时监视其访问尝试。	第 67 页中的 “如何限制和监视超级用户登录”

监视和限制超级用户

另一种使用超级用户帐户的方法是设置基于角色的访问控制。基于角色的访问控制称为 RBAC。有关 RBAC 的概述信息，请参见第 169 页中的 “基于角色的访问控制（概述）” 。要设置 RBAC，请参见第 9 章。

▼ 如何监视正在使用 su 命令的用户

sulog 文件列出了 su 命令的每次使用情况，而不仅仅包括用于从用户切换到超级用户的 su 尝试。

◗ **定期监视 /var/adm/sulog 文件的内容。**

```
# more /var/adm/sulog

SU 12/20 16:26 + pts/0 stacey-root

SU 12/21 10:59 + pts/0 stacey-root

SU 01/12 11:11 + pts/0 root-rimmer

SU 01/12 14:56 + pts/0 pmorph-root

SU 01/12 14:57 + pts/0 pmorph-root
```

这些项显示以下信息：

- 输入命令的日期和时间。
- 尝试是否成功。加号 (+) 表明尝试成功。减号 (-) 表明尝试失败。
- 发出命令的端口。
- 用户名称和切换身份的名称。

缺省情况下，通过 /etc/default/su 文件中的以下项启用此文件中的 su 记录：

```
SULOG=/var/adm/sulog
```

▼ 如何限制和监视超级用户登录

此方法可立即检测访问本地系统的超级用户尝试。

1 **查看** /etc/default/login **文件中的** CONSOLE **项。**

```
CONSOLE=/dev/console
```

缺省情况下，控制台设备设置为 /dev/console。使用此设置，root 可以登录到控制台。root 无法远程登录。

2 **检验** root **是否可以远程登录。**

从远程系统，尝试以超级用户身份登录。

```
mach2 % rlogin -l root mach1

Password:      <键入 mach1 的超级用户口令>

Not on system console

Connection closed.
```

3 **监视成为超级用户的尝试。**

缺省情况下，会使用 SYSLOG 实用程序在控制台上列显成为超级用户的尝试。

a. **在桌面上打开终端控制台。**

b. **在另一个窗口中，使用** su **命令成为超级用户。**

```
% su

Password:      <键入超级用户口令>

#
```

将在终端控制台上列显一条消息。

```
Sep 7 13:22:57 mach1 su: 'su root' succeeded for jdoe on /dev/pts/6
```

示例 3–7 记录超级用户访问尝试

在此示例中，SYSLOG 不会记录超级用户尝试。因此，管理员将通过删除 /etc/default/su 文件中的 #CONSOLE=/dev/console 项的注释来记录这些尝试。

```
# CONSOLE determines whether attempts to su to root should be logged

# to the named device

#
```

```
CONSOLE=/dev/console
```

当用户尝试成为超级用户时，将在终端控制台上列显该尝试。

```
SU 09/07 16:38 + pts/8 jdoe-root
```

故障排除 要在 /etc/default/login 文件包含缺省 CONSOLE 项的情况下从远程系统成为超级用户，用户必须先使用其用户名登录。使用其用户名登录后，用户便可以使用 su 命令成为超级用户。

如果控制台显示类似于 `Mar 16 16:20:36 mach1 login: ROOT LOGIN /dev/pts/14 FROM mach2.Example.COM` 的项，则系统允许远程 root 登录。要禁止远程超级用户访问，请将 /etc/default/login 文件中的 #CONSOLE=/dev/console 项更改为 CONSOLE=/dev/console。

SPARC: 控制对系统硬件的访问（任务列表）

以下任务列表说明如何使 PROM 免受不需要的访问：

任务	说明	参考
禁止用户更改系统硬件设置	修改 PROM 设置需要提供口令。	第 68 页中的 “如何要求硬件访问口令”
禁用中止序列	禁止用户访问 PROM。	第 69 页中的 “如何禁用系统的中止序列”

控制对系统硬件的访问

可以通过要求在访问硬件设置时提供口令来保护物理计算机，也可以通过禁止用户使用中止序列离开窗口系统来保护计算机。

▼ 如何要求硬件访问口令

在 x86 系统上，保护 PROM 即是保护 BIOS。有关如何保护 BIOS，请参阅计算机手册。

1. **成为超级用户或承担拥有设备安全配置文件、维护和修复配置文件或系统管理员配置文件的角色。**

   系统管理员配置文件包括维护和修复配置文件。要创建拥有系统管理员配置文件的角色并将该角色指定给用户，请参见第 186 页中的 “配置 RBAC（任务列表）”。

2. **在终端窗口中，键入 PROM 安全模式。**

   ```
   # eeprom security-mode=command
   ```

```
Changing PROM password:

New password:      <键入口令>

Retype new password:      <重新键入口令>
```

选择值 command 或 full。有关更多详细信息，请参见 eeprom(1M) 手册页。

如果键入上述命令时未提示输入 PROM 口令，则表明系统已拥有 PROM 口令。

3 （可选的）要更改 PROM 口令，请键入以下命令：

```
# eeprom security-password=      按回车键

Changing PROM password:

New password:      <键入口令>

Retype new password:      <重新键入口令>
```

新的 PROM 安全模式和口令将立即生效。但是，很可能在下次引导时才会对这些更改进行通知。

注意 – 切勿忘记 PROM 口令。如果没有此口令，硬件将不可用。

▼ 如何禁用系统的中止序列

某些服务器系统具有键开关。如果将键开关设置在安全位置，则该开关将覆盖软件键盘中止设置。因此，使用以下过程进行的任何更改都可能无法实现。

1 承担主管理员角色，或成为超级用户。

主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2 将 KEYBOARD_ABORT 的值更改为 disable。

注释掉 /etc/default/kbd 文件中的 enable 行。然后，添加 disable 行：

```
# cat /etc/default/kbd

...

# KEYBOARD_ABORT affects the default behavior of the keyboard abort
```

```
# sequence, see kbd(1) for details.  The default value is "enable".

# The optional value is "disable".  Any other value is ignored.

...

#KEYBOARD_ABORT=enable

KEYBOARD_ABORT=disable
```

3 更新键盘缺省值。

```
# kbd -i
```

◆ ◆ ◆ 第 4 章

控制对设备的访问（任务）

本章介绍了保护设备的逐步说明，并包含一个参考部分。以下是本章中的信息列表：

- 第 71 页中的 “配置设备（任务列表）”
- 第 71 页中的 “配置设备策略（任务列表）”
- 第 76 页中的 “管理设备分配（任务列表）”
- 第 81 页中的 “分配设备（任务列表）”
- 第 87 页中的 “设备保护（参考）”

有关设备保护的概述信息，请参见第 41 页中的 “控制对设备的访问”。

配置设备（任务列表）

以下任务列表介绍了管理设备访问的任务。

任务	参考
管理设备策略	第 71 页中的 “配置设备策略（任务列表）”
管理设备分配	第 76 页中的 “管理设备分配（任务列表）”
使用设备分配	第 81 页中的 “分配设备（任务列表）”

配置设备策略（任务列表）

以下任务列表介绍了与设备策略相关的设备配置过程。

任务	说明	参考
查看系统上设备的设备策略	列出设备及其设备策略。	第 72 页中的 “如何查看设备策略”

任务	说明	参考
要求使用设备的权限	使用权限保护设备。	第 73 页中的 “如何更改现有设备上的设备策略”
删除设备的权限要求	删除或减少访问设备所需的权限。	示例 4–3
审计设备策略中的更改	在审计跟踪中记录设备策略的更改。	第 74 页中的 “如何审计设备策略中的更改”
访问 /dev/arp	获取 Solaris IP MIB-II 信息。	第 75 页中的 “如何从 /dev/* 设备检索 IP MIB-II 信息”

配置设备策略

设备策略会限制或防止对作为系统整体部分的设备进行访问。策略在内核中实施。

▼ 如何查看设备策略

◗ **显示系统上所有设备的设备策略。**

```
% getdevpolicy | more

DEFAULT

        read_priv_set=none

        write_priv_set=none

ip:*

        read_priv_set=net_rawaccess

        write_priv_set=net_rawaccess

...
```

示例 4–1 查看特定设备的设备策略

此示例显示了三个设备的设备策略。

```
% getdevpolicy /dev/allkmem /dev/ipsecesp /dev/hme

/dev/allkmem

        read_priv_set=all
```

```
        write_priv_set=all

/dev/ipsecesp

        read_priv_set=sys_net_config

        write_priv_set=sys_net_config

/dev/hme

        read_priv_set=net_rawaccess

        write_priv_set=net_rawaccess
```

▼ 如何更改现有设备上的设备策略

1 **承担拥有设备安全权限配置文件的角色或成为超级用户。**

主管理员角色拥有设备安全权限配置文件。还可以将设备安全权限配置文件指定给所创建的角色。有关如何创建角色并将其指定给用户的信息，请参见示例 9–3。

2 **向设备中添加策略。**

`# update_drv -a -p` *policy device-driver*

`-a` 指定 *device-driver* 的 *policy*。

`-p` *policy* *device-driver* 的设备策略。设备策略指定两组权限。一组用于读取设备，另一组用于写入设备。

device-driver 设备驱动程序。

有关更多信息，请参见 `update_drv`(1M) 手册页。

示例 4–2 向现有设备中添加策略

以下示例向 `ipnat` 设备中添加设备策略。

```
# getdevpolicy /dev/ipnat

/dev/ipnat

        read_priv_set=none

        write_priv_set=none

# update_drv -a \
```

```
-p 'read_priv_set=net_rawaccess write_priv_set=net_rawaccess' ipnat
# getdevpolicy /dev/ipnat
/dev/ipnat
        read_priv_set=net_rawaccess
        write_priv_set=net_rawaccess
```

示例 4–3 删除设备的策略

以下示例从 ipnat 设备的设备策略中删除读取的一组权限。

```
# getdevpolicy /dev/ipnat
/dev/ipnat
        read_priv_set=net_rawaccess
        write_priv_set=net_rawaccess
# update_drv -a -p write_priv_set=net_rawaccess ipnat
# getdevpolicy /dev/ipnat
/dev/ipnat
        read_priv_set=none
        write_priv_set=net_rawaccess
```

▼ 如何审计设备策略中的更改

缺省情况下，as 审计类包括 AUE_MODDEVPLCY 审计事件。

1. **承担主管理员角色，或成为超级用户。**
   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2 **预选包括 AUE_MODDEVPLCY 审计事件的审计类。**

将 as 类添加到 audit_control 文件的 flags 行中。此文件的显示将与以下内容类似：

```
# audit_control file

dir:/var/audit

flags:lo,as

minfree:20

naflags:lo
```

有关详细说明，请参见第 536 页中的 “如何修改 audit_control 文件” 。

▼ 如何从 /dev/* 设备检索 IP MIB-II 信息

检索 Solaris IP MIB-II 信息的应用程序应该打开 /dev/arp，而不是 /dev/ip。

1 **确定 /dev/ip 和 /dev/arp 上的设备策略。**

```
% getdevpolicy /dev/ip /dev/arp

/dev/ip

        read_priv_set=net_rawaccess

        write_priv_set=net_rawaccess

/dev/arp

        read_priv_set=none

        write_priv_set=none
```

请注意，读写 /dev/ip 需要具有 net_rawaccess 权限，而 /dev/arp 不需要权限。

2 **打开 /dev/arp 并推送 tcp 和 udp 模块。**

此方法不需要权限，与打开 /dev/ip 并推送 arp、tcp 和 udp 模块等效。由于现在打开 /dev/ip 需要权限，因此首选使用 /dev/arp 方法。

管理设备分配（任务列表）

以下任务列表介绍了启用和配置设备分配的过程。缺省情况下不会启用设备分配。启用设备分配之后，请参见第 81 页中的“分配设备（任务列表）”。

任务	说明	参考
使设备可分配	可以一次将一个设备分配给一个用户。	第 76 页中的“如何使设备可分配”
授权用户分配设备	将设备分配授权指定给用户。	第 77 页中的“如何授权用户来分配设备”
查看系统上的可分配设备	列出可分配的设备及其状态。	第 78 页中的“如何查看有关设备的分配信息”
强制分配设备	将设备分配给有即时需要的用户	第 78 页中的“强制分配设备”
强制解除设备分配	解除当前分配给某用户设备的分配	第 79 页中的“强制解除设备分配”
更改设备的分配属性	更改分配设备的要求	第 79 页中的“如何更改可以分配的设备”
创建设备清理脚本	清除物理设备中的数据。	第 92 页中的“编写新的设备清理脚本”
禁用设备分配	删除所有设备中的分配限制。	第 555 页中的“如何禁用审计”
审计设备分配	在审计跟踪中记录设备分配	第 81 页中的“如何审计设备分配”

管理设备分配

设备分配会限制或防止对外围设备进行访问。限制在用户分配时实施。缺省情况下，用户必须具有授权才能访问可分配设备。

▼ 如何使设备可分配

如果已经运行 `bsmconv` 命令启用了审计，则已经在系统上启用了设备分配。有关更多信息，请参见 `bsmconv`(1M) 手册页。

1. **承担拥有审计控制权限配置文件的角色或成为超级用户。**

   主管理员角色拥有审计控制权限配置文件。还可以将审计控制权限配置文件指定给所创建的角色。有关如何创建角色并将其指定给用户的信息，请参见示例 9–3。

2. **启用设备分配。**

   ```
   # bsmconv

   This script is used to enable the Basic Security Module (BSM).
   ```

```
Shall we continue with the conversion now? [y/n] y

bsmconv: INFO: checking startup file.

bsmconv: INFO: move aside /etc/rc3.d/S81volmgt.

bsmconv: INFO: turning on audit module.

bsmconv: INFO: initializing device allocation files.


The Basic Security Module is ready.

If there were any errors, please fix them now.

Configure BSM by editing files located in /etc/security.

Reboot this system now to come up with BSM enabled.
```

注 – 此命令禁用 Volume Management 守护进程 (`/etc/rc3.d/S81volmgt`)。

▼ 如何授权用户来分配设备

1. **承担主管理员角色，或成为超级用户。**

   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2. **创建包含适当授权和命令的权限配置文件。**

   通常，可以创建包括 `solaris.device.allocate` 授权的权限配置文件。请按照第 205 页中的“如何创建或更改权限配置文件”中的说明执行。授予权限配置文件适当属性，例如：

   - 权限配置文件名称： `Device Allocation`
   - 授予的授权： `solaris.device.allocate`
   - 带有安全属性的命令：带有 `sys_mount` 权限的 `mount` 和带有 `sys_mount` 权限的 `umount`

3. **创建权限配置文件的角色。**

   请按照第 188 页中的 “如何使用 GUI 创建和指定角色”中的说明执行。使用以下角色属性作为指南：

   - 角色名： `devicealloc`
   - 角色全名： `Device Allocator`

- 角色说明： `Allocates and mounts allocated devices`
- 权限配置文件： `Device Allocation`

  此权限配置文件必须在包括于角色中的配置文件列表的顶部。

4. **将此角色指定给允许分配设备的每个用户。**

5. **为用户讲授如何使用设备分配。**

   有关分配可移除介质的示例，请参见第 82 页中的 “如何分配设备”。

   由于 Volume Management 守护进程 (`vold`) 未运行，因此不会自动挂载可移除介质。有关挂载已分配设备的示例，请参见第 83 页中的 “如何挂载已分配的设备”。

▼ 如何查看有关设备的分配信息

开始之前 必须启用设备分配，此过程才会成功。有关如何启用设备分配的信息，请参见第 76 页中的 “如何使设备可分配”。

1. **承担拥有设备安全权限配置文件的角色或成为超级用户。**

   主管理员角色拥有设备安全权限配置文件。还可以将设备安全权限配置文件指定给所创建的角色。有关如何创建角色并将其指定给用户的信息，请参见示例 9–3。

2. **显示有关系统上可分配设备的信息。**

   ```
   # list_devices device-name
   ```

   其中，*device-name* 是以下各项之一：

   - `audio`[*n*]－麦克风和扬声器。
   - `fd`[*n*]－软盘驱动器。
   - `sr`[*n*]－CD-ROM 驱动器。
   - `st`[*n*]－磁带机。

故障排除 如果 `list_devices` 命令返回一条类似于以下内容的错误消息，则表明未启用设备分配，或者您不具有足够权限来检索此信息。

```
list_devices: No device maps file entry for specified device.
```

为使此命令成功执行，请启用设备分配并承担具有 `solaris.device.revoke` 授权的角色。

▼ 强制分配设备

强制分配应在某个用户忘记解除设备分配时使用，也可以在用户对设备有即时需要时使用。

开始之前 此用户或角色必须具有 `solaris.device.revoke` 授权。

1. **确定角色中是否具有适当授权。**

```
$ auths

solaris.device.allocate solaris.device.revoke
```

2. **将设备强制分配给需要此设备的用户。**

   此示例将磁带机强制分配给用户 jdoe。

```
$ allocate -U jdoe
```

▼ 强制解除设备分配

在进程终止或用户注销时，不会自动解除对用户的设备分配。用户忘记解除设备分配时，应使用强制解除分配。

开始之前 此用户或角色必须具有 solaris.device.revoke 授权。

1. **确定角色中是否具有适当授权。**

```
$ auths

solaris.device.allocate solaris.device.revoke
```

2. **强制解除设备分配。**

   此示例强制解除打印机分配。现在，其他用户可以分配此打印机。

```
$ deallocate -f /dev/lp/printer-1
```

▼ 如何更改可以分配的设备

1. **承担拥有设备安全权限配置文件的角色或成为超级用户。**

   主管理员角色拥有设备安全权限配置文件。还可以将设备安全权限配置文件指定给所创建的角色。有关如何创建角色并将其指定给用户的信息，请参见示例 9–3。

2. **指定是否需要授权，或者指定 solaris.device.allocate 授权。**

   更改 device_allocate 文件中设备项的第五个字段。

```
audio;audio;reserved;reserved;solaris.device.allocate;/etc/security/lib/audio_clean

fd0;fd;reserved;reserved;solaris.device.allocate;/etc/security/lib/fd_clean

sr0;sr;reserved;reserved;solaris.device.allocate;/etc/security/lib/sr_clean
```

   其中，solaris.device.allocate 指示用户必须具有 solaris.device.allocate 授权才能使用此设备。

示例 4–4 允许任何用户分配设备

在以下示例中，系统上的任何用户都可以分配所有设备。device_allocate 文件中每个设备项的第五个字段都更改为一个 at 符号 (@)。

```
$ whoami

devicesec

$ vi /etc/security/device_allocate

audio;audio;reserved;reserved;@;/etc/security/lib/audio_clean

fd0;fd;reserved;reserved;@;/etc/security/lib/fd_clean

sr0;sr;reserved;reserved;@;/etc/security/lib/sr_clean

...
```

示例 4–5 防止使用某些外围设备

在以下示例中，不能使用音频设备。device_allocate 文件中音频设备项的第五个字段更改为一个星号 (*)。

```
$ whoami

devicesec

$ vi /etc/security/device_allocate

audio;audio;reserved;reserved;*;/etc/security/lib/audio_clean

fd0;fd;reserved;reserved;solaris device.allocate;/etc/security/lib/fd_clean

sr0;sr;reserved;reserved;solaris device.allocate;/etc/security/lib/sr_clean

...
```

示例 4–6 防止使用所有外围设备

在以下示例中，不能使用外围设备。device_allocate 文件中每个设备项的第五个字段都更改为一个星号 (*)。

```
$ whoami

devicesec
```

```
$ vi /etc/security/device_allocate

audio;audio;reserved;reserved;*;/etc/security/lib/audio_clean

fd0;fd;reserved;reserved;*;/etc/security/lib/fd_clean

sr0;sr;reserved;reserved;*;/etc/security/lib/sr_clean

...
```

▼ 如何审计设备分配

缺省情况下，设备分配命令位于 `other` 审计类中。

1. **承担主管理员角色，或成为超级用户。**

   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2. **预选 `ot` 类进行审计。**

   将 `ot` 类添加到 `audit_control` 文件的 `flags` 行中。此文件的显示与以下信息类似：

   ```
   # audit_control file

   dir:/var/audit

   flags:lo,ot

   minfree:20

   naflags:lo
   ```

   有关详细说明，请参见第 536 页中的 “如何修改 `audit_control` 文件” 。

分配设备（任务列表）

以下任务列表介绍了向用户说明如何分配设备的过程。

任务	说明	参考
分配设备	使某用户可以用某设备，同时防止任何其他用户使用此设备。	第 82 页中的 “如何分配设备”

任务	说明	参考
挂载已分配的设备	使用户可以查看需要挂载的设备，例如 CD-ROM 或软盘。	第 83 页中的 “如何挂载已分配的设备 ”
解除设备分配	使其他用户可以使用可分配设备。	第 85 页中的 “如何解除设备分配”

分配设备

设备分配使设备每次只能被一个用户使用。必须挂载需要挂载点的设备。

▼ 如何分配设备

开始之前 必须启用设备分配，如第 76 页中的 “如何使设备可分配” 中所述。如果需要授权，则用户必须具有此授权。

1. **分配设备。**

   根据设备名称指定设备。

   ```
   % allocate device-name
   ```

2. **检验是否已分配此设备。**

   运行相同命令。

   ```
   % allocate device-name

   allocate. Device already allocated.
   ```

示例 4–7 分配麦克风

在此示例中，用户 `jdoe` 分配麦克风 `audio`。

```
% whoami

jdoe

% allocate audio
```

示例 4–8 分配打印机

在此示例中，某用户分配打印机。该用户解除打印机分配或打印机被强制分配给其他用户之前，任何其他用户都不能打印到 `printer-1`。

```
% allocate /dev/lp/printer-1
```

有关强制解除分配的示例，请参见第 79 页中的 “强制解除设备分配” 。

示例 4–9 分配磁带机

在此示例中，用户 jdoe 分配磁带机 st0。

```
% whoami

jdoe

% allocate st0
```

故障排除 如果 allocate 命令不能分配设备，则会在控制台窗口显示一条错误消息。有关分配错误消息的列表，请参见 allocate(1) 手册页。

▼ 如何挂载已分配的设备

开始之前 用户或角色已分配了此设备。要挂载设备，用户或角色必须具有挂载此设备所需的权限。有关如何提供所需权限的信息，请参见第 77 页中的 “如何授权用户来分配设备”。

1. **承担可以分配和挂载设备的角色。**

   ```
   % su role-name

   Password:        <键入 role-name 的口令>

   $
   ```

2. **在角色起始目录中创建并保护挂载点。**

   只需在首次需要挂载点的时候执行此步骤。

   ```
   $ mkdir mount-point ; chmod 700 mount-point
   ```

3. **列出可分配的设备。**

   ```
   $ list_devices -l

   List of allocatable devices
   ```

4. **分配设备。**

   根据设备名称指定设备。

   ```
   $ allocate device-name
   ```

5. **挂载设备。**

   ```
   $ mount -o ro -F filesystem-type device-path mount-point
   ```

   其中，

   `-o ro` 指示将此设备挂载为只读。使用 `-o rw` 指示应该可以写入此设备。

`-F` *filesystem-type*	指示设备的文件系统格式。通常，使用 HSFS 文件系统格式化 CD-ROM。而软盘通常使用 PCFS 文件系统进行格式化。
device-path	指示设备的路径。`list_devices -l` 命令的输出包括 *device-path*。
mount-point	指示在步骤 2 中创建的挂载点。

示例 4–10 分配软盘驱动器

在此示例中，用户承担可以分配和挂载软盘驱动器 `fd0` 的角色。软盘已使用 PCFS 文件系统进行了格式化。

```
% roles

devicealloc

% su devicealloc

Password:      <键入 devicealloc 的口令>

$ mkdir /home/devicealloc/mymnt

$ chmod 700 /home/devicealloc/mymnt

$ list_devices -l

...

device: fd0 type: fd files: /dev/diskette /dev/rdiskette /dev/fd0a

...

$ allocate fd0

$ mount -o ro -F pcfs /dev/diskette /home/devicealloc/mymnt

$ ls /home/devicealloc/mymnt
```

List of the contents of diskette

示例 4–11 分配 CD-ROM 驱动器

在此示例中，用户承担可以分配和挂载 CD-ROM 驱动器 `sr0` 的角色。此驱动器按照 HSFS 文件系统进行格式化。

```
% roles
```

```
devicealloc

% su devicealloc

Password:      <键入 devicealloc 的口令>

$ mkdir /home/devicealloc/mymnt

$ chmod 700 /home/devicealloc/mymnt

$ list_devices -l

...

device: sr0 type: sr files: /dev/sr0 /dev/rsr0 /dev/dsk/c0t2d0s0 ...

...

$ allocate sr0

$ mount -o ro -F hsfs /dev/sr0 /home/devicealloc/mymnt

$ cd /home/devicealloc/mymnt ; ls

List of the contents of CD-ROM
```

故障排除 如果 `mount` 命令不能挂载设备，则会显示一条错误消息： `mount:insufficient privileges`。检查以下各项：

- 确保正在配置文件 shell 中执行 `mount` 命令。如果已经承担了某角色，则此角色具有一个配置文件 shell。如果您是被指定了带有 `mount` 命令的配置文件的用户，则必须创建一个配置文件 shell。命令 `pfsh`、`pfksh` 和 `pfcsh` 可创建配置文件 shell。
- 确保拥有指定的挂载点。您应该具有挂载点的读取、写入和执行权限。

如果仍然不能挂载已分配的设备，请与管理员联系。

▼ 如何解除设备分配

解除分配使其他用户可以分配和使用此设备。

开始之前 必须已经分配此设备。

1. **如果此设备已挂载，请将其卸载。**

   ```
   $ cd $HOME
   ```

   ```
   $ umount mount-point
   ```

2. **解除设备分配。**

   ```
   $ deallocate device-name
   ```

示例 4–12 解除麦克风分配

在此示例中，用户 jdoe解除麦克风 audio 的分配。

```
% whoami
```

```
jdoe
```

```
% deallocate audio
```

示例 4–13 解除 CD-ROM 驱动器的分配

在此示例中，设备分配器角色解除 CD-ROM 驱动器的分配。消息列显后，会弹出 CD-ROM。

```
$ whoami
```

```
devicealloc
```

```
$ cd /home/devicealloc
```

```
$ umount /home/devicealloc/mymnt
```

```
$ ls /home/devicealloc/mymnt
```

```
$
```

```
$ deallocate sr0
```

```
/dev/sr0:      326o
```

```
/dev/rsr0:     326o
```

```
...
```

```
sr_clean: Media in sr0 is ready.  Please, label and store safely.
```

设备保护（参考）

设备策略保护 Solaris OS 中的设备。设备分配可以保护外围设备。设备策略由内核实施。可以选择启用设备分配，并在用户级别实施。

设备策略命令

设备管理命令管理本地文件的设备策略。设备策略可以包括权限要求。只有超级用户或具有等效功能的角色才能管理设备。

下表列出了设备管理命令。

表 4–1 设备管理命令

命令	目的	手册页
`devfsadm`	在运行的系统上管理设备和设备驱动程序。还装入设备策略。 使用 `devfsadm` 命令，可以清除指向磁盘、磁带、端口、音频设备和伪设备的不稳定 `/dev` 链接。还可以重新配置已命名驱动程序的设备。	`devfsadm`(1M)
`getdevpolicy`	显示与一个或多个设备关联的策略。任何用户都可以运行此命令。	`getdevpolicy`(1M)
`add_drv`	将新设备驱动程序添加到正在运行的系统。包含将设备策略添加到此新设备的选项。通常，正在安装设备驱动程序时会在脚本中调用此命令。	`add_drv`(1M)
`update_drv`	更新现有设备驱动程序的属性。包含为此设备更新设备策略的选项。通常，正在安装设备驱动程序时会在脚本中调用此命令。	`update_drv`(1M)
`rem_drv`	删除设备或设备驱动程序。	`rem_drv`(1M)

设备分配

设备分配可以保护站点免受数据丢失、计算机病毒以及其他安全问题的破坏。与设备策略不同，设备分配是可选的。只有在 `bsmconv` 脚本运行之后才可分配设备。。设备分配使用授权限制对可分配设备的访问。

设备分配的组成

设备分配机制的组成如下所示：

- `allocate`、`deallocate`、 `dminfo` 和 `list_devices` 命令。有关更多信息，请参见第 88 页中的 “设备分配命令”。

- 各个可分配设备的设备清理脚本。

这些命令和脚本使用以下本地文件实现设备分配：

- /etc/security/device_allocate 文件。有关更多信息，请参见 device_allocate(4) 手册页。
- /etc/security/device_maps 文件。有关更多信息，请参见 device_maps(4) 手册页。
- /etc/security/dev 目录中每个可分配设备的锁定文件。
- 与每个可分配设备关联的锁定文件的已更改属性。

注 – Solaris OS 的将来发行版可能不支持 /etc/security/dev 目录。

设备分配命令

与大写选项一起使用时，allocate、deallocate 和 list_devices 命令是管理命令。否则，这些命令是用户命令。下表列出了设备分配命令。

表 4–2 设备分配命令

命令	目的	手册页
bsmconv	创建数据库以处理设备分配。还启用审计服务。您必须是超级用户或承担主管理员角色。	bsmconv(1M)
dminfo	根据设备类型、设备名称以及全路径名搜索可分配设备。	dminfo(1M)
list_devices	列出可分配设备的状态。	list_devices(1)
	列出与 device_maps 文件中列出的任何设备关联的所有设备特定文件。	
list_devices -U	列出可分配或分配给指定用户 ID 的设备。使用此选项可以检查可分配或分配给其他用户的设备。必须具有 solaris.device.revoke 授权。	
allocate	保留一个可分配设备以供一个用户使用。	allocate(1)
	缺省情况下，用户必须具有 solaris.device.allocate 授权才能分配设备。可以修改 device_allocate 文件，使其不需要用户授权。然后，系统上的任何用户都可以请求分配设备以供使用。	
deallocate	删除设备的分配保留。	deallocate(1)

分配命令授权

缺省情况下，用户必须具有 solaris.device.allocate 授权才能保留可分配设备。有关如何创建包括 solaris.device.allocate 授权的权限配置文件的信息，请参见第 77 页中的 “如何授权用户来分配设备”。

管理员必须具有 solaris.device.revoke 授权才能更改任何设备的分配状态。例如，allocate 和 list_devices 命令的 -U 选项，以及 deallocate 命令的 -F 选项需要 solaris.device.revoke 授权。

有关更多信息，请参见第 224 页中的 “要求授权的命令” 。

分配错误状态

当 deallocate 命令解除分配失败，或 allocate 命令分配失败时，设备将被置于**分配错误状态**。可分配设备处于分配错误状态时，必须强制解除对此设备的分配。只有超级用户或者具有设备管理权限配置文件或设备安全权限配置文件的角色才能处理分配错误状态。

deallocate 命令和 -F 选项一起使用可强制解除分配。或者，可以使用 allocate -U 将设备指定给用户。设备分配后，便可查明出现的任何错误消息。更正设备的所有问题后，可强制解除其分配。

device_maps 文件

设置设备分配时会创建设备映射。启用审计服务后，bsmconv 命令会创建缺省 /etc/security/device_maps 文件。可以为站点自定义此初始 device_maps 文件。device_maps 文件包括与每个可分配设备关联的设备名称、设备类型和设备特定文件。

device_maps 文件定义每个设备的设备特定文件映射，这在许多情况下并不是直观的。使用此文件，程序可以查看设备特定文件和设备之间的映射关系。例如，可以使用 dminfo 命令，在设置可分配设备时检索要指定的设备名称、设备类型和设备特定文件。dminfo 命令使用 device_maps 文件来报告此信息。

每个设备由仅占一行的项表示，该项格式如下：

device-name:*device-type*:*device-list*

示例 4–14 device_maps 项样例

以下是 device_maps 文件中软盘驱动器 fd0 的项的示例：

```
fd0:\
        fd:\
        /dev/diskette /dev/rdiskette /dev/fd0a /dev/rfd0a \
/dev/fd0b /dev/rfd0b /dev/fd0c /dev/fd0 /dev/rfd0c /dev/rfd0:\
```

device_maps 文件中的行可以用反斜杠 (\) 结束，以在下一行继续项。也可以包括注释。井号 (#) 注释前面未紧跟反斜杠的下一个新行之前的所有后续文本。所有字段中都允许前导空格和后缀空格。字段定义如下：

device-name 指定此设备的名称。有关当前设备名称的列表，请参见第 78 页中的 “如何查看有关设备的分配信息” 。

device-type 指定通用设备类型。通用名称是设备类的名称，例如 st、fd 或 audio。*device-type* 字段在逻辑上将相关设备分组。

device-list 列出与物理设备关联的设备特定文件。*device-list* 必须包含所有允许访问特定设备的特定文件。如果此列表不完整，则恶意用户仍可以获取或修改专用信息。*device-list* 字段的有效项反映位于 /dev 目录中的设备文件。

device_allocate 文件

启用审计服务后，bsmconv 命令会创建初始 /etc/security/device_allocate 文件。此初始 device_allocate 文件可以用作起始点。可以修改 device_allocate 文件，以将设备从可分配更改为不可分配，或者添加新设备。以下是一个 device_allocate 文件样例。

```
st0;st;;;;/etc/security/lib/st_clean

fd0;fd;;;;/etc/security/lib/fd_clean

sr0;sr;;;;/etc/security/lib/sr_clean

audio;audio;;;*;/etc/security/lib/audio_clean
```

device_allocate 文件中的项并不表示此设备是可分配的，除非此项专门声明此设备是可分配的。请注意 device_allocate 文件样例中音频设备项第五个字段中的星号 (*)。第五个字段中的星号向系统指示此设备不是可分配的。因此，不能使用此设备。如果此字段中存在其他值或没有值，则指示可以使用此设备。

在 device_allocate 文件中，每个设备由仅占一行的项表示，该项的格式如下：

device-name;*device-type*;reserved;reserved;*auths*;*device-exec*

device_allocate 文件中的行可以用反斜杠 (\) 结束，以在下一行继续项。也可以包括注释。井号 (#) 注释前面未紧跟反斜杠的下一个新行之前的所有后续文本。所有字段都允许前导空格和后缀空格。字段定义如下：

device-name 指定此设备的名称。有关当前设备名称的列表，请参见第 78 页中的 “如何查看有关设备的分配信息”。

device-type 指定通用设备类型。通用名称是设备类的名称，例如 st、fd 和 sr。*device-type* 字段在逻辑上将相关设备分组。使设备可分配后，可 device_maps 文件的 *device-type* 字段中检索设备名称。

reserved Sun 保留两个标记为 reserved 的字段以供将来使用。

auths 指定此设备是否可分配。此字段中的星号 (*) 指示此设备是不可分配的。授权字符串或空字段指示此设备是可分配的。例如，*auths* 字段中的字符串 solaris.device.allocate 指示需要 solaris.device.allocate 授权才能分配此设备。此文件中的 at 符号 (@) 指示任何用户都可以分配此设备。

device-exec 提供为特殊处理（例如分配进程期间的清除和对象重用保护）而调用的脚本的路径名称。每当 `deallocate` 命令对此设备执行操作时，都会运行 *device-exec* 脚本。

例如，`sr0` 设备的以下项指示具有 `solaris.device.allocate` 授权的用户可以分配 CD-ROM 驱动器：

```
sr0;sr;reserved;reserved;solaris.device.allocate;/etc/security/lib/sr_clean
```

可以决定接受缺省设备及其已定义特征。安装新设备后，可以修改这些项。任何需要在使用之前分配的设备都必须在该设备系统的 `device_allocate` 和 `device_maps` 文件中定义。当前，将盒式磁带机、软盘驱动器、CD-ROM 驱动器和音频芯片视为可分配的。这些设备类型具有设备清理脚本。

注 – Xylogics™ 磁带机或归档磁带机还使用为 SCSI 设备提供的 `st_clean` 脚本。需要为其他设备（如调制解调器、终端、图形输入板和其他可分配设备）创建自己的设备清理脚本。脚本必须满足此类型设备的对象重用要求。

设备清理脚本

设备分配满足称为对象重用的部分要求。**设备清理**脚本说明安全要求，即在重用之前从物理设备中清除所有可用数据。清除数据后，其他用户才可分配此设备。缺省情况下，盒式磁带机、软盘驱动器、CD-ROM 驱动器和音频设备需要设备清理脚本。Solaris OS 提供了这些脚本。本节介绍设备清理脚本的功能。

磁带的设备清理脚本

`st_clean` 设备清理脚本支持三种磁带设备：

- SCSI ¼ 英寸磁带
- 归档 ¼ 英寸磁带
- 开放式卷盘 ½ 英寸磁带

`st_clean` 脚本使用 `mt` 命令的 `rewoffl` 选项来清除此设备。有关更多信息，请参见 `mt`(1) 手册页。如果此脚本在系统引导期间运行，则会查询此设备以确定它是否联机。如果设备联机，则此脚本确定设备中是否有介质。内含介质的 ¼ 英寸磁带设备会被置于分配错误状态。分配错误状态强制管理员手动清除此设备。

系统正常操作期间，在交互模式下执行 `deallocate` 命令时，将提示用户删除此介质。从设备中删除此介质之后才会进行解除分配。

软盘驱动器和 CD-ROM 驱动器的设备清理脚本

为软盘驱动器和 CD-ROM 驱动器提供以下设备清理脚本：

- **`fd_clean` 脚本** – 软盘的设备清理脚本。

- **sr_clean 脚本**－CD-ROM 驱动器的设备清理脚本。

这些脚本使用 eject 命令从驱动器中删除介质。如果 eject 命令失败，则会将此设备置于分配错误状态。有关更多信息，请参见 eject(1) 手册页。

音频的设备清理脚本

使用 audio_clean 脚本清除音频设备。此脚本执行 AUDIO_GETINFO ioctl 系统调用来读取设备。然后，此脚本执行 AUDIO_SETINFO ioctl 系统调用将此设备的配置重置为缺省值。

编写新的设备清理脚本

如果将更多可分配设备添加到系统，则可能需要创建自己的设备清理脚本。deallocate 命令将参数传送到设备清理脚本中。在此显示的参数是包含设备名称的字符串。有关更多信息，请参见 device_allocate(4) 手册页。

clean-script `-[I|i|f|S]` *device-name*

设备清理脚本在成功时一定会返回 "0"，而在失败时返回大于 "0" 的值。-I、-f 和 -S 选项确定脚本的运行模式：

`-I` 只有在系统引导期间才需要此选项。所有输出必须转到系统控制台。强制弹出介质失败或无法强制弹出介质必定会将此设备置于分配错误状态。

`-i` 与 -I 选项类似，但不显示输出。

`-f` 用于强制清除。此选项是交互的，并且假定用户可以响应提示。如果一部分清除操作失败，则带有此选项的脚本必须尝试完成清除。

`-S` 用于标准清除。此选项是交互的，并且假定用户可以响应提示。

◆ ◆ ◆ 第 5 章

使用基本审计报告工具（任务）

本章介绍如何在系统上创建文件清单以及如何使用此清单来检查系统的完整性。使用基本审计报告工具 (Basic Audit Reporting Tool, BART)，您可以通过在一段时间内对系统执行文件层检查来全面地验证系统。

以下是本章中信息的列表：

- 第 95 页中的 “使用 BART（任务列表）”
- 第 93 页中的 “基本审计报告工具（概述）”
- 第 96 页中的 “使用 BART（任务）”
- 第 113 页中的 “BART 清单、Rules 文件和报告（参考）”

基本审计报告工具（概述）

BART 是一种完全在文件系统层运行的文件跟踪工具。使用 BART，可以迅速、轻松、可靠地收集有关安装在已部署的系统上的软件栈组件的信息。使用 BART，可以通过简化耗时的管理任务来显著降低管理系统网络的成本。

使用 BART，可以根据已知的基准确定系统上所进行的文件层更改。可以使用 BART 根据完全安装并配置的系统创建基准或**控制**清单。然后可将此基准与系统快照进行比较，将生成一个列出从系统安装以来所进行的文件层更改的报告。

`bart` 命令是标准 UNIX 命令。您可以将 `bart` 命令的输出重定向到文件以便进行后续处理。

BART 功能

BART 在设计上侧重于既有效又灵活的简单语法。使用此工具，可以生成给定系统在一段时间内的清单。然后，需要验证此系统的文件时，可以通过比较新旧清单来生成报告。使用 BART 的另一种方法是生成若干个相似系统的清单，然后进行系统间的比较。BART 与现有审计工具的主要区别在于 BART 在跟踪信息和报告信息方面都非常灵活。

BART 的其他优点和用法包括：

- 提供了一种为运行 Solaris 软件的系统在文件层编制目录的有效而简便的方法。
- 使用 BART，可以定义要监视的文件，还可以在必要时修改配置文件。借助这种灵活性，可以监视本地的自定义项，并可轻松、有效地重新配置软件。
- 确保系统运行可靠的软件。
- 允许监视一段时间内系统在文件层的变化，从而帮助找到损坏或异常的文件。
- 帮助对系统性能问题进行疑难解答。

BART 组件

BART 有两个主要组件和一个可选组件：

- BART 清单
- BART 报告
- BART Rules 文件

BART 清单

您可以使用 `bart create` 命令在特定时间拍摄系统的文件层快照。输出是名为**清单**的关于文件和文件属性的目录。此清单列出了有关系统上所有文件或特定文件的信息。它包含了有关文件属性的信息，其中可以包括一些唯一标识的信息，如 MD5 校验和。有关 MD5 校验和的更多信息，请参见 `md5`(3EXT) 手册页。清单可以进行存储，并可以在客户机和服务器系统间传送。

注 – BART **不会**跨越文件系统边界，但同一类型的文件系统除外。此约束使 `bart create` 命令的输出更容易预测。例如，在不带参数的情况下，`bart create` 命令编制根 (`/`) 目录下所有 UFS 文件系统的目录。但是，不会对 NFS 或 TMPFS 文件系统或已挂载的 CD-ROM 编制目录。创建清单时，请勿尝试审计网络中的文件系统。请注意，使用 BART 监视联网的文件系统会占用大量的资源而生成价值很小的清单。

有关 BART 清单的更多信息，请参见第 113 页中的 “BART 清单文件格式”。

BART 报告

此报告工具有三项输入：两份要比较的清单和一个可选 rules 文件，此 rules 文件由用户提供，用于指明要标记的差异。

可以使用 `bart compare` 命令比较两份清单，一份是**控制清单**，另一份是**测试清单**。准备这些清单使用的文件系统、选项和 rules 文件必须与使用 `bart create` 命令时相同。

`bart compare` 命令的输出是一份报告，其中按文件列出了两份清单间的差异。**差异**是指上述两份清单中所列出的某个特定文件的任何属性的变化。两份清单间文件项的添加或删除也被视为差异。

报告差异时使用两个控制级别：

- 在生成清单时
- 在生成报告时

这些控制级别是特意设置的，因为生成清单所需的开销比报告两份清单间差异所需的开销大。创建清单之后，即可通过使用不同的 rules 文件运行 `bart compare` 命令来从各个方面比较清单。

有关 BART 报告的更多信息，请参见第 116 页中的 “BART 报告” 。

BART Rules 文件

rules **文件**是一个运行 `bart` 命令时可选的作为输入的文本文件。此文件使用包含和排除规则。rules 文件用于创建自定义清单和报告。使用 rules 文件，可以用简洁的语法表达要编制目录的文件集以及要监视的任何给定文件集的属性。在比较清单时，使用 rules 文件有助于标记清单间的差异。使用 rules 文件是一种收集有关系统上文件的特定信息的有效方法。

可以使用文本编辑器创建 rules 文件。通过 rules 文件，可以执行以下任务：

- 使用 `bart create` 命令创建列出有关系统上所有文件或特定文件的信息的清单。
- 使用 `bart compare` 命令生成监视文件系统的特定属性的报告。

注 – 您可以创建若干个用于不同用途的 rules 文件。但是，如果使用 rules 文件创建清单，则比较清单时必须使用同一个 rules 文件。如果比较清单时所用的 rules 文件与创建清单时所用的 rules 文件不同，则 `bart compare` 命令的输出会列出许多无效的差异。

rules 文件中也可能包含由于用户错误而导致的语法错误和其他不明确的信息。如果 rules 文件确实包含错误信息，则也会报告这些错误。

使用 rules 文件来监视系统上的特定文件和文件属性需要进行规划。创建 rules 文件之前，请首先确定系统上要监视的文件和文件属性。根据要完成的目标，可以使用 rules 文件来创建清单、比较清单或执行其他操作。

有关 BART rules 文件的更多信息，请参见第 114 页中的 “BART Rules 文件格式” 和 `bart_rules`(4) 手册页。

使用 BART（任务列表）

任务	说明	参考
创建清单。	获取一个列出有关系统上安装的所有文件的信息的清单。	第 96 页中的 “如何创建清单”

任务	说明	参考
创建自定义清单。	通过以下方法之一来获取一个列出有关系统上安装的特定文件的信息的清单： ■ 通过指定子树 ■ 通过指定文件名 ■ 通过使用 rules 文件	第 99 页中的 “如何自定义清单”
比较同一系统在一段时间内的清单。或者，将不同系统的清单与控制系统清单进行比较。	获取比较系统在一段时间内的更改的报告。或者，获取一个或几个系统与控制系统的相比较的报告。	第 104 页中的 “如何比较同一系统在一段时间内的清单” 第 107 页中的 “如何比较不同系统的清单与控制系统的清单”
（可选）自定义 BART 报告。	通过以下方法之一获取自定义 BART 报告： ■ 通过指定属性。 ■ 通过使用 rules 文件。	第 111 页中的 “如何通过指定文件属性自定义 BART 报告” 第 111 页中的 “如何通过使用 Rules 文件自定义 BART 报告”

使用 BART（任务）

您可以以普通用户、超级用户或承担主管理员角色的用户的身份来运行 `bart` 命令。 如果作为普通用户运行 `bart` 命令，则只能列出和监视有权访问的文件，如有关起始目录中文件的信息。在运行 `bart` 命令时成为超级用户的优势是您所创建的清单会包含有关要监视的隐藏文件和专用文件的信息。如果需要列出和监视有关具有限制性权限的文件（例如 `/etc/passwd` 或 `/etc/shadow` 文件）的信息，请以超级用户或承担等效角色的用户的身份来运行 `bart` 命令。有关使用基于角色的访问控制的更多信息，请参见第 186 页中的 “配置 RBAC（任务列表）”。

BART 安全注意事项

以超级用户的身份运行 `bart` 命令会使任何人都能读取输出。此输出可能包含专用文件名。如果您在运行 `bart` 命令时成为超级用户，请采取相应的措施保护输出。例如，使用可生成具有限制性权限的输出文件的选项。

注 – 本章中的过程和示例显示了由超级用户运行的 `bart` 命令。除非另有指定，否则以超级用户的身份运行 `bart` 命令为可选操作。

▼ 如何创建清单

可以在 Solaris 软件的初始安装之后立即创建系统清单。此类清单可提供用于比较同一系统在一段时间内的更改的基准。或者，可以将此清单与不同系统的清单进行比较。例如，如果为网络中的每个系统拍摄快照，然后将每个测试清单与控制清单进行比较，则可以迅速确定需要执行哪些操作来实现测试系统与基准配置的同步。

1. **承担主管理员角色，或成为超级用户。**

   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2. **安装 Solaris 软件之后，创建一个控制清单并将输出重定向到文件。**

   `# bart create` *options* > *control-manifest*

   `-R` 指定清单的根目录。所有由规则指定的路径都会被解释为此目录的相对路径。所有由清单报告的路径均为此目录的相对路径。

   `-I` 无论是从命令行执行此选项，还是从标准输入中读取此选项，它都会接受要列出的单个文件的列表。

   `-r` 此清单的 rules 文件的名称。请注意，– 在与 `-r` 选项一起使用时，会从标准输入读取 rules 文件。

   `-n` 禁用文件列表中所有常规文件的内容签名。此选项可用于改善性能。或者，可以在需要更改文件列表的内容时使用此选项，这与系统日志文件的情况类似。

3. **检查清单内容。**

4. **保存清单以便将来使用。**

   为清单选择一个有意义的名称。例如，使用系统名称及此清单的创建日期。

示例 5–1 创建可列出有关系统上所有文件的信息的清单

如果运行不带任何选项的 `bart create` 命令，则会列出有关系统上安装的所有文件的信息。当从核心映像安装多个系统时，可使用此类清单作为基准。或者，在需要确保安装完全一致的情况下使用此类清单运行比较。

例如：

```
# bart create

! Version 1.0

! Thursday, December 04, 2003 (16:17:39)

# Format:

#fname D size mode acl dirmtime uid gid

#fname P size mode acl mtime uid gid

#fname S size mode acl mtime uid gid
```

```
#fname F size mode acl mtime uid gid contents

#fname L size mode acl lnmtime uid gid dest

#fname B size mode acl mtime uid gid devnode

#fname C size mode acl mtime uid gid devnode

/ D 1024 40755 user::rwx,group::r-x,mask:r-x,other:r-x 3fd9ea47 0 0

/.java D 512 40755 user::rwx,group::r-x,mask:r-x,other:r-x 3f8dc04d 0 10

/.java/.userPrefs D 512 40700 user::rwx,group::---,mask:---

other:--- 3f8dc06b 010

/.java/.userPrefs/.user.lock.root F 0 100600 user::rw-

group::---,mask:---,other:--- 3f8dc06b 0 10 -

/.java/.userPrefs/.userRootModFile.root F 0 100600 user::rw-,

group::---,mask:---,other:--- 3f8dc0a1 0 10 -

/.smc.properties F 1389 100644 user::rw-,group::r--,mask:r--

other:r-- 3f8dca0c0 10

.

.

.

/var/sadm/pkg/SUNWdtmad/install/depend F 932 100644 user::rw-,

group::r--,mask:r--,other:r-- 3c23a19e 0 0 -

/var/sadm/pkg/SUNWdtmad/pkginfo F 594 100644 user::rw-

group::r--,mask:r--,other:r-- 3f81e416 0 0 -

/var/sadm/pkg/SUNWdtmad/save D 512 40755 user::rwx,group::r-x

mask:r-x,other:r-x 3f81e416 0 0

/var/sadm/pkg/SUNWdtmaz D 512 40755 user::rwx,group::r-x
```

```
mask:r-x,other:r-x 3f81e41b 0 0

/var/sadm/pkg/TSIpgxw/save D 512 40755 user::rwx

group::r-x,mask:r-x,other:r-x 3f81e892 0 0

.

.

.
```

每份清单都包括头和项。每个清单文件项均是单独的一行，具体取决于文件类型。例如，对于上面输出的每个清单项而言，类型 `F` 指定文件，而类型 `D` 指定目录。同时还列出了有关大小、内容、用户 ID、组 ID 和权限的信息。输出中的文件项按文件名的编码版本排序，从而能够正确地处理特殊字符。所有项均按文件名的升序排列。所有非标准文件名（例如那些包含嵌入的换行符或制表符的文件名）都在排序之前将非标准字符引起来。

以 `!` 开头的行提供有关清单的元数据。清单版本行指示清单规格版本。日期行以日期格式显示清单的创建日期。请参见 `date`(1) 手册页。清单比较工具会忽略某些行。被忽略的行包括空白行、仅包含空格的行，以及以 `#` 开头的注释。

▼ 如何自定义清单

可以通过以下方法之一自定义清单：

- 通过指定子树

  为系统上的单个子树（而不是大型目录的全部内容）创建清单是监视特定文件更改的有效方法。您可以创建系统上特定子树的基准清单，然后定期创建同一子树的测试清单。使用 `bart compare` 命令将控制清单与测试清单进行比较。使用此选项，可以有效地监视重要的文件系统以确定是否有任何文件受到入侵者的威胁。

- 通过指定文件名

  由于创建列出整个系统的清单耗时更多，占用空间的更多，所需的开销更大，因此可在仅需要列出有关系统上一个或多个特定文件的信息时选择使用此 `bart` 命令选项。

- 通过使用 rules 文件

  可以使用 rules 文件创建列出有关给定系统上特定文件和特定子树的信息的自定义清单，也可以使用 rules 文件来监视特定的文件属性。使用 rules 文件创建和比较清单使您能够灵活地为多个文件或子树指定多个属性。而使用命令行只能指定应用于所创建的每份清单或所生成的每个报告的所有文件的全局属性定义。

1 确定要列出和监视的文件。

2. **承担主管理员角色，或成为超级用户。**
   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

3. **安装 Solaris 软件之后，使用以下选项之一创建自定义清单：**
   - 通过指定子树：

     ```
     # bart create -R root-directory
     ```
   - 通过指定一个或多个文件名：

     ```
     # bart create -I filename...
     ```

     例如：

     ```
     # bart create -I /etc/system /etc/passwd /etc/shadow
     ```
   - 通过使用 rules 文件：

     ```
     # bart create -r rules-file
     ```

4. **检查清单内容。**

5. **保存清单以便将来使用。**

示例 5–2 通过指定子树创建清单

此示例说明了如何创建仅包含有关 `/etc/ssh` 子树中文件的信息的清单。

```
# bart create -R /etc/ssh
! Version 1.0
! Saturday, November 29, 2003 (14:05:36)
# Format:
#fname D size mode acl dirmtime uid gid
#fname P size mode acl mtime uid gid
#fname S size mode acl mtime uid gid
#fname F size mode acl mtime uid gid contents
```

```
#fname L size mode acl lnmtime uid gid dest

#fname B size mode acl mtime uid gid devnode

#fname C size mode acl mtime uid gid devnode

/ D 512 40755 user::rwx,group::r-x,mask:r-x,other:r-x 3f81eab9 0 3

/ssh_config F 861 100644 user::rw-,group::r--,mask:r--,

other:r-- 3f81e504 0 3 422453ca0e2348cd9981820935600395

/ssh_host_dsa_key F 668 100600 user::rw-,group::---,mask:---,

other:--- 3f81eab9 0 0 5cc28cdc97e833069fd41ef89e4d9834

/ssh_host_dsa_key.pub F 602 100644 user::rw-,group::r--,mask:r--,

other:r-- 3f81eab9 0 0 16118c736995a4e4754f5ab4f28cf917

/ssh_host_rsa_key F 883 100600 user::rw-,group::---,mask:---,

other:--- 3f81eaa2 0 0 6ff17aa968ecb20321c448c89a8840a9

/ssh_host_rsa_key.pub F 222 100644 user::rw-,group::r--,mask:r--,

other:r-- 3f81eaa2 0 0 9ea27617efc76058cb97aa2caa6dd65a

.

.

.
```

示例 5–3 通过指定文件名自定义清单

此示例说明了如何创建仅列出有关系统上 /etc/passwd 和 /etc/shadow 文件的信息的清单。

```
# bart create -I /etc/passwd /etc/shadow

! Version 1.0

! Monday, December 15, 2003 (16:28:55)

# Format:

#fname D size mode acl dirmtime uid gid
```

```
#fname P size mode acl mtime uid gid

#fname S size mode acl mtime uid gid

#fname F size mode acl mtime uid gid contents

#fname L size mode acl lnmtime uid gid dest

#fname B size mode acl mtime uid gid devnode

#fname C size mode acl mtime uid gid devnode

/etc/passwd F 542 100444 user::r--,group::r--,mask:r--,

other:r-- 3fcfd45b 0 3 d6

84554f85d1de06219d80543174ad1a

/etc/shadow F 294 100400 user::r--,group::---,mask:---,

other:--- 3f8dc5a0 0 3 fd

c3931c1ae5ee40341f3567b7cf15e2
```

下面以比较的方式列出了同一系统上 `/etc/passwd` 和 `/etc/shadow` 文件的 `ls - al` 命令的标准输出。

```
# ls -al /etc/passwd

-r--r--r--   1 root     sys          542 Dec  4 17:42 /etc/passwd

# ls -al /etc/shadow

-r--------   1 root     sys          294 Oct 15 16:09 /etc/shadow
```

示例 5–4 通过使用 Rules 文件自定义清单

此示例说明了如何通过使用 rules 文件仅列出 `/etc` 目录中的文件来创建清单。同一个 rules 文件还包括由 `bart compare` 命令用于监视 `/etc/system` 文件的 `acl` 属性更改的指令。

- 使用文本编辑器创建仅列出 `/etc` 目录中文件的 rules 文件。

  ```
  # List information about all the files in the /etc directory.
  ```

```
CHECK all

/etc


# Check only acl changes in the /etc/system file


IGNORE all

CHECK acl

/etc/system
```

有关创建 rules 文件的更多信息，请参见第 95 页中的 “BART Rules 文件”。

- 使用已创建的 rules 文件创建控制清单。

```
# bart create -r etc.rules-file > etc.system.control-manifest

! Version 1.0

! Thursday, December 11, 2003 (21:51:32)

# Format:

#fname D size mode acl dirmtime uid gid

#fname P size mode acl mtime uid gid

#fname S size mode acl mtime uid gid

#fname F size mode acl mtime uid gid contents

#fname L size mode acl lnmtime uid gid dest

#fname B size mode acl mtime uid gid devnode

#fname C size mode acl mtime uid gid devnode

/etc/system F 1883 100644 user::rw-,group::r--,mask:r--,

other:r-- 3f81db61 0 3
```

- 在每次需要监视系统的更改时创建测试清单。使用相同的 `bart` 选项和同一个 rules 文件并按照与控制清单完全相同的方式来准备测试清单。

- 使用同一个 rules 文件比较清单。

▼ 如何比较同一系统在一段时间内的清单

当您要监视同一系统在一段时间内的文件层更改时，请使用此过程。此类清单可以帮助找到损坏或异常的文件，检测安全性破坏，或对系统的性能问题进行疑难解答。

1 承担主管理员角色，或成为超级用户。

主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2 安装 Solaris 软件之后，创建系统上要监视的文件的控制清单。

```
# bart create -R /etc > control-manifest
```

3 在每次需要监视系统更改时创建一个在准备方式上与控制清单完全相同的测试清单。

```
# bart create -R /etc > test-manifest
```

4 将控制清单与测试清单进行比较。

```
# bart compare options control-manifest test-manifest > bart-report
```

`-r`	此比较的 rules 文件的名称。将 `-r` 选项和 – 一起使用意味着从标准输入中读取指令。
`-i`	允许用户从命令行设置全局 `IGNORE` 指令。
`-p`	生成用于进行程序分析的标准非本地化输出的程序模式。
control-manifest	控制系统的 `bart create` 命令输出。
test-manifest	测试系统的 `bart create` 命令输出。

5 检查 BART 报告中的异常情况。

示例 5–5 比较同一系统在一段时间内的清单

此示例说明了如何监视两个时间点之间 `/etc` 目录中发生的更改。此类比较使您可以迅速确定系统上的重要文件是否受到威胁。

- 创建控制清单。

```
# bart create -R /etc > system1.control.121203

! Version 1.0

! Friday, December 12, 2003 (08:34:51)
```

```
# Format:
#fname D size mode acl dirmtime uid gid
#fname P size mode acl mtime uid gid
#fname S size mode acl mtime uid gid
#fname F size mode acl mtime uid gid contents
#fname L size mode acl lnmtime uid gid dest
#fname B size mode acl mtime uid gid devnode
#fname C size mode acl mtime uid gid devnode
/ D 4096 40755 user::rwx,group::r-x,mask:r-x,other:r-x 3fd9dfb4 0 3
/.cpr_config F 2236 100644 user::rw-,group::r--,mask:r--,other:r--
3fd9991f 0 0
67cfa2c830b4ce3e112f38c5e33c56a2
/.group.lock F 0 100600 user::rw-,group::---,mask:---,other:--- 3f81f14d
0 1 d41
d8cd98f00b204e9800998ecf8427e
/.java D 512 40755 user::rwx,group::r-x,mask:r-x,other:r-x 3f81dcb5 0 2
/.java/.systemPrefs D 512 40755 user::rwx,group::r-x,mask:r-x,
other:r-x 3f81dcb7
.
.
.
```

- 在需要监视 /etc 目录的更改时创建测试清单。

```
# bart create -R /etc > system1.test.121503
Version 1.0
```

```
! Monday, December 15, 2003 (08:35:28)

# Format:

#fname D size mode acl dirmtime uid gid

#fname P size mode acl mtime uid gid

#fname S size mode acl mtime uid gid

#fname F size mode acl mtime uid gid contents

#fname L size mode acl lnmtime uid gid dest

#fname B size mode acl mtime uid gid devnode

#fname C size mode acl mtime uid gid devnode

/ D 4096 40755 user::rwx,group::r-x,mask:r-x,other:r-x 3fd9dfb4 0 3

/.cpr_config F 2236 100644 user::rw-,group::r--,mask:r--,other:r--

3fd9991f 0 0

67cfa2c830b4ce3e112f38c5e33c56a2

/.group.lock F 0 100600 user::rw-,group::---,mask:---,other:---

3f81f14d 0 1 d41d8cd98f00b204e9800998ecf8427e

/.java D 512 40755 user::rwx,group::r-x,mask:r-x,other:r-x 3f81dcb5 0 2

/.java/.systemPrefs D 512 40755 user::rwx,group::r-x,mask:r-x,

other:r-x 3f81dcb70 2

/.java/.systemPrefs/.system.lock F 0 100644 user::rw-,group::r--

,mask:r--,other:

r-- 3f81dcb5 0 2 d41d8cd98f00b204e9800998ecf8427e

/.java/.systemPrefs/.systemRootModFile F 0 100644 user::rw-,

group::r--,mask:r--,
```

```
other:r-- 3f81dd0b 0 2 d41d8cd98f00b204e9800998ecf8427e

.

.

.
```

- 将控制清单与测试清单进行比较。

```
# bart compare system1.control.121203 system1.test.121503

/vfstab:

  mode  control:100644  test:100777

  acl  control:user::rw-,group::r--,mask:r--,other:r--  test:user::rwx,

group::rwx,mask:rwx,other:rwx
```

上面的输出指示 `vfstab` 文件的权限自创建了控制清单以来已发生更改。此报告可以用于检查拥有权、日期、内容或任何其他文件属性是否已发生变化。具备此类信息有助于跟踪可能的文件篡改者和更改可能发生的时间。

▼ 如何比较不同系统的清单与控制系统的清单

您可以运行系统间比较，这样可以迅速确定在基准系统和其他系统之间是否存在任何文件层差异。例如，如果您已经在基准系统上安装了特定版本的 Solaris 软件，并且需要了解其他系统是否也安装了相同的软件包，则可以创建那些系统的清单，然后将测试清单与控制清单进行比较。此类比较会列出与控制系统比较的每个测试系统在文件内容方面的任何差异。

1 承担主管理员角色，或成为超级用户。

主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2 安装 Solaris 软件之后，创建控制清单。

```
# bart create options > control-manifest
```

3 保存控制清单。

4 在测试系统上，使用相同的 `bart` 选项创建清单，并将输出重定向到文件中。

```
# bart create options > test1-manifest
```

为测试清单选择一个特殊而有意义的名称。

5 **将测试清单一直保存在系统的中心位置，直至准备比较清单为止。**

6 **需要比较清单时，将控制清单复制到测试清单的位置。或者，将测试清单复制到控制系统。**

例如：

cp *control-manifest /net/test-server/bart/manifests*

如果测试系统不是已挂载 NFS 系统，则使用 FTP 或一些其他某个可靠方法将控制清单复制到测试系统。

7 **将控制清单与测试清单进行比较并将输出重定向到文件。**

 # bart compare *control-manifest test1-manifest > test1.report*

8 **检查 BART 报告中的异常情况。**

9 **对于每个需要与控制清单比较的测试清单，重复执行步骤 4 至步骤 9。**

对于每个测试系统使用相同的 bart 选项。

示例 5–6 比较不同系统的清单与控制系统的清单

此示例介绍了如何通过比较控制清单与不同系统的测试清单来监视 /usr/bin 目录内容的更改。

- 创建控制清单。

```
# bart create -R /usr/bin > control-manifest.121203

!Version 1.0

! Friday, December 12, 2003 (09:19:00)

# Format:

#fname D size mode acl dirmtime uid gid

#fname P size mode acl mtime uid gid

#fname S size mode acl mtime uid gid

#fname F size mode acl mtime uid gid contents

#fname L size mode acl lnmtime uid gid dest
```

```
#fname B size mode acl mtime uid gid devnode
#fname C size mode acl mtime uid gid devnode
/ D 13312 40755 user::rwx,group::r-x,mask:r-x,other:r-x 3fd9e925 0 2
/.s F 14200 104711 user::rwx,group::--x,mask:--x,other:--x
3f8dbfd6 0 1 8ec7e52d8a35ba3b054a6394cbf71cf6
/ControlPanel L 28 120777 - 3f81dc71 0 1 jre/bin/ControlPanel
/HtmlConverter L 25 120777 - 3f81dcdc 0 1 bin/HtmlConverter
/acctcom F 28300 100555 user::r-x,group::r-x,mask:r-x,other:r-x
3f6b5750 0 2 d6e99b19c847ab4ec084d9088c7c7608
/activation-client F 9172 100755 user::rwx,group::r-x,mask:r-x,
other:r-x 3f5cb907 0 1 b3836ad1a656324a6e1bd01edcba28f0
/adb F 9712 100555 user::r-x,group::r-x,mask:r-x,other:r-x
3f6b5736 0 2 5e026413175f65fb239ee628a8870eda
/addbib F 11080 100555 user::r-x,group::r-x,mask:r-x,other:r-x
3f6b5803 0 2 a350836c36049febf185f78350f27510
.
.
.
```

- 为需要与控制系统进行比较的系统创建测试清单。

```
# bart create -R /usr/bin > system2-manifest.121503
! Version 1.0
! Friday, December 15, 2003 (13:30:58)
# Format:
#fname D size mode acl dirmtime uid gid
```

```
#fname P size mode acl mtime uid gid

#fname S size mode acl mtime uid gid

#fname F size mode acl mtime uid gid contents

#fname L size mode acl lnmtime uid gid dest

#fname B size mode acl mtime uid gid devnode

#fname C size mode acl mtime uid gid devnode

/ D 13312 40755 user::rwx,group::r-x,mask:r-x,other:r-x 3fd9ea9c 0 2

/.s F 14200 104711 user::rwx,group::--x,mask:--x,other:--x

3f8dbfd6 0 1 8ec7e52d8a35ba3b054a6394cbf71cf6

/ControlPanel L 28 120777 - 3f81dc71 0 1 jre/bin/ControlPanel

/HtmlConverter L 25 120777 - 3f81dcdc 0 1 bin/HtmlConverter

/acctcom F 28300 100555 user::r-x,group::r-x,mask:r-x,other:

r-x 3f6b5750 0 2 d6e99b19c847ab4ec084d9088c7c7608

.

.

.
```

- 需要比较清单时，将清单复制到同一位置。

```
# cp control-manifest /net/system2.central/bart/manifests
```

- 将控制清单与测试清单进行比较。

```
# bart compare control-manifest  system2.test > system2.report

/su:

  gid  control:3  test:1

/ypcat:

  mtime  control:3fd72511  test:3fd9eb23
```

上面的输出指示了 /usr/bin 目录中 su 文件的组 ID 与控制系统中的组 ID 不同。此信息有助于确定测试系统上是否安装了不同版本的软件或是否有人篡改了文件。

▼ 如何通过指定文件属性自定义 BART 报告

此过程为可选过程，它介绍了如何通过从命令行指定文件属性来自定义 BART 报告。如果创建了列出有关系统上所有文件或特定文件的信息的基准清单，则可以在需要监视特定目录、子目录、一个或多个文件的更改的时运行 bart compare 命令，并指定不同的属性。您可以通过从命令行指定不同的文件属性来针对同一清单运行不同类型的比较。

1. **确定需要监视的文件属性。**

2. **承担主管理员角色，或成为超级用户。**

   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

3. **安装 Solaris 软件之后，创建控制清单。**

4. **在需要监视更改时创建测试清单。**

   按照准备控制清单的方式准备测试清单。

5. **比较清单。**

   例如：

   ```
   # bart compare -i dirmtime,lnmtime,mtime control-manifest.121503 \

   test-manifest.010504 > bart.report.010504
   ```

   请注意，使用逗号分隔在命令行语法中指定的各个属性。

6. **检查 BART 报告中的异常情况。**

▼ 如何通过使用 Rules 文件自定义 BART 报告

此过程也是可选过程，它介绍了如何通过使用 rules 文件作为 bart compare 命令的输入来自定义 BART 报告。通过使用 rules 文件，可以自定义 BART 报告，从而使您能够灵活地为多个文件或子树指定多个属性。可以使用不同的 rules 文件来针对同一清单运行不同的比较。

1. **承担主管理员角色，或成为超级用户。**

   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2. **确定需要监视的文件和文件属性。**

3. **使用文本编辑器创建具有相应指令的 rules 文件。**

4. **安装 Solaris 软件之后，使用所创建的 rules 文件创建一个控制清单。**

   ```
   # bart create -r rules-file > control-manifest
   ```

5. **创建在准备方式上与控制清单完全相同的测试清单。**

   ```
   # bart create -r rules-file > test-manifest
   ```

6. **使用同一个 rules 文件比较控制清单与测试清单。**

   ```
   # bart compare -r rules-file control-manifest test-manifest > bart.report
   ```

7. **检查 BART 报告中的异常情况。**

示例 5–7 通过使用 Rules 文件自定义 BART 报告

以下 rules 文件同时包括 `bart create` 和 `bart compare` 命令的指令。此 rules 文件指示 `bart create` 命令列出有关 `/usr/bin` 目录内容的信息。此外，rules 文件还指示 `bart compare` 命令仅跟踪同一目录中大小和内容方面的更改。

```
# Check size and content changes in the /usr/bin directory.

# This rules file only checks size and content changes.

# See rules file example.



IGNORE all

CHECK size contents

/usr/bin
```

- 使用已创建的 rules 文件创建控制清单。

  ```
  # bart create -r bartrules.txt > usr_bin.control-manifest.121003
  ```

- 在需要监视 `/usr/bin` 目录的更改时创建测试清单。

  ```
  # bart create -r bartrules.txt > usr_bin.test-manifest.121103
  ```

- 使用同一个 rules 文件比较清单。

```
# bart compare -r bartrules.txt usr_bin.control-manifest \
usr_bin.test-manifest
```

- 检查 bart compare 命令的输出。

```
 /usr/bin/gunzip:  add
/usr/bin/ypcat:
  delete
```

在上面的输出中，bart compare 命令报告 /usr/bin 目录中的差异。此输出指示已删除了 /usr/bin/ypcat 文件，并添加了 /usr/bin/gunzip 文件。

BART 清单、Rules 文件和报告（ 参考 ）

本节包含以下参考信息：

- 第 113 页中的 “BART 清单文件格式”
- 第 114 页中的 “BART Rules 文件格式”
- 第 116 页中的 “BART 报告”

BART 清单文件格式

每个清单文件项均是单独的一行，具体取决于文件类型。每个项都以 *fname*（即文件名）开头。为了避免分析文件名中嵌入的特殊字符所导致的问题，已对文件名进行了编码。有关更多信息，请参见第 114 页中的 “BART Rules 文件格式”。

后续字段表示以下文件属性：

type 文件类型，可能值为：

- B 表示块设备节点
- C 表示字符设备节点
- D 表示目录
- F 表示文件
- L 表示符号链接
- P 表示管道
- S 表示套接字

size 以字节为单位的文件大小。

mode 表示文件权限的八进制数。

acl 文件的 ACL 属性。对于具有 ACL 属性的文件，它包含了 acltotext() 的输出。

uid 此项的属主的数字用户 ID。

gid 此项的属主的数字组 ID。

dirmtime 目录上次修改的时间，以秒为单位，从 1970 年 1 月 1 日 00:00:00 UTC（国际协调时间）开始计算。

lnmtime 链接上次修改的时间，以秒为单位，从 1970 年 1 月 1 日 00:00:00 UTC 开始计算。

mtime 文件上次修改的时间，以秒为单位，从 1970 年 1 月 1 日 00:00:00 UTC 开始计算。

contents 文件校验和的值。此属性仅为常规文件指定。如果关闭上下文检查，或者无法计算校验和，则此字段的值为 –。

dest 符号链接的目标。

devnode 设备节点值。此属性仅用于字符设备文件和块设备文件。

有关 BART 清单的更多信息，请参见 `bart_manifest(4)` 手册页。

BART Rules 文件格式

`bart` 命令的输入文件为文本文件。这些文件由行组成，行中指定了要包括在清单中的文件和要包括在报告中的文件属性。同一个输入文件可同时在两项 BART 功能中使用。工具将忽略以 `#` 开头的行、空白行以及包含空格的行。

输入文件包含三种类型的指令：

- 子树指令，带有可选的模式匹配修饰符
- `CHECK` 指令
- `IGNORE` 指令

示例 5–8 Rules 文件格式

```
<Global CHECK/IGNORE Directives>

<subtree1> [pattern1..]

<IGNORE/CHECK Directives for subtree1>



<subtree2> [pattern2..]

subtree3> [pattern3..]

subtree4> [pattern4..]
```

示例 5–8 Rules 文件格式 （续）

```
<IGNORE/CHECK Directives for subtree2, subtree3, subtree4>
```

注 – 所有指令均会按顺序进行读取，后面的指令可能会覆盖前面的指令。

每行都有一个子树指令。指令**必须**以绝对路径名开头，后面跟有零个或多个模式匹配语句。

Rules 文件属性

`bart` 命令使用 `CHECK` 和 `IGNORE` 语句定义要跟踪或忽略的属性。每个属性都有一个关联的关键字。

属性**关键字**有：

- `acl`
- `all`
- `contents`
- `dest`
- `devnode`
- `dirmtime`
- `gid`
- `lnmtime`
- `mode`
- `mtime`
- `size`
- `type`
- `uid`

`all` 关键字是指所有文件属性。

引用语法

BART 所用的 rules 文件规范语言是用于表示非标准文件名的标准 UNIX 引用语法。嵌入的制表符、空格、换行符或特殊字符以八进制格式编码，以保证工具能够读取文件名。这种不一致的引用语法会阻止在命令管道中正确处理某些文件名，如包含嵌入的回车的文件名。使用 rules 规范语言可表达复杂的文件名过滤条件，这些条件如果仅使用 shell 语法会很难并且无法有效表达。

有关 BART rules 文件或 BART 所用的引用语法的更多信息，请参见 `bart_rules(4)` 手册页。

BART 报告

在缺省模式下，bart compare 命令会检查系统上安装的除已修改的目录时间标记 (dirmtime) 外所有文件，如以下示例所示：

```
CHECK all

IGNORE    dirmtime
```

如果提供了 rules 文件，则全局指令 CHECK all 和 IGNORE dirmtime 会按照以上顺序自动前置到 rules 文件之前。

BART 输出

将返回以下退出值：

0　成功

1　处理文件时出现非致命错误，如权限问题

>1　出现致命错误，如无效的命令行选项

报告机制可提供两种类型的输出：详细输出和程序输出：

- 详细输出为缺省输出，已本地化并出现在多行中。详细输出已经过国际化并具有可读性。如果使用 bart compare 命令对两份系统清单进行比较，则会生成一个文件差异列表。

  例如：

  filename attribute control:xxxx test:yyyy

  filename　在控制清单和测试清单中各不相同的文件的名称。

  attribute　在进行比较的清单中各不相同的文件属性的名称。*xxxx* 是控制清单的属性值，*yyyy* 是测试清单的属性值。如果同一个文件中的多个属性出现差异，则每个差异都将记录在单独的一行中。

  下面是 bart compare 命令的缺省输出的示例。/etc/passwd 文件中出现属性差异。输出指明 size、mtime 和 contents 属性已发生变化。

  ```
  /etc/passwd:

  size    control:74    test:81

  mtime control:3c165879    test:3c165979

  contents    control:daca28ae0de97afd7a6b91fde8d57afa

  test:84b2b32c4165887355317207b48a6ec7
  ```

- 如果在运行 `bart compare` 命令时使用 `-p` 选项，则会生成程序输出。此输出以适合程序操作的格式生成。程序输出可以由其他程序轻松分析并且旨在用作其他工具的输入。

  例如：

  filename attribute control-val test-val [*attribute control-val test-val*]*

filename	与缺省格式中的 *filename* 属性相同
attribute control-val test-val	每个文件在控制清单和测试清单中不同的文件属性的说明

有关 `bart` 命令支持的属性的列表，请参见第 115 页中的 “Rules 文件属性”。

有关 BART 的更多信息，请参见 `bart`(1M) 手册页。

◆ ◆ ◆ 第 6 章

控制对文件的访问（任务）

本章介绍如何保护 Solaris 操作系统 (Solaris Operating System, Solaris OS) 中的文件，还将介绍如何防范其权限可能危及系统安全的文件。

以下是本章中信息的列表：

- 第 119 页中的 “使用 UNIX 权限保护文件”
- 第 125 页中的 “使用访问控制列表保护文件”
- 第 127 页中的 “防止可执行文件危及安全”
- 第 127 页中的 “保护文件（任务列表）”
- 第 128 页中的 “使用 UNIX 权限保护文件（任务列表）”
- 第 134 页中的 “使用 ACL 保护文件（任务列表）”
- 第 140 页中的 “防止程序受到安全风险（任务列表）”

使用 UNIX 权限保护文件

通过 UNIX 文件权限和 ACL 可保证文件安全。带 sticky 位的文件和可执行文件要求特殊的安全措施。

用于查看和保证文件安全的命令

下表给出了用于监视以及保证文件和目录安全的命令。

表 6–1 保证文件和目录安全的命令

命令	说明	手册页
`ls`	列出目录中的文件及其有关信息。	`ls`(1)
`chown`	更改文件的拥有权。	`chown`(1)

表 6–1 保证文件和目录安全的命令 （续）

命令	说明	手册页
chgrp	更改文件的组拥有权。	chgrp(1)
chmod	更改文件的权限。可以使用符号模式（使用字母和符号）或绝对模式（使用八进制数字）更改文件的权限。	chmod(1)

文件和目录的拥有权

传统 UNIX 文件权限可以为三类用户指定拥有权：

- **用户 –** 文件或目录的属主，通常为创建该文件的用户。文件的属主可以决定谁拥有读取文件、写入文件（对文件进行更改）或执行文件（如果该文件为命令）的权限。
- **组 –** 一组用户的成员。
- **其他用户 –** 所有其他不是文件属主和组成员的用户。

文件属主通常可以指定或修改文件权限。此外，具有管理功能的用户或角色（如超级用户或主管理员角色）可以更改文件的拥有权。要覆盖系统策略，请参见示例 6–2。

文件可以是七种类型之一。每种类型由一个符号显示：

- （减号） 文本或程序

b 块特殊文件

c 字符特殊文件

d 目录

l 符号链接

s 套接字

D 门

P 命名管道 (FIFO)

UNIX 文件权限

下表列出并说明了可以为文件或目录的每类用户授予的权限。

表 6–2 文件和目录权限

符号	权限	对象	说明
r	读	文件	指定的用户可以打开和读取文件内容。

表 6–2 文件和目录权限 （续）

符号	权限	对象	说明
		目录	指定的用户可以列出目录中的文件。
w	写	文件	指定的用户可以修改文件的内容或删除该文件。
		目录	指定的用户可以在目录中添加文件或链接。这些用户也可以删除目录中的文件或链接。
x	执行	文件	指定的用户可以执行文件（如果该文件为程序或 shell 脚本）。这些用户也可以使用一个 exec(2) 系统调用来运行程序。
		目录	指定的用户可以打开或执行目录中的文件。这些用户也可以使该目录以及该目录下的目录成为当前目录。
-	拒绝	文件和目录	指定的用户无法读写或执行文件。

这些文件权限可应用于常规文件，也可应用于特殊文件（如设备、套接字和命名管道 (FIFO)）。

对于符号链接，所应用的权限为链接指向的文件权限。

通过对目录设置受限文件权限，可以保护该目录及其子目录中的文件。但是请注意，超级用户有权访问系统中的所有文件和目录。

特殊文件权限（ setuid、setgid 和 Sticky 位）

可执行文件和公共目录可以使用三种特殊类型的权限：setuid、setgid 和 sticky 位。设置这些权限之后，运行可执行文件的任何用户都应采用该可执行文件属主（或组）的 ID。

设置特殊权限时必须非常小心，因为特殊权限会带来安全风险。例如，通过执行将用户 ID (user ID, UID) 设置为 0（root 的 UID）的程序，用户可以获取超级用户功能。此外，所有用户可以为其拥有的文件设置特殊权限，这会带来其他安全问题。

应对系统中未经授权使用 setuid 权限和 setgid 权限获取超级用户功能的情况进行监视。可疑权限为用户而不是 root 或 bin 授予管理程序的拥有权。要搜索并列出所有使用此特殊权限的文件，请参见第 140 页中的 “如何使用特殊文件权限查找文件”。

setuid 权限

对可执行文件设置 setuid 权限时，将对运行该文件的进程授予基于文件属主的访问权限。该访问权限**不是**基于正在运行可执行文件的用户。使用此特殊权限，用户可以访问通常只有属主才可访问的文件和目录。

例如，passwd 命令的 setuid 权限使用户可以更改口令。拥有 setuid 权限的 passwd 命令与以下类似：

```
-r-sr-sr-x   3 root     sys        28144 Jun 17 12:02 /usr/bin/passwd
```

此特殊权限会带来安全风险。一些确定的用户甚至可以在 setuid 进程执行完毕后，找到保持由该进程授予他们的权限的方法。

注 – 在程序中使用具有保留 UID (0–100) 的 setuid 权限可能无法正确设置有效的 UID。请使用 shell 脚本或避免将保留的 UID 用于 setuid 权限。

---

setgid 权限

setgid 权限与 setuid 权限类似。可将进程的有效组 ID (group ID, GID) 更改为拥有该文件的组，并基于授予该组的权限对用户授权访问权限。/usr/bin/mail 命令拥有 setgid 权限：

```
-r-x--s--x   1 root    mail     67504 Jun 17 12:01 /usr/bin/mail
```

将 setgid 权限应用于目录时，该目录中已创建的文件将属于该目录所属于的组。这些文件不属于创建进程所属于的组。在目录中拥有写和执行权限的任何用户都可以在其中创建文件。但是，文件将属于拥有该目录的组，而不是用户所属于的组。

应对系统中未经授权使用 setgid 权限获取超级用户功能的情况进行监视。可疑权限为非常规组而不是 root 或 bin 授予对此类程序的访问权限。要搜索并列出所有使用此权限的文件，请参见第 140 页中的 “如何使用特殊文件权限查找文件”。

Sticky 位

sticky **位**是保护目录中文件的权限位。如果对目录设置了 sticky 位，则只有文件属主、目录属主或特权用户才可以删除文件。root 用户和主管理员角色即是特权用户。sticky 位禁止用户从公共目录（如 /tmp）中删除其他用户的文件：

```
drwxrwxrwt 7  root  sys   400 Sep  3 13:37 tmp
```

在 TMPFS 文件系统中设置公共目录时，务必手动设置 sticky 位。有关说明，请参见示例 6–5。

缺省 umask 值

创建文件或目录时，将使用一组缺省权限进行创建。系统缺省值为空。文本文件拥有 666 权限，该权限对所有用户授予读写权限。目录和可执行文件拥有 777 权限，该权限对所有用户授予读写和执行权限。通常，用户会覆盖其 /etc/profile 文件、.cshrc 文件或 .login 文件中的系统缺省值。

由 umask 命令指定的值将从缺省值中减去。此进程的作用是以 chmod 命令授予权限的相同方式拒绝这些权限。例如，chmod 022 对组和其他用户授予写权限。umask 022 命令拒绝组和其他用户的写权限。

下表给出了一些典型 umask 设置及其对可执行文件的影响。

表 6–3 不同安全级别的 umask 设置

安全级别	umask 设置	禁用的权限
许可 (744)	022	w（组和其他用户）
中等 (740)	027	w（组），rwx（其他用户）
中等 (741)	026	w（组），rw（其他用户）
严重 (700)	077	rwx（组和其他用户）

有关设置 umask 值的更多信息，请参见 umask(1) 手册页。

文件权限模式

使用 chmod 命令，可以更改文件的权限。要更改文件的权限，您必须是超级用户或是文件或目录的属主。

可以使用 chmod 命令按照以下两种模式之一设置权限：

- **绝对模式** – 使用数字表示文件权限。使用绝对模式更改权限时，由八进制模式数字表示每个三元字节权限。绝对模式是设置权限的最常用方法。
- **符号模式** – 使用字母和符号的组合来添加或删除权限。

下表列出了在绝对模式下设置文件权限的八进制值。可按顺序以三个一组的形式，使用这些数字来设置属主、组和其他用户的权限。例如，值 644 为属主设置读写权限，为组和其他用户设置只读权限。

表 6–4 在绝对模式下设置文件权限

八进制值	设置文件权限	权限说明
0	---	无权限
1	--x	仅执行权限
2	-w-	只写权限
3	-wx	写和执行权限
4	r--	只读权限
5	r-x	读和执行权限
6	rw-	读写权限
7	rwx	读写和执行权限

下表列出了用于在符号模式下设置文件权限的符号。符号可以指定要设置或更改其权限的用户、要执行的操作，以及要指定或更改的权限。

表 6–5 在符号模式下设置文件权限

符号	功能	说明
u	*who*	用户（属主）
g	*who*	组
o	*who*	其他用户
a	*who*	所有
=	*operator*	赋值
+	*operator*	添加
-	*operator*	删除
r	*permissions*	读
w	*permissions*	写
x	*permissions*	执行
l	*permissions*	强制锁定，`setgid` 位打开，组执行位关闭
s	*permissions*	`setuid` 或 `setgid` 位打开
t	*permissions*	Sticky 位打开，对于其他用户，执行位打开

功能列中的名称 *who operator permissions* 指定用于更改文件或目录的权限的符号。

who 指定要更改其权限的用户。

operator 指定要执行的操作。

permissions 指定要更改的权限。

可以在绝对模式或符号模式下设置文件的特殊权限。但是，必须使用符号模式设置或删除目录的 `setuid` 权限。在绝对模式下，通过在权限三元字节的左侧添加新的八进制值，可设置特殊权限。下表列出了用于对文件设置特殊权限的八进制值。

表 6–6 在绝对模式下设置特殊文件权限

八进制值	特殊文件权限
1	Sticky 位
2	`setgid`
4	`setuid`

使用访问控制列表保护文件

传统 UNIX 文件保护可为以下三类用户提供读写和执行权限：文件属主、文件组和其他用户。访问控制列表 (Access Control List, ACL) 通过允许您执行以下操作来提供更好的文件安全性：

- 为文件属主、组、其他用户、特定用户和特定组定义文件权限
- 为上面的每一种类别定义缺省权限

例如，如果想要组中的每个用户都能够读取某文件，则只需要授予该组对该文件的读取权限即可。现在，假设您希望组中只有一个用户能够写入该文件。标准 UNIX 不提供该级别的文件安全性。但是，ACL 可提供此级别的文件安全性。

ACL 项定义文件的 ACL。这些项通过 `setfacl` 命令设置。 ACL 项由以下字段组成并使用冒号进行分隔：

entry-type:[*uid*|*gid*]:*perms*

entry-type 设置文件权限的 ACL 项的类型。例如，*entry-type* 可以是 `user`（文件属主）或 `mask`（ACL 掩码）。有关 ACL 项的列表，请参见表 6–7 和表 6–8。

uid 用户名或用户 ID (user ID, UID)。

gid 组名或组 ID (group ID, GID)。

perms 表示 *entry-type* 中设置的权限。*perms* 可以由符号字符 `rwx` 或八进制数字表示。这些数字与用于 `chmod` 命令的数字相同。

在以下示例中，ACL 项为用户 `stacey` 设置读写权限。

```
user:stacey:rw-
```

注意 – 仅 UFS 文件系统支持 UFS 文件系统属性，例如 ACL。因此，如果将具有 ACL 项的文件恢复或复制到 `/tmp` 目录（通常挂载为 TMPFS 文件系统）中，则这些 ACL 项将丢失。使用 `/var/tmp` 目录临时存储 UFS 文件。

文件的 ACL 项

下表列出了对文件设置 ACL 时可以使用的有效 ACL 项。前三个 ACL 项提供基本的 UNIX 文件保护。

表 6–7 文件的 ACL 项

ACL 项	说明
`u[ser]::`*perms*	文件属主权限。

表 6–7 文件的 ACL 项 （续）

ACL 项	说明
g[roup]::*perms*	文件组权限。
o[ther]:*perms*	文件属主或文件组成员之外的用户的权限。
m[ask]:*perms*	ACL 掩码。掩码项表示允许用户（属主除外）和组拥有的最大权限。掩码是一种可快速更改所有用户和组的权限的方法。 例如，mask:r-- 掩码项表示，用户和组只能拥有读取权限，即使他们可能拥有写和执行权限。
u[ser]:*uid*:*perms*	特定用户的权限。对于 *uid*，可以指定用户名或数字 UID。
g[roup]:*gid*:*perms*	特定组的权限。对于 *gid*，可以指定组名或数字 GID。

目录的 ACL 项

除表 6–7 中说明的 ACL 项外，还可以对目录设置缺省 ACL 项。在具有缺省 ACL 项的目录中创建的文件或目录将具有与缺省 ACL 项相同的 ACL 项。表 6–8 列出了目录的缺省 ACL 项。

首次为特定用户和组设置目录的缺省 ACL 项时，还必须为文件属主、文件组、其他用户和 ACL 掩码设置缺省 ACL 项。这些项是必需的。这些项是下表中前四个缺省 ACL 项。

表 6–8 目录的缺省 ACL 项

缺省 ACL 项	说明
d[efault]:u[ser]::*perms*	缺省文件属主权限。
d[efault]:g[roup]::*perms*	缺省文件组权限。
d[efault]:o[ther]:*perms*	文件属主或文件组成员之外的用户的缺省权限。
d[efault]:m[ask]:*perms*	缺省 ACL 掩码。
d[efault]:u[ser]:*uid*:*perms*	特定用户的缺省权限。对于 *uid*，可以指定用户名或数字 UID。
d[efault]:g[roup]:*gid*:*perms*	特定组的缺省权限。对于 *gid*，可以指定组名或数字 GID。

用于管理 ACL 的命令

以下命令可管理文件或目录的 ACL。

setfacl 命令　设置、添加、修改和删除 ACL 项。有关更多信息，请参见 setfacl(1) 手册页。

getfacl 命令　显示 ACL 项。有关更多信息，请参见 getfacl(1) 手册页。

防止可执行文件危及安全

在将可执行栈的权限设置为读写和执行时，很多安全错误与缺省可执行栈有关。虽然允许栈拥有执行权限，但大多数程序可以在不使用可执行栈的情况下正常运行。

使用 noexec_user_stack 变量，可以指定栈映射是否可执行。从 Solaris 2.6 发行版开始，可以使用该变量。缺省情况下，该变量被设置为零（64 位应用程序中除外），这将提供兼容 ABI 的行为。如果将该变量设置为非零值，则系统会将系统中每个进程的栈标记为可读写，但不可执行。

设置此变量后，将向尝试执行其栈中代码的程序发送一个 SIGSEGV 信号。此信号通常将导致程序终止，同时进行核心转储。这样的程序还将生成一条警告消息，该消息中包括违例程序的名称、进程 ID 和运行该程序的用户的实际 UID。例如：

```
a.out[347] attempt to execute code on stack by uid 555
```

将 syslog kern 功能设置为 notice 级别时，该消息由 syslog 守护进程记录。缺省情况下，将在 syslog.conf 文件中设置此日志，这意味着，消息将发送到控制台和 /var/adm/messages 文件。有关更多信息，请参见 syslogd(1M) 和 syslog.conf(4) 手册页。

syslog 消息用于观察可能的安全问题。通过设置此变量，该消息还将确定依赖于可执行栈（已被禁止，无法正确执行操作）的有效程序。如果不想记录任何消息，则可以在 /etc/system 文件中将 noexec_user_stack_log 变量设置为零。即使未记录消息，SIGSEGV 信号仍可能会导致执行程序终止，同时进行核心转储。

如果希望程序显式将其栈标记为可执行，则可以使用 mprotect() 功能。有关更多信息，请参见 mprotect(2) 手册页。

由于硬件限制，在大多数基于 x86 的系统中不能使用捕获和报告可执行栈问题的功能。AMD64 产品系列中的系统可以捕获和报告可执行栈问题。

保护文件（任务列表）

以下任务列表说明一组保护文件的过程。

任务	说明	参考
使用 UNIX 权限保护文件	查看文件的 UNIX 权限。使用 UNIX 权限保护文件。	第 128 页中的 “使用 UNIX 权限保护文件（任务列表）”
使用 ACL 保护文件	添加 ACL，以便在比 UNIX 权限更周密的级别上保护文件。	第 134 页中的 “使用 ACL 保护文件（任务列表）”
防止系统受到来自文件的安全风险	查找具有可疑拥有权的可执行文件。禁用可能会破坏系统的文件。	第 140 页中的 “防止程序受到安全风险（任务列表）”

使用 UNIX 权限保护文件（任务列表）

以下任务列表说明列出文件权限、更改文件权限，以及使用特殊文件权限保护文件的过程。

任务	参考
显示文件信息	第 128 页中的 “如何显示文件信息”
更改文件拥有权	第 129 页中的 “如何更改文件的属主” 第 130 页中的 “如何更改文件的组拥有权”
更改文件权限	第 131 页中的 “如何在符号模式下更改文件权限” 第 132 页中的 “如何在绝对模式下更改文件权限” 第 133 页中的 “如何在绝对模式下更改特殊文件权限”

▼ 如何显示文件信息

使用 `ls` 命令显示有关目录中所有文件的信息。

◗ **键入以下命令以显示当前目录中所有文件的长列表。**

```
% ls -la
```

`-l` 显示包括用户拥有权、组拥有权和文件权限的长格式。

`-a` 显示所有文件，包括以点 (.) 开头的隐藏文件。

示例 6–1 显示文件信息

在以下示例中，显示了 `/sbin` 目录中部分文件的列表。

```
% cd /sbin

% ls -la

total 13456

drwxr-xr-x   2 root     sys          512 Sep  1 14:11 .

drwxr-xr-x  29 root     root        1024 Sep  1 15:40 ..

-r-xr-xr-x   1 root     bin       218188 Aug 18 15:17 autopush

lrwxrwxrwx   1 root     root          21 Sep  1 14:11 bpgetfile -> ...
```

```
-r-xr-xr-x   1 root     bin       505556 Aug 20 13:24 dhcpagent

-r-xr-xr-x   1 root     bin       456064 Aug 20 13:25 dhcpinfo

-r-xr-xr-x   1 root     bin       272360 Aug 18 15:19 fdisk

-r-xr-xr-x   1 root     bin       824728 Aug 20 13:29 hostconfig

-r-xr-xr-x   1 root     bin       603528 Aug 20 13:21 ifconfig

-r-xr-xr-x   1 root     sys       556008 Aug 20 13:21 init

-r-xr-xr-x   2 root     root      274020 Aug 18 15:28 jsh

-r-xr-xr-x   1 root     bin       238736 Aug 21 19:46 mount

-r-xr-xr-x   1 root     sys         7696 Aug 18 15:20 mountall

   .

   .

   .
```

每一行按以下顺序显示了有关文件的信息：

- 文件的类型－例如，`d`。有关文件类型的列表，请参见第 120 页中的 “文件和目录的拥有权”。
- 权限－例如，`r-xr-xr-x`。有关说明，请参见第 120 页中的 “文件和目录的拥有权”。
- 硬链接数－例如，`2`。
- 文件的属主－例如，`root`。
- 文件的组－例如，`bin`。
- 文件的大小（以字节为单位）－例如，`7696`。
- 创建文件的日期或上次更改文件的日期－例如，`Aug 18 15:20`。
- 文件名－例如，`mountall`。

▼ 如何更改文件的属主

文件属主、主管理员角色或超级用户可以更改任何文件的拥有权。

1 显示文件的权限。

```
% ls -l example-file

-rw-r--r--   1 janedoe   staff   112640 May 24 10:49 example-file
```

2 **承担主管理员角色，或成为超级用户。**

主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

3 **更改文件的属主。**

```
# chown stacey example-file
```

4 **检验文件的属主是否已更改。**

```
# ls -l example-file

-rw-r--r--   1 stacey   staff   112640 May 26 08:50 example-file
```

示例 6–2 允许用户更改其他用户拥有的文件的拥有权

安全注意事项 – 您应该有合理理由通过将 `rstchown` 变量设置为零来覆盖系统安全策略。访问系统的任何用户都可以更改系统中任何文件的拥有权。

在此示例中，在 `/etc/system` 文件中将 `rstchown` 变量的值设置为零。通过此设置，文件属主可以使用 `chown` 命令将文件的拥有权更改为另一用户。通过此设置，文件属主还可以使用 `chgrp` 命令将文件的组拥有权设置为非其所在的组。重新引导系统后，更改将生效。

```
set rstchown = 0
```

有关更多信息，请参见 `chown`(1) 和 `chgrp`(1) 手册页。

此外，请注意，已挂载 NFS 的文件系统对更改拥有权和组有更多限制。有关限制对已挂载 NFS 的系统的访问的更多信息，请参见《System Administration Guide: Network Services》中的第 6 章，“Accessing Network File Systems (Reference)”。

▼ 如何更改文件的组拥有权

1 **承担主管理员角色，或成为超级用户。**

主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2 **更改文件的组拥有权。**

```
$ chgrp scifi example-file
```

有关设置组的信息，请参见《System Administration Guide: Basic Administration》中的第 4 章，“Managing User Accounts and Groups (Overview)”。

3 **检验文件的组拥有权是否已更改。**

```
$ ls -l example-file

 -rw-r--r--   1 stacey   scifi   112640 June 20 08:55  example-file
```

另请参见示例 6–2。

▼ 如何在符号模式下更改文件权限

1 **如果您不是文件或目录的属主，请成为超级用户或承担等效角色。**

只有当前属主或超级用户可以使用 chmod 命令更改文件或目录的文件权限。

2 **在符号模式下更改权限。**

```
% chmod who operator permissions filename
```

who	指定要更改其权限的用户。
operator	指定要执行的操作。
permissions	指定要更改的权限。有关有效的符号列表，请参见表 6–5。
filename	指定文件或目录。

3 **检验文件的权限是否已更改。**

```
% ls -l filename
```

示例 6–3 在符号模式下更改权限

在以下示例中，将解除其他用户的读取权限。

```
% chmod o-r example-file1
```

在以下示例中，将为用户、组和其他用户添加读和执行权限。

```
$ chmod a+rx example-file2
```

在以下示例中，将为组指定读写和执行权限。

```
$ chmod g=rwx example-file3
```

▼ 如何在绝对模式下更改文件权限

1 **如果您不是文件或目录的属主，请成为超级用户或承担等效角色。**

只有当前属主或超级用户可以使用 chmod 命令更改文件或目录的文件权限。

2 **在绝对模式下更改权限。**

```
% chmod nnn filename
```

nnn 按照该顺序指定将表示文件属主、文件组和其他用户的权限的八进制值。有关有效八进制值的列表，请参见表 6–4。

filename 指定文件或目录。

注 – 使用 chmod 命令更改具有 ACL 项的文件的文件组权限时，文件组权限和 ACL 掩码都将更改为新权限。请注意，新 ACL 掩码权限可以更改在文件中具有 ACL 项的其他用户和组的权限。使用 getfacl 命令以确保为所有 ACL 项都设置了适当的权限。有关更多信息，请参见 getfacl(1) 手册页。

---

3 **检验文件的权限是否已更改。**

```
% ls -l filename
```

示例 6–4 在绝对模式下更改权限

在以下示例中，将公共目录的权限从 744（读、写、执行；只读；只读）更改为 755（读、写、执行；读和执行；读和执行）。

```
# ls -ld public_dir

drwxr--r--  1 ignatz   staff    6023 Aug  5 12:06 public_dir

# chmod 755 public_dir

# ls -ld public_dir

drwxr-xr-x  1 ignatz   staff    6023 Aug  5 12:06 public_dir
```

在以下示例中，将可执行 shell 脚本的权限从读写更改为读写和执行。

```
% ls -l my_script

-rw------- 1 ignatz   staff    6023 Aug  5 12:06 my_script

% chmod 700 my_script
```

```
% ls -l my_script

-rwx------ 1 ignatz   staff    6023 Aug  5 12:06 my_script
```

▼ 如何在绝对模式下更改特殊文件权限

1. **如果您不是文件或目录的属主，请成为超级用户或承担等效角色。**
   只有当前属主或具有超级用户功能的用户可以使用 chmod 命令更改文件或目录的特殊权限。

2. **在绝对模式下更改特殊权限。**
   `% chmod` *nnnn filename*

   nnnn 指定用于更改文件或目录的权限的八进制值。最左侧的八进制值设置文件的特殊权限。有关特殊权限的有效八进制值的列表，请参见表 6–6。

   filename 指定文件或目录。

   注 – 使用 chmod 命令更改具有 ACL 项的文件的文件组权限时，文件组权限和 ACL 掩码都将更改为新权限。请注意，新 ACL 掩码权限可以更改在文件中具有 ACL 项的其他用户和组的权限。使用 getfacl 命令以确保为所有 ACL 项都设置了适当的权限。有关更多信息，请参见 getfacl(1) 手册页。

3. **检验文件的权限是否已更改。**
   `% ls -l` *filename*

示例 6–5 在绝对模式下设置特殊文件权限

在以下示例中，将对 dbprog 文件设置 setuid 权限。

```
# chmod 4555 dbprog

# ls -l dbprog

-r-sr-xr-x   1 db     staff        12095 May  6 09:29 dbprog
```

在以下示例中，将对 dbprog2 文件设置 setgid 权限。

```
# chmod 2551 dbprog2

# ls -l dbprog2

-r-xr-s--x   1 db     staff       24576 May  6 09:30 dbprog2
```

在以下示例中，将对 public_dir 目录设置 sticky 位权限。

```
# chmod 1777 public_dir

# ls -ld public_dir

drwxrwxrwt   2 ignatz   staff          512 May 15 15:27 public_dir
```

使用 ACL 保护文件（ 任务列表 ）

以下任务列表说明列出文件的 ACL、更改 ACL，以及将 ACL 复制到另一个文件的过程。

任务	参考
确定文件是否具有 ACL	第 134 页中的 “如何检查文件是否具有 ACL”
将 ACL 添加到文件	第 135 页中的 “如何将 ACL 项添加到文件”
复制 ACL	第 137 页中的 “如何复制 ACL”
修改 ACL	第 137 页中的 “如何更改文件的 ACL 项”
删除文件的 ACL	第 138 页中的 “如何删除文件的 ACL 项”
显示文件的 ACL	第 138 页中的 “如何显示文件的 ACL 项”

▼ 如何检查文件是否具有 ACL

◗ **检查文件是否具有 ACL。**

```
% ls -l filename
```

其中，*filename* 指定文件或目录。

在输出中，模式字段右侧的加号 (+) 表示该文件具有 ACL。

注 – 除非已添加了用于扩展 UNIX 文件权限的 ACL 项，否则会将文件视为具有“琐碎” ACL，并且不会显示加号 (+)。

示例 6–6 检查文件是否具有 ACL

在以下示例中，ch1.sgm 文件具有 ACL。ACL 由模式字段右侧的加号 (+) 表示。

```
% ls -l ch1.sgm

-rwxr-----+  1 stacey   techpubs      167 Nov 11 11:13 ch1.sgm
```

▼ 如何将 ACL 项添加到文件

1 使用 `setfacl` 命令设置文件的 ACL。

```
% setfacl -s user::perms,group::perms,other:perms,mask:perms,acl-entry-list filename ...
```

`-s`	设置文件的 ACL。如果文件已具有 ACL，则会替换该 ACL。此选项要求至少有 `user::`、`group::` 和 `other::` 项。
`user::`*perms*	指定文件属主权限。
`group::`*perms*	指定组属主权限。
`other:`*perms*	为文件属主或组成员之外的用户指定权限。
`mask:`*perms*	指定 ACL 掩码的权限。掩码表示允许用户（属主除外）和组拥有的最大权限。
acl-entry-list	指定文件或目录中要为特定用户和组设置的一个或多个 ACL 项的列表。也可以设置目录的缺省 ACL 项。表 6–7 和表 6–8 显示了有效的 ACL 项。
filename ...	指定要对其设置 ACL 的一个或多个文件或目录。多个 *filename* 由空格分隔。

注意 – 如果该文件已存在 ACL，则 `-s` 选项将使用新的 ACL 替换整个 ACL。

有关更多信息，请参见 `setfacl`(1) 手册页。

2 检验是否已对文件设置了 ACL 项。

```
% getfacl filename
```

有关更多信息，请参见第 134 页中的 “如何检查文件是否具有 ACL”。

示例 6–7 设置文件的 ACL

在以下示例中，会在 `ch1.sgm` 文件中将文件属主权限设置为读写、将文件组权限设置为只读，并将其他用户权限设置为无。此外，还在文件中为用户 `anusha` 指定读写权限。将 ACL 掩码权限设置为读写，这意味着任何用户或组都没有执行权限。

```
% setfacl -s user::rw-,group::r--,other:---,mask:rw-,user:anusha:rw- ch1.sgm

% ls -l

total 124

-rw-r-----+  1 stacey  techpubs   34816 Nov 11 14:16 ch1.sgm

-rw-r--r--   1 stacey  techpubs   20167 Nov 11 14:16 ch2.sgm
```

```
-rw-r--r--   1 stacey  techpubs    8192 Nov 11 14:16 notes
% getfacl ch1.sgm
# file: ch1.sgm
# owner: stacey
# group: techpubs
user::rw-
user:anusha:rw-    #effective:rw-
group::r--         #effective:r--
mask:rw-
other:---
```

在以下示例中，会将文件属主权限设置为读写和执行，将文件组权限设置为只读，并将其他用户权限设置为无。此外，还会在 ch2.sgm 文件中将 ACL 掩码权限设置为读。最后，将为用户 anusha 指定读写权限。但是，由于 ACL 掩码的原因，anusha 的权限为只读。

```
% setfacl -s u::7,g::4,o:0,m:4,u:anusha:7 ch2.sgm
% getfacl ch2.sgm
# file: ch2.sgm
# owner: stacey
# group: techpubs
user::rwx
user:anusha:rwx         #effective:r--
group::r--              #effective:r--
mask:r--
other:---
```

▼ 如何复制 ACL

◗ **通过重定向 `getfacl` 输出，将文件的 ACL 复制到另一个文件。**

```
% getfacl filename1 | setfacl -f - filename2
```

filename1 指定将从其中复制 ACL 的文件。

filename2 指定要对其设置所复制的 ACL 的文件。

示例 6–8 复制 ACL

在以下示例中，会将 `ch2.sgm` 中的 ACL 复制到 `ch3.sgm`。

```
% getfacl ch2.sgm | setfacl -f - ch3.sgm
```

▼ 如何更改文件的 ACL 项

1 **使用 `setfacl` 命令修改文件的 ACL 项。**

```
% setfacl -m acl-entry-list filename ...
```

`-m` 修改现有的 ACL 项。

acl-entry-list 指定文件或目录中要修改的一个或多个 ACL 项的列表。也可以修改目录的缺省 ACL 项。表 6–7 和表 6–8 显示了有效的 ACL 项。

filename ... 指定一个或多个文件或目录，由空格分隔。

2 **检验是否已修改文件的 ACL 项。**

```
% getfacl filename
```

示例 6–9 修改文件的 ACL 项

在以下示例中，将用户 `anusha` 的权限修改为读写。

```
% setfacl -m user:anusha:6 ch3.sgm

% getfacl ch3.sgm

# file: ch3.sgm

# owner: stacey

# group: techpubs
```

```
user::rw-

user::anusha:rw-        #effective:r--

group::r-               #effective:r--

mask:r--

other:r-
```

在以下示例中，将组 `staff` 的缺省权限修改为对 `book` 目录的读取权限。此外，还将缺省 ACL 掩码权限修改为读写。

```
% setfacl -m default:group:staff:4,default:mask:6 book
```

▼ 如何删除文件的 ACL 项

1 删除文件的 ACL 项。

```
% setfacl -d acl-entry-list filename ...
```

`-d` 删除指定的 ACL 项。

acl-entry-list 指定文件或目录中要删除的 ACL 项（未指定权限）的列表。只能删除特定用户和组的 ACL 项和缺省 ACL 项。表 6–7 和表 6–8 显示了有效的 ACL 项。

filename ... 指定一个或多个文件或目录，由空格分隔。

或者，可以使用 `setfacl -s` 命令删除文件的所有 ACL 项，并使用所指定的新 ACL 项替换它们。

2 检验是否已删除文件的 ACL 项。

```
% getfacl filename
```

示例 6–10 删除文件的 ACL 项

在以下示例中，将从 `ch4.sgm` 文件中删除用户 `anusha`。

```
% setfacl -d user:anusha ch4.sgm
```

▼ 如何显示文件的 ACL 项

◗ **使用 `getfacl` 命令显示文件的 ACL 项。**

```
% getfacl [-a | -d] filename ...
```

`-a` 显示指定文件或目录的文件名、文件属主、文件组和 ACL 项。

`-d` 显示指定目录的文件名、文件属主、文件组和缺省 ACL 项（如果存在）。

filename ... 指定一个或多个文件或目录，由空格分隔。

如果在命令行中指定多个文件名，则会在每两个 ACL 项之间显示一个空白行。

示例 6–11 显示文件的 ACL 项

在以下示例中，将显示文件 `ch1.sgm` 的所有 ACL 项。用户和组项旁边的 `#effective:` 注释表示由 ACL 掩码修改后的权限。

```
% getfacl ch1.sgm


# file: ch1.sgm
# owner: stacey
# group: techpubs
user::rw-
user:anusha:r-        #effective:r--
group::rw-            #effective:rw-
mask:rw-
other:---
```

在以下示例中，将显示目录 `book` 的缺省 ACL 项。

```
% getfacl -d book


# file: book
# owner: stacey
# group: techpubs
user::rwx
user:anusha:r-x       #effective:r-x
```

```
group::rwx              #effective:rwx

mask:rwx

other:---

default:user::rw-

default:user:anusha:r--

default:group::rw-

default:mask:rw-

default:other:---
```

防止程序受到安全风险（任务列表）

以下任务列表说明查找系统中的危险可执行程序，以及禁止程序利用可执行栈的过程。

任务	说明	参考
使用特殊权限查找文件	查找设置了 `setuid` 位，但不归 `root` 用户拥有的文件。	第 140 页中的 “如何使用特殊文件权限查找文件”
防止可执行栈溢出	防止程序利用可执行栈。	第 142 页中的 “如何禁止程序使用可执行栈”
防止记录可执行栈消息	关闭记录可执行栈消息	示例 6–13

▼ 如何使用特殊文件权限查找文件

应对系统中未经授权在程序中使用 `setuid` 和 `setgid` 权限的情况进行监视。使用 `setuid` 和 `setgid` 权限，普通用户可以获取超级用户功能。可疑可执行文件为用户而不是 `root` 或 `bin` 授予拥有权。

1 承担主管理员角色，或成为超级用户。

主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2 使用 `find` 命令查找拥有 `setuid` 权限的文件

```
# find directory -user root -perm -4000 -exec ls -ldb {} \; >/tmp/filename
```

`find` *directory*	检查以指定的 *directory*（可以是 root (`/`)、`sys`、 `bin` 或 `mail`）开头的所有挂载路径。
`-user root`	仅显示由 `root` 拥有的文件。
`-perm -4000`	仅显示权限被设置为 `4000` 的文件。
`-exec ls -ldb`	以 `ls -ldb` 格式显示 `find` 命令的输出。
`>/tmp/`*filename*	包含 `find` 命令的结果的文件。

3 在 `/tmp/`*filename* 中显示结果。

```
# more /tmp/filename
```

有关 `setuid` 权限的背景信息，请参见第 121 页中的 “`setuid` 权限”。

示例 6–12 使用 `setuid` 权限查找文件

以下示例的输出显示，名为 `rar` 的用户创建了一份 `/usr/bin/sh` 的个人副本，并将权限设置为 `root` 的 `setuid`。因此，`/usr/rar/bin/sh` 程序将使用 `root` 权限运行。

通过将文件从 `/tmp` 目录中移出，可以保存此输出以供将来参考。

```
# find / -user root -perm -4000 -exec ls -ldb {} \; > /var/tmp/ckprm

# cat /var/tmp/ckprm

-r-sr-xr-x 1 root bin 38836 Aug 10 16:16 /usr/bin/at

-r-sr-xr-x 1 root bin 19812 Aug 10 16:16 /usr/bin/crontab

---s--x--x 1 root sys 46040 Aug 10 15:18 /usr/bin/ct

-r-sr-xr-x 1 root sys 12092 Aug 11 01:29 /usr/lib/mv_dir

-r-sr-sr-x 1 root bin 33208 Aug 10 15:55 /usr/lib/lpadmin

-r-sr-sr-x 1 root bin 38696 Aug 10 15:55 /usr/lib/lpsched

---s--x--- 1 root rar 45376 Aug 18 15:11 /usr/rar/bin/sh

-r-sr-xr-x 1 root bin 12524 Aug 11 01:27 /usr/bin/df

-rwsr-xr-x 1 root sys 21780 Aug 11 01:27 /usr/bin/newgrp

-r-sr-sr-x 1 root sys 23000 Aug 11 01:27 /usr/bin/passwd

-r-sr-xr-x 1 root sys 23824 Aug 11 01:27 /usr/bin/su
```

```
# mv /var/tmp/ckprm /export/sysreports/ckprm
```

▼ 如何禁止程序使用可执行栈

有关可执行栈的安全风险的说明，请参见第 127 页中的“防止可执行文件危及安全”。

1. **承担主管理员角色，或成为超级用户。**

   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2. **编辑 `/etc/system` 文件并添加以下行：**

   ```
   set noexec_user_stack=1
   ```

3. **重新引导系统。**

   ```
   # init 6
   ```

示例 6–13 禁止记录可执行栈消息

在此示例中，将禁止记录可执行栈消息，然后重新引导系统。

```
# cat /etc/system

set noexec_user_stack=1

set noexec_user_stack_log=0

# init 6
```

◆ ◆ ◆ 第 7 章

使用自动安全性增强工具（任务）

本章介绍了如何使用自动安全性增强工具 (Automated Security Enhancement Tool, ASET) 来监视或限制对系统文件和目录的访问。

以下是本章中的逐步说明列表：

- 第 143 页中的 “自动安全性增强工具 (Automated Security Enhancement Tool, ASET)”
- 第 159 页中的 “运行 ASET（任务列表）”
- 第 164 页中的 “解决 ASET 问题”

如需一种比 ASET 更为全面的工具，请使用 Solaris 安全工具包。Solaris 安全工具包提供了用于强化和最小化 Solaris 系统的框架。此工具包包括文件配置工具、报告工具和撤消功能。此工具包是免费的，可以从 Sun Web 站点 `http://wwws.sun.com/security/jass` 下载。此 Web 站点包含指向联机文档的链接。

Alex Noordergraaf 和 Glenn Brunette 合著的《Securing Systems with the Solaris Security Toolkit》（ISBN 0-13-141071-7，2003 年 6 月）中对此工具包进行了详细说明。本书是 Sun Microsystems Press 出版的 Sun BluePrints 系列的一部分。

自动安全性增强工具 (Automated Security Enhancement Tool, ASET)

Solaris 操作系统包括自动安全性增强工具 (Automated Security Enhancement Tool, ASET)。ASET 通过自动执行原本需要手动执行的任务来帮助监视和控制系统安全性。

ASET 安全软件包提供可用于控制和监视系统安全性的自动管理工具。您可以指定运行 ASET 时的安全级别。安全级别包括低、中和高。级别越高，ASET 的文件控制功能越强，从而可限制文件访问并提高系统安全性。

ASET 可以运行七项任务。每项任务都针对系统文件执行特定的检查和调整。ASET 任务可以加强对文件权限的控制，检查关键系统文件内容中的安全漏洞以及监视重要区域。ASET 还可以通过将防火墙系统的基本要求应用于网关系统来保护网络。请参见第 146 页中的 “防火墙设置”。

ASET 使用主文件进行配置。主文件、报告和其他 ASET 文件都位于 /usr/aset 目录中。可以更改这些文件以适应您站点的特定要求。

每项任务都会生成一个报告。此报告将记录检测到的安全漏洞以及任务对系统文件所做的任何更改。以最高安全级别运行时，ASET 将尝试修复所有系统安全漏洞。如果 ASET 无法更正某个潜在的安全问题，则会报告存在此问题。

您可以通过交互地使用 /usr/aset/aset 命令来启动 ASET 会话。或者，也可以通过向 crontab 文件中加入一项，将 ASET 设置为定期运行。

ASET 任务会占用大量磁盘空间，并且会干扰常规活动。要将对系统性能的影响降至最小，可以安排 ASET 在系统活动量最少时运行。例如，每 24 或 48 小时在午夜运行一次 ASET。

ASET 安全级别

可以将 ASET 设置为在三种安全级别之一下运行：低、中或高。级别越高，ASET 的文件控制功能就越强，从而可限制文件访问并提高系统安全性。这些功能范围很广，从在不限制用户访问文件的情况下监视系统安全性，到逐渐加强对访问权限的控制直到系统完全安全为止。

下表概括了这三种安全级别。

安全级别	说明
低	确保系统文件的属性设置为标准发行版的值。ASET 会执行几项检查，然后报告潜在的安全漏洞。在此级别，ASET 不执行任何操作，因此不会影响系统服务。
中	为大多数环境提供足够的安全控制。ASET 会修改系统文件和参数的一些设置。ASET 限制系统访问以降低来自安全攻击的风险。ASET 报告安全漏洞以及它对限制访问所做的任何修改。在此级别，ASET 不会影响系统服务。
高	实现一个高度安全的系统。ASET 会调整许多系统文件和参数设置，以最大限度地减少访问权限。大多数系统应用程序和命令将继续正常运行。但在此级别，安全方面的考虑优先于其他系统行为。

注 – ASET 不会通过更改文件的权限来降低文件的安全性，除非您降低安全级别。您也可以专门将系统恢复到运行 ASET 之前存在的设置。

ASET 任务列表

本节介绍 ASET 所执行的任务。您应该了解每项 ASET 任务。通过了解 ASET 的目标、ASET 执行的操作以及 ASET 影响的系统组件，可以有效地理解和使用其报告。

ASET 报告文件包含的消息尽可能详细地说明了每项 ASET 任务发现的所有问题。这些消息有助于诊断和更正这些问题。但是，要成功使用 ASET，您需要对系统管理和系统组件有大致的了解。如果您是初级管理员，可以参阅其他 Solaris 系统管理文档。您可以参阅相关的手册页来学习 ASET 管理。

`taskstat` 实用程序用于标识已完成的任务，以及仍在运行的任务。每项完成的任务都会生成一个报告文件。有关 `taskstat` 实用程序的完整说明，请参阅 `taskstat`(1M)。

系统文件权限调优

此任务会将系统文件的权限设置为指定的安全级别。此任务在安装系统时运行。如果您稍后决定更改先前建立的级别，请再次运行此任务。在低安全级别，权限会设置为适合于开放式信息共享环境的值。在中安全级别，会加强对权限的控制，以便为大多数环境提供足够高的安全性。在高安全级别，会控制权限以严格限制访问。

此任务对系统文件权限或参数设置所做的任何修改都会报告在 `tune.rpt` 文件中。有关 ASET 在设置权限时参考的文件的示例，请参见，请参见第 158 页中的 “调优文件示例” 。

系统文件检查

此任务会检查系统文件，并将每个文件与主文件中相应文件的说明进行比较。主文件是 ASET 首次运行此任务时创建的。主文件包含 `checklist` 针对指定安全级别执行的系统文件设置。

针对每种安全级别，将定义要检查其文件的目录的列表。可以使用缺省列表，也可以修改列表，以便针对每个级别指定不同的目录。

对于每个文件，会检查以下条件：

- 属主和组
- 权限位
- 大小以及校验和
- 链接数目
- 上次修改时间

ASET 发现的任何差异都会报告在 `cklist.rpt` 文件中。此文件包含将系统文件大小、权限及校验和的值与主文件进行比较的结果。

用户和组检查

此任务会检查用户帐户和组的一致性和完整性。此任务使用 `passwd` 和 `group` 文件中的定义。此任务检查本地口令文件、NIS 口令文件或 NIS+ 口令文件，并报告 NIS+ 的口令文件问题，但不会进行更正。此任务将检查以下违规：

- 重复的名称或 ID
- 格式不正确的项
- 缺少口令的帐户
- 无效的登录目录
- `nobody` 帐户

- 空的组口令
- NIS 服务器或 NIS+ 服务器上的 /etc/passwd 文件中的加号 (+)

差异会报告在 usrgrp.rpt 文件中。

系统配置文件检查

在执行此任务的过程中，ASET 会检查各种系统表，其中大多数系统表位于 /etc 目录中。这些文件包括：

- /etc/default/login
- /etc/hosts.equiv
- /etc/inetd.conf
- /etc/aliases
- /var/adm/utmpx
- /.rhosts
- /etc/vfstab
- /etc/dfs/dfstab
- /etc/ftpd/ftpusers

ASET 会对这些文件执行各种检查和修改，并在 sysconf.rpt 文件中报告问题。

环境变量检查

此任务会检查为超级用户和其他用户设置 PATH 和 UMASK 环境变量的方式，并检查 /.profile、/.login 和 /.cshrc 文件。

检查环境安全性的结果会报告在 env.rpt 文件中。

eeprom 检查

此任务会检查 eeprom 安全参数的值，以确保此参数设置在合适的安全级别。 可以将 eeprom 安全参数设置为 none、command 或 full。

ASET 不会更改此设置，但会在 eeprom.rpt 文件中报告其建议。

防火墙设置

此任务确保系统可以安全地用作网络中继。此任务通过将专用系统设置为防火墙，保护内部网络不受外部公共网络的干扰，第 50 页中的 “防火墙系统”对此进行了说明。防火墙系统可分隔两个网络。在这种情况下，每个网络都作为不可信对象访问另一个网络。防火墙设置任务将禁止 Internet 协议 (Internet Protocol, IP) 包的转发。防火墙还会对外部网络隐藏路由信息。

防火墙任务可以在所有安全级别运行，但是仅在最高级别执行操作。如果要在高安全级别运行ASET，但发现系统不需要防火墙保护，则可以取消防火墙任务。可以通过编辑 asetenv 文件删除此任务。

所做的任何更改都会报告在 firewall.rpt 文件中。

ASET 执行日志

无论 ASET 以交互方式运行还是在后台运行，它都会生成执行日志。缺省情况下，ASET 会生成标准输出形式的日志文件。执行日志可确认 ASET 是否在指定时间运行，并且还包含所有执行错误消息。`aset -n` 命令指示将日志通过电子邮件发送到指定的用户。有关 ASET 选项的完整列表，请参见 aset(1M) 手册页。

ASET 执行日志文件的示例

```
ASET running at security level low



Machine=example; Current time = 0325_08:00




aset: Using /usr/aset as working directory



Executing task list...

        firewall

        env

        sysconfig

        usrgrp

        tune

        cklist

        eeprom

All tasks executed. Some background tasks may still be running.



Run /usr/aset/util/taskstat to check their status:

     $/usr/aset/util/taskstat     aset_dir

Where aset_dir is ASET's operating directory, currently=/usr/aset
```

```
When the tasks complete, the reports can be found in:

     /usr/aset/reports/latest/*.rpt

You can view them by:

more /usr/aset/reports/latest/*.rpt
```

执行日志首先显示运行 ASET 的系统和时间。然后，执行日志会列出启动任务时的每项任务。

ASET 将针对这些任务中的每一项调用后台进程，这在第 144 页中的 “ASET 任务列表”中进行了说明。启动任务时，执行日志中会列出此任务。列出此任务并不意味着任务已完成。要检查后台任务的状态，请使用 `taskstat` 命令。

ASET 报告

由 ASET 任务生成的所有报告文件都存储在 `/usr/aset/reports` 目录下的子目录中。本节介绍 `/usr/aset/reports` 目录的结构，并提供管理报告文件的指南。

ASET 将报告文件放在子目录中，这些子目录的名称可反映报告生成时间和日期。通过此约定，可以按顺序跟踪记录，这些记录用于在系统状态随 ASET 执行的不同而变化的过程中记载系统状态。可以监视和比较这些报告以确定系统安全的可靠性。

下图显示了 `reports` 目录结构的示例。

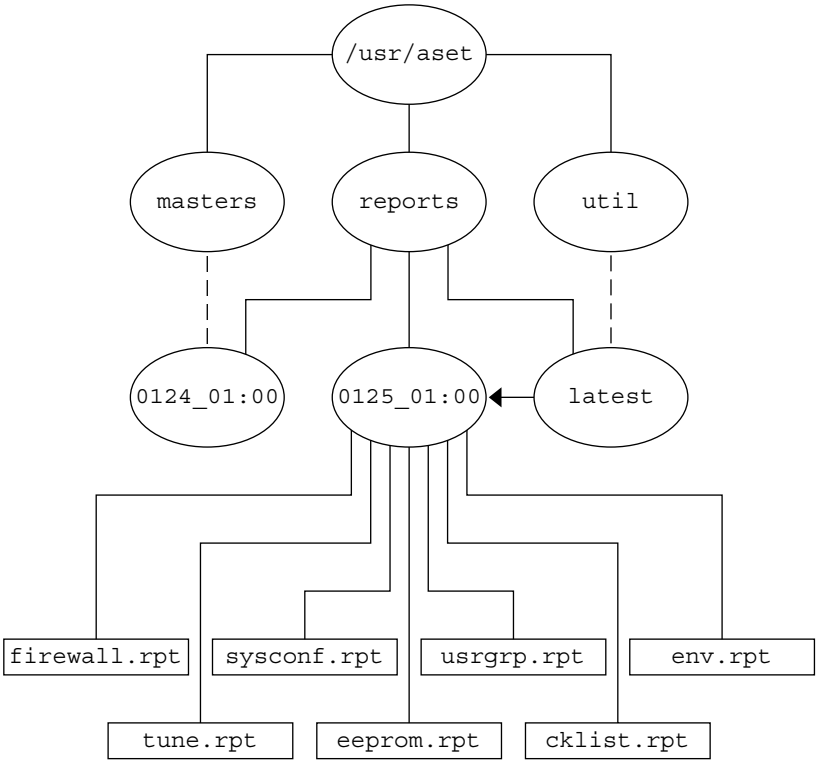


图 7–1 ASET `reports` 目录的结构

此示例说明了两个报告子目录。

- `0124_01:00`
- `0125_01:00`

子目录的名称指明了报告的生成日期和时间。每个报告子目录名称都具有以下格式：

monthdate_hour:minute

month、*date*、*hour* 和 *minute* 都是两位数字。例如，`0125_01:00` 表示 1 月 25 日凌晨 1 点。

这两个报告子目录都包含执行一次 ASET 所生成的报告集合。

`latest` 目录是始终指向包含最新报告的子目录的符号链接。因此，要查看 ASET 生成的最新报告，可以转至 `/usr/aset/reports/latest` 目录。此目录中包含 ASET 在最近一次执行期间所执行的每项任务的报告文件。

ASET 报告文件的格式

每个报告文件都以生成此报告的任务命名。下表列出了各项任务及其报告。

表 7–1 ASET 任务和生成的报告

任务	报告
系统文件权限调优 (tune)	tune.rpt
系统文件检查 (cklist)	cklist.rpt
用户和组检查 (usrgrp)	usrgrp.rpt
系统配置文件检查 (sysconf)	sysconf.rpt
环境变量检查 (env)	env.rpt
eeprom 检查 (eeprom)	eeprom.rpt
防火墙设置 (firewall)	firewall.rpt

每个报告文件中，消息括在开始标题行和结束标题行中。有时，任务会提前结束。例如，意外删除或损坏 ASET 的组件时，任务便会提前结束。在这类情况下，报告文件通常会在结尾附近包含一条消息，指明提前终止的原因。

以下是一个报告文件样例 usrgrp.rpt。

```
*** Begin User and Group Checking ***



Checking /etc/passwd ...

Warning! Password file, line 10, no passwd

:sync::1:1::/:/bin/sync

..end user check; starting group check ...

Checking /etc/group...

*** End User And group Checking ***
```

检查 ASET 报告文件

初次运行或重新配置 ASET 之后，应该严密地检查报告文件。重新配置包括修改 asetenv 文件或 masters 子目录中的主文件，或者更改运行 ASET 的安全级别。

报告会记录在重新配置 ASET 时引起的所有错误。通过严密地查看报告，可以在出现问题时做出反应并解决问题。

比较 ASET 报告文件

在对报告文件监视一段时间（其间没有进行配置更改或系统更新）之后，您可能会发现报告的内容开始趋于稳定。如果报告包含的意外信息很少，可使用 diff 实用程序来比较这些报告。

ASET 主文件

ASET 的主文件 tune.high、tune.low 、tune.med 和 uid_aliases 位于 /usr/aset/masters 目录中。ASET 使用这些主文件来定义安全级别。有关更多详细信息，请参见 asetmasters(4) 手册页。

调优文件

tune.low、tune.med 和 tune.high 主文件用于定义可用的 ASET 安全级别。这些文件可指定系统文件在每个级别的属性，用于比较和参考。

uid_aliases 文件

uid_aliases 文件包含共享同一用户 ID (user ID, UID) 的多个用户帐户的列表。通常，由于这种做法降低了可说明性，因此 ASET 会针对此类多个用户帐户发出警告。您可以通过在 uid_aliases 文件中列出例外情况来允许此规则存在例外。如果 uid_aliases 文件中指定了具有重复 UID 的项，则 ASET 不会在 passwd 文件中报告这些项。

请避免使多个用户帐户共享同一 UID。您应该考虑使用其他方法来实现目标。例如，如果要让多个用户共享一个权限集，则可以创建一个组帐户 。您还可以创建角色。仅当其他方法都无法实现目标时，才应使用共享 UID 这一方法。

可以使用 UID_ALIASES 环境变量来指定备用的别名文件。缺省文件为 /usr/aset/masters/uid_aliases。

核对表文件

系统文件检查所使用的主文件在首次执行 ASET 时生成，也可在更改安全级别之后运行 ASET 时生成。

以下环境变量定义了此任务检查的文件：

- CKLISTPATH_LOW
- CKLISTPATH_MED
- CKLISTPATH_HIGH

ASET 环境文件 (asetenv)

环境文件 asetenv 包含影响 ASET 任务的环境变量的列表。可以更改其中一些变量来修改 ASET 操作。有关 asetenv 文件的详细信息，请参见 asetenv(4)。

配置 ASET

本节介绍了如何配置 ASET，还介绍了 ASET 的运行环境。

ASET 所需的管理和配置操作最少。在大多数情况下，可以使用缺省值运行 ASET。但是，也可以微调某些影响 ASET 操作和行为的参数，以便最大程度发挥此工具的优点。更改缺省值之前，应了解 ASET 如何运行以及它如何影响系统组件。

ASET 依靠四个配置文件来控制其任务的行为：

- `/usr/aset/asetenv`
- `/usr/aset/masters/tune.low`
- `/usr/aset/masters/tune.med`
- `/usr/aset/masters/tune.high`

修改环境文件 (asetenv)

`/usr/aset/asetenv` 文件包含两个主要部分：

- 用户可配置的环境变量部分
- 内部环境变量部分

您可以更改用户可配置的参数部分。但是，内部环境变量部分中的设置仅供内部使用，不应修改这些设置。

您可以编辑用户可配置部分中的各项以执行以下操作：

- 选择要运行的任务
- 指定系统文件检查任务的目录
- 安排 ASET 执行
- 指定 UID 别名文件
- 将检查扩展到 NIS+ 表

选择要运行的任务： TASKS

ASET 执行的每项任务都会监视系统安全的特定方面。在大多数系统环境中，必须执行所有任务以便在各方面都可保证安全性。但是，您可能会决定删除一项或多项任务。

例如，防火墙任务可以在所有安全级别下运行，但是仅在高安全级别下执行操作。您可能希望在高安全级别运行 ASET，但是不需要防火墙保护。

可以将 ASET 设置为在没有防火墙功能的情况下在高安全级别运行。为此，可编辑 `asetenv` 文件中的环境变量的 `TASKS` 列表。缺省情况下，`TASKS` 列表包含所有的 ASET 任务。要删除某项任务，请从此文件中删除与此任务相关的环境变量。在这种情况下，可从列表中删除 `firewall` 环境变量。下次运行 ASET 时，便不会执行已排除的任务。

以下示例显示了包含所有 ASET 任务的 `TASKS` 列表。

```
TASKS="env sysconfig usrgrp tune cklist eeprom firewall"
```

指定系统文件检查任务的目录： CKLISTPATH

系统文件检查会检查选定系统目录中的文件属性。可以使用以下环境变量来定义要检查的目录。

CKLISTPATH_LOW 变量定义要在低安全级别检查的目录。CKLISTPATH_MED 和 CKLISTPATH_HIGH 环境变量可分别在中安全级别和高安全级别实现类似的功能。

环境变量在较低安全级别定义的目录列表应该是在下一个较高级别定义的目录列表的子集。例如，为 CKLISTPATH_LOW 指定的所有目录应该包括在 CKLISTPATH_MED 中。同样，为 CKLISTPATH_MED 指定的所有目录应该包括在 CKLISTPATH_HIGH 中。

针对这些目录执行的检查并不是递归的。ASET 仅检查在环境变量中显式列出的那些目录，而不检查其子目录。

可以编辑这些环境变量定义，以添加或删除需要 ASET 检查的目录。请注意，这些核对表仅适用于通常不会每日更改的系统文件。例如，用户的起始目录通常动态地频繁更新，因此不适合选择用作核对表的目录。

安排 ASET 执行： PERIODIC_SCHEDULE

您可以交互地启动 ASET，也可以使用 -p 选项来请求 ASET 任务在预定时间运行。您可以定期在系统需求较少时运行 ASET。例如，ASET 可参阅 PERIODIC_SCHEDULE 来确定执行 ASET 任务的频率以及运行这些任务的时间。有关设置 ASET 使其定期运行的详细说明，请参见第 161 页中的 “如何定期运行 ASET” 。

PERIODIC_SCHEDULE 的格式遵循 crontab 项的格式。有关完整信息，请参见 crontab(1)。

指定别名文件： UID_ALIASES

UID_ALIASES 变量可指定用于列出共享 UID 的别名文件。缺省文件为 /usr/aset/masters/uid_aliases。

将检查扩展到 NIS+ 表： YPCHECK

YPCHECK 环境变量可指定 ASET 是否也应该检查系统配置文件表。YPCHECK 为布尔变量。 只能将 YPCHECK 指定为 true 或 false。缺省值为 false，它将禁用 NIS+ 表检查。

要了解此环境变量如何运行，请考虑其对 passwd 文件的影响。设置为 false 时，ASET 会检查本地 passwd 文件。如果设置 true，此任务还将检查系统域的 NIS+ passwd 表。

注 – 尽管 ASET 会自动修复本地文件，但是 ASET 仅报告 NIS+ 表中的潜在问题，而不会更改这些表。

修改调优文件

ASET 使用三个主调优文件 tune.low、tune.med 和 tune.high 来放松或加强对关键系统文件的访问。这些主文件位于 /usr/aset/masters 目录中。可以修改这些文件以适合您的环境。有关示例，请参见第 158 页中的 “调优文件示例” 。

tune.low 文件可将权限设置为适合于缺省系统设置的值。tune.med 文件可进一步限制这些权限。tune.med 文件还包括 tune.low 中没有的项。tune.high 文件可更进一步限制这些权限。

注 – 可以通过添加或删除文件项来修改调优文件中的设置。无法有效地将权限设置为比当前设置限制少的值。除非将系统安全降至更低的级别，否则 ASET 任务不会放松权限。

恢复 ASET 修改的系统文件

首次执行 ASET 时，ASET 会保存并归档初始系统文件。aset.restore 实用程序可重新恢复这些文件。如果当前安排 ASET 定期执行，则此实用程序还会取消对 ASET 的安排。aset.restore 命令位于 ASET 操作目录 /usr/aset 中。

运行 aset.restore 命令时，对系统文件所做的更改会丢失。

应在以下情况下使用 aset.restore 命令：

- 要删除 ASET 更改并恢复初始系统。

  如果先前已将 aset 命令添加到根目录的 crontab，则要永久取消激活 ASET 时，可以从 cron 调度中删除 ASET。有关如何使用 cron 删除自动执行的任务的说明，请参见第 162 页中的 “如何停止定期运行 ASET”。
- 试用 ASET 一段时间后，希望恢复初始系统状态。
- 某些主要系统功能没有正常运行，并且您怀疑问题是由 ASET 引起时。

使用 NFS 系统进行网络操作

一般情况下，ASET 在单机模式下使用，即使在网络所包含的系统中也是如此。作为独立系统的系统管理员，您要负责系统的安全。因此，您将负责运行和管理 ASET 以保护系统。

您还可以在 NFS 分布式环境中使用 ASET。作为网络管理员，您要负责为所有客户机安装、运行和管理各种管理任务。为了便于在多个客户机系统中进行 ASET 管理，可以对全局应用于所有客户机的配置进行更改。通过全局应用更改，无需登录到每个系统即可重复配置更改。

决定如何在联网系统中设置 ASET 时，应该考虑希望谁来控制安全性。您可能希望用户控制其各自系统的部分安全性，还可能希望集中负责安全控制。

为每种安全级别提供全局配置

要设置多个网络配置时可能会出现这种情况。例如，您可能要为那些指定为低安全级别的客户机设置一种配置，为中级别的客户机设置一种配置，为高级别的客户机设置另一种配置。

如果需要为每种安全级别创建单独的 ASET 网络配置，可以在服务器上创建三种 ASET 配置，为每种级别创建一种配置。可以将每种配置导出到具有相应安全级别的客户机中。对于所有三种配置都相同的某些 ASET 组件可使用链接来共享。

收集 ASET 报告

您不仅可以在服务器上集中 ASET 组件，还可以在服务器上设置中央目录以收集所有的 ASET 报告。具有或不具有超级用户权限的客户机均可访问此服务器。有关设置收集机制的说明，请参见第 162 页中的 “如何在服务器上收集 ASET 报告” 。

通过在服务器上设置报告收集，可以从一个位置查看所有客户机的报告。无论客户机是否具有超级用户权限，都可以使用此方法。或者，如果要用户监视自己的 ASET 报告，可以将报告目录保留在本地系统上。

ASET 环境变量

以下是 ASET 环境变量和这些变量指定的值的列表。

`ASETDIR`	指定 ASET 工作目录
`ASETSECLEVEL`	指定安全级别
`PERIODIC_SCHEDULE`	指定定期安排
`TASKS`	指定要运行的 ASET 任务
`UID_ALIASES`	指定别名文件
`YPCHECK`	确定是否将检查扩展到 NIS 映射和 NIS+ 表
`CKLISTPATH_LOW`	低安全级别的目录列表
`CKLISTPATH_MED`	中安全级别的目录列表
`CKLISTPATH_HIGH`	高安全级别的目录列表

以下各节中列出的环境变量位于 `/usr/aset/asetenv` 文件中。`ASETDIR` 和 `ASETSECLEVEL` 变量为可选变量。这些变量只能使用 `/usr/aset/aset` 命令通过 shell 来设置。其他环境变量可以通过编辑此文件来设置。

ASETDIR 环境变量

`ASETDIR` 用于指定 ASET 工作目录。

从 C shell 中，键入：

```
% setenv ASETDIR pathname
```

从 Bourne shell 或 Korn shell 中，键入：

```
$ ASETDIR=pathname
```

```
$ export ASETDIR
```

将 *pathname* 设置为 ASET 工作目录的全路径名。

ASETSECLEVEL 环境变量

ASETSECLEVEL 变量指定执行 ASET 任务的安全级别。

从 C shell 中，键入：

```
% setenv ASETSECLEVEL level
```

从 Bourne shell 或 Korn shell 中，键入：

```
$ ASETSECLEVEL=level
```

```
$ export ASETSECLEVEL
```

在这些命令中，可以将 *level* 设置为以下各项之一：

`low` 低安全级别

`med` 中安全级别

`high` 高安全级别

PERIODIC_SCHEDULE 环境变量

PERIODIC_SCHEDULE 的值与 `crontab` 文件遵循相同的格式。将变量值指定为用双引号引起的五个字段的字符串，其中各个字段用一个空格分隔：

"*minutes hours day-of-month month day-of-week*"

minutes hours 以（整小时数之后经过的）分钟数 (0-59) 和小时数 (0-23) 指定开始时间。

day-of-month 指定应该运行 ASET 的月日期，值的范围为 1-31。

month 指定应该运行 ASET 的月份，值的范围为 1-12。

day-of-week 指定应该运行 ASET 的周日期，值的范围为 0-6。星期天为 0 日。

为 ASET 创建定期安排时，可应用以下规则：

- 可以为任何字段指定值的列表，各个值用逗号分隔。
- 可以将值指定为数字，也可以将值指定为一个范围。范围是由连字符连接的一对数字。范围表明了 ASET 任务应该在此范围所包括的所有时间执行。
- 可以将星号 (*) 指定为任何字段的值。星号可概括性地指定字段所有可能的值。

如果使用 PERIODIC_SCHEDULE 变量的缺省项，则 ASET 每天在午夜 12:00 执行：

```
PERIODIC_SCHEDULE=”0 0 * * *”
```

TASKS 环境变量

TASKS 变量用于列出 ASET 所执行的任务。缺省设置为列出所有七项任务：

```
TASKS=”env sysconfig usrgrp tune cklist eeprom firewall”
```

UID_ALIASES 环境变量

UID_ALIASES 变量用于指定别名文件。如果存在，则 ASET 会参阅此文件以获取允许的多个别名的列表。别名格式为 UID_ALIASES=*pathname*，其中 *pathname* 为别名文件的全路径名。

缺省设置如下：

```
UID_ALIASES=${ASETDIR}/masters/uid_aliases
```

YPCHECK 环境变量

YPCHECK 变量用于将检查系统表的任务扩展到包括 NIS 或 NIS+ 表。YPCHECK 变量为布尔变量，可以设置为 true 或 false。

缺省设置为 false，表示将检查限定在本地系统表：

```
YPCHECK=false
```

CKLISTPATH_*level* 环境变量

三个核对表路径变量用于列出系统文件检查任务要检查的目录。缺省情况下，将设置以下变量定义。这些定义说明了不同级别变量之间的关系：

```
CKLISTPATH_LOW=${ASETDIR}/tasks:${ASETDIR}/util:${ASETDIR}/masters:/etc

CKLISTPATH_MED=${CKLISTPATH_LOW}:/usr/bin:/usr/ucb

CKLISTPATH_HIGH=${CKLISTPATH_MED}:/usr/lib:/sbin:/usr/sbin:/usr/ucblib
```

核对表路径环境变量的值类似于 shell 路径变量的值。与 shell 路径变量相同，核对表路径环境变量也是目录名称的列表。这些目录名称用冒号分隔。您可以使用等号 (=) 将变量名称与其值相连。

ASET 文件示例

本节介绍一些 ASET 文件的示例，包括调优文件和别名文件。

调优文件示例

ASET 可维护三个调优文件。调优文件中的每项都占用一行。每项中的字段按以下顺序排列：

pathname mode owner group type

pathname	文件的全路径名
mode	用于表示权限设置的五位数
owner	文件的属主
group	文件的组属主
type	文件的类型

编辑调优文件时，可应用以下规则：

- 可在路径名中使用常规的 shell 通配符（例如星号 (*) 和问号 (?)），以便引用多个路径。有关更多信息，请参见 sh(1)。
- *mode* 表示限制最少的值。如果当前设置已经比指定值的限制多，则 ASET 不会放松权限设置。例如，如果指定的值为 00777，则权限保持不变，因为 00777 始终比当前设置的限制少。

  此过程说明 ASET 如何处理模式设置。如果降低安全级别或删除 ASET，则此过程有所不同。从先前执行时的级别降低安全级别时，或是要将系统文件恢复到首次执行 ASET 之前的状态时，ASET 可识别正在执行的操作并降低保护级别。
- 必须使用 *owner* 和 *group* 的名称，而不是使用数字 ID。
- 可以使用问号 (?) 来代替 *owner*、*group* 和 *type*，以防止 ASET 更改这些参数的现有值。
- *type* 可以是 symlink、目录或文件。symlink 是符号链接。
- 较高安全级别的调优文件可将文件权限重置为至少与较低级别的文件权限具有相同限制。另外，在较高的安全级别，还会向列表中添加其他文件。
- 一个文件可以与多个调优文件项相匹配。例如，etc/passwd 与 etc/pass* 和 /etc/* 项相匹配。
- 如果两个项具有不同权限，则文件权限将设置为限制性最高的值。在以下示例中，/etc/passwd 文件的权限设置为 00755，这是 00755 和 00770 中限制性较高的值。

```
/etc/pass* 00755 ? ? file

/etc/* 00770 ? ? file
```

- 如果两个项指定的 *owner* 或 *group* 不同，则后一项优先级更高。在以下示例中，`/usr/sbin/chroot` 的属主设置为 `root`。

```
/usr/sbin/chroot 00555 bin bin file

/usr/sbin/chroot 00555 root bin file
```

别名文件示例

别名文件包含共享同一用户 ID 的别名列表。

每项都具有如下格式：

uid=alias1 =alias2=alias3=...

uid 共享的 UID。

aliasn 共享一个 UID 的用户帐户。

例如，以下项列出了 UID `0`。`sysadm` 和 `root` 帐户共享此 UID：

```
0=root=sysadm
```

运行 ASET（任务列表）

任务	说明	参考
从命令行运行 ASET	以指定的 ASET 级别保护系统。查看执行日志可了解更改。	第 160 页中的 “如何交互运行 ASET”
按固定间隔以批处理模式运行 ASET	设置 cron（时钟守护进程）作业以确保 ASET 保护系统。	第 161 页中的 “如何定期运行 ASET”
停止以批处理模式运行 ASET	删除 ASET cron 作业。	第 162 页中的 “如何停止定期运行 ASET”
将 ASET 报告存储在服务器上	收集来自客户机的 ASET 报告以便集中进行监视。	第 162 页中的 “如何在服务器上收集 ASET 报告”

要设置 ASET 中的变量，请参见第 155 页中的 “ASET 环境变量”。要配置 ASET，请参见第 152 页中的 “配置 ASET”。

▼ 如何交互运行 ASET

1 **成为超级用户或承担等效角色。**

角色包含授权和具有一定权限的命令。有关角色的更多信息，请参见第 186 页中的 “配置 RBAC（任务列表）”。

2 **使用 `aset` 命令交互运行 ASET。**

```
# /usr/aset/aset -l level -d pathname
```

level 指定安全级别。有效值为 `low`、`medium` 或 `high`。缺省设置为 `low`。有关安全级别的详细信息，请参见第 144 页中的 “ASET 安全级别”。

pathname 指定 ASET 的工作目录。缺省设置为 `/usr/aset`。

3 **查看屏幕上显示的 ASET 执行日志，验证 ASET 是否正在运行。**

执行日志消息可确定正在运行的任务。

示例 7–1 交互运行 ASET

在以下示例中，ASET 在低安全级别运行，并且使用缺省工作目录。

```
# /usr/aset/aset -l low

======= ASET Execution Log =======


ASET running at security level low


Machine = jupiter; Current time = 0111_09:26


aset: Using /usr/aset as working directory


Executing task list ...

    firewall

    env

    sysconf
```

```
    usrgrp

    tune

    cklist

    eeprom


All tasks executed. Some background tasks may still be running.


Run /usr/aset/util/taskstat to check their status:

 /usr/aset/util/taskstat [aset_dir]


where aset_dir is ASET's operating

directory,currently=/usr/aset.


When the tasks complete, the reports can be found in:

 /usr/aset/reports/latest/*.rpt


You can view them by:

 more /usr/aset/reports/latest/*.rpt
```

▼ 如何定期运行 ASET

1 **成为超级用户或承担等效角色。**

角色包含授权和具有一定权限的命令。有关角色的更多信息，请参见第 186 页中的 “配置 RBAC（任务列表）” 。

2 **如有必要，设置希望 ASET 定期运行的时间。**

应该在系统需求较少时运行 ASET。/usr/aset/asetenv 文件中的 PERIODIC_SCHEDULE 环境变量用于设置 ASET 的定期运行时间。缺省情况下，时间设置为每天午夜。

如果希望设置其他时间，请编辑 /usr/aset/asetenv 文件中的 PERIODIC_SCHEDULE 变量。有关设置 PERIODIC_SCHEDULE 变量的详细信息，请参见第 156 页中的 “PERIODIC_SCHEDULE 环境变量”。

3 **使用 aset 命令向 crontab 文件中添加项。**

```
# /usr/aset/aset -p
```

-p 选项可在 crontab 文件中插入一行，使 ASET 在 /usr/aset/asetenv 文件中的 PERIODIC_SCHEDULE 环境变量确定的时间开始运行。

4 **显示 crontab 项以检验安排 ASET 运行的时间。**

```
# crontab -l root
```

▼ 如何停止定期运行 ASET

1 **承担主管理员角色，或成为超级用户。**

主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2 **编辑 crontab 文件。**

```
# crontab -e root
```

3 **删除 ASET 项。**

4 **保存更改并退出。**

5 **显示 crontab 项，检验 ASET 项是否已删除。**

```
# crontab -l root
```

▼ 如何在服务器上收集 ASET 报告

1 **承担主管理员角色，或成为超级用户。**

主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2 **在服务器上设置目录：**

a. **转到** `/usr/aset` **目录。**

```
mars# cd /usr/aset
```

b. **创建** *rptdir* **目录。**

```
mars# mkdir rptdir
```

c. **转到** *rptdir* **目录并创建** *client_rpt* **目录。**

此步骤可为客户机创建 *client_rpt* 子目录。对于每台需要收集报告的客户机，请重复此步骤。

```
mars# cd rptdir

mars# mkdir client_rpt
```

在以下示例中，创建了目录 `all_reports` 以及子目录 `pluto_rpt` 和 `neptune_rpt`。

```
mars# cd /usr/aset

mars# mkdir all_reports

mars# cd all_reports

mars# mkdir pluto_rpt

mars# mkdir neptune_rpt
```

3 **将** *client_rpt* **目录添加到** `/etc/dfs/dfstab` **文件中。**

这些目录应可选择进行读取还是写入。

例如，可使用读写权限共享 `dfstab` 文件中的以下各项。

```
share -F nfs -o rw=pluto /usr/aset/all_reports/pluto_rpt

share -F nfs -o rw=neptune /usr/aset/all_reports/neptune_rpt
```

4 **使** `dfstab` **文件中的资源可供客户机使用。**

```
# shareall
```

5 **在每台客户机上，在挂载点** `/usr/aset/masters/reports` **上挂载服务器中的客户机子目录。**

```
# mount server:/usr/aset/client_rpt /usr/aset/masters/reports
```

6 **编辑 /etc/vfstab 文件以在系统引导时自动挂载目录。**

neptune 上的 /etc/vfstab 中的以下样例项列出了要从 mars、/usr/aset/all_reports/neptune_rpt 以及 neptune 上的挂载点 /usr/aset/reports 挂载的目录。系统引导时，vfstab 中列出的目录会自动挂载。

```
mars:/usr/aset/all_reports/neptune.rpt /usr/aset/reports nfs - yes hard
```

解决 ASET 问题

本节介绍 ASET 生成的错误消息。

ASET 错误消息

```
ASET failed: no mail program found.
```

原因: 系统指示 ASET 将执行日志发送给用户，但是无法找到邮件程序。

解决方法: 安装邮件程序。

```
Usage: aset [-n user[@host]]]] in /bin/mail or /usr/ucb/mail.
```

```
Cannot decide current and previous security levels.
```

原因: ASET 无法确定当前调用和先前调用的安全级别。

解决方法: 确保通过命令行选项或 ASETSECLEVEL 环境变量设置了当前安全级别。此外，还应确保 ASETDIR/archives/asetseclevel.arch 的最后一行正确反映先前的安全级别。如果未设置这些值，或者这些值不正确，请输入正确的值。

```
ASET working directory undefined.
```

```
To specify, set ASETDIR environment variable or use command line option -d.
```

```
ASET startup unsuccessful.
```

原因: ASET 工作目录未定义，或者定义错误。此工作目录为操作目录。

解决方法: 使用 ASETDIR 环境变量或 -d 命令行选项来更正此错误，然后重新启动 ASET。

```
ASET working directory $ASETDIR missing.
```

```
ASET startup unsuccessful.
```

原因: ASET 工作目录未定义，或者定义错误。此工作目录为操作目录。此问题可能是由于 ASETDIR 变量引用了不存在的目录而引起，或是 -d 命令行选项可能引用了不存在的目录。

解决方法: 确保正确引用了正确的目录，即包含 ASET 目录分层结构的目录。

`Cannot expand $ASETDIR to full pathname.`
原因: ASET 无法将由 `ASETDIR` 变量或 `-d` 命令行选项提供的目录名称扩展为全路径名。

解决方法: 确保目录名称正确。确保目录引用了用户可以访问的现有目录。

`aset: invalid/undefined security level.`

`To specify, set ASETSECLEVEL environment variable or use command line option -l, with argument= low/med/high.`
原因: 安全级别未定义或者定义错误。仅有值 `low`、`med` 或 `high` 可以接受。

解决方法: 使用 `ASETSECLEVEL` 变量或 `-l` 命令行选项来指定这三个值中的其中一个。

`ASET environment file asetenv not found in $ASETDIR.`

`ASET startup unsuccessful.`
原因: ASET 在其工作目录中无法找到 `asetenv` 文件。

解决方法: 确保 ASET 的工作目录中存在 `asetenv` 文件。有关此文件的详细信息，请参见 `asetenv`(4) 手册页。

`filename doesn’t exist or is not readable.`
原因: 通过 *filename* 引用的文件不存在或无法读取。使用 `-u` 选项时可能会出现此问题。通过此选项，可以指定一个包含要检查的用户列表的文件。

解决方法: 确保 `-u` 选项的参数存在并可读取。

`ASET task list TASKLIST undefined.`
原因: 未定义应在 `asetenv` 文件中定义的 ASET 任务列表。此消息表示 `asetenv` 文件错误。

解决方法: 检查 `asetenv` 文件。确保在 `User Configurable` 部分中定义了任务列表。另外，还要检查此文件的其他部分以确保文件的完整性。有关有效的 `asetenv` 文件的内容，请参见 `asetenv`(4) 手册页。

`ASET task list $TASKLIST missing.`

`ASET startup unsuccessful.`
原因: 未定义应在 `asetenv` 文件中定义的 ASET 任务列表。此消息表示 `asetenv` 文件错误。

解决方法: 检查 `asetenv` 文件。确保在 `User Configurable` 部分中定义了任务列表。另外，还要检查此文件的其他部分以确保文件的完整性。有关有效的 `asetenv` 文件的内容，请参见 `asetenv`(4) 手册页。

`Schedule undefined for periodic invocation.`

`No tasks executed or scheduled. Check asetenv file.`
原因: 使用 `-p` 选项请求了 ASET 调度，但未在 `asetenv` 文件中定义环境变量 `PERIODIC_SCHEDULE`。

解决方法: 检查 asetenv 文件的 User Configurable 部分以确保定义了此变量。确保此变量的格式正确。

```
Warning! Duplicate ASET execution scheduled.
```

```
Check crontab file.
```

原因: 安排 ASET 运行多次。换句话说，在一个 ASET 调度仍有效时请求了另一个调度。如果确实需要多个调度，则此消息不一定就表示错误。在这种情况下，此消息仅用作警告。如果需要多个调度，则应该使用正确的调度格式以及 crontab 命令。有关更多信息，请参见 crontab(1) 手册页。

解决方法: 通过 crontab 命令检验正确的调度是否有效。确保 ASET 不存在不必要的 crontab 项。

第 3 部分

角色、权限配置文件和权限

本节介绍了基于角色的访问控制 (role-based access control, RBAC) 和进程权利管理。RBAC 组件包括角色、权限配置文件和授权。进程权利管理通过权限实现。在 RBAC 中使用权限可以提供一种比通过超级用户角色管理系统更为安全的管理替代方法。

◆ ◆ ◆ 第 8 章

使用角色和权限（概述）

Solaris 基于角色的访问控制 (role-based access control, RBAC) 和权限提供了更为安全的超级用户替代项。本章概述了有关 RBAC 以及相应权限的信息。

以下是本章中概述信息的列表：

- 第 169 页中的 “基于角色的访问控制（概述）”
- 第 177 页中的 “权限（概述）”

基于角色的访问控制（概述）

基于角色的访问控制 (Role-based access control, RBAC) 是一种安全功能，用于控制用户访问那些通常仅限于超级用户访问的任务。通过对进程和用户应用安全属性，RBAC 可以在多个管理员之间划分超级用户的功能。进程权利管理通过**权限**实现。用户权利管理通过 RBAC 实现。

- 有关进程权利管理的介绍，请参见第 177 页中的 “权限（概述）” 。
- 有关 RBAC 任务的信息，请参见第 9 章。
- 有关参考信息，请参见第 10 章。

RBAC：超级用户模型的替代项

在传统的 UNIX 系统中，`root` 用户（也称为超级用户）可执行所有功能。以 `root` 身份运行的程序或 `setuid` 程序可执行所有功能。`root` 用户可以读取和写入任何文件，运行所有程序，以及向任何进程发送中止信号。实际上，任何可成为超级用户的用户都能够修改站点的防火墙，更改审计跟踪，读取机密记录以及关闭整个网络。被非法修改的 `setuid` 程序可以在系统上执行任何操作。

基于角色的访问控制 (Role-based access control, RBAC) 提供了更为安全的全有或全无型超级用户模型替代项。使用 RBAC，可以在划分更精细的级别上强制执行安全策略。RBAC 使用**最低权限**的安全原则。最低权限表示用户仅具有执行某项作业所需的权限。普通用户具有

足够权限来使用其应用程序、检查其作业状态、列显文件、创建新文件等。超出普通用户功能以外的功能将分为多个权限配置文件。预期要执行需要某些超级用户功能的作业的用户将承担拥有相应权限配置文件的角色。

RBAC 会将超级用户功能收集到**权限配置文件**中。这些权限配置文件将指定给称为**角色**的特殊用户帐户。然后，用户可以承担执行需要某些超级用户功能的作业的角色。预定义的权限配置文件是 Solaris 软件附带的。您可以创建角色并指定相应的配置文件。

权限配置文件可以提供广泛的功能。例如，主管理员权限配置文件等效于超级用户。权限配置文件还可以从狭义范围进行定义。例如，Cron 管理权限配置文件可管理 `at` 和 `cron` 作业。创建角色时，可以决定是创建具有广泛功能的角色，还是创建具有有限功能的角色，抑或是同时创建这两种角色。

在 RBAC 模型中，超级用户可创建一种或多种角色。这些角色基于权限配置文件。然后，超级用户会将这些角色指定给可以放心委任执行角色任务的用户。这些用户使用其用户名进行登录。登录之后，用户将承担可以运行有限管理命令和图形用户界面 (graphical user interface, GUI) 工具的角色。

由于可以灵活地设置角色，因此可启用各种安全策略。尽管 Solaris 操作系统 (Solaris OS) 没有附带任何角色，但是您可以轻松地配置三种建议的角色。这些角色基于以下同名的权限配置文件：

- **主管理员** – 等效于 `root` 用户或超级用户的功能强大的角色。
- **系统管理员** – 用于执行与安全性无关的管理任务的角色，其功能相对较弱。此角色可以管理文件系统、邮件以及软件安装，但是不能设置口令。
- **操作员** – 用于执行备份和打印机管理等操作的初级管理员角色。

并不一定要实现这三种角色。角色是一种实现组织安全性需要的功能。可以为各个领域（如安全性、联网或防火墙管理）中具有特殊目的的管理员设置角色。另一种策略是创建一种功能强大的管理员角色和一种高级用户角色。高级用户角色将用于那些允许修复其自身系统的各部分的用户。

超级用户模型可以与 RBAC 模型共存。下表汇总了 RBAC 模型中可能存在的、从超级用户到受限普通用户的各个等级。该表包括可以在两种模型中跟踪的管理操作。有关权限对系统的单独影响的概述，请参见表 8–2。

表 8–1 超级用户模型与具有权限的 RBAC 模型

系统上的用户功能	超级用户模型	RBAC 模型
可以成为具有全部超级用户功能的超级用户	是	是
可以使用具有全部用户功能的用户身份登录	是	是
可以成为具有有限功能的超级用户	否	是

表 8–1 超级用户模型与具有权限的 RBAC 模型 （续）

系统上的用户功能	超级用户模型	RBAC 模型
可以使用用户身份登录，有时具有超级用户功能	是，仅使用 setuid 程序	是，使用 setuid 程序和 RBAC
可以使用具有管理功能的用户身份登录，但是没有全部超级用户功能	否	是，使用 RBAC 以及直接指定的权限和授权
可以使用功能少于普通用户的用户身份登录	否	是，使用 RBAC 和已删除的权限
可以跟踪超级用户操作	是，通过审计 su 命令	是，通过审计配置文件 shell 命令 另外，如果禁用了 root 用户，则审计跟踪中会出现已经承担 root 角色的用户的名称

Solaris RBAC 元素和基本概念

Solaris OS 中的 RBAC 模型引入了以下元素：

- **授权** – 一种可使用户或角色执行一类可能影响安全性的操作的权限。例如，安装过程中的安全策略会为普通用户提供 solaris.device.cdrw 授权。用户可使用此授权来读取和写入 CD-ROM 设备。有关授权的列表，请参见 /etc/security/auth_attr 文件。
- **权限** – 可以授予命令、用户、角色或系统的单项权利。进程可使用权限来成功执行。例如，proc_exec 权限允许进程调用 execve()。普通用户具有基本权限。要查看基本权限，请运行 ppriv -vl basic 命令。
- **安全属性** – 可供进程用于执行操作的属性。在典型的 UNIX 环境中，进程可使用安全属性来执行原本禁止普通用户执行的操作。例如，setuid 和 setgid 程序具有安全属性。在 RBAC 模型中，普通用户执行的操作可能需要安全属性。除了 setuid 和 setgid 程序之外，授权和权限也是 RBAC 模型中的安全属性。例如，具有 solaris.device.allocate 授权的用户可以分配供独占使用的设备。具有 sys_time 权限的进程可以处理系统时间。
- **特权应用程序** – 可以通过检查**安全属性**来覆盖系统控制的应用程序或命令。在典型的 UNIX 环境和 RBAC 模型中，使用 setuid 和 setgid 的程序都是特权应用程序。在 RBAC 模型中，需要权限或授权才能成功执行的程序也是特权应用程序。有关更多信息，请参见第 174 页中的 “特权应用程序和 RBAC”。
- **权限配置文件** – 可以指定给角色或用户的管理功能的集合。一个权限配置文件由授权、具有安全属性的命令以及其他权限配置文件组成。权限配置文件提供了一种便捷的方法来对安全属性进行分组。
- **角色** – 用于运行特权应用程序的特殊身份。这种特殊身份只能由指定的用户承担。在由角色运行的系统中，超级用户并不是必需的。超级用户功能会分配给不同的角色。例如，在具有两种角色的系统中，将由安全角色处理安全任务，而第二种角色则处理与安全性无关的系统管理任务。角色可以进行更为精细的划分。例如，系统可能包括各种用于处理加密框架、打印机、系统时间、文件系统以及审计的管理角色。

下图说明了各 RBAC 元素如何协同工作。

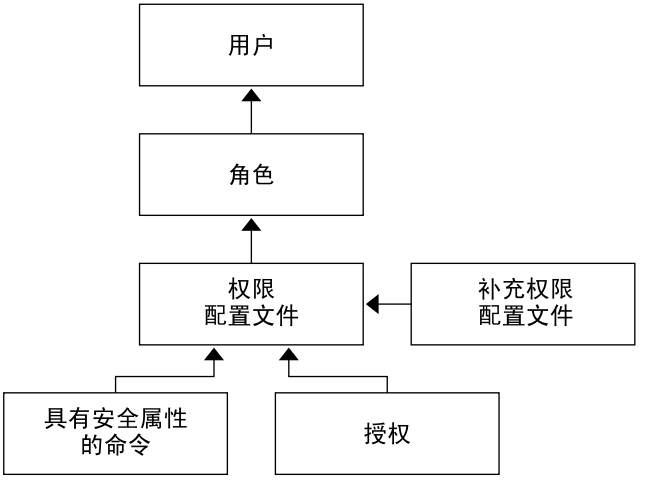


图 8–1 Solaris RBAC 元素关系

在 RBAC 中，将角色指定给用户。用户承担某种角色时，便可使用此角色的功能。角色从权限配置文件中获取其功能。权限配置文件可以包含授权、特权命令以及其他补充权限配置文件。特权命令是指那些使用安全属性执行的命令。

下图使用操作员角色、操作员权限配置文件以及打印机管理权限配置文件来说明 RBAC 的各种关系。

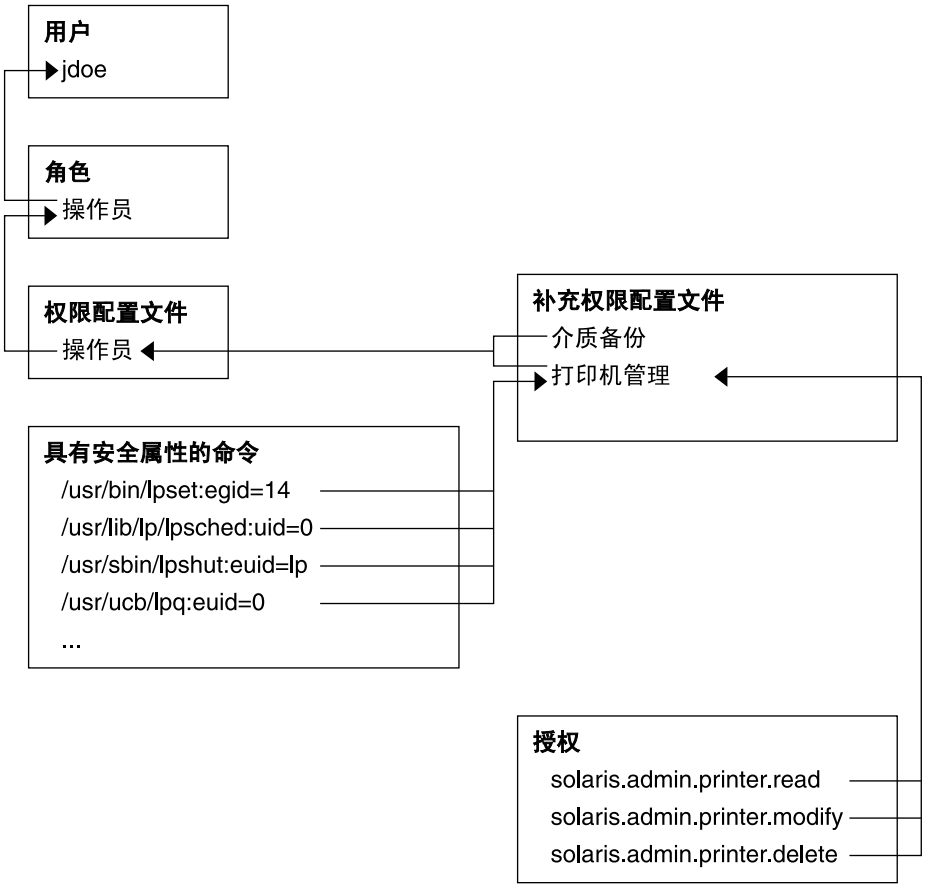


图 8–2 Solaris RBAC 元素关系的示例

操作员角色用于维护打印机以及执行介质备份。此角色会指定给用户 `jdoe`。`jdoe` 通过切换到此角色然后提供角色口令，便可承担此角色。

操作员权限配置文件已指定给操作员角色。操作员权限配置文件包含两个辅助配置文件：打印机管理和介质备份。这些辅助配置文件反映了角色的主要任务。

打印机管理权限配置文件用于管理打印机、打印守护进程和假脱机程序。打印机管理权限配置文件包括三种授权：`solaris.admin.printer.read`、`solaris.admin.printer.delete` 和 `solaris.admin.printer.modify`。角色和用户可使用这些授权来处理打印机队列中的信息。打印机管理权限配置文件还包括一些具有安全属性的命令，例如 `euid=lp` 的 `/usr/sbin/lpshut` 以及 `euid=0` 的 `/usr/ucb/lpq`。

RBAC 授权

授权是指可以授予角色或用户的单项权利。授权可在用户应用程序级别强制执行策略。可以将授权直接指定给角色或用户。通常，授权包括在权限配置文件中，而角色拥有权限配置文件，并且角色会被指定给用户。有关示例，请参见图 8–2。

RBAC 兼容的应用程序可以先检查用户的授权，然后再授予访问应用程序或应用程序内特定操作的权限。此检查取代了传统的 UNIX 应用程序中对 `UID=0` 的检查。有关授权的更多信息，请参见以下各节：

- 第 215 页中的 “授权命名和委托”
- 第 219 页中的 “`auth_attr` 数据库”
- 第 224 页中的 “要求授权的命令”

授权和权限

权限可在内核中强制执行安全策略。授权和权限之间的差别会影响强制执行安全策略的级别。如果没有正确的权限，内核可能会阻止进程执行特权操作。如果没有正确的授权，可能会阻止用户使用特权应用程序或执行特权应用程序内与安全性相关的操作。有关权限的更全面的介绍，请参见第 177 页中的 “权限（概述）”。

特权应用程序和 RBAC

可以覆盖系统控制的应用程序和命令被视为特权应用程序。可以使用安全属性（如 `UID=0`）、权限和授权将应用程序变为特权应用程序。

检查 UID 和 GID 的应用程序

在 UNIX 环境中，长期以来都存在检查 `root` (`UID=0`) 或其他某个特殊 UID 或 GID 的特权应用程序。使用权限配置文件机制，可以将需要特定 ID 的命令分离出来。您可以将具有执行安全属性的命令放在权限配置文件中，而不是针对任何用户均可访问的命令更改 ID。这样，拥有此权限配置文件的用户或角色不必成为超级用户便可运行程序。

可以指定实际 ID 或有效 ID。指定有效 ID 优先于指定实际 ID。有效 ID 等效于文件权限位中的 `setuid` 功能，它还可以标识 UID 以进行审计。但是，由于某些 shell 脚本和程序需要 `root` 的实际 UID，因此也可设置实际 UID。例如，`pkgadd` 命令需要实际 UID 而不是有效 UID。如果使用有效 ID 不足以运行命令，则需要将此 ID 更改为实际 ID。有关过程，请参见第 205 页中的 “如何创建或更改权限配置文件”。

检查权限的应用程序

特权应用程序可以检查权限的使用。使用 RBAC 权限配置文件机制，可以指定特定命令的权限。您可以将具有执行安全属性的命令单独放在权限配置文件中，而无需具有使用应用程序或命令的超级用户功能。这样，拥有此权限配置文件的用户或角色便可使用成功执行命令所需的那几种权限来运行此命令。

用于检查权限的命令包括：

- Kerberos 命令，如 kadmin、kprop 和 kdb5_util
- 网络命令，如 ifconfig、routeadm 和 snoop
- 文件和文件系统命令，如 chmod、chgrp 和 mount
- 用于控制进程的命令，如 kill、pcred 和 rcapadm

要向权限配置文件中添加具有权限的命令，请参见第 205 页中的 “如何创建或更改权限配置文件”。要确定特殊配置文件中用于检查权限的命令，请参见第 236 页中的 “确定已指定的权限”。

检查授权的应用程序

Solaris OS 还提供了用于检查授权的命令。通过定义，root 用户可具有所有授权。因此，root 用户可以运行任何应用程序。用于检查授权的应用程序包括：

- 整个 Solaris Management Console 的工具套件
- 审计管理命令，如 auditconfig 和 auditreduce
- 打印机管理命令，如 lpadmin 和 lpfilter
- 与批处理作业相关的命令，如 at、atq、batch 和 crontab
- 面向设备的命令，如 allocate、deallocate、list_devices 和 cdrw。

要针对授权测试脚本或程序，请参见示例 9–19。要编写需要授权的程序，请参见《Solaris Security for Developers Guide》中的“About Authorizations”。

RBAC 权限配置文件

权限配置文件是指可以指定给角色或用户的系统覆盖值的集合。一个权限配置文件可以包括授权、具有指定安全属性的命令以及其他权限配置文件。权限配置文件信息分放在 prof_attr 和 exec_attr 数据库中。权限配置文件的名称和授权位于 prof_attr 数据库中。权限配置文件名称和具有指定安全属性的命令位于 exec_attr 数据库中。有关权限配置文件的更多信息，请参见以下各节：

- 第 211 页中的 “权限配置文件的内容”
- 第 220 页中的 “prof_attr 数据库”
- 第 221 页中的 “exec_attr 数据库”

RBAC 角色

角色是一种特殊类型的用户帐户，通过此帐户可运行特权应用程序。角色与用户帐户使用相同的常规方式创建。角色具有起始目录、组指定和口令等。权限配置文件和授权可为角色提供管理功能。角色不能从其他角色或其他用户那里继承功能。各角色分配有相应的超级用户功能，因此可以实现更为安全的管理。

用户承担某种角色时，此角色的属性将替换所有用户属性。角色信息存储在 passwd、shadow 和 user_attr 数据库中。可以将角色信息添加到 audit_user 数据库中。有关设置角色的详细信息，请参见以下各节：

- 第 187 页中的 “如何规划 RBAC 实现”
- 第 191 页中的 “如何通过命令行创建角色”
- 第 203 页中的 “如何更改角色的属性”

可以为多个用户分配一种角色。所有可以承担同一角色的用户都具有同一角色起始目录，在同一环境中运行，并且可访问相同文件。用户可以通过从命令行运行 `su` 命令并提供角色名称和口令来承担角色，还可以在 Solaris Management Console 工具中承担角色。

角色无法直接进行登录。用户需要首先登录，然后才能承担角色。用户承担一种角色之后，如果不首先退出当前角色，则无法承担其他角色。用户退出此角色之后，便可承担其他角色。

可以通过将 `root` 用户更改为角色（如第 196 页中的 “如何使 `root` 用户成为角色”中所示），防止匿名 `root` 进行登录。 如果要审计配置文件 shell 命令 `pfexec` ，则审计跟踪需要包含登录用户的实际 UID、用户承担的角色以及相应角色执行的操作。要针对角色操作来审计系统或特定用户，请参见第 195 页中的 “如何审计角色” 。

Solaris 软件没有附带任何预定义的角色。

- 要配置主管理员角色，请参见《System Administration Guide: Basic Administration》中的“Using the Solaris Management Tools With RBAC (Task Map)”。
- 要配置其他角色，请参见第 188 页中的 “如何使用 GUI 创建和指定角色” 。
- 要在命令行中创建角色，请参见第 202 页中的 “管理 RBAC（任务列表）” 。

RBAC 中的配置文件 Shell

角色可以通过 Solaris Management Console 启动器或**配置文件** *shell* 来运行特权应用程序。配置文件 shell 是一种特殊的 shell，可用于识别权限配置文件包括的安全属性。当用户运行 `su` 命令来承担角色时，便会启动配置文件 shell。配置文件 shell 包括 `pfsh`、`pfcsh` 和 `pfksh`。这些 shell 分别对应于 Bourne shell (`sh`)、C shell (`csh`) 和 Korn shell (`ksh`)。

已直接指定有权限配置文件的用户必须调用配置文件 shell 才能运行具有安全属性的命令。有关可用性和安全性的注意事项，请参见第 177 页中的 “直接指定安全属性时的安全注意事项” 。

可以对所有在配置文件 shell 中执行的命令进行审计。有关更多信息，请参见第 195 页中的 “如何审计角色” 。

名称服务范围和 RBAC

名称服务范围是用于了解 RBAC 的一个重要概念。角色的范围可能会限定为单独的主机。或者，此范围还可能包括所有由名称服务（如 NIS、NIS+ 或 LDAP）提供服务的主机。文件 `/etc/nsswitch.conf` 指定了系统的名称服务范围。遇到第一个匹配项时，查找便会停止。例如，如果某个权限配置文件存在于两个名称服务范围中，则只使用第一个名称服务范围中的各项。如果 `files` 是第一个匹配项，则角色的范围将限定为本地主机。

直接指定安全属性时的安全注意事项

通常，用户通过角色来获取管理功能。授权和特权命令将分组到一个权限配置文件中，而角色拥有此权限配置文件，并且角色会指定给用户。

还可以直接指定权限配置文件和安全属性：

- 可以将权限配置文件、权限和授权直接指定给用户。
- 可以将权限和授权直接指定给角色。

但是，直接指定的做法并不安全。具有直接指定的权限的用户和角色可能会在内核需要此权限的情况下覆盖安全策略。如果某权限是权限配置文件中某个命令的安全属性，则此权限仅供拥有此权限配置文件的用户用于此命令。不能将此权限用于用户或角色可能运行的其他命令。

由于授权在用户级别执行，因此，与直接指定权限相比，直接指定授权的危险性会较小。但是，用户可使用授权来执行高度安全的任务，如委托设备管理。

直接指定给用户的权限配置文件存在的可用性问题要多于安全性问题。权限配置文件中具有安全属性的命令只能在配置文件 shell 中成功执行。用户必须打开配置文件 shell，然后再键入命令。指定有权限配置文件的角色会自动获取配置文件 shell。因此，命令可在此角色的 shell 中成功执行。

权限配置文件提供了一种可扩展的便捷方法，用于针对特定管理任务将安全特征进行分组。

权限（概述）

进程权利管理允许在命令、用户、角色和系统四个级别上对进程加以限制。Solaris OS 通过**权限**实现进程权利管理。权限可以降低与（在系统中具有完全超级用户功能的）某个用户或某个进程相关的安全风险。权限和 RBAC 为传统的超级用户模型提供了功能强大的替代模型。

- 有关 RBAC 的信息，请参见第 169 页中的 “基于角色的访问控制（概述）”。
- 有关如何管理权限的信息，请参见第 11 章。
- 有关权限的参考信息，请参见第 12 章。

权限保护内核进程

权限是进程执行某项操作所需的单项权利。权利是在内核中实施的。在 Solaris 权限**基本集**范围内运行的程序可在系统安全策略范围内运行。`setuid` 程序是在系统安全策略范围之外运行的程序示例。通过使用权限，程序可以不必调用 `setuid`。

权限会分别枚举针对系统的可能的操作种类。程序可以使用可使其成功执行的确切权限运行。例如，用来设置日期并将其写入管理文件的程序可能需要 `file_dac_write` 和 `sys_time` 权限。借助此功能，可以不必以 `root` 身份运行任何程序。

以前，系统并未遵循权限模型，而是使用超级用户模型。在超级用户模型中，进程以 root 或用户身份运行。用户进程被限定为对用户的目录和文件执行操作，root 进程则可在系统中的任何位置创建目录和文件。需要在用户目录外部创建目录的进程将使用 UID=0（即作为 root）运行。安全策略依靠 DAC（discretionary access control，自主访问控制）来保护系统文件。设备节点受到 DAC 的保护。例如，组 sys 拥有的设备只能由 sys 组的成员打开。

但是，setuid 程序、文件权限和管理帐户很容易被误用。setuid 进程所允许的操作数比该进程完成其运行过程所需的操作数多。入侵者会破坏 setuid 程序，然后以可执行所有功能的 root 用户身份运行。同样，可访问 root 口令的任何用户都可能会破坏整个系统的安全。

与之相反，通过权限来实施策略的系统允许在用户功能和 root 功能之间划分等级。可以授予用户执行超出普通用户功能的活动，并将 root 加以限制，使 root 具有的权限比目前要少。借助 RBAC，可以将使用权限运行的命令单独放在权限配置文件中，并指定给一个用户或角色。表 8–1 汇总了 RBAC 和权限模型提供的用户功能和超级用户功能之间的等级。

权限模型比超级用户模型具有更大的安全性。已从进程中删除的权限不会被利用。进程权限防止程序或管理帐户获取对所有功能的访问权限。进程权限可为敏感文件提供额外的保护措施，而 DAC 防护功能本身可被用来获取访问权限。

之后，权限可将程序和进程限制为仅具备程序所需的功能。此功能称为**最低权限原则**。在实现最低权限的系统上，捕获某个进程的入侵者只能访问该进程所具有的那些权限，不会破坏其余部分的系统安全。

权限说明

可以根据权限范围对权限进行逻辑分组。

- FILE **权限** – 以字符串 file 开头并作用于文件系统对象的权限。例如，file_dac_write 权限可在写入文件时覆盖自主访问控制。
- IPC **权限** – 以字符串 ipc 开头并覆盖 IPC 对象访问控制的权限。例如，进程可使用 ipc_dac_read 权限来读取受 DAC 保护的远程共享内存。
- NET **权限** – 以字符串 net 开头并提供对特定网络功能进行访问的权限。例如，设备可使用 net_rawaccess 权限连接到网络。
- PROC **权限** – 以字符串 proc 开头并允许进程修改其自身受限制属性的权限。PROC 权限包括影响非常有限的权限。例如，进程可借助 proc_clock_highres 权限来使用高分辨率的计时器。
- SYS **权限** – 以字符串 sys 开头并为进程提供对各种系统属性进行无限制访问的权限。例如，进程可使用 sys_linkdir 权限来建立和断开指向目录的硬链接。

某些权限对系统具有有限的影响，而某些权限则具有广泛的影响。proc_taskid 权限的定义指明了其有限的影响：

```
proc_taskid

        Allows a process to assign a new task ID to the calling process.
```

file_setid 权限的定义指明了其广泛的影响：

```
net_rawaccess

        Allow a process to have direct access to  the  network layer.
```

privileges(5) 手册页中提供了每个权限的描述。ppriv -lv 命令会将对每个权限的描述显示在标准输出中。

具有权限的系统的管理差别

具有权限的系统与没有权限的系统之间存在多种明显差别。下表列出了部分差别。

表 8–2 具有权限的系统与没有权限的系统之间的明显差别

功能	没有权限	权限
守护进程	以 root 身份运行的守护进程。	以用户 daemon 身份运行的守护进程。 例如，以下守护进程已指定有相应的权限，并以 daemon 身份运行：lockd、mountd、nfsd 和 rpcbind。
日志文件拥有权	日志文件由 root 拥有。	现在，日志文件由创建了此日志文件的 daemon 拥有。root 用户不拥有此文件。
错误消息	错误消息涉及超级用户。 例如，chroot:not superuser。	错误消息反映权限的使用。 例如，chroot 故障的等效错误消息为 chroot:exec failed。
setuid 程序	程序使用 setuid 来完成不允许普通用户执行的任务。	许多 setuid 程序都已更改为使用权限运行。 例如，以下实用程序会使用权限：ufsdump、ufsrestore、rsh、rlogin、rcp、rdist、ping、traceroute 和 newtask。
文件权限	设备权限受 DAC 控制。例如，sys 组的成员可以打开 /dev/ip。	文件权限 (DAC) 不会预测可以打开设备的对象。设备通过 DAC 和设备策略进行保护。 例如，/dev/ip 文件具有 666 种权限，但是设备只能由具有相应权限的进程打开。原始套接字仍受 DAC 保护。
审计事件	对 su 命令的使用进行审计涉及许多管理功能。	对权限的使用进行审计涉及大多数管理功能。pm 和 as 审计类包括用于配置设备策略的审计事件以及用于设置权限的审计事件。
进程	进程受进程属主保护。	进程受权限保护。进程权限和进程标志可显示为 /proc/<pid> 目录中的一个新项 priv。

表 8–2 具有权限的系统与没有权限的系统之间的明显差别 （续）

功能	没有权限	权限
调试	核心转储中不引用任何权限。	核心转储的 ELF 注释部分包括有关 NT_PRPRIV 和 NT_PRPRIVINFO 注释中的进程权限和标志的信息。 ppriv 实用程序和其他实用程序可显示大小合适的集的正确数目。这些实用程序会将位集中的位正确映射为权限名称。

如何实现权限

每个进程都有四个权限集，用于确定进程是否可以使用特定权限。内核会自动计算权限的**有效集**。可以修改权限的初始**可继承集**。通过编码来使用权限的程序可以减小程序的权限**允许集**。可以缩小权限的**限制集**。

- **有效权限集 (E)** – 当前有效的权限集。进程可以将允许集中的权限添加到有效集，还可以从 E 中删除权限。
- **允许权限集 (P)** – 可用的权限集。权限可通过继承或指定来供程序使用。执行配置文件便是一种将权限指定给程序的方法。setuid 命令可将 root 具有的所有权限指定给程序。可从允许集中删除权限，但不能向该集中添加权限。从 P 中删除的权限会自动从 E 中删除。

  权限识别程序会从程序的允许集中删除该程序从不使用的权限。通过这种方法，程序或恶意进程便无法使用不必要的权限。有关可识别权限的程序的更多信息，请参见《Solaris Security for Developers Guide》中的第 2 章，“Developing Privileged Applications”。
- **可继承权限集 (I)** – 进程可以通过调用 exec 而继承的权限集。调用 exec 之后，允许集和有效集便会相同，但是 setuid 程序的特殊情况除外。

  对于 setuid 程序，调用 exec 之后，可继承集会首先受限制集的限制。然后，将继承的权限集 (I) 减去限制集 (L) 中的所有权限后的权限指定给此进程的 P 和 E。
- **限制权限集 (L)** – 对可用于进程及其子进程的权限的外部限制。缺省情况下，限制集为所有权限。进程可以缩小限制集，但是永远不能扩展限制集。L 用于限制 I。因此，L 会在调用 exec 时限制 P 和 E。

  如果已为用户指定的配置文件中包括已指定有权限的程序，则此用户通常可以运行此程序。在未修改的系统上，为此程序指定的权限位于此用户的限制集中。已为程序指定的权限会成为此用户允许集的一部分。要运行已指定有权限的程序，用户必须从配置文件 shell 运行此程序。

内核可识别**基本权限集**。在未修改的系统上，每个用户的初始可继承集等效于登录时获取的基本集。可以修改用户的初始可继承集，但不能修改基本集。

在未修改的系统上，用户在登录时的权限集将显示以下类似信息：

```
E (Effective): basic

I (Inheritable): basic
```

```
P (Permitted): basic

L (Limit): all
```

因此，登录时所有用户在其各自的可继承集、允许集和有效集中包含基本集。用户的限制集包含所有权限。要在用户的有效集中加入更多权限，必须为该用户指定一个权限配置文件。此权限配置文件将包括已向其中添加了权限的命令。还可以将权限直接指定给用户或角色，尽管这种权限指定可能会存在风险。有关风险的介绍，请参见第 177 页中的 “直接指定安全属性时的安全注意事项” 。

进程如何获取权限

进程可以继承权限。或者，可以为进程指定权限。进程从其父进程继承权限。登录时，用户的初始可继承权限集确定可用于此用户进程的权限。用户初始登录的所有子进程都可继承此权限集。

还可以将权限直接指定给程序、用户和角色。当某个程序需要权限时，可以在权限配置文件中将权限指定给此程序的可执行文件。允许运行此程序的用户或角色会被指定包括此程序的配置文件。登录或进入配置文件 shell 时，如果在配置文件 shell 中键入程序的可执行文件，则可使用权限运行此程序。例如，拥有对象访问管理配置文件的角色可以使用 `file_chown` 权限运行 `chmod` 命令。

当某个角色或用户运行已直接指定有其他权限的程序时，指定的权限会添加到此角色或用户的可继承集中。指定有权限的程序的子进程会继承父进程的权限。如果子进程需要的权限比父进程的权限多，则必须为子进程直接指定这些权限。

通过编码来使用权限的程序称为可识别权限的程序。可识别权限的程序可在程序执行过程中启用和禁用权限。要在生产环境中成功执行，必须为程序指定其启用和禁用的权限。

有关可识别权限的代码的示例，请参见《Solaris Security for Developers Guide》中的第 2 章，“Developing Privileged Applications”。要为需要权限的程序指定权限，请参见第 232 页中的 “如何为命令添加权限” 。

指定权限

您作为系统管理员应负责指定权限。通常，可在权限配置文件中将权限指定给命令。然后，将权限配置文件指定给角色或用户。Solaris Management Console 提供了图形用户界面 (graphical user interface, GUI) 来指定权限。还可以使用 `smuser` 和 `smrole` 等命令来指定权限。有关如何使用 GUI 来指定权限的更多信息，请参见第 9 章。

还可以将权限直接指定给用户。如果您相信某些用户在其整个会话过程中会负责地使用某种权限，则可以直接指定此权限。适合直接指定的权限是具有有限影响的权限，如 `proc_clock_highres`。不适合直接指定的权限是具有广泛影响的权限，如 `file_dac_write`。

还可以拒绝为用户或系统指定权限。从用户或系统的初始可继承集或限制集中删除权限时必须谨慎。

扩展用户或角色的权限

用户和角色具有可继承权限集以及限制权限集。限制集不能扩展，因为限制集最初包括所有权限。可以针对用户、角色和系统扩展初始可继承集。还可以将不在可继承集中的权限指定给进程。

按进程指定权限是最精确的添加权限方法。可以通过允许某用户承担某种角色，扩展此用户可执行的特权操作的数目。可以为角色指定包括具有已添加权限的命令的配置文件。用户承担角色时，便会获取此角色的配置文件 shell。通过在角色的 shell 中键入角色配置文件中的命令，便可使用已添加的权限执行这些命令。

还可以将配置文件指定给用户而不是用户承担的角色。配置文件可包括具有已添加权限的命令。用户打开配置文件 shell（如 `pfksh`）时，便可使用权限执行用户配置文件中的命令。在常规 shell 中，不使用权限执行命令。特权进程只能在特权 shell 中执行。

扩展用户、角色或系统的初始可继承权限集是一种风险性较高的指定权限方法。可继承集中的所有权限都位于允许集和有效集中。用户或角色在 shell 中键入的所有命令都可以使用直接指定的权限。使用直接指定的权限，用户或角色可以轻松执行可能超出其管理职责范围的操作。

向某系统上的初始可继承权限集中添加权限后，所有登录到此系统的用户都会具有更大的基本权限集。通过这种直接指定，系统的所有用户都可以轻松执行可能超出普通用户执行范围的操作。

限制用户或角色的权限

通过删除权限，可以防止用户和角色执行特定的任务。可以从初始可继承集和限制集中删除权限。分配小于缺省集的初始可继承集或限制集之前，应谨慎地对权限删除进行测试。通过从初始可继承集中删除权限，可能会使用户无法登录。从限制集中删除权限后，传统的 `setuid` 程序可能会失败，因为此程序需要的权限已被删除。

向脚本指定权限

脚本与命令一样，也是可执行文件。因此，在权限配置文件中，可以将权限添加到脚本中，就像将权限添加到命令中一样。已指定有配置文件的用户或角色在配置文件 shell 中执行脚本时，将会使用已添加的权限来运行脚本。如果脚本包含需要权限的命令，则具有已添加权限的命令也应该位于配置文件中。

可识别权限的程序可以限制每个进程的权限。包含可识别权限的程序的作业是指为可执行文件指定此程序所需的确切权限。然后测试此程序，查看此程序是否成功执行了其任务。还可以检查此程序是否误用了其权限。

权限和设备

权限模型使用权限来保护系统接口，这些接口在超级用户模型中单独由文件权限保护。在具有权限的系统中，文件权限太小，因此无法保护这些接口。`proc_owner` 等权限可以覆盖文件权限，然后提供对所有系统的完全访问权限。

因此，具有设备目录的拥有权不足以打开设备。例如，不再自动允许 sys 组的成员打开 /dev/ip 设备。/dev/ip 的文件权限为 0666，但是需要 net_rawaccess 权限才能打开设备。

设备策略受权限控制。getdevpolicy 命令可显示每个设备的设备策略。设备配置命令 devfsadm 可用于安装设备策略。devfsadm 命令可将权限集与 open 绑定，以便设备读取或写入。有关更多信息，请参见 getdevpolicy(1M) 和 devfsadm(1M) 手册页。

使用设备策略，可以更灵活地为打开的设备授予权限。您可以要求与缺省设备策略不同的权限，或者比其更多的权限。可以针对设备策略和驱动程序适当地修改权限要求。可以在安装、添加或更新设备驱动程序时修改权限。

add_drv 和 update_drv 命令可以修改设备策略项以及驱动程序特定的权限。必须使用完整的权限集运行进程才能更改设备策略。有关更多信息，请参见 add_drv(1M) 和 update_drv(1M) 手册页。

权限和调试

Solaris OS 提供了各种工具来调试权限故障。ppriv 命令和 truss 命令可提供调试输出。有关示例，请参见 ppriv(1) 手册页。有关过程，请参见第 230 页中的 “如何确定程序所需的权限”。

◆ ◆ ◆ 第 9 章

使用基于角色的访问控制（任务）

本章介绍与使用各种不同的角色分配超级用户功能相关的任务。角色可以使用的机制包括权限配置文件、授权和特权。以下是本章中的任务列表：

- 第 185 页中的 “使用 RBAC（任务列表）”
- 第 186 页中的 “配置 RBAC（任务列表）”
- 第 198 页中的 “使用角色（任务列表）”
- 第 202 页中的 “管理 RBAC（任务列表）”

有关 RBAC 的概述，请参见第 169 页中的 “基于角色的访问控制（概述）” 。有关参考信息，请参见第 10 章。要通过 RBAC 或不通过 RBAC 使用特权，请参见第 11 章。

使用 RBAC（任务列表）

要使用 RBAC，需要规划和配置 RBAC 以及了解如何承担角色。熟悉角色后，便可以进一步自定义 RBAC 以处理新操作。以下任务列表列出了这些主要任务：

任务	说明	参考
规划和配置 RBAC	在站点上配置 RBAC。	第 186 页中的 “配置 RBAC（任务列表）”
使用角色	通过命令行和在 Solaris Management Console GUI 中承担角色。	第 198 页中的 “使用角色（任务列表）”
自定义 RBAC	为站点自定义 RBAC。	第 202 页中的 “管理 RBAC（任务列表）”

配置 RBAC（任务列表）

要有效地使用 RBAC，需要进行规划。使用以下任务列表可在站点上规划并初步实现 RBAC。

任务	说明	参考
1. 规划 RBAC	涉及检查站点的安全要求，以及确定如何在站点上使用 RBAC。	第 187 页中的 “如何规划 RBAC 实现”
2. 学习使用 Solaris Management Console	涉及熟悉 Solaris Management Console。	《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”
3. 配置第一个用户和角色	使用 Solaris Management Console 的 RBAC 配置工具创建用户和角色，并将角色指定给用户。	《System Administration Guide: Basic Administration》中的“Using the Solaris Management Tools With RBAC (Task Map)”
4.（可选）创建可以承担角色的其他用户	确保存在可以承担管理角色的用户。	《System Administration Guide: Basic Administration》中的“Using the Solaris Management Tools With RBAC (Task Map)”
5.（建议）创建其他角色并将其指定给用户	使用 RBAC 工具为特定管理区域创建角色，并将这些角色指定给用户。	第 188 页中的 “如何使用 GUI 创建和指定角色”
		示例 9–5
	使用命令行创建角色，并将这些角色指定给用户。	第 191 页中的 “如何通过命令行创建角色”
		第 194 页中的 “如何将角色指定给本地用户”
6.（建议）审计角色操作	预先选择包括记录角色操作的审计事件的审计类。	第 195 页中的 “如何审计角色”
7.（可选）使 root 用户成为角色	防止匿名 root 登录，这是一个安全漏洞。	第 196 页中的 “如何使 root 用户成为角色”

配置 RBAC

可以使用以下实用程序配置 RBAC：

- **Solaris Management Console GUI** – 执行与 RBAC 相关的任务的首选方法是通过 GUI。用于管理 RBAC 元素的控制台工具包含在用户工具集中。
- **Solaris Management Console 命令** – 使用 Solaris Management Console 命令行界面（如 `smrole`），可对任何名称服务进行操作。Solaris Management Console 命令需要进行验证才能连接到服务器。因此，这些命令并不适合在脚本中使用。
- **本地命令** – 使用 `user*` 和 `role*` 组合这两类命令行界面（如 `useradd`），仅可对本地文件进行操作。对本地文件进行操作的命令必须由超级用户或具有相应特权的角色来运行。

▼ 如何规划 RBAC 实现

RBAC 可以作为组织管理其信息资源方式的组成部分。进行规划时，需要全面了解 RBAC 功能以及组织的安全要求。

1 了解 RBAC 基本概念。

请阅读第 169 页中的 “基于角色的访问控制（概述）” 。使用 RBAC 来管理系统与使用常规的 UNIX 管理做法完全不同。开始实现之前，应先对 RBAC 概念进行了解。有关更详细的信息，请参见第 10 章。

2 检查安全策略。

您组织的安全策略应详细说明系统面临的潜在威胁，衡量每种威胁的风险并制订应对这些威胁的策略。通过 RBAC 隔离与安全相关的任务可以作为该策略的一部分。虽然可以按照缺省设置安装系统建议的角色及其配置，但是您可能需要自定义 RBAC 配置以符合安全策略。

3 确定组织需要 RBAC 的程度。

根据安全需要，选择如何使用 RBAC，如下所述：

- **无 RBAC** – 您可以以 root 用户身份执行所有任务。在此配置中，您将以自己的身份登录。然后，在选择 Solaris Management Console 工具时，键入 root 作为用户。
- **仅单一角色** – 此方法添加一个角色。该角色会被指定主管理员权限配置文件。此方法与超级用户模型类似，因为该角色具有超级用户功能。但是，通过此方法可以跟踪已承担该角色的用户。
- **建议的角色** – 此方法创建三个基于以下权限配置文件的角色：主管理员、系统管理员和操作员。这些角色适用于管理员具有不同责任级别的组织。
- **自定义角色** – 您可以创建自己的角色以满足组织的安全要求。新角色可以基于现有或自定义的权限配置文件。
- **使超级用户成为角色** – 此方法可防止任何用户以 root 身份登录。相反，在承担 root 角色之前，用户必须以普通用户身份登录。有关详细信息，请参见第 196 页中的 “如何使 root 用户成为角色” 。

4 确定适用于组织的建议角色。

请查看建议的角色的功能和缺省的权限配置文件。通过缺省的权限配置文件，管理员可以使用单个配置文件配置建议的角色。以下三个缺省的权限配置文件可用于配置建议的角色：

- **主管理员权限配置文件** – 用于配置可以执行所有管理任务、为其他用户授予权限以及编辑与管理角色关联的权限的角色。该角色中的用户可将该角色指定给其他用户，并可为其他用户授予权限。
- **系统管理员权限配置文件** – 用于配置可执行大多数与安全无关的管理任务的角色。例如，系统管理员可以添加新的用户帐户，但不能设置口令或为其他用户授予权限。
- **操作员权限配置文件** – 用于配置可以执行介质备份和打印机维护等简单管理任务的角色。

要进一步检查权限配置文件，请阅读以下内容之一：

- 在 /etc/security 目录中，阅读 prof_attr 数据库和 exec_attr 数据库的内容。
- 在 Solaris Management Console 中，使用“权限”工具显示权限配置文件的内容。
- 在本书中，参阅第 211 页中的 “权限配置文件的内容”以了解某些典型的权限配置文件的摘要。

5 确定是否有任何其他角色或权限配置文件适用于组织。

请在站点上查找可能从受限制访问中受益的其他应用程序或应用程序系列。合适的 RBAC 候选对象包括：影响安全的应用程序、可能导致服务被拒绝的应用程序，或需要对管理员进行特殊培训的应用程序。您可以自定义角色和权限配置文件，以处理组织的安全要求。

a. 确定新任务所需的命令。

b. 确定适用于此任务的权限配置文件。

检查现有权限配置文件是否可以处理此任务，或是否需要创建单独的权限配置文件。

c. 确定适用于此权限配置文件的角色。

确定是否应将此任务的权限配置文件指定给现有角色，或是否应创建新角色。如果使用现有角色，请检查其他权限配置文件是否适用于将被指定该角色的用户。

6 确定应将哪些用户指定给可用角色。

根据最低特权的原则，应将用户指定给适合其信任级别的角色。如果禁止用户访问用户无需执行的任务，则可以减少潜在的问题。

▼ 如何使用 GUI 创建和指定角色

您可以以超级用户身份，也可以使用主管理员角色来创建新角色。在此过程中，新角色的创建者会承担主管理员的角色。

开始之前

- 您已在站点上创建可承担角色的用户。如果尚未创建这些用户，请按照《System Administration Guide: Basic Administration》中的“Using the Solaris Management Tools With RBAC (Task Map)”的说明进行创建。
- 已按照《System Administration Guide: Basic Administration》中的“Using the Solaris Management Tools With RBAC (Task Map)”的过程，为您指定了主管理员角色。

1 启动 Solaris Management Console。

```
# /usr/sbin/smc &
```

有关登录说明，请参见第 201 页中的 “如何在 Solaris Management Console 中承担角色”。

2 单击“管理角色”图标。

3 从“操作”菜单中选择“添加管理角色”。

4 **填写一系列对话框中的字段以创建新角色。**

有关可能的角色，请参见示例 9–1 至示例 9–4。

提示 – Solaris Management Console 中的所有工具都会在页面底部或向导面板的左侧显示信息。您可随时选择“帮助”，以查找有关在此界面中执行任务的其他信息。

---

5 **将角色指定给用户。**

提示 – 填写角色的属性后，最后一个对话框将提示您为该角色指定一个用户。

---

6 **在终端窗口中，重新启动名称服务高速缓存守护进程。**

```
# svcadm restart system/name-service-cache
```

有关更多信息，请参见 `svcadm`(1M) 和 `nscd`(1M) 手册页。

示例 9–1 为系统管理员权限配置文件创建角色

在本示例中，新角色可以执行与安全无关的系统管理任务。该角色是通过执行上述过程创建的，其参数如下：

- 角色名称：`sysadmin`
- 角色全名：`System Administrator`
- 角色说明：`Performs non-security admin tasks`
- 权限配置文件：`System Administrator`

  此权限配置文件位于该角色具有的配置文件列表的顶部。

示例 9–2 为操作员权限配置文件创建角色

操作员权限配置文件可以管理打印机并将系统备份到脱机介质。您可能希望将该角色指定给各个班次上的某个用户。为此，可在“步骤 1：进入‘角色名’对话框”中选择角色邮件列表选项。该角色是通过执行上述过程创建的，其参数如下：

- 角色名称：`operadm`
- 全名：`Operator`
- 描述：`Backup operator`
- 权限配置文件：`Operator`

  此权限配置文件必须位于该角色中具有的配置文件列表的顶部。

示例 9–3 为与安全相关的权限配置文件创建角色

缺省情况下，仅有主管理员配置文件包含与安全相关的命令和权限。如果要创建功能不如主管理员强大，但可处理某些与安全相关的任务的角色，则必须创建该角色。

在以下示例中，该角色可保护设备。该角色是通过执行上述过程创建的，其参数如下：

- 角色名称： devicesec
- 全名： Device Security
- 描述： Configures Devices
- 权限配置文件： Device Security

在以下示例中，该角色可确保系统和主机在网络上的安全。该角色是通过执行上述过程创建的，其参数如下：

- 角色名称： netsec
- 全名：Network Security
- 描述： Handles IPsec, IKE, and SSH
- 权限配置文件：Network Security

示例 9–4 为具有有限范围的权限配置文件创建角色

许多权限配置文件的范围都是有限的。在本示例中，该角色的唯一任务是管理 DHCP。该角色是通过执行上述过程创建的，其参数如下：

- 角色名称： dhcpmgt
- 全名： DHCP Management
- 描述： Manages Dynamic Host Config Protocol
- 权限配置文件： DHCP Management

示例 9–5 修改用户的角色指定

在本示例中，将向现有用户添加角色。您可以修改用户的角色指定，方法是在 Solaris Management Console 的“用户”工具中单击“用户帐户”图标，双击相应用户，然后按照联机帮助的说明将角色添加到该用户的功能。

故障排除 如果角色不具有应有的功能，请检查以下情况：

- 角色的权限配置文件是否按功能从高到低的顺序在 GUI 中列出？

  例如，如果 All 权限配置文件位于列表顶部，则不会运行具有安全性属性的命令。包含具有安全性属性的命令的配置文件在列表中必须位于 All 权限配置文件的前面。

- 角色的权限配置文件中的命令是否具有相应的安全性属性？

  例如，如果策略为 suser，则某些命令会要求 uid=0，而不是要求 euid=0。

- 是否在相应的名称服务范围中定义了权限配置文件？角色是否在定义权限配置文件的名称服务范围内运行？

- 名称服务高速缓存 svc:/system/name-service-cache 是否已重新启动？

  nscd 守护进程可以具有很长的生存时间间隔。通过重新启动此守护进程，可使用当前数据更新该名称服务。

▼ 如何通过命令行创建角色

Solaris Management Console GUI 是管理 RBAC 的首选方法。要使用该 GUI，请参见第 188 页中的 “如何使用 GUI 创建和指定角色”。另外，还可以使用命令行界面，如此过程中所述。

注 – 请勿尝试同时使用命令行和图形用户界面来管理 RBAC。这样可能会导致对配置所做的更改出现冲突，从而使得行为不可预测。您可以使用这两种工具来管理 RBAC，但是不能同时使用二者。

开始之前 要创建角色，必须承担具有主管理员权限配置文件的角色，或切换到用户 `root`。

1. **承担主管理员角色，或成为超级用户。**

   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2. **选择以下命令之一，在命令行上创建角色。**

   - **对于本地名称服务范围内的角色，请使用 `roleadd` 命令。**

     注 – 与 Solaris Management Console GUI 或命令行界面相比，`roleadd` 命令的限制更多。运行 `roleadd` 命令后，还必须运行 `usermod` 命令才能将角色指定给用户。然后，用户还必须为角色设置口令，如第 194 页中的 “如何将角色指定给本地用户”中所述。

     ```
     # roleadd -c comment \
     -g group -m homedir -u UID -s shell \
     -P profile rolename
     ```

`-c` *comment*	描述 *rolename* 的注释。
`-g` *group*	*rolename* 的组指定。
`-m` *homedir*	*rolename* 的起始目录的路径。
`-u` *UID*	*rolename* 的 UID。
`-s` *shell*	*rolename* 的登录 shell。此 shell 必须是配置文件 shell。
`-P` *profile*	*rolename* 的一个或多个权限配置文件。
rolename	新本地角色的名称。

- **使用 `smrole add` 命令。**
  此命令可在 NIS、NIS+ 或 LDAP 等分布式名称服务中创建角色。此命令将作为 Solaris Management Console 服务器的客户机运行。

  ```
  $ /usr/sadm/bin/smrole -D domain-name \

  -r admin-role -l <Type admin-role password> \

  add -- -n rolename -a rolename -d directory\

  -F full-description -p profile
  ```

`-D` *domain-name*	要管理的域的名称。
`-r` *admin-role*	可以修改角色的管理角色的名称。管理角色必须具有 `solaris.role.assign` 授权。如果要修改已承担的角色，则该角色必须具有 `solaris.role.delegate` 授权。
`-l`	*admin-role* 的口令输入提示。
`--`	验证选项和子命令选项之间必需的分隔符。
`-n` *rolename*	新角色的名称。
`-c` *comment*	描述角色功能的注释。
`-a` *username*	可以承担 *rolename* 的用户的名称。
`-d` *directory*	*rolename* 的起始目录。
`-F` *full-description*	*rolename* 的完整说明。此说明显示在 Solaris Management Console GUI 中。
`-p` *profile*	*rolename* 的功能中包括的权限配置文件。此选项可为角色提供具有管理功能的命令。您可以指定多个 `-p` *profile* 选项。

3. **要使更改生效，请参见第 194 页中的“ 如何将角色指定给本地用户 ”。**

示例 9–6 使用 `smrole` 命令创建自定义操作员角色

`smrole` 命令可在名称服务中指定新角色及其属性。在以下示例中，主管理员创建了一个新版本的操作员角色。该角色具有标准的操作员权限配置文件以及介质恢复权限配置文件。请注意，此命令会提示您输入新角色的口令。

```
% su primaryadm

Password:       <键入 primaryadm 的口令>

$ /usr/sadm/bin/smrole add -H myHost -- -c "Backup and Restore Operator" \

-n operadm2 -a janedoe -d /export/home/operadm \
```

```
-F "Backup/Restore Operator" -p "Operator" -p "Media Restore"
Authenticating as user: primaryadm

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password ::      <键入 primaryadm 的口令>

Loading Tool: com.sun.admin.usermgr.cli.role.UserMgrRoleCli from myHost
Login to myHost as user primaryadm was successful.
Download of com.sun.admin.usermgr.cli.role.UserMgrRoleCli from myHost was successful.

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password ::     <键入 operadm2 的口令>

$ svcadm restart system/name-service-cache
```

包含 `list` 子命令的 `smrole` 命令用于显示新角色：

```
$ /usr/sadm/bin/smrole list --
Authenticating as user: primaryadm

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password ::      <键入 primaryadm 的口令>

Loading Tool: com.sun.admin.usermgr.cli.role.UserMgrRoleCli from myHost
Login to myHost as user primaryadm was successful.
```

```
Download of com.sun.admin.usermgr.cli.role.UserMgrRoleCli from myHost was successful.

root                    0            Superuser

primaryadm              100          Most powerful role

sysadmin                101          Performs non-security admin tasks

operadm                 102          Backup Operator

operadm2                103          Backup/Restore Operator
```

▼ 如何将角色指定给本地用户

此过程会将本地角色指定给本地用户、重新启动名称高速缓存守护进程，然后说明用户如何承担该角色。

要将角色指定给分布式名称服务中的用户，请参见第 191 页中的 “如何通过命令行创建角色”和第 203 页中的 “如何更改角色的属性”。

开始之前 您已按照第 191 页中的 “如何通过命令行创建角色”中所述添加了本地角色。您必须已承担主管理员角色或已切换到超级用户。

1 将角色指定给本地用户。

如果已使用 `roleadd` 命令添加了本地角色，则必须执行此步骤。使用 `smrole` 命令和 Solaris Management Console 创建角色时，此步骤为可选步骤。

```
# usermod -u UID -R rolename
```

`-u` *UID* 用户的 UID。

`-R` *rolename* 指定给用户的角色。

2 要使更改生效，请重新启动名称服务高速缓存守护进程。

```
# svcadm restart system/name-service-cache
```

如果已使用 Solaris Management Console 界面添加了角色，请转至第 198 页中的 “使用角色（任务列表）”。否则，请继续执行下一步。

3 （可选的）要解除锁定角色帐户，用户必须创建口令。

如果已使用 `roleadd` 命令添加了本地角色，则必须执行此步骤。

```
% su rolename

Password:       <键入 rolename 的口令>

Confirm Password:       <重新键入 rolename 的口令>
```

```
$
```

示例 9–7 通过命令行创建和指定本地角色

在本示例中，将创建管理 Solaris 加密框架的角色。加密管理权限配置文件中包含用于管理本地系统中的硬件和软件加密服务的 `cryptoadm` 命令。

```
# roleadd -c "Cryptographic Services manager" \
-g 14 -m /export/home/cryptoadm -u 104 -s pfksh \
-P "Crypto Management" cryptomgt
# usermod -u 1111 -R cryptomgt
# svcadm restart system/name-service-cache
% su cryptomgt
Password:       <键入 cryptomgt 的口令>
Confirm Password:       <重新键入 cryptomgt 的口令>
$ /usr/ucb/whoami
cryptomgt
$
```

有关 Solaris 加密框架的信息，请参见第 13 章。要管理该框架，请参见第 266 页中的 “管理加密框架（任务列表）” 。

▼ 如何审计角色

可以审计角色执行的操作。审计记录中包括承担角色的用户的登录名、角色名和角色执行的操作。 `6180:AUE_prof_cmd:profile command:ua,as` 审计事件用于收集该信息。通过预先选择 `as` 类或 `ua` 类，可以审计角色操作。

1. **规划审计并编辑审计配置文件。**

   有关更多信息，请参见第 535 页中的 “Solaris 审计（任务列表）” 。

2 **在 audit_control 文件的 flags 行中包括 ua 类或 as 类。**

```
# audit_control file

dir:/var/audit

flags:lo,as

minfree:20

naflags:lo
```

ua 类和 as 类包括其他审计事件。要查看类中包括的审计事件，请阅读 audit_event 文件。另外，还可以使用 bsmrecord 命令，如示例 29–22 中所示。

3 **完成审计服务的配置，然后启用审计。**

有关更多信息，请参见第 546 页中的 “配置和启用审计服务” 。

▼ 如何使 root 用户成为角色

此过程说明如何将 root 从登录用户更改为角色。完成此过程后，将无法再以 root 身份登录到系统，但在单用户模式下除外。如果已为您指定 root 角色，则可以对 root 执行 su。

通过将 root 用户更改为角色，可以防止匿名 root 登录。由于用户必须首先登录，**然后**才能承担 root 角色，因此用户的登录 ID 将提供给审计服务并位于 sulog 文件中。

开始之前 如果将 root 用户更改为角色，但未将该角色指定给有效用户，或当前没有与 root 用户等效的现有角色，则任何用户都不能成为超级用户。

- 为安全起见，至少应为一个本地用户指定 root 角色。
- 如果以 root 身份登录，则无法执行此过程。您必须以自身身份登录，然后才能对 root 执行 su。

1 **以普通用户身份登录到目标主机。**

2 **承担主管理员角色，或成为超级用户。**

主管理员角色拥有主管理员配置文件。要创建该角色并将其指定给用户，请参见《System Administration Guide: Basic Administration》中的“Using the Solaris Management Tools With RBAC (Task Map)”。

3 **创建可承担 root 角色的本地用户。**

```
$ useradd -c comment -d homedir username
```

-c *comment*　描述用户的注释。

-d *homedir*　用户的起始目录。此目录应位于本地系统中。

username 新本地用户的名称。

```
# useradd -c "Local administrative user" -d /export/home1 admuser
```

4 为用户指定口令。

```
# passwd -r files admuser

New Password:        <键入口令>

Re-enter new Password:     <重新键入口令>

passwd: password successfully changed for admuser

#
```

5 确保您未以 root 身份登录。

```
# who

jdoe    console      May 24 13:51    (:0)

jdoe    pts/5        May 24 13:51    (:0.0)

jdoe    pts/4        May 24 13:51    (:0.0)

jdoe    pts/10       May 24 13:51    (:0.0)
```

6 将 root 用户更改为角色。

```
# usermod -K type=role root
```

7 验证 root 是否为角色。

user_attr 文件中 root 项的显示应与以下内容类似：

```
root::::type=role;auths=solaris.*,solaris.grant;profiles=Web Console

Management,All;lock_after_retries=no
```

8 将 root 角色指定给本地管理用户。

```
# usermod -R root admuser
```

9 配置在失败时返回的名称服务。

a. 打开新的终端窗口并承担 root 角色。

```
% whoami

jdoe
```

```
% su admuser

Enter password:        <键入 admuser 的口令>

% roles

root

% su root

Enter password:        <键入 root 的口令>

#
```

b. 编辑 nsswitch.conf 文件。

例如，nsswitch.conf 文件中的以下各项将允许返回名称服务。

```
passwd:  files nis [TRYAGAIN=0 UNAVAIL=return NOTFOUND=return]

group:  files nis [TRYAGAIN=0 UNAVAIL=return NOTFOUND=return]
```

10 将 root 角色指定给名称服务中选定的用户帐户。

有关过程，请参见第 207 页中的 “如何更改用户的 RBAC 属性”。

使用角色（任务列表）

以下任务列表列出了用于在指定角色后使用角色的过程：

任务	说明	参考
使用 Solaris Management Console	以角色身份对自身进行验证，以便在 Solaris Management Console 中执行管理任务。	第 201 页中的 “如何在 Solaris Management Console 中承担角色”
在终端窗口中承担角色	在配置文件 shell 中执行命令行管理任务。	第 199 页中的 “如何在终端窗口中承担角色”

使用角色

使用缺省的 Solaris 权限配置文件设置角色并将其指定给用户后，便可以使用这些角色。可以通过命令行来承担角色。在 Solaris Management Console 中，还可使用角色以本地方式和通过网络来管理系统。

▼ 如何在终端窗口中承担角色

开始之前 必须已为您指定了角色，并且必须使用该信息更新名称服务。

1 在终端窗口中，确定可以承担的角色。

```
% roles
```

Comma-separated list of role names is displayed

2 使用 su 命令承担角色。

```
% su rolename
```

```
Password:        <键入 rolename 的口令>
```

```
$
```

带有角色名的 su 命令会将 shell 更改为该角色的配置文件 shell。配置文件 shell 可识别安全性属性（授权、特权和集 ID 位）。

3 验证您现在是否已承担某种角色。

```
$ /usr/ucb/whoami
```

rolename

您现在可在此终端窗口中执行角色任务。

4 （可选的）查看角色的功能。

有关过程，请参见第 240 页中的 “如何确定角色可以运行的特权命令”。

示例 9–8 承担主管理员角色

在以下示例中，用户承担主管理员角色。在缺省配置中，该角色与超级用户等效。然后，该角色会查看哪些特权可供在角色的配置文件 shell 中键入的任何命令使用。

```
% roles

sysadmin,oper,primaryadm

% su primaryadm

Password:        <键入 primaryadm 口令>

$ /usr/ucb/whoami       提示符更改为角色提示符

primaryadm

$ ppriv $$
```

```
1200:   pfksh

flags = <none>

        E (Effective): all

        I (Inheritable): basic

        P (Permitted): all

        L (Limit): all
```

有关特权的信息，请参见第 177 页中的 “权限（概述）” 。

示例 9–9 承担 root 角色

在以下示例中，用户承担 root 角色。该角色是在第 196 页中的 “如何使 root 用户成为角色” 中创建的。

```
% roles

root

% su root

Password:       <键入 root 的口令>

# /usr/ucb/whoami       提示符更改为角色提示符

root

$ ppriv $$

1200:   pfksh

flags = <none>

        E: all

        I: basic

        P: all

        L: all
```

有关特权的信息，请参见第 177 页中的 “权限（概述）” 。

示例 9–10 承担系统管理员角色

在以下示例中，用户承担系统管理员的角色。与主管理员角色相反，系统管理员角色在其有效集中具有基本的特权集。

```
% roles

sysadmin,oper,primaryadm

% su sysadmin

Password:       <键入 sysadmin 的口令>

$ /usr/ucb/whoami       提示符更改为角色提示符

sysadmin

$ ppriv $$

1200:   pfksh

flags = <none>

        E: basic

        I: basic

        P: basic

        L: all
```

有关特权的信息，请参见第 177 页中的 “权限（概述）”。有关该角色功能的简短说明，请参见第 212 页中的 “系统管理员权限配置文件”。

▼ 如何在 Solaris Management Console 中承担角色

要在 Solaris Management Console GUI 中更改信息，需要具有管理功能。角色可为您提供管理功能。如果要查看信息，则必须具有 `solaris.admin.usermgr.read` 授权。基本 Solaris 用户权限配置文件包括此授权。

开始之前 必须已为您指定了可以更改用户或角色属性的管理角色。例如，主管理员角色可更改用户或角色的属性。

1 启动 Solaris Management Console。

```
% /usr/sbin/smc &
```

有关详细说明，请参见《System Administration Guide: Basic Administration》中的“Using the Solaris Management Tools With RBAC (Task Map)”。

2 **根据任务选择工具箱。**

导航至包含相应名称服务范围内的工具或集合的工具箱，然后单击该图标。这些范围包括文件（本地）、NIS、NIS+ 和 LDAP。如果导航窗格中未显示相应的工具箱，请从“控制台”菜单中选择“打开工具箱”并装入相关的工具箱。

3 **选择要使用的工具。**

导航到该工具或集合，然后单击相应图标。用于管理 RBAC 元素的工具位于“用户”工具中，如下图所示：

4 **在“登录：用户名”对话框中键入用户名和口令。**

5 **在“登录：角色”对话框中对自身进行验证。**

该对话框中的“角色”选项菜单显示了为您指定的角色。请选择角色并键入角色口令。

管理 RBAC（任务列表）

以下任务列表列出了用于在初步实现基于角色的访问控制 (role-based access control, RBAC) 后自定义 RBAC 的过程。

任务	说明	参考
修改角色的属性	修改角色的功能（特权、特权命令、配置文件或授权）。	第 203 页中的 “如何更改角色的属性”
创建或更改权限配置文件	创建权限配置文件。或修改授权、特权命令或权限配置文件中的补充权限配置文件。	第 205 页中的 “如何创建或更改权限配置文件”
更改用户的管理功能	向普通用户添加角色、权限配置文件、授权或特权。	第 207 页中的 “如何更改用户的 RBAC 属性”
确保传统应用程序安全	为传统应用程序启用集 ID 权限。脚本可以包含具有集 ID 的命令。传统应用程序可检查授权（如果适用）。	第 209 页中的 “如何为传统应用程序添加 RBAC 属性”

这些过程可用于管理 RBAC 中使用的元素。有关用户管理的过程，请参阅《System Administration Guide: Basic Administration》中的第 5 章，“Managing User Accounts and Groups (Tasks)”。

管理 RBAC

Solaris Management Console GUI 是管理 RBAC 的首选方法。

注 – 请勿尝试同时使用命令行和图形用户界面来管理 RBAC。这样可能会导致对配置所做的更改出现冲突，从而使得行为不可预测。这两种工具都可以管理 RBAC，但是不能同时使用二者。

▼ 如何更改角色的属性

开始之前 要更改角色属性，您必须已承担主管理员角色或已切换到超级用户。角色属性包括口令、权限配置文件和授权。

◗ **使用以下方法之一更改角色属性。**

- **使用 Solaris Management Console 中的 “ 用户 ” 工具。**

  要启动该控制台，请参见第 201 页中的 “如何在 Solaris Management Console 中承担角色 ” 。请按照左侧窗格中的说明在 “管理角色” 中修改角色。有关更详细的信息，请参见联机帮助。

- **使用 `rolemod` 命令。**

  此命令可修改本地名称服务中定义的角色的属性。

  ```
  $ rolemod -c comment -P profile-list rolename
  ```

  `-c` *comment*　　描述角色功能的新注释。

`-P` *profile-list* 角色具有的配置文件的列表。此列表将替换当前的配置文件列表。

rolename 要修改的现有本地角色的名称。

有关更多命令选项，请参见 `rolemod`(1M) 手册页。

- **使用包含 `modify` 子命令的 `smrole` 命令。**

  此命令可修改 NIS、NIS+ 或 LDAP 等分布式名称服务中的角色的属性。此命令将作为 Solaris Management Console 服务器的客户机运行。

  ```
  $ /usr/sadm/bin/smrole -D domain-name \

  -r admin-role -l <Type admin-role password> \

  modify -- -n rolename  -r username -u username
  ```

  `-D` *domain-name* 要管理的域的名称。

  `-r` *admin-role* 可以修改角色的管理角色的名称。管理角色必须具有 `solaris.role.assign` 授权。如果要修改已承担的角色，则该角色必须具有 `solaris.role.delegate` 授权。

  `-l` *admin-role* 的口令输入提示。

  `--` 验证选项和子命令选项之间必需的分隔符。

  `-n` *rolename* 新角色的名称。

  `-r` *username* 无法再承担 *rolename* 的用户的名称。

  `-u` *username* 现在可以承担 *rolename* 的用户的名称。

  有关更多命令选项，请参见 `smrole`(1M) 手册页。

示例 9–11 使用 `rolemod` 命令更改本地角色的属性

在本示例中，将修改 `operadm` 角色以使其具有介质恢复权限配置文件。

```
$ rolemod -c "Handles printers, backup, AND restore" \

-P "Printer Management,Media Backup,Media Restore,All" operadm
```

示例 9–12 使用 `smrole modify` 命令更改本地角色的属性

在以下示例中，将修改 `operadm` 角色以添加介质恢复权限配置文件。

```
$ /usr/sadm/bin/smrole -r primaryadm -l <Type primaryadm password> \

modify -- -n operadm -c "Handles printers, backup, AND restore" \
```

```
-p "Media Restore"
```

示例 9–13 使用 smrole modify 命令更改域中的角色

在以下示例中，将更改 clockmgr 角色。ID 为 108 的 NIS 用户无法再承担该角色。ID 为 110 的 NIS 用户可以承担 clockmgr 角色。

```
$ /usr/sadm/bin/smrole -D nis:/examplehost/example.domain \

-r primaryadm -l <Type primaryadm password> \

modify -- -n clockmgr -r 108 -u 110
```

▼ 如何创建或更改权限配置文件

权限配置文件是一种角色属性。如果 prof_attr 数据库不包含满足您需求的权限配置文件，则应创建或更改权限配置文件。要了解有关权限配置文件的更多信息，请参见第 175 页中的 “RBAC 权限配置文件”。

开始之前 要创建或更改权限配置文件，您必须已承担主管理员的角色或已切换到超级用户。

◗ **使用以下方法之一更改角色的属性。**

- **使用 Solaris Management Console 中的 “ 用户 ” 工具。**

  要启动该控制台，请参见第 201 页中的 “如何在 Solaris Management Console 中承担角色”。请按照左侧窗格中的说明在 “权限” 中创建或更改权限配置文件。有关更详细的信息，请参见联机帮助。

- **使用 smprofile 命令。**

  使用此命令可以添加、修改、列出或删除权限配置文件。此命令可在文件以及 NIS、NIS+ 或 LDAP 等分布式名称服务中运行。smprofile 命令将作为 Solaris Management Console 服务器的客户机运行。

  ```
  $ /usr/sadm/bin/smprofile -D domain-name \

  -r admin-role -l <Type admin-role password> \

  add | modify -- -n profile-name \

  -d description   -m help-file -p supplementary-profile
  ```

  -D *domain-name*　　要管理的域的名称。

-r *admin-role*	可以修改角色的管理角色的名称。管理角色必须具有 solaris.role.assign 授权。如果要修改已承担的角色，则该角色必须具有 solaris.role.delegate 授权。
-l	*admin-role* 的口令输入提示。
--	验证选项和子命令选项之间必需的分隔符。
-n *profile-name*	新配置文件的名称。
-d *description*	配置文件的简短说明。
-m *help-file*	已创建并放置在 /usr/lib/help/profiles/locale/C 目录中的 HTML 帮助文件的名称。
-p *supplementary-profile*	此权限配置文件中包括的现有权限配置文件的名称。您可以指定多个 -p *supplementary-profile* 选项。

有关更多命令选项，请参见 smprofile(1M) 手册页。

示例 9–14 通过命令行修改权限配置文件

在以下示例中，网络管理权限配置文件充当网络安全权限配置文件的补充配置文件。包含网络安全配置文件的角色现在可以配置网络和主机，而且还可以运行与安全相关的命令。

```
$ /usr/sadm/bin/smprofile -D nisplus:/example.host/example.domain \

-r primaryadm -l <Type primaryadm password> \

modify -- -n "Network Security" \

-d "Manage network and host configuration and security" \

-m RtNetConfSec.html -p "Network Management"
```

运行此命令之前，管理员创建了一个新的帮助文件 RtNetConfSec.html，并将其放置在 /usr/lib/help/profiles/locale/C 目录中。

示例 9–15 使用权限工具创建新的权限配置文件

下表列出了名为“生成管理员”的假设的权限配置文件的样例数据。此权限配置文件包括子目录 /usr/local/swctrl/bin 中的命令。这些命令的有效 UID 为 0。生成管理员权限配置文件适用于管理软件开发的生成和版本控制的管理员。

选项卡	字段	示例
一般	名称	生成管理员

选项卡	字段	示例
	说明	用于管理软件生成和版本控制。
	帮助文件名称	`BuildAdmin.html`
命令	添加目录	单击“添加目录”，在对话框中键入 `/usr/local/swctrl/bin`，然后单击“确定”。
	命令遭拒/许可的命令	将 `/usr/local/swctrl/bin` 移动到“许可的命令”列。
	设置安全性属性	选择 `/usr/local/swctrl/bin`，单击“设置安全性属性”，然后将“有效 UID”设置为 `root`。
授权	排除的授权/包括的授权	**无授权。**
辅助权限	排除的权限/包括的权限	**无补充权限配置文件。**

故障排除 如果权限配置文件没有为角色提供所需的功能，请检查以下情况：

- 角色的权限配置文件是否按功能从高到低的顺序在 GUI 中列出？

  例如，如果 `All` 权限配置文件位于列表顶部，则不会运行具有安全性属性的命令。包含具有安全性属性的命令的配置文件在列表中必须位于 `All` 权限配置文件的前面。
- 角色的权限配置文件中是否多次列出了某个命令？如果是这样，第一个命令实例是否具有所需的全部安全性属性？

  例如，某个命令可以要求该命令特定选项的特权。为使要求特权的选项成功运行，列表中的最高权限配置文件的第一个命令实例必须具有指定的特权。
- 角色的权限配置文件中的命令是否具有相应的安全性属性？

  例如，如果策略为 `suser`，则某些命令要求 `uid=0` 而非 `euid=0` 才能成功运行。
- 名称服务高速缓存 `svc:/system/name-service-cache` 是否已重新启动？

  `nscd` 守护进程可以具有很长的生存时间间隔。通过重新启动此守护进程，可使用当前数据更新该名称服务。

▼ 如何更改用户的 RBAC 属性

用户属性包括口令、权限配置文件和授权。为用户提供管理功能的最安全的方法是将角色指定给用户。有关说明，请参见第 177 页中的 “直接指定安全属性时的安全注意事项”。

开始之前 要更改用户属性，您必须已承担主管理员角色或已切换到超级用户。

◗ **使用以下方法之一更改用户的 RBAC 属性。**

- **使用 Solaris Management Console 中的“ 用户 ”工具。**

  要启动该控制台，请参见第 201 页中的 “如何在 Solaris Management Console 中承担角色”。请按照左侧窗格中的说明在“用户帐户”中修改用户。有关更详细的信息，请参见联机帮助。

提示 – 为用户直接指定授权、特权或权限配置文件并不是一个好的做法。首选方法是将角色指定给用户。然后，用户承担角色以执行特权操作。

- **使用 `usermod` 命令。**

  此命令可修改本地名称服务中定义的用户的属性。

  ```
  $ usermod -R rolename username
  ```

  `-R` *rolename* 现有本地角色的名称。

  username 要修改的现有本地用户的名称。

  有关更多命令选项，请参见 `usermod`(1M) 手册页。

- **使用包含 `modify` 子命令的 `smuser` 命令。**

  此命令可修改 NIS、NIS+ 或 LDAP 等分布式名称服务中的用户的属性。此命令将作为 Solaris Management Console 服务器的客户机运行。

  ```
  $ /usr/sadm/bin/smuser -D domain-name \

  -r admin-role -l <Type admin-role password> \

  modify -- -n username -a rolename
  ```

  `-D` *domain-name* 要管理的域的名称。

  `-r` *admin-role* 可以修改角色的管理角色的名称。管理角色必须具有 `solaris.role.assign` 授权。如果要修改已承担的角色，则该角色必须具有 `solaris.role.delegate` 授权。

  `-l` *admin-role* 的口令输入提示。

  `--` 验证选项和子命令选项之间必需的分隔符。

  `-n` *username* 指定了 *rolename* 的用户的名称。

  `-a` *rolename* 将指定给 *username* 的角色的名称。您可以指定多个 `-a` *rolename* 选项。

  有关更多命令选项，请参见 `smuser`(1M) 手册页。

示例 9–16 通过命令行修改本地用户的 RBAC 属性

在本示例中，用户 `jdoe` 现在可以承担系统管理员的角色。

```
$ usermod -R sysadmin jdoe
```

示例 9–17 使用 smuser 命令修改用户的 RBAC 属性

在本示例中，为用户 jdoe 指定了两个角色：系统管理员和操作员。由于该用户和这两个角色是在本地定义的，因此不必使用 -D 选项。

```
$ /usr/sadm/bin/smuser -r primaryadm -l <Type primaryadm password> \
modify -- -n jdoe -a sysadmin -a operadm
```

在以下示例中，该用户是在 NIS 名称服务中定义的。因此，需要使用 -D 选项。两个角色是在名称服务中定义的。角色 root 是在本地定义的。

```
$ /usr/sadm/bin/smuser -D nis:/examplehost/example.domain \
-r primaryadm -l <Type primaryadm password> \
modify -- -n jdoe -a sysadmin -a operadm -a root
```

▼ 如何为传统应用程序添加 RBAC 属性

传统应用程序是一个命令或一组命令。先针对权限配置文件中的每个命令设置安全性属性。然后，在角色中包括该权限配置文件。承担该角色的用户便可以运行具有安全性属性的传统应用程序。

要将传统应用程序添加到 Solaris Management Console，请参见《System Administration Guide: Basic Administration》中的“Adding Tools to the Solaris Management Console”。

开始之前 要更改权限配置文件中命令的安全性属性，您必须已承担主管理员角色或已切换到超级用户。

1. **使用 Solaris Management Console 中的“ 用户 ”工具。**

   要启动该控制台，请参见第 201 页中的 “如何在 Solaris Management Console 中承担角色 ”。请按照左侧窗格中的说明在“权限”中修改权限配置文件。有关更详细的信息，请参见联机帮助。

2. **向实现传统应用程序的命令添加安全性属性。**

   向传统应用程序添加安全性属性的方式与为任何命令添加安全性属性的方式相同。您必须将具有安全性属性的命令添加到权限配置文件。对于传统命令，请为命令指定 euid=0 或 uid=0 安全性属性。有关该过程的详细信息，请参见第 205 页中的 “如何创建或更改权限配置文件”。

3. **将传统应用程序添加到权限配置文件后，在角色的配置文件列表中包括该权限配置文件。**

   要向角色添加权限配置文件，请参见第 203 页中的 “如何更改角色的属性”。

示例 9–18 向脚本中的命令添加安全性属性

如果某个脚本中的命令需要设置 setuid 位或 setgid 位才能成功运行，则将必须在权限配置文件中为该脚本的可执行脚本**和**命令添加安全性属性。然后，在角色中包括该权限配置文件，并将该角色指定给用户。当用户承担该角色并执行该脚本时，此命令便会以安全性属性运行。

要将安全性属性添加到命令或 shell 脚本，请参见第 205 页中的 “如何创建或更改权限配置文件” 。

示例 9–19 检查脚本或程序中的授权

要检查脚本授权，需要添加一项基于 auths 命令的测试。有关此命令的详细信息，请参见 auths(1) 手册页。

例如，以下行会测试用户是否具有作为 $1 参数提供的授权：

```
if [ ‘/usr/bin/auths|/usr/xpg4/bin/grep $1‘ ]; then

        echo Auth granted

else

        echo Auth denied

fi
```

要使测试更加完整，应在测试中包括检查其他使用通配符的授权的逻辑。例如，要测试用户是否具有 solaris.admin.usermgr.write 授权，需要检查以下字符串：

- solaris.admin.usermgr.write
- solaris.admin.usermgr.*
- solaris.admin.*
- solaris.*

如果您要编写程序，请使用函数 getauthattr() 对授权进行测试。

◆ ◆ ◆ 第 10 章

基于角色的访问控制（参考）

本章介绍了有关 RBAC 的参考资料。以下是本章中参考信息的列表：

- 第 211 页中的 “权限配置文件的内容”
- 第 215 页中的 “授权命名和委托”
- 第 216 页中的 “支持 RBAC 的数据库”
- 第 223 页中的 “RBAC 命令”

有关使用 RBAC 的信息，请参见第 9 章。有关概述信息，请参见第 169 页中的 “基于角色的访问控制（概述）”。

权限配置文件的内容

本节介绍了一些典型的权限配置文件。权限配置文件可以包括授权、带有安全属性的命令，以及补充权限配置文件。权限配置文件根据功能的强弱从高到低列出。有关如何将权限配置文件分配给站点中角色的建议，请参见第 187 页中的 “如何规划 RBAC 实现”。

- **主管理员权限配置文件** – 在一个配置文件中提供了超级用户的功能。
- **系统管理员权限配置文件** – 提供可以执行与安全性无关的大多数任务的配置文件。此配置文件包括一些其他配置文件以创建功能强大的角色。
- **操作员权限配置文件** – 提供有限的功能以管理文件和脱机介质。此配置文件包括用于创建简单角色的补充权限配置文件。
- **打印机管理权限配置文件** – 提供有限数量的命令和授权以处理打印。此配置文件是涉及单个管理区域的数个配置文件之一。
- **基本 Solaris 用户权限配置文件** – 使用此配置文件，用户可以在安全策略的限定范围内使用系统。缺省情况下，会在 `policy.conf` 文件中列出此配置文件。
- **所有权限配置文件** – 为角色提供访问不具有安全属性的命令的权限。

每个权限配置文件都具有关联的帮助文件。帮助文件以 HTML 形式提供，是可自定义的。这些文件驻留在 `/usr/lib/help/auths/locale/C` 目录中。

主管理员权限配置文件

将主管理员权限配置文件指定给系统上功能最强的角色。拥有主管理员权限配置文件的角色具有超级用户功能。

- `solaris.*` 授权有效地指定由 Solaris 软件提供的所有授权。
- 通过 `solaris.grant` 授权，角色可以为任何权限配置文件、角色或用户指定任何授权。
- 命令指定 `*:uid=0;gid=0` 提供了使用 `UID=0` 和 `GID=0` 运行任何命令的功能。

如有必要，可以为站点自定义帮助文件 `RtPriAdmin.html`。帮助文件存储在 `/usr/lib/help/auths/locale/C` 目录中。

另请注意，如果主管理员权限配置文件与站点的安全策略不一致，则可以修改配置文件或者根本就不指定配置文件。但是，需要在一个或多个其他权限配置文件中处理主管理员权限配置文件中的安全功能。然后将这些权限配置文件指定给角色。

表 10–1 主管理员权限配置文件的内容

目的	内容
执行所有管理任务	**命令**： `*:uid=0;gid=0` **授权**： `solaris.*`、`solaris.grant` **帮助文件**： `RtPriAdmin.html`

系统管理员权限配置文件

系统管理员权限配置文件适用于系统管理员角色。由于系统管理员不具有主管理员具有的广泛功能，因此不使用通配符。相反，此配置文件是一组不涉及安全性的独立的补充管理权限配置文件。显示其中一个补充权限配置文件中带有安全属性的命令。

请注意，将在补充权限配置文件列表的末尾指定所有权限配置文件。

表 10–2 系统管理员权限配置文件的内容

目的	内容
执行大多数非安全性的管理任务	**补充权限配置文件**：审计查看、打印机管理、计时程序管理、设备管理、文件系统管理、邮件管理、维护和修复、介质备份、介质恢复、名称服务管理、网络管理、对象访问管理、进程管理、软件安装、用户管理、所有 **帮助文件**： `RtSysAdmin.html`

表 10–2 系统管理员权限配置文件的内容 （续）

目的	内容
补充配置文件之一中的命令	**对象访问管理权限配置文件**，solaris 策略：/usr/bin/chgrp:privs=file_chown、/usr/bin/chmod:privs=file_chown、/usr/bin/chown:privs=file_chown、/usr/bin/setfacl:privs=file_chown suser 策略：/usr/bin/chgrp:euid=0、/usr/bin/chmod:euid=0、/usr/bin/chown:euid=0、 /usr/bin/getfacl:euid=0、/usr/bin/setfacl:euid=0

操作员权限配置文件

操作员权限配置文件是一个功能较弱的配置文件，提供执行备份和打印机维护的功能。恢复文件的功能与安全性相关更密切。因此，在此配置文件中，缺省值将不包括恢复文件的功能。

表 10–3 操作员权限配置文件的内容

目的	内容
执行简单的管理任务	**补充权限配置文件**：打印机管理、介质备份、所有 **帮助文件**：RtOperator.html

打印机管理权限配置文件

打印机管理是适用于特定任务区域的典型权限配置文件。此配置文件包括授权和命令。下表显示了部分命令列表。

表 10–4 打印机管理权限配置文件的内容

目的	内容
管理打印机、守护进程和假脱机	**授权**：`solaris.admin.printer.delete`、`solaris.admin.printer.modify`、`solaris.admin.printer.read` **命令**：`/usr/bin/cancel:euid=lp;uid=lp`，`/usr/bin/lpset:egid=14`，`/usr/bin/lpstat:euid=0`，`/usr/lib/lp/local/lpadmin:uid=lp;gid=8`，`/usr/lib/lp/lpsched:uid=0`，`/usr/sbin/lpadmin:egid=14;uid=lp;gid=8`，`/usr/sbin/lpfilter:euid=lp;uid=lp`，`/usr/ucb/lprm:euid=0` **帮助文件**：`RtPrntMngmnt.html`

基本 Solaris 用户权限配置文件

缺省情况下，会通过 `policy.conf` 文件将基本 Solaris 用户权限配置文件自动指定给所有用户。此配置文件提供了正常操作中有用的基本授权。请注意，基本 Solaris 用户权限配置文件提供的便利必须与站点的安全要求平衡。需要更严格安全性的站点可能希望从 `policy.conf` 文件中删除此配置文件。

表 10–5 基本 Solaris 用户权限配置文件的内容

目的	内容
自动将权限指定给所有用户	**授权**：`solaris.profmgr.read`、`solaris.jobs.users`、`solaris.mail.mailq`、`solaris.admin.usermgr.read`、`solaris.admin.logsvc.read`、`solaris.admin.fsmgr.read`、`solaris.admin.serialmgr.read`、`solaris.admin.diskmgr.read`、`solaris.admin.procmgr.user`、`solaris.compsys.read`、`solaris.admin.printer.read`、`solaris.admin.prodreg.read`、`solaris.admin.dcmgr.read`、`solaris.snmp.read`、`solaris.project.read`、`solaris.admin.patchmg.read`、`solaris.network.hosts.read`、`solaris.compsys.read`、`solaris.admin.volmgr.read` **补充权限配置文件**：所有 **帮助文件**：`RtDefault.html`

所有权限配置文件

所有权限配置文件使用通配符包括所有命令。此配置文件提供了可访问未在其他权限配置文件中显式指定的所有命令的角色。如果没有所有权限配置文件或使用通配符的其他权限配置文件，则角色只能访问显式指定的命令。如此有限的命令集不是很实用。

所有权限配置文件（如果使用）应该是指定的最终权限配置文件。此最后一个位置可确保不会意外覆盖其他权限配置文件中的显式安全属性指定。

表 10–6 所有权限配置文件的内容

目的	内容
以用户或角色的身份执行任何命令	**命令**： * **帮助文件**： RtAll.html

权限配置文件的顺序

权限配置文件中的命令按顺序进行解释。第一次出现的命令版本是用于此角色或用户的命令的唯一版本。不同的权限配置文件可以包括相同命令。因此，配置文件列表中权限配置文件的顺序至关重要。应该首先列出具有最多功能的权限配置文件。

权限配置文件在 Solaris Management Console GUI 和 `prof_attr` 文件中列出。在 Solaris Management Console GUI 中，具有最多功能的权限配置文件应该是指定权限配置文件列表中最顶部的配置文件。在 `prof_attr` 文件中，具有最多功能的权限配置文件应该是补充配置文件列表中的第一个配置文件。此放置方法可确保带有安全属性的命令列在不带安全属性的相同命令之前。

查看权限配置文件的内容

Solaris Management Console 权限工具提供了一种检查权限配置文件内容的方法。

`prof_attr` 和 `exec_attr` 文件提供了划分更细的视图。`prof_attr` 文件包含在系统上定义的每个权限配置文件的名称。此文件还包括每个配置文件的授权和补充权限配置文件。`exec_attr` 文件包含权限配置文件的名称及其带有安全属性的命令。

授权命名和委托

RBAC **授权**是可以授予角色或用户的独立权限。在用户获取对应用程序或应用程序内特定操作的访问权限之前，将通过 RBAC 兼容应用程序检查授权。此检查替换了常规 UNIX 应用程序中对 `UID=0` 的测试。

授权命名约定

授权具有在内部以及文件中使用的名称。例如，`solaris.admin.usermgr.pswd` 是一个授权的名称。授权具有简短说明，此说明出现在图形用户界面 (graphical user interface, GUI) 中。例如，`Change Passwords` 是 `solaris.admin.usermgr.pswd` 授权的说明。

根据约定，授权名称由顺序颠倒过来的供应商 Internet 名称、主题区域（任何子区域）和功能组成。授权名称的各个部分以点分隔。com.xyzcorp.device.access 便是一个示例。此约定的例外是 Sun Microsystems, Inc. 的授权，它使用前缀 solaris 代替 Internet 名称。使用命名约定，管理员可以用分层方式应用授权。通配符 (*) 可以表示点右侧的所有字符串。

授权粒度示例

可将以下情况视为如何使用授权的示例：操作员角色中的用户可能限于 solaris.admin.usermgr.read 授权，此授权只提供可对用户配置文件的读取访问，不提供写入访问。系统管理员角色自然地具有 solaris.admin.usermgr.read 和 solaris.admin.usermgr.write 授权，以对用户文件进行更改。但是，如果没有 solaris.admin.usermgr.pswd 授权，系统管理员就不能更改口令。主管理员具有所有这三个授权。

需要 solaris.admin.usermgr.pswd 授权才能在 Solaris Management Console 用户工具中更改口令。使用 smuser、smmultiuser 和 smrole 命令中的口令修改选项时也需要此授权。

授权中的授权委托

使用以后缀 grant 结束的授权，用户或角色可将以相同前缀开头的任何指定授权委托给其他用户。

例如，具有授权 solaris.admin.usermgr.grant 和 solaris.admin.usermgr.read 的角色可将 solaris.admin.usermgr.read 授权委托给其他用户。具有 solaris.admin.usermgr.grant 和 solaris.admin.usermgr.* 授权的角色可将具有 solaris.admin.usermgr 前缀的任何授权委托给其他用户。

支持 RBAC 的数据库

以下四个数据库存储 RBAC 元素的数据：

- **扩展用户属性数据库** (user_attr) – 将用户与具有授权和权限的角色相关联
- **权限配置文件属性数据库** (prof_attr) – 定义权限配置文件，列出配置文件的指定授权，并标识关联的帮助文件
- **授权属性数据库** (auth_attr) – 定义授权及其属性，并标识关联的帮助文件
- **执行属性数据库** (exec_attr) – 标识指定给特定权限配置文件的带有安全属性的命令

policy.conf 数据库包含应用于所有用户的授权、权限和权限配置文件。有关更多信息，请参见第 222 页中的 “policy.conf 文件”。

RBAC 数据库关系

每个 RBAC 数据库都使用 *key=value* 语法存储属性。此方法可以适应将来的数据库扩展。此外，使用此方法，系统可以在遇到其策略未知的关键字时继续运行。*key=value* 内容将文件链接起来。四个数据库中的以下链接项说明了 RBAC 数据库协同工作的方式。

示例 10–1 显示 RBAC 数据库连接

在以下示例中，通过为用户 jdoe 指定角色 filemgr，使此用户获取文件系统管理配置文件的功能。

1. 在 user_attr 数据库的 jdoe 用户项中为用户 jdoe 指定角色 filemgr。

```
# user_attr - user definition

jdoe::::type=normal;roles=filemgr
```

2. 在 user_attr 数据库的角色项中为角色 filemgr 指定权限配置文件文件系统管理。

```
# user_attr - role definition

filemgr::::profiles=File System Management;type=role
```

   用户和角色在本地系统上的 passwd 和 shadow 文件中（或者在分布式名称服务的等效数据库中）唯一定义。

3. 文件系统管理权限配置文件在 prof_attr 数据库中定义。此数据库还为文件系统管理项指定了三组授权。

```
# prof_attr - rights profile definitions and assigned authorizations

File System Management:::Manage, mount, share file systems:

help=RtFileSysMngmnt.html;

auths=solaris.admin.fsmgr.*,solaris.admin.diskmgr.*,solaris.admin.volmgr.*
```

4. 这些授权在 auth_attr 数据库中定义。

```
# auth_attr - authorization definitions

solaris.admin.fsmgr.:::Mounts and Shares::help=AuthFsmgrHeader.html

solaris.admin.fsmgr.read:::View Mounts and Shares::help=AuthFsmgrRead.html

solaris.admin.fsmgr.write:::Mount and Share Files::help=AuthFsmgrWrite.html
```

5. 在 exec_attr 数据库中为文件系统管理权限配置文件指定带有安全属性的命令。

```
# exec_attr - rights profile names with secured commands
```

示例 10–1 显示 RBAC 数据库连接 （续）

```
File System Management:suser:cmd:::/usr/sbin/mount:uid=0

File System Management:suser:cmd:::/usr/sbin/dfshares:euid=0

...

File System Management:solaris:cmd:::/usr/sbin/mount:privs=sys_mount

...
```

RBAC 数据库和名称服务

RBAC 数据库的名称服务范围只能应用于本地主机。此范围还可以包括由 NIS、NIS+ 或 LDAP 之类的名称服务提供服务的所有主机。在 /etc/nsswitch.conf 文件中为每个数据库设置具有优先级的名称服务。

- auth_attr 项－设置 auth_attr 数据库的名称服务优先级。
- passwd 项－设置 user_attr 数据库的名称服务优先级。
- prof_attr 项－设置 prof_attr 数据库的名称服务优先级。此外，还设置 exec_attr 数据库的名称服务优先级。

  例如，如果将带有安全属性的命令指定给存在于两个名称服务范围中的权限配置文件，则只会使用第一个名称服务范围中的项。

user_attr 数据库

user_attr 数据库包含补充 passwd 和 shadow 数据库的用户和角色信息。user_attr 数据库包含授权、权限配置文件和指定角色之类的扩展用户属性。 user_attr 数据库中的字段以冒号分隔，如下所示：

```
user:qualifier:res1:res2:attr
```

这些字段具有以下含义：

user
: passwd 数据库中指定的用户或角色的名称。

qualifier:res1:res2
: 保留这些字段供将来使用。

attr
: 以分号 (;) 分隔的关键字-值对的可选列表，用于说明将在用户运行命令时应用的安全属性。四个有效关键字为 type、auths、profiles 和 roles。

  - 如果此帐户属于普通用户，则可以将 type 关键字设置为 normal。如果此帐户属于角色，则 type 是 role。

- auths 关键字指定从 auth_attr 数据库中定义的名称中选择的以逗号分隔的授权名称列表。授权名称可以包括星号 (*) 字符作为通配符。例如，solaris.device.* 表示所有的 Solaris 设备授权。
- profiles 关键字指定 prof_attr 数据库中排序的逗号分隔权限配置文件名称列表。权限配置文件的排序方式与 UNIX 搜索路径的排序方式类似。列表中包含要执行的命令的第一个配置文件定义将应用于命令的安全属性（如果存在）。
- 可以通过以逗号分隔的角色名称列表将 roles 关键字指定给用户。请注意，角色在同一 user_attr 数据库中定义。通过将类型值设置为 role 来表示角色。不能将角色指定给其他角色。

以下示例演示了如何在典型的 user_attr 数据库中定义操作员角色。此示例显示了如何将角色指定给用户 jdoe。通过 type 关键字区分角色和用户。

```
% grep operator /etc/user_attr

jdoe::::type=normal;roles=operator

operator::::profiles=Operator;type=role
```

auth_attr 数据库

所有授权都存储在 auth_attr 数据库中。可以将授权指定给用户、角色或权限配置文件。首选方法是将授权放置在权限配置文件中，将配置文件包括在角色的配置文件列表中，然后将角色指定给用户。

auth_attr 数据库中的字段以冒号分隔，如下所示：

```
authname:res1:res2:short_desc:long_desc:attr
```

这些字段具有以下含义：

authname
: 用于以 *prefix*.[*suffix*] 格式标识授权的唯一字符串。Solaris OS 的授权使用 solaris 作为前缀。所有其他授权应使用与创建授权的组织的 Internet 域名顺序相反的名称开头的前缀（例如，com.xyzcompany）。后缀指示要授权的内容，通常是功能区域和操作。

  当 authname 由前缀和功能区域组成并以句点结束时，authname 将用作应用程序 GUI 中使用的标题。由两部分组成的 authname 不是实际授权。authname 的值 solaris.printmgr. 便是一个标题示例。

  当 authname 以单字 "grant" 结束时，authname 将用作授予授权。使用授予授权，用户可以将具有相同前缀和功能区域的授权委托给其他用户。authname 的值 solaris.printmgr.grant 便是一个授予授权示例。solaris.printmgr.grant 授予用户将 solaris.printmgr.admin 和 solaris.printmgr.nobanner 之类的授权委托给其他用户的权限。

res1:res2
: 保留以供将来使用。

short_desc　授权的短名称。此短名称适于在用户界面中（如 GUI 中的滚动列表中）显示。

long_desc　详细说明。此字段标识授权的目的、使用授权的应用程序以及可能使用授权的用户类型。可以在应用程序的帮助文本中显示详细说明。

attr　以分号 (;) 分隔的关键字-值对的可选列表，用于说明授权属性。可以指定零个或多个关键字。

关键字 help 标识 HTML 形式的帮助文件。可以通过 /usr/lib/help/auths/locale/C 目录中的 index.html 文件访问帮助文件。

以下示例显示了带有一些典型值的 auth_attr 数据库：

```
% grep printer /etc/security/auth_attr

solaris.admin.printer.:::Printer Information::help=AuthPrinterHeader.html

solaris.admin.printer.delete:::Delete Printer Information::help=AuthPrinterDelete.html

solaris.admin.printer.modify:::Update Printer Information::help=AuthPrinterModify.html

solaris.admin.printer.read:::View Printer Information::help=AuthPrinterRead.html
```

请注意，solaris.admin.printer. 被定义为标题，这是因为授权名称以点 (.) 结束。GUI 使用标题组织授权系列。

prof_attr 数据库

prof_attr 数据库存储名称、说明、帮助文件的位置以及指定给权限配置文件的授权。指定给权限配置文件的命令和安全属性存储在 exec_attr 数据库中。有关更多信息，请参见第 221 页中的 “exec_attr 数据库”。prof_attr 数据库中的字段以冒号分隔，如下所示：

```
profname:res1:res2:desc:attr
```

这些字段具有以下含义：

profname　权限配置文件的名称。权限配置文件名称区分大小写。user_attr 数据库也使用此名称指示指定给角色和用户的配置文件。

res1:res2　保留以供将来使用。

desc　详细说明。此字段应介绍权限配置文件的目的，包括有兴趣使用此配置文件的用户类型。详细说明适于在应用程序的帮助文本中显示。

attr　以分号 (;) 分隔的关键字-值对的可选列表，用于说明在执行时应用于对象的安全属性。可以指定零个或多个关键字。两个有效关键字为 help 和 auths。

关键字 help 标识 HTML 形式的帮助文件。可以通过 /usr/lib/help/auths/locale/C 目录中的 index.html 文件访问帮助文件。

关键字 auths 指定从 auth_attr 数据库中定义的那些名称中选择的以逗号分隔的授权名称列表。可以使用星号 (*) 字符作为通配符来指定授权名称。

以下示例显示了两个典型的 prof_attr 数据库项。请注意，打印机管理权限配置文件是操作员权限配置文件的补充权限配置文件。根据显示的需要，另起一行来显示此示例。

```
% grep 'Printer Management' /etc/security/prof_attr

Printer Management:::                    权限配置文件的名称

Manage printers, daemons, spooling:      说明

help=RtPrntAdmin.html;                   帮助文件

auths=solaris.admin.printer.read,        授权

solaris.admin.printer.modify,solaris.admin.printer.delete

...

Operator:::                              权限配置文件的名称

Can perform simple administrative tasks:     说明

profiles=Printer Management,             补充权限配置文件

Media Backup,All;

help=RtOperator.html                     帮助文件
```

exec_attr 数据库

exec_attr 数据库定义需要安全属性才能成功运行的命令。这些命令是权限配置文件的一部分。具有安全属性的命令可以由为其指定了此配置文件的角色运行。

exec_attr 数据库中的字段以冒号分隔，如下所示：

```
name:policy:type:res1:res2:id:attr
```

这些字段具有以下含义：

profname　　权限配置文件的名称。权限配置文件名称区分大小写。该名称指的是 prof_attr 数据库中的配置文件。

policy　与此项相关联的安全策略。目前，suser 和 solaris 是有效项。solaris 策略可识别权限，而 suser 策略则不能。

type　指定的实体类型。目前，唯一有效的实体类型是 cmd（命令）。

res1:res2　保留以供将来使用。

id　标识实体的字符串。命令应该具有全路径或带有通配符 (*) 的路径。要指定参数，请编写具有这些参数的脚本并使 id 指向此脚本。

attr　以分号 (;) 分隔的关键字-值对的可选列表，用于说明将在执行时应用于实体的安全属性。可以指定零个或多个关键字。有效关键字的列表取决于强制执行的策略。

对于 suser 策略，四个有效关键字为 euid、uid、egid 和 gid。

- euid 和 uid 关键字包含单个用户名或数字用户 ID (user ID, UID)。通过 euid 指定的命令使用提供的 UID 运行，这与在可执行文件上设置 setuid 位类似。通过 uid 指定的命令使用实际 UID 和有效 UID 运行。
- egid 和 gid 关键字包含单个组名或数字组 ID (group ID, GID)。通过 egid 指定的命令使用提供的 GID 运行，这与在可执行文件上设置 setgid 位类似。通过 gid 指定的命令使用实际 GID 和有效 GID 运行。

对于 solaris 策略，有效关键字为 privs。值由以逗号分隔的权限列表组成。

以下示例显示了 exec_attr 数据库中的一些典型值：

```
% grep 'File System Management' /etc/security/exec_attr

File System Management:suser:cmd:::/usr/sbin/ff:euid=0

File System Management:solaris:cmd:::/usr/sbin/mount:privs=sys_mount

...
```

policy.conf 文件

policy.conf 文件提供了向所有用户授予特定权限配置文件、特定授权和特定权限的方法。文件中的相关项由 *key=value* 对组成：

- AUTHS_GRANTED=*authorizations*－指一个或多个授权。
- PROFS_GRANTED=*rights profiles*－指一个或多个权限配置文件。
- PRIV_DEFAULT=*privileges*－指一个或多个权限。
- PRIV_LIMIT=*privileges*－指所有权限。

以下示例显示了 policy.conf 数据库中的一些典型值：

```
# grep AUTHS /etc/security/policy

AUTHS_GRANTED=solaris.device.cdrw


# grep PROFS /etc/security/policy

PROFS_GRANTED=Basic Solaris User


# grep PRIV /etc/security/policy


#PRIV_DEFAULT=basic

#PRIV_LIMIT=all
```

有关权限的更多信息，请参见第 177 页中的 “权限（概述）”。

RBAC 命令

本节列出了用于管理 RBAC 的命令，还提供了一个命令表，其中命令的访问可以由授权控制。

管理 RBAC 的命令

虽然可以手动编辑本地 RBAC 数据库，但是强烈建议不要进行此类编辑。以下命令可用于管理对具有 RBAC 的任务进行访问。

表 10–7 RBAC 管理命令

命令的手册页	说明
auths(1)	显示用户的授权。
makedbm(1M)	生成 dbm 文件。
nscd(1M)	名称服务高速缓存守护进程，适用于高速缓存 user_attr、prof_attr 和 exec_attr 数据库。使用 svcadm 命令重新启动守护进程。
pam_roles(5)	PAM 的角色帐户管理模块。检查承担角色的授权。

表 10–7 RBAC 管理命令 （续）

命令的手册页	说明
pfexec(1)	由配置文件 shell 使用以执行在 exec_attr 数据库中指定的带有安全属性的命令。
policy.conf(4)	系统安全策略的配置文件。列出授予的授权、授予的权限和其他安全信息。
profiles(1)	显示指定用户的权限配置文件。
roles(1)	显示指定用户可以承担的角色。
roleadd(1M)	向本地系统中添加角色。
roledel(1M)	从本地系统中删除角色。
rolemod(1M)	在本地系统上修改角色的属性。
smattrpop(1M)	将源安全属性数据库合并到目标数据库。用于需要将本地数据库合并到名称服务的情况。还用于未提供转换脚本的升级。
smexec(1M)	管理 exec_attr 数据库中的项。要求验证。
smmultiuser(1M)	管理对用户帐户的批量操作。要求验证。
smprofile(1M)	管理 prof_attr 和 exec_attr 数据库中的权限配置文件。要求验证。
smrole(1M)	管理角色帐户中的角色和用户。要求验证。
smuser(1M)	管理用户项。要求验证。
useradd(1M)	向系统中添加用户帐户。-P 选项将角色指定给用户帐户。
userdel(1M)	从系统中删除用户的登录。
usermod(1M)	修改系统上的用户帐户属性。

要求授权的命令

下表提供了在 Solaris 系统上如何使用授权限制命令选项的示例。有关授权的更多介绍，请参见第 215 页中的 “授权命名和委托”。

表 10–8 命令和关联的授权

命令的手册页	授权要求
at(1)	所有选项所需的 solaris.jobs.user（at.allow 和 at.deny 文件都不存在时）
atq(1)	所有选项所需的 solaris.jobs.admin
cdrw(1)	所有选项所需的 solaris.device.cdrw，缺省情况下在 policy.conf 文件中授予

表 10–8 命令和关联的授权 （续）

命令的手册页	授权要求
crontab(1)	选项提交作业所需的 solaris.jobs.user（crontab.allow 和 crontab.deny 文件都不存在时）
	选项列出或修改其他用户的 crontab 文件所需的 solaris.jobs.admin
allocate(1)	分配设备所需的 solaris.device.allocate（或在 device_allocate 文件中指定的其他授权）
	将设备分配给其他用户（-F 选项）所需的 solaris.device.revoke（或在 device_allocate 文件中指定的其他授权）
deallocate(1)	解除其他用户的设备分配所需的 solaris.device.allocate（或在 device_allocate 文件中指定的其他授权）
	强制解除指定设备（-F 选项）或所有设备的分配（-I 选项）所需的 solaris.device.revoke（或在 device_allocate 中指定的其他授权）
list_devices(1)	列出其他用户的设备（-U 选项）所需的 solaris.device.revoke
sendmail(1M)	访问邮件子系统功能所需的 solaris.mail；查看邮件队列所需的 solaris.mail.mailq

◆ ◆ ◆ 第 11 章

11 权限（任务）

本章提供了在系统上管理和使用权限的逐步说明。以下是本章中信息的列表：

- 第 227 页中的 “管理和使用权限（任务列表）”
- 第 227 页中的 “管理权限（任务列表）”
- 第 236 页中的 “确定权限（任务列表）”

有关权限的概述，请参见第 177 页中的 “权限（概述）” 。有关参考信息，请参见第 12 章。

管理和使用权限（任务列表）

以下任务列表介绍了管理权限和使用权限的任务列表。

任务	说明	参考
在站点使用权限	涉及指定、删除、添加和调试权限的使用。	第 227 页中的 “管理权限（任务列表）”
运行命令时使用权限	涉及使用已指定给您的权限。	第 236 页中的 “确定权限（任务列表）”

管理权限（任务列表）

以下任务列表介绍了查看权限、指定权限以及运行包含特权命令的脚本的过程。

任务	说明	参考
确定进程中的权限	列出进程的有效权限集、可继承权限集、允许权限集和限制权限集。	第 228 页中的 “如何确定进程的权限”
确定进程缺少的权限	列出失败进程成功运行所需的权限。	第 230 页中的 “如何确定程序所需的权限”
为命令添加权限	为权限配置文件的命令添加权限。可以将权限配置文件指定给用户或角色。然后，用户可以在配置文件 shell 中运行具有指定权限的命令。	第 232 页中的 “如何为命令添加权限”
为用户指定权限	扩展用户或角色的可继承权限集。使用此过程时应谨慎。	第 233 页中的 “如何将权限指定给用户或角色”
限制用户的权限	限制用户的基本权限集。使用此过程时应谨慎。	第 234 页中的 “如何限制用户或角色的权限”
运行特权 shell 脚本	为 shell 脚本和 shell 脚本中的命令添加权限。然后，在配置文件 shell 中运行此脚本。	第 235 页中的 “如何运行具有特权命令的 Shell 脚本”

管理权限

管理用户和角色权限的最安全的方法是将权限使用限制在权限配置文件内的命令中。然后，此权限配置文件就包括在某个角色中。此角色会指定给某个用户。当此用户承担指定的角色时，特权命令便可在配置文件 shell 中运行。以下过程显示了如何指定权限、删除权限以及调试权限的使用。

▼ 如何确定进程的权限

此过程说明如何确定可用于进程的权限。列出内容不包括已经指定给特定命令的权限。

◗ **列出可用于 shell 进程的权限。**

```
% ppriv pid

$ ppriv -v pid
```

pid 进程号。使用双美元符号 (`$$`) 将父 shell 的进程号传递到命令。

`-v` 提供权限名称的详细列表。

示例 11–1 确定当前 shell 中的权限

在以下示例中，列出了用户 shell 进程的父进程中的权限。在下面第二个示例中，列出了权限的全名。输出中的单个字母指代以下权限集：

`E`
有效权限集。

I
   可继承权限集。

P
   允许权限集。

L
   限制权限集。

```
% ppriv $$

1200:   -csh

flags = <none>

        E: basic

        I: basic

        P: basic

        L: all

% ppriv -v $$

1200:   -csh

flags = <none>

        E: file_link_any,proc_exec,proc_fork,proc_info,proc_session

        I: file_link_any,proc_exec,proc_fork,proc_info,proc_session

        P: file_link_any,proc_exec,proc_fork,proc_info,proc_session

        L: cpc_cpu,dtrace_kernel,dtrace_proc,dtrace_user,...,sys_time
```

示例 11–2 确定可承担的角色的权限

角色使用管理 shell 或配置文件 shell。必须承担角色并使用此角色的 shell 列出已直接指定给此角色的权限。在以下示例中，角色 `sysadmin` 不具有直接指定的权限。

```
% su sysadmin

Password:       <键入 sysadmin 的口令>

$ /usr/ucb/whoami
```

```
sysadmin

$ ppriv -v $$

1400:   pfksh

flags = <none>

        E: file_link_any,proc_exec,proc_fork,proc_info,proc_session

        I: file_link_any,proc_exec,proc_fork,proc_info,proc_session

        P: file_link_any,proc_exec,proc_fork,proc_info,proc_session

        L: cpc_cpu,dtrace_kernel,dtrace_proc,dtrace_user,...,sys_time
```

▼ 如何确定程序所需的权限

此过程确定成功运行命令或进程所需的权限。

开始之前 命令或进程必须已经失败，才执行此过程。

1 键入失败的命令作为 ppriv 调试命令的参数。

```
% ppriv -eD touch /etc/acct/yearly

touch[11365]: missing privilege "file_dac_write"

     (euid = 130, syscall = 224) needed at ufs_direnter_cm+0x27c

touch: /etc/acct/yearly cannot create
```

2 通过在 /etc/name_to_sysnum 文件中查找 syscall 编号来确定失败的系统调用。

```
% grep 224 /etc/name_to_sysnum

creat64                 224
```

示例 11–3 使用 truss 命令检查权限使用

truss 命令可以在常规 shell 中调试权限的使用。例如，以下命令调试失败的 touch 进程：

```
% truss -t creat touch /etc/acct/yearly

creat64("/etc/acct/yearly", 0666)

                        Err#13 EACCES [file_dac_write]
```

```
touch: /etc/acct/yearly cannot create
```

扩展的 /proc 接口在 truss 输出中的错误代码后面报告缺少的权限。

示例 11–4 使用 ppriv 命令检查配置文件 Shell 中的权限使用

ppriv 命令可以在配置文件 shell 中调试权限的使用。如果将权限配置文件指定给用户，并且此权限配置文件包括具有权限的命令，则必须在配置文件 shell 中键入这些命令。在常规 shell 中键入特权命令时，这些命令执行时不使用特权。

在此示例中，jdoe 用户可以承担角色 objadmin。objadmin 角色拥有对象访问管理配置文件。使用此权限配置文件，objadmin 角色可以更改不属于 objadmin 的文件的权限。

在以下摘录中，jdoe 无法更改 useful.script 文件的权限：

```
jdoe% ls -l useful.script

-rw-r--r--  1 aloe  staff  2303 Mar 11 05:29 useful.script

jdoe% chown objadmin useful.script

chown: useful.script: Not owner

jdoe% ppriv -eD chown objadmin useful.script

chown[11444]: missing privilege "file_chown"

            (euid = 130, syscall = 16) needed at ufs_setattr+0x258

chown: useful.script: Not owner
```

当 jdoe 承担 objadmin 角色时，更改了该文件的权限：

```
jdoe% su objadmin

Password:      <键入 objadmin 的口令>

$ ls -l useful.script

-rw-r--r--  1 aloe  staff  2303 Mar 11 05:29 useful.script

$ chown objadmin useful.script

$ ls -l useful.script

-rw-r--r--  1 objadmin  staff  2303 Mar 11 05:29 useful.script
```

```
$ chgrp admin useful.script

$ ls -l objadmin.script

-rw-r--r--  1 objadmin  admin  2303 Mar 11 05:31 useful.script
```

示例 11–5 更改 root 用户拥有的文件

此示例说明防止权限升级的方法。有关说明，请参见第 247 页中的 “防止权限升级”。此文件归 root 用户所有。由于权限较低的 objadmin 角色需要所有权限才能更改文件的拥有权，因此操作失败。

```
jdoe% su objadmin

Password:       <键入 objadmin 的口令>

$ cd /etc; ls -l system

-rw-r--r--  1 root  sys   1883 Mar 20 14:04 system

$ chown objadmin system

chown: system: Not owner

$ ppriv -eD chown objadmin system

chown[11481]: missing privilege "ALL"

     (euid = 101, syscall = 16) needed at ufs_setattr+0x258

chown: system: Not owner
```

▼ 如何为命令添加权限

应在将命令添加到权限配置文件时为此命令添加权限。使用这些权限，拥有权限配置文件的角色可以运行管理命令，但不会获取任何其他超级用户功能。

开始之前 命令或程序必须可识别权限。有关更全面的介绍，请参见第 181 页中的 “进程如何获取权限”。

1. **成为超级用户或承担等效角色。**

   角色包含授权和具有一定权限的命令。有关角色的更多信息，请参见第 186 页中的 “配置 RBAC（任务列表）”。

2. **打开 Solaris Management Console GUI。**

   有关说明，请参见第 201 页中的 “如何在 Solaris Management Console 中承担角色”。

3 **使用权限工具更新相应的配置文件。**

选择要包括的命令。对于每个包括的命令，添加此命令所需的权限。

注意 – 在权限配置文件中添加命令并向命令中添加权限后，这些命令在配置文件 shell 中运行时会使用这些权限执行。

配置文件的顺序很重要。配置文件 shell 使用帐户配置文件列表内的最早配置文件中指定的安全属性执行命令或操作。例如，如果 `chgrp` 命令位于具有权限的对象访问管理权限配置文件中，并且对象访问管理配置文件是包含 `chgrp` 命令的第一个配置文件，则 `chgrp` 命令执行时使用在对象访问管理配置文件中指定的权限。

▼ 如何将权限指定给用户或角色

您可能始终信任某些具有特定权限的用户。只有对系统影响非常小的仅有的几个特定权限才适合指定给用户。有关直接指定权限所涉及内容的介绍，请参见第 177 页中的 “直接指定安全属性时的安全注意事项” 。

用户 `jdoe` 可以通过以下过程使用高分辨率计时器。

1 **承担主管理员角色，或成为超级用户。**

主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2 **将影响高分辨率时间的权限添加到用户的初始可继承权限集。**

```
$ usermod -K defaultpriv=basic,proc_clock_highres jdoe
```

3 **读取产生的 `user_attr` 项。**

```
$ grep jdoe /etc/user_attr

jdoe::::type=normal;defaultpriv=basic,proc_clock_highres
```

示例 11–6 创建具有权限的角色来配置系统时间

此示例创建了一个角色，其唯一任务是处理系统上的时间。

```
$ /usr/sadm/bin/smrole -D nisplus:/examplehost/example.domain \

-r primaryadm -l <Type primaryadm password> \

add -- -n clockmgr \

-c "Role that sets system time" \
```

```
-F "Clock Manager" \

-s /bin/pfksh \

-u 108 \

-P <Type clockmgr password> \

-K defaultpriv=basic,proc_priocntl,sys_cpu_config,

proc_clock_highres,sys_time
```

-K 行换行以便显示。

如果角色在本地创建，则此角色 user_attr 项的显示可能与以下信息类似：

```
clockmgr:::Role that sets system time:

type=role;defaultpriv=basic,proc_priocntl,sys_cpu_config,

proc_clock_highres,sys_time
```

▼ 如何限制用户或角色的权限

可以通过减少基本集或减少限制集来限制用户或角色可用的权限。由于此类限制可能产生预料不到的副作用，因此不是非常必要时不要使用此方法限制用户权限。

注意 – 为某个用户修改了基本集或限制集时，应该彻底测试任何用户的功能。

- 当基本集少于缺省值时，可阻止用户使用此系统。
- 当限制集少于所有权限时，需要使用有效 UID=0 运行的进程可能会失败。

1. **确定用户基本集和限制集中的权限。**

   有关过程，请参见第 228 页中的 “如何确定进程的权限”。

2. **（可选的）从基本集中删除一项权限。**

   ```
   $ usermod -K defaultpriv=basic,!priv-name username
   ```

   通过删除 proc_session 权限，可以防止用户检查其当前会话以外的任何进程。通过删除 file_link_any 权限，可以防止用户生成指向不归其所有的文件的硬链接。

注意 – 请勿删除 proc_fork 或 proc_exec 权限。如果没有这些权限，用户将无法使用系统。事实上，只能从不对其他进程执行 fork() 或 exec() 操作的守护进程中删除这两个权限。

3 （可选的）从限制集中删除一项权限。

```
$ usermod -K limitpriv=all,!priv-name username
```

4 测试 *username* 的功能。

以 *username* 的身份登录，并尝试执行 *username* 必须在系统上执行的任务。

示例 11–7 从用户的限制集中删除权限

在以下示例中，防止所有源自 jdoe 初始登录的会话使用 sys_linkdir 权限。也就是说，即使在用户运行 su 命令之后，也不能生成指向目录的硬链接，并且也不能解除目录链接。

```
$ usermod -K limitpriv=all,!sys_linkdir jdoe

$ grep jdoe /etc/user_attr

jdoe::::type=normal;defaultpriv=basic;limitpriv=all,!sys_linkdir
```

示例 11–8 从用户的基本集中删除权限

在以下示例中，防止所有源自 jdoe 初始登录的会话使用 proc_session 权限。也就是说，即使在用户运行 su 命令之后，也不能检查此用户会话以外的任何进程。

```
$ usermod -K defaultpriv=basic,!proc_session jdoe

$ grep jdoe /etc/user_attr

jdoe::::type=normal;defaultpriv=basic,!proc_session;limitpriv=all
```

▼ 如何运行具有特权命令的 Shell 脚本

注 – 创建运行具有继承权限的命令的 shell 脚本时，相应的权限配置文件必须包含具有指定权限的命令。

1 以 /bin/pfsh 或任何其他配置文件 shell 作为此脚本的第一行。

```
#!/bin/pfsh

# Copyright (c) 2003 by Sun Microsystems, Inc.
```

2 确定脚本中的命令所需的权限。

```
% ppriv -eD script-full-path
```

3 **打开 Solaris Management Console GUI。**

有关说明，请参见第 201 页中的 “如何在 Solaris Management Console 中承担角色” 。选择可以创建权限配置文件的角色，如主管理员。

4 **使用权限工具创建或更新相应的配置文件。**

选择脚本，并在此权限配置文件中包括需要权限才能运行的 shell 脚本中的每个命令。对于每个包括的命令，添加此命令所需的权限。

注意 – 权限配置文件的顺序很重要。配置文件 shell 执行配置文件列表中最早的命令实例。例如，如果 `chgrp` 命令位于对象访问管理权限配置文件中，并且对象访问管理配置文件是包含 `chgrp` 命令的第一个配置文件，则执行 `chgrp` 命令时使用在对象访问管理配置文件中指定的权限。

5 **向角色中添加权限配置文件并将此角色指定给用户。**

为了执行此配置文件，用户会承担角色并在此角色的配置文件 shell 中运行脚本。

确定权限（任务列表）

以下任务列表介绍了有关使用已指定给您的权限的过程。

任务	说明	参考
以用户的身份在任何 shell 中查看权限	显示已直接指定给您的权限。由您运行的所有进程都将以这些权限运行。	第 236 页中的 “如何确定已直接指定给您的权限”
确定可以使用权限运行的命令	将权限指定给权限配置文件中的可执行文件后，必须在配置文件 shell 中键入此可执行文件。	第 238 页中的 “如何确定可以运行的特权命令”
确定角色可以使用权限运行的命令	承担此角色来确定其可以使用权限运行的命令。	第 240 页中的 “如何确定角色可以运行的特权命令”

确定已指定的权限

直接将权限指定给用户时，这些权限在每个 shell 中都有效。未直接将权限指定给用户时，则此用户必须打开一个配置文件 shell。例如，当具有指定权限的命令位于用户权限配置文件列表内的权限配置文件中时，此用户必须在配置文件 shell 中执行此命令。

▼ 如何确定已直接指定给您的权限

以下过程显示如何确定是否已直接为您指定权限。

注意 – 不当使用直接指定的权限可能导致无意的安全性破坏。有关介绍，请参见第 177 页中的 “直接指定安全属性时的安全注意事项” 。

1. **列出进程可以使用的权限。**

   有关过程，请参见第 228 页中的 “如何确定进程的权限” 。

2. **在任何 shell 中调用操作并运行命令。**

   有效集中列出的权限在整个会话中都有效。如果已为您直接指定了除基本集之外的权限，则会在有效集中列出这些权限。

示例 11–9 确定直接指定给您的权限

如果已经为您直接指定权限，则基本集包含的权限会多于缺省基本集。在此示例中，用户始终能够访问 `proc_clock_highres` 权限。

```
% /usr/ucb/whoami

jdoe

% ppriv -v $$

1800:   pfksh

flags = <none>

        E: file_link_any,...,proc_clock_highres,proc_session

        I: file_link_any,...,proc_clock_highres,proc_session

        P: file_link_any,...,proc_clock_highres,proc_session

        L: cpc_cpu,dtrace_kernel,dtrace_proc,dtrace_user,...,sys_time

% ppriv -vl proc_clock_highres

        Allows a process to use high resolution timers.
```

示例 11–10 确定直接指定给角色的权限

角色使用管理 shell 或配置文件 shell。承担角色的用户可以使用此角色的 shell 列出已直接指定给此角色的权限。在以下示例中，已经直接将处理日期和时间程序的权限指定给角色 `realtime`。

```
% su realtime
```

```
Password:      <键入 realtime 的口令>

$ /usr/ucb/whoami

realtime

$ ppriv -v $$

1600:   pfksh

flags = <none>

        E: file_link_any,...,proc_clock_highres,proc_session,sys_time

        I: file_link_any,...,proc_clock_highres,proc_session,sys_time

        P: file_link_any,...,proc_clock_highres,proc_session,sys_time

        L: cpc_cpu,dtrace_kernel,dtrace_proc,dtrace_user,...,sys_time
```

▼ 如何确定可以运行的特权命令

未直接将权限指定给用户时，此用户可通过权限配置文件获取特权命令的访问权限。必须在配置文件 shell 中执行权限配置文件中的命令。

开始之前 验证到 Solaris Management Console 的用户或角色必须具有 `solaris.admin.usermgr.read` 授权。基本 Solaris 用户权限配置文件包括此授权。

1 确定已指定给您的权限配置文件。

```
$ /usr/sadm/bin/smuser list -- -n username -l

Authenticating as user: admin

... Please enter a string value for: password ::

...

User name:      username

User ID (UID):  130

Primary group:  staff

Secondary groups:

Comment: object mgt jobs
```

```
Login Shell: /bin/sh

Home dir server: system

Home directory: /export/home/username

AutoHome setup: True

Mail server: system

Rights: Object Access Management

Assigned Roles:
```

2 **找到以 "Rights:" 开头的行。**

"Rights" 行列出已直接指定给您的权限配置文件的名称。

3 **在 exec_attr 数据库中查找权限配置文件的名称。**

```
$ cd /etc/security

$ grep "Object Access Management" exec_attr

Object Access Management:solaris:cmd:::/usr/bin/chgrp:privs=file_chown

Object Access Management:solaris:cmd:::/usr/bin/chown:privs=file_chown

Object Access Management:suser:cmd:::/usr/bin/chgrp:euid=0

Object Access Management:suser:cmd:::/usr/bin/chmod:euid=0

...
```

具有已添加权限的命令列在 `solaris` 策略项的末尾。

4 **在配置文件 shell 中键入需要权限的命令。**

在常规 shell 中键入这些命令时，它们不会使用权限运行，因而不会成功运行。

```
% pfsh

$
```

示例 11–11 在配置文件 Shell 中运行特权命令

在以下示例中，用户 `jdoe` 不能从其常规 shell 中更改文件的组权限。但是，在配置文件 shell 中键入命令时，`jdoe` 可以更改这些权限。

```
% whoami

jdoe

% ls -l useful.script

-rwxr-xr-- 1 nodoe eng 262 Apr 2 10:52 useful.script

chgrp staff useful.script

chgrp: useful.script: Not owner

% pfksh

$ /usr/ucb/whoami

jdoe

$ chgrp staff useful.script

$ chown jdoe useful.script

$ ls -l useful.script

-rwxr-xr-- 1 jdoe staff 262 Apr 2 10:53 useful.script
```

▼ 如何确定角色可以运行的特权命令

角色通过包含具有指定权限的命令的权限配置文件来获取特权命令的访问权限。为用户提供特权命令访问权限的最安全的方法是为用户指定一个角色。承担此角色之后，用户便可执行所有包括在此角色权限配置文件中的特权命令。

开始之前 验证到 Solaris Management Console 的用户或角色必须具有 `solaris.admin.usermgr.read` 授权。基本 Solaris 用户权限配置文件包括此授权。

1 确定可以承担的角色。

```
$ /usr/sadm/bin/smuser list -- -n username -l

Authenticating as user: primadmin

...

User name:      username

User ID (UID):  110
```

```
Primary group:  staff

Secondary groups:

Comment: Has admin roles

Login Shell: /bin/sh

...

Rights:

Assigned Roles: primadmin, admin
```

2 找到以 **"Assigned Roles:"** 开头的行。

"Assigned Roles" 行列出可以承担的角色。

3 确定角色之一拥有的权限配置文件。

```
$ /usr/sadm/bin/smuser list -- -n admin -l

Authenticating as user: primadmin

...

User name:      admin

User ID (UID):  101

Primary group:  sysadmin

Secondary groups:

Comment: system administrator

Login Shell: /bin/pfksh

...

Rights: System Administrator

Assigned Roles:
```

4 在 **"Rights:"** 行中找到该角色的权限配置文件的名称。

5. **在 prof_attr 数据库中查找权限配置文件。**

   由于系统管理员配置文件是配置文件的集合，因此需要在系统管理员配置文件中列出这些配置文件。

```
$ cd /etc/security

$ grep "System Administrator" prof_attr

System Administrator:::Can perform most non-security administrative

tasks:profiles=Audit Review,Printer Management,Cron Management,

Device Management,File System Management,Mail Management,Maintenance

and Repair,Media Backup,Media Restore,Name Service Management,Network

Management,Object Access Management,Process Management,Software

Installation,User Management,All;help=RtSysAdmin.html
```

6. **对于每个权限配置文件，在 exec_attr 数据库中查找与其相关的权限配置文件。**

   例如，Network Management 配置文件是 System Administrator 配置文件的补充配置文件。Network Management 配置文件包括一些特权命令。

```
$ cd /etc/security

$ grep "Network Management" exec_attr

Network Management:solaris:cmd:::/usr/sbin/ifconfig:privs=sys_net_config

Network Management:solaris:cmd:::/usr/sbin/route:privs=sys_net_config

...
```

   这些命令及其指定权限是 solaris 策略项的最后两个字段。可以在角色的配置文件 shell 中运行这些命令。

示例 11–12 在角色中运行特权命令

当用户承担角色时，其 shell 成为配置文件 shell。因此，将使用指定给命令的权限来执行这些命令。在以下示例中，admin 角色可以更改 useful.script 文件的权限。

```
% whoami

jdoe

% ls -l useful.script

-rwxr-xr-- 1 elsee eng 262 Apr 2 10:52 useful.script
```

```
chgrp admin useful.script
chgrp: useful.script: Not owner
% su admin
Password:      <键入 admin 的口令>
$ /usr/ucb/whoami
admin
$ chgrp admin useful.script
$ chown admin useful.script
$ ls -l useful.script
-rwxr-xr-- 1 admin admin 262 Apr 2 10:53 useful.script
```

◆ ◆ ◆ 第 12 章

12 权限（参考）

以下是本章中参考信息的列表：

- 第 245 页中的 “用于处理权限的管理命令”
- 第 246 页中的 “包含权限信息的文件”
- 第 247 页中的 “权限和审计”
- 第 247 页中的 “防止权限升级”
- 第 248 页中的 “传统应用程序和权限模型”

有关如何使用权限的信息，请参见第 11 章。有关概述信息，请参见第 177 页中的 “权限（概述）”。

用于处理权限的管理命令

下表列出了可用于处理权限的命令。

表 12–1 用于处理权限的命令

目的	命令	手册页
检查进程权限	`ppriv -v` *pid*	`ppriv`(1)
设置进程权限	`ppriv -s` *spec*	
列出系统上的权限	`ppriv -l`	
列出权限及其说明	`ppriv -lv` *priv*	
调试权限故障	`ppriv -eD` *failed-operation*	
为新的本地用户指定权限	`useradd`	`useradd`(1M)
为现有本地用户添加权限	`usermod`	`usermod`(1M)
为名称服务中的用户指定权限	`smuser`	`smuser`(1M)

表 12–1 用于处理权限的命令 （续）

目的	命令	手册页
将权限指定给新的本地角色	`roleadd`	`roleadd`(1M)
为现有本地角色添加权限	`rolemod`	`rolemod`(1M)
为名称服务中的角色指定权限	`smrole`	`smrole`(1M)
查看设备策略	`getdevpolicy`	`getdevpolicy`(1M)
设置设备策略	`devfsadm`	`devfsadm`(1M)
在打开的设备上更新设备策略	`update_drv -p` *policy driver*	`update_drv`(1M)
向设备中添加设备策略	`add_drv -p` *policy driver*	`add_drv`(1M)

Solaris Management Console GUI 是用于为命令、用户和角色指定权限的首选工具。有关更多信息，请参见第 201 页中的 “如何在 Solaris Management Console 中承担角色”。

包含权限信息的文件

以下文件包含权限信息。

表 12–2 包含权限信息的文件

文件和手册页	关键字	说明
`/etc/security/policy.conf`	`PRIV_DEFAULT`	系统的可继承权限集
`policy.conf`(4)	`PRIV_LIMIT`	系统的限制权限集
`/etc/user_attr`	用户项或角色项中的 `defaultpriv` 关键字	用户或角色的可继承权限集
`user_attr`(4)	其值通常在 Solaris Management Console GUI 中设置	
	用户项或角色项中的 `limitpriv` 关键字	用户或角色的限制权限集
	其值通常在 Solaris Management Console GUI 中设置	
`/etc/security/exec_attr`	命令的配置文件项中的 `privs` 关键字	为权限配置文件中的命令指定的权限列表
`exec_attr`(4)	命令的策略必须是 `solaris`	
`syslog.conf`	调试消息的系统日志文件	权限调试日志
`syslog.conf`(4)	在 `priv.debug` 项中设置的路径	

注 – 请勿直接编辑 exec_attr 和 user_attr 数据库。要管理权限，请使用 Solaris Management Console 或诸如 smuser 的命令。有关更多信息，请参见 smc(1M) 和 smuser(1M) 手册页。有关过程，请参见第 227 页中的 “管理权限（任务列表）” 。

权限和审计

可以审计权限的使用。当进程使用权限时，将在审计跟踪中记录权限的使用。权限将在其文本说明中记录。以下审计事件记录权限的使用：

- AUE_SETPPRIV **审计事件** – 更改权限集时，此事件会生成审计记录。AUE_SETPPRIV 审计事件位于 pm 类中。
- AUE_MODALLOCPRIV **审计事件** – 从内核外部添加权限时，此审计事件会生成审计记录。AUE_MODALLOCPRIV 审计事件位于 ad 类中。
- AUE_MODDEVPLCY **审计事件** – 更改设备策略时，此审计事件会生成审计记录。AUE_MODDEVPLCY 审计事件位于 ad 类中。
- AUE_prof_cmd **审计事件** – 在配置文件 shell 中执行命令时，此审计事件会生成审计记录。AUE_prof_cmd 审计事件位于 as 和 ua 审计类中。

不会审计基本集中权限的成功使用。尝试使用已从用户基本集中删除的基本权限时会进行审计。

防止权限升级

Solaris 内核可防止**权限升级**。权限升级是指某项权限使进程执行的操作多于其原本能够执行的操作。要防止进程获取的权限超出其应该获得的权限，必须具有完整的权限集才能进行特定系统修改。例如，只有具有完整权限集的进程才能更改 root (UID=0) 拥有的文件或进程。root 用户不需要权限就能更改 root 拥有的文件。但是，非超级用户必须具有所有权限才能更改 root 拥有的文件。

同样，提供设备访问权限的操作需要有效集中的所有权限。

file_chown_self 和 proc_owner 权限可进行权限升级。file_chown_self 权限允许进程放弃其文件。proc_owner 权限允许进程检查不归其拥有的进程。

可通过 rstchown 系统变量限制 file_chown_self 权限。将 rstchown 变量设置为零时，会从系统和所有用户的初始可继承集中删除 file_chown_self 权限。有关 rstchown 系统变量的更多信息，请参见 chown(1) 手册页。

file_chown_self 权限以最安全的方式指定给特定命令，放置在配置文件中，并指定给角色以在配置文件 shell 中使用。

proc_owner 权限不足以将进程 UID 切换为 0。将进程从任何 UID 切换为 UID=0 需要所有权限。由于 proc_owner 权限授予无限制读取系统上所有文件的权限，因此可以非常安全地将该权限指定给特定命令，将其放置在配置文件中，指定给角色以在配置文件 shell 中使用。

注意 – 可修改用户帐户，以便在该用户的初始可继承集中包含 `file_chown_self` 权限或 `proc_owner` 权限。您应当有充分的安全理由将这些功能强大的权限放置在任何用户、角色或系统的可继承权限集中。

有关如何针对设备防止权限升级的详细信息，请参见第 182 页中的 “权限和设备”。

传统应用程序和权限模型

为了适应传统应用程序，权限的实现使用超级用户模型和权限模型。内核会自动跟踪 `PRIV_AWARE` 标志，此标志指示已指定某个程序使用权限。请考虑不能识别权限的子进程。从父进程继承的所有权限均可在子进程的允许集和有效集中找到。如果子进程将 UID 设置为 0，则其可能不具有全部超级用户功能。进程的有效集和允许集会被限制为子进程限制集中包含的那些权限。这样，可识别权限进程的限制集会限制不能识别权限的子进程的超级用户权限。

第 4 部 分

Solaris 加密服务

本部分介绍 Solaris OS 所提供的集中的加密服务。

◆ ◆ ◆ 第 13 章

Solaris 加密框架（概述）

本章介绍 Solaris™ 加密框架。以下是本章中信息的列表：

- 第 251 页中的 “Solaris 加密框架的新增功能”
- 第 252 页中的 “Solaris 加密框架”
- 第 252 页中的 “Solaris 加密框架术语”
- 第 253 页中的 “Solaris 加密框架的范围”
- 第 254 页中的 “Solaris 加密框架中的管理命令”
- 第 254 页中的 “Solaris 加密框架中的用户级命令”
- 第 255 页中的 “Solaris 加密框架插件”
- 第 255 页中的 “加密服务和区域”

要管理和使用 Solaris 加密框架，请参见第 14 章。

Solaris 加密框架的新增功能

Solaris 10 1/06：框架库 `libpkcs11.so` 包含新组件**元插槽**。元插槽充当一个虚拟插槽，它具有框架中安装的所有令牌和槽的组合功能。实际上，元插槽使应用程序通过单个插槽与任何可用的加密服务透明地连接。

- 有关更多信息，请参见第 252 页中的 “Solaris 加密框架术语”中插槽、元插槽和令牌的定义。
- 要管理元插槽，请参见 `cryptoadm`(1M) 手册页。
- 有关 Solaris 新增功能的完整列表以及 Solaris 发行版的描述，请参见《Solaris 10 What’s New》。

Solaris 加密框架

Solaris 加密框架提供用于处理加密要求的算法和 PKCS #11 库的公共存储区。PKCS #11 库按照以下标准实现：RSA Security Inc. 推出的 PKCS #11 加密令牌接口 (Cryptographic Token Interface, Cryptoki)。

在内核级别，框架当前处理 Kerberos 和 IPsec 的加密要求。用户级使用者包括 `libsasl` 和 IKE。

美国出口法要求应限制使用开放式加密接口。Solaris 加密框架通过要求对内核加密提供器和 PKCS #11 加密提供器签名来满足当前法律规定。有关进一步介绍，请参见第 254 页中的“第三方软件的二进制文件签名”。

框架使加密服务的**提供器**可以让 Solaris 操作系统 中的许多**使用者**使用其服务。提供器的另外一个名称是**插件**。框架允许使用三种类型的插件：

- **用户级插件** – 使用 PKCS #11 库（如 `pkcs11_softtoken.so.1`）提供服务的共享对象。
- **内核级插件** – 提供软件中加密算法（如 AES）的实现的内核模块。

  框架中的许多算法针对 x86（使用 SSE2 指令集）和 SPARC 硬件进行了优化。
- **硬件插件** – 设备驱动程序及其关联的硬件加速器。硬件加速器使操作系统不再处理高开销的加密功能。Sun Crypto 加速器 1000 板就是这样一个示例。

框架为用户级提供器实现了标准接口 PKCS #11 v2.11 库。第三方应用程序可以使用该库来访问提供器。第三方也可以向框架中添加签名库、签名内核算法模块和签名设备驱动程序。`pkgadd` 实用程序安装该第三方软件时将添加这些插件。有关框架的主要组件图，请参见《Solaris Security for Developers Guide》中的第 8 章，“Introduction to the Solaris Cryptographic Framework”。

Solaris 加密框架术语

以下列出的定义和示例在使用加密框架时非常有用。

- **Algorithms（算法）** – 加密算法。这些算法是已建立的用于对输入执行加密或散列操作的递归计算过程。加密算法可以是对称算法，也可是非对称算法。对称算法对于加密和解密使用相同的密钥。非对称算法用于公钥加密中，它要求两个密钥。散列函数也是算法。

  算法示例包括：

  - 对称算法，如 AES 和 ARCFOUR
  - 非对称算法，例如 Diffie-Hellman 和 RSA
  - 散列函数，如 MD5
- **Consumers（使用者）** – 来自提供器的加密服务的用户。使用者可以是应用程序、最终用户或内核操作。

  使用者示例包括：

  - 应用程序，如 IKE

  - 最终用户，如运行 encrypt 命令的普通用户
  - 内核操作，例如 IPsec
- **Mechanism（机制）**－针对特定目的算法模式的应用。

  例如，应用于验证的 DES 机制（如 CKM_DES_MAC）与应用于加密的 DES 机制（如 CKM_DES_CBC_PAD）不同。
- **Metaslot（元插槽）**－提供框架中装入的其他插槽的功能组合的单个插槽。元插槽简化了处理框架中提供的所有提供器功能的工作。使用元插槽的应用程序请求某操作时，元插槽将确定应执行该操作的实际插槽。元插槽功能可以配置，但不需要配置。缺省情况下，元插槽处于打开状态。要配置元插槽，请参见 cryptoadm(1M) 手册页。
- **Mode（模式）**－加密算法的版本。例如，模式 CBC（Cipher Block Chaining，密码块链接）与模式 ECB（Electronic Code Book，电子源码书）不同。AES 算法包含两种模式：CKM_AES_ECB 和 CKM_AES_CBC。
- **Policy（策略）**－由管理员做出的、要使哪些机制可用的选择。缺省情况下，所有提供器和所有机制都可用。禁用任何机制即是对策略的应用。启用已禁用的机制也是对策略的应用。
- **Providers（提供器）**－使用者使用的加密服务。提供器插入到框架中，因此也称为**插件**。

  提供器示例包括：

  - PKCS #11 库，如 pkcs11_softtoken.so
  - 加密算法模块，如 aes 和 arcfour
  - 设备驱动程序及其关联的硬件加速器，例如 dca/0 加速器
- **Slot（插槽）**－一种或多种加密设备的接口。对应于物理读取器或其他设备接口的每个插槽可能包含令牌。令牌提供框架中加密设备的逻辑视图。
- **Token（令牌）**－在插槽中，令牌提供框架中加密设备的逻辑视图。

Solaris 加密框架的范围

框架为给出提供器的管理员、用户和开发者提供命令：

- **管理命令**－cryptoadm 命令提供用于列出可用提供器及其功能的 list 子命令。普通用户可以运行 cryptoadm list 和 cryptoadm --help 命令。

  所有其他 cryptoadm 子命令要求您承担包括加密管理权限配置文件的角色或者成为超级用户。子命令（如 disable、install 和 uninstall）可用于管理框架。有关更多信息，请参见 cryptoadm(1M) 手册页。

  svcadm 命令用于管理 kcfd 守护进程和刷新内核中的加密策略。有关更多信息，请参见 svcadm(1M) 手册页。
- **用户级命令**－digest 和 mac 命令提供文件完整性服务。encrypt 和 decrypt 命令保护文件不被窃听。要使用这些命令，请参见第 257 页中的 “使用 Solaris 加密框架保护文件（任务列表）”。

- **第三方提供器的二进制文件签名** – `elfsign` 命令使第三方可以对要在框架中使用的二进制文件签名。可以添加到框架中的二进制文件有 PKCS #11 库、内核算法模块和硬件设备驱动程序。要使用 `elfsign` 命令，请参见 《Solaris Security for Developers Guide》中的附录 F，“Packaging and Signing Cryptographic Providers”。

Solaris 加密框架中的管理命令

`cryptoadm` 命令管理正在运行的加密框架。该命令是加密管理权限配置文件的一部分。可以将此配置文件指定给用于安全管理加密框架的角色。`cryptoadm` 命令管理以下方面：

- 显示加密提供器的信息
- 禁用或启用提供器机制
- Solaris 10 1/06：禁用或启用元插槽

`svcadm` 命令用于启用、刷新和禁用加密服务守护进程 `kcfd`。此命令是 Solaris 服务管理工具 `smf` 的一部分。`svc:/system/cryptosvcs` 是加密框架的服务实例。有关更多信息，请参见 `smf`(5) 和 `svcadm`(1M) 手册页。

Solaris 加密框架中的用户级命令

Solaris 加密框架提供了用于检查文件完整性、加密文件和解密文件的用户级命令。独立命令 `elfsign` 使提供器可以对要在框架中使用的二进制文件签名。

- `digest` **命令** – 计算用于一个或多个文件或 stdin 的 message digest（消息摘要）。摘要用于验证文件的完整性。摘要功能的示例有 SHA1 和 MD5。
- `mac` **命令** – 计算一个或多个文件或 stdin 的 message authentication code, MAC（消息验证代码）。MAC 将数据与经过验证的消息相关联。MAC 使接收者可以验证消息是否来自发送者，以及消息是否被篡改。`sha1_mac` 和 `md5_hmac` 机制可以计算 MAC。
- `encrypt` **命令** – 使用对称密码加密文件或 stdin。 `encrypt -l` 命令列出可用的算法。用户级库中列出的机制可用于 `encrypt` 命令。框架提供了可进行用户加密的 AES、DES、3DES（Triple-DES，三重 DES）和 ARCFOUR 机制。
- `decrypt` **命令** – 解密使用 `encrypt` 命令加密的文件或 stdin。`decrypt` 命令使用的密钥和机制与用于加密原始文件的密钥和机制相同。

第三方软件的二进制文件签名

`elfsign` 命令提供了对要用于 Solaris 加密框架的提供器签名的方法。通常，此命令由提供器的开发者运行。

`elfsign` 命令具有请求 Sun 提供的证书和对二进制文件进行签名的子命令。另外一个子命令验证签名。未签名的二进制文件无法被 Solaris 加密框架使用。要对一个或多个提供器进行签名，需要 Sun 提供的证书和用于请求该证书的私钥。有关更多信息，请参见 《Solaris Security for Developers Guide》中的附录 F，“Packaging and Signing Cryptographic Providers”。

Solaris 加密框架插件

第三方可以将其提供器插入到 Solaris 加密框架中。第三方提供器可以是以下对象之一：

- PKCS #11 共享库
- 可装入的内核软件模块，如加密算法、MAC 函数或摘要功能
- 硬件加速器的内核设备驱动程序

必须使用 Sun 提供的证书对提供器对象进行签名。证书请求基于第三方选择的私钥和 Sun 提供的证书。证书请求发送到 Sun，Sun 将注册该第三方并发布证书。第三方然后使用 Sun 提供的证书对其提供器对象进行签名。

可装入的内核软件模块和硬件加速器的内核设备驱动程序也必须在内核中注册。注册通过 Solaris 加密框架 SPI（service provider interface，服务提供器接口）进行。

要安装提供器，第三方应提供用于安装签名的对象和 Sun 提供的证书的软件包。该软件包必须包括证书，并允许管理员将该证书放到安全目录中。有关更多信息，请参见《Solaris Security for Developers Guide》中的附录 F，“Packaging and Signing Cryptographic Providers”。

加密服务和区域

全局区域和每个非全局的区域都有独立的 `/system/cryptosvc` 服务。在全局区域中启用或刷新加密服务时，`kcfd` 守护进程会在该全局区域中启动，全局区域的用户级策略将被设置，系统的内核策略也将被设置。在非全局区域中启用或刷新服务时，`kcfd` 守护进程将在该区域中启动，用户级策略将在该区域中设置。内核策略由全局区域设置。

有关区域的更多信息，请参见《System Administration Guide: Solaris Containers-Resource Management and Solaris Zones》中的第二部分，“Zones”。有关用于管理持久性应用程序的服务管理工具的更多信息，请参见《System Administration Guide: Basic Administration》中的第 14 章“Managing Services (Overview)” 和 `smf`(5) 手册页。

◆ ◆ ◆ 第 14 章

Solaris 加密框架（任务）

本章介绍如何使用 Solaris 加密框架。以下是本章中信息的列表。

- 第 257 页中的 “使用加密框架（任务列表）”
- 第 257 页中的 “使用 Solaris 加密框架保护文件（任务列表）”
- 第 266 页中的 “管理加密框架（任务列表）”

使用加密框架（任务列表）

以下任务列表指向要使用加密框架的任务。

任务	说明	参考
保护单个文件或文件集	确保文件内容未被篡改。阻止侵入者读取文件。这些过程可以由普通用户完成。	第 257 页中的 “使用 Solaris 加密框架保护文件（任务列表）”
管理框架	添加、配置和删除软件提供器。禁用和启用硬件提供器机制。这些过程为管理过程。	第 266 页中的 “管理加密框架（任务列表）”
对提供器签名	使提供器添加到 Solaris 加密框架。 这些过程为开发者过程。	《Solaris Security for Developers Guide》中的附录 F，“Packaging and Signing Cryptographic Providers”。

使用 Solaris 加密框架保护文件（任务列表）

Solaris 加密框架可以帮助您保护文件。以下任务列表指向用于列出可用算法和通过加密保护文件的过程。

任务	说明	参考
生成对称密钥	生成用于 encrypt 命令或 mac 命令的随机密钥。	第 258 页中的 “如何生成对称密钥”
提供用于确保文件完整性的校验和	验证接收者的文件副本是否与发送的文件相同。	第 260 页中的 “如何计算文件摘要”
使用消息验证代码 (message authentication code, MAC) 保护文件	向消息接收者验证您是发送者。	第 261 页中的 “如何计算文件的 MAC”
加密文件，然后将此已加密的文件解密	通过加密文件保护文件内容。提供用于解密文件的加密参数。	第 263 页中的 “如何加密和解密文件”

使用 Solaris 加密框架保护文件

本节介绍如何生成对称密钥、如何创建文件完整性校验和以及如何避免文件遭到窃听。本节中的命令可以由普通用户运行。开发者可以编写使用这些命令的脚本。

▼ 如何生成对称密钥

加密文件以及生成文件的 MAC 需要使用密钥。密钥应派生于随机数池。

如果您的站点具有随机数生成器，请使用该生成器。或者，可以使用以 Solaris /dev/urandom 设备作为输入的 dd 命令。有关更多信息，请参见 dd(1M) 手册页。

1 确定算法要求的密钥长度。

a. 列出可用算法。

```
% encrypt -l

Algorithm       Keysize:  Min   Max (bits)

----------------------------------------

aes                       128   128

arcfour                     8   128

des                        64    64

3des                      192   192
```

```
% mac -l

Algorithm       Keysize:  Min   Max (bits)

----------------------------------------

des_mac                    64    64

sha1_hmac                   8   512

md5_hmac                    8   512
```

b. 确定要传递到 dd 命令的密钥长度（以字节为单位）。

将最小密钥大小和最大密钥大小除以 8。最小密钥大小和最大密钥大小不同时，可以使用中间密钥大小。例如，可以将值 8、16 或 64 传递到 sha1_hmac 和 md5_hmac 函数的 dd 命令。

2 生成对称密钥。

```
% dd if=/dev/urandom of=keyfile bs=n count=n
```

if=*file*	输入文件。对于随机密钥，请使用 /dev/urandom 文件。
of=*keyfile*	存储已生成密钥的输出文件。
bs=*n*	密钥大小（以字节为单位）。要获取字节长度，请将密钥位的比特长度除以 8。
count=*n*	输入块的计数。*n* 的数值应为 1。

3 将密钥存储在受保护的目录中。

密钥文件不应被除用户之外的任何人读取。

```
% chmod 400 keyfile
```

示例 14–1 创建用于 AES 算法的密钥

在以下示例中，将创建用于 AES 算法的密钥。也将存储该密钥用于以后的解密。AES 机制使用 128 位的密钥。该密钥在 dd 命令中表示为 16 字节。

```
% ls -al ~/keyf

drwx------   2 jdoe  staff        512 May 3 11:32 ./

% dd if=/dev/urandom of=$HOME/keyf/05.07.aes16 bs=16  count=1

% chmod 400 ~/keyf/05.07.aes16
```

示例 14–2 创建用于 DES 算法的密钥

在以下示例中，将创建用于 DES 算法的密钥。也将存储该密钥用于以后的解密。DES 机制使用 64 位的密钥。该密钥在 dd 命令中表示为 8 字节。

```
% dd if=/dev/urandom of=$HOME/keyf/05.07.des8 bs=8  count=1

% chmod 400 ~/keyf/05.07.des8
```

示例 14–3 创建用于 3DES 算法的密钥

在以下示例中，将创建用于 3DES 算法的密钥。也将存储该密钥用于以后的解密。3DES 机制使用 192 位的密钥。该密钥在 dd 命令中表示为 24 字节。

```
% dd if=/dev/urandom of=$HOME/keyf/05.07.3des.24 bs=24 count=1

% chmod 400 ~/keyf/05.07.3des.24
```

示例 14–4 创建用于 MD5 算法的密钥

在以下示例中，将创建用于 MD5 算法的密钥。也将存储该密钥用于以后的解密。 该密钥在 dd 命令中表示为 64 字节。

```
% dd if=/dev/urandom of=$HOME/keyf/05.07.mack64 bs=64 count=1

% chmod 400 ~/keyf/05.07.mack64
```

▼ 如何计算文件摘要

计算文件摘要时，可以通过比较摘要输出来查看文件是否被篡改。摘要不会修改原始文件。

1 列出可用摘要算法。

```
% digest -l

md5

sha1
```

2 计算文件摘要并保存摘要列表。

为 digest 命令提供一种算法。

```
% digest -v -a algorithm input-file > digest-listing
```

-v 显示以下格式的输出：

algorithm (*input-file*) = *digest*

-a *algorithm* 用于计算文件摘要的算法。键入与步骤 1 的输出中显示的算法相同的算法。

input-file `digest` 命令的输入文件。

digest-listing `digest` 命令的输出文件。

示例 14–5 使用 MD5 机制计算摘要

在以下示例中，`digest` 命令使用 MD5 机制计算电子邮件附件的摘要。

```
% digest -v -a md5 email.attach >>  $HOME/digest.emails.05.07

% cat  ~/digest.emails.05.07

md5 (email.attach) = 85c0a53d1a5cc71ea34d9ee7b1b28b01
```

未使用 `-v` 选项时，将不会与摘要一起保存以下伴随信息：

```
% digest -a md5 email.attach >>  $HOME/digest.emails.05.07

% cat  ~/digest.emails.05.07

85c0a53d1a5cc71ea34d9ee7b1b28b01
```

示例 14–6 使用 SHA1 机制计算摘要

在以下示例中，`digest` 命令使用 SHA1 机制提供目录列表。结果保存于文件中。

```
% digest -v -a sha1 docs/* > $HOME/digest.docs.legal.05.07

% more ~/digest.docs.legal.05.07

sha1 (docs/legal1) = 1df50e8ad219e34f0b911e097b7b588e31f9b435

sha1 (docs/legal2) = 68efa5a636291bde8f33e046eb33508c94842c38

sha1 (docs/legal3) = 085d991238d61bd0cfa2946c183be8e32cccf6c9

sha1 (docs/legal4) = f3085eae7e2c8d008816564fdf28027d10e1d983
```

▼ 如何计算文件的 MAC

消息验证代码或 MAC 会计算文件的摘要并使用密钥进一步保护该摘要。MAC 不会修改原始文件。

1. **列出可用机制。**

```
% mac -l

Algorithm       Keysize:  Min   Max

-----------------------------------

des_mac                    64    64

sha1_hmac                   8   512

md5_hmac                    8   512
```

2. **生成相应长度的对称密钥。**

   您有两种选择。可以提供将根据其生成密钥的 passphrase（口令短语）。或者，可以提供密钥。

   - 如果提供口令短语，则必须存储或记住该口令短语。如果您在线存储口令短语，则只有您才可以读取该口令短语文件。
   - 如果提供密钥，则它必须是对应于机制的正确大小。有关过程，请参见第 258 页中的 “如何生成对称密钥”。

3. **创建文件的 MAC。**

   为 `mac` 命令提供密钥并使用对称密钥算法。

   `% mac -v -a` *algorithm* `[ -k` *keyfile* `]` *input-file*

`-v`	显示以下格式的输出： *algorithm* (*input-file*) = *mac*
`-a` *algorithm*	用于计算 MAC 的算法。键入与 `mac -l` 命令的输出中显示的算法相同的算法。
`-k` *keyfile*	包含算法所指定长度的密钥的文件。
input-file	MAC 的输入文件。

示例 14–7 使用 DES_MAC 和口令短语计算 MAC

在以下示例中，将使用 DES_MAC 机制和派生于口令短语的密钥对电子邮件附件进行验证。MAC 列表将保存到文件中。如果口令短语存储在某个文件中，则除了该用户之外，其他任何人都不能读取该文件。

```
% mac -v -a des_mac email.attach

Enter key:      <键入口令短语>
```

```
des_mac (email.attach) = dd27870a

% echo "des_mac (email.attach) = dd27870a" >> ~/desmac.daily.05.07
```

示例 14–8 使用 MD5_HMAC 和密钥文件计算 MAC

在以下示例中，将使用 MD5_HMAC 机制和密钥对电子邮件附件进行验证。MAC 列表将保存到文件中。

```
% mac -v -a md5_hmac -k $HOME/keyf/05.07.mack64 email.attach

md5_hmac (email.attach) = 02df6eb6c123ff25d78877eb1d55710c

% echo "md5_hmac (email.attach) = 02df6eb6c123ff25d78877eb1d55710c" \

>> ~/mac.daily.05.07
```

示例 14–9 使用 SHA1_HMAC 和密钥文件计算 MAC

在以下示例中，将使用 SHA1_HMAC 机制和密钥对目录清单进行验证。结果保存于文件中。

```
% mac -v -a sha1_hmac \

-k $HOME/keyf/05.07.mack64 docs/* > $HOME/mac.docs.legal.05.07

% more ~/mac.docs.legal.05.07

sha1_hmac (docs/legal1) = 9b31536d3b3c0c6b25d653418db8e765e17fe07a

sha1_hmac (docs/legal2) = 865af61a3002f8a457462a428cdb1a88c1b51ff5

sha1_hmac (docs/legal3) = 076c944cb2528536c9aebd3b9fbe367e07b61dc7

sha1_hmac (docs/legal4) = 7aede27602ef6e4454748cbd3821e0152e45beb4
```

▼ 如何加密和解密文件

加密文件时，不会删除或更改原始文件。输出文件将被加密。

有关 `encrypt` 命令的常见错误的解决方案，请参见示例后面的部分。

1. **创建适当长度的对称密钥。**

   您有两种选择。可以提供将根据其生成密钥的 passphrase（口令短语）。或者，可以提供密钥。

   - 如果提供口令短语，则必须存储或记住该口令短语。如果您在线存储口令短语，则只有您才可以读取该口令短语文件。
   - 如果提供密钥，则它必须是对应于机制的正确大小。有关过程，请参见第 258 页中的 “如何生成对称密钥”。

2. **加密文件。**

   为 `encrypt` 命令提供密钥并使用对称密钥算法。

   `% encrypt -a` *algorithm* `[ -k` *keyfile* `] -i` *input-file* `-o` *output-file*

   `-a` *algorithm* — 用于加密文件的算法。 键入与 `encrypt -l` 命令的输出中显示的算法相同的算法。

   `-k` *keyfile* — 包含算法所指定长度的密钥的文件。 每种算法的密钥长度（以位为单位）列在 `encrypt -l` 命令的输出中。

   `-i` *input-file* — 要加密的输入文件。命令不会更改此文件。

   `-o` *output-file* — 加密格式的输入文件的输出文件。

示例 14–10 使用 AES 和口令短语进行加密和解密

在以下示例中，将使用 AES 算法加密文件。密钥根据口令短语生成。如果口令短语存储在某个文件中，则除了该用户之外，其他任何人都不能读取该文件。

```
% encrypt -a aes -i ticket.to.ride -o ~/enc/e.ticket.to.ride

Enter key:      <键入口令短语>
```

输入文件 `ticket.to.ride` 仍然以其原始格式存在。

要解密输出文件，用户应使用加密该文件的相同口令短语和加密机制。

```
% decrypt -a aes -i ~/enc/e.ticket.to.ride -o ~/d.ticket.to.ride

Enter key:      <键入口令短语>
```

示例 14–11 使用 AES 和密钥文件进行加密和解密

在以下示例中，将使用 AES 算法加密文件。AES 机制使用 128 位（16 字节）的密钥。

```
% encrypt -a aes -k ~/keyf/05.07.aes16 \

-i ticket.to.ride -o ~/enc/e.ticket.to.ride
```

输入文件 ticket.to.ride 仍然以其原始格式存在。

要解密输出文件，用户应使用加密该文件的相同密钥和加密机制。

```
% decrypt -a aes -k ~/keyf/05.07.aes16  \
-i ~/enc/e.ticket.to.ride -o ~/d.ticket.to.ride
```

示例 14–12 使用 ARCFOUR 和密钥文件进行加密和解密

在以下示例中，使用 ARCFOUR 算法加密文件。ARCFOUR 算法接受 8 位（1 字节）、64 位（8 字节）或 128 位（16 字节）的密钥。

```
% encrypt -a arcfour -i personal.txt \
-k ~/keyf/05.07.rc4.8 -o ~/enc/e.personal.txt
```

要解密输出文件，用户应使用加密该文件的相同密钥和加密机制。

```
% decrypt -a arcfour -i ~/enc/e.personal.txt \
-k ~/keyf/05.07.rc4.8 -o ~/personal.txt
```

示例 14–13 使用 3DES 和密钥文件进行加密和解密

在以下示例中，将使用 3DES 算法加密文件。3DES 算法要求 192 位（24 字节）的密钥。

```
% encrypt -a 3des -k ~/keyf/05.07.des24 \
-i ~/personal2.txt -o ~/enc/e.personal2.txt
```

要解密输出文件，用户应使用加密该文件的相同密钥和加密机制。

```
% decrypt -a 3des -k ~/keyf/05.07.des24 \
-i ~/enc/e.personal2.txt -o ~/personal2.txt
```

故障排除 以下消息说明，所使用的算法不接受提供给 encrypt 命令的密钥。

- encrypt: unable to create key for crypto operation: CKR_ATTRIBUTE_VALUE_INVALID
- encrypt: failed to initialize crypto operation: CKR_KEY_SIZE_RANGE

如果所提供的密钥不满足算法要求，则必须提供正确的密钥。

- 一种选择是使用口令短语。框架然后提供满足要求的密钥。

- 第二种选择是提供算法接受的密钥大小。例如，DES 算法要求 64 位的密钥。3DES 算法要求 192 位的密钥。

管理加密框架（任务列表）

以下任务列表指向用于管理 Solaris 加密框架中的软件和硬件提供器的过程。

任务	说明	参考
列出 Solaris 加密框架中的提供器	列出可在 Solaris 加密框架中使用的算法、库和硬件设备。	第 266 页中的 “如何列出可用提供器”
添加软件提供器	将 PKCS #11 库或内核模块添加到 Solaris 加密框架中。必须对提供器签名。	第 270 页中的 “如何添加软件提供器”
禁止使用用户级机制	删除不使用的软件机制。可以再次启用该机制。	第 272 页中的 “如何禁止使用用户级机制”
临时禁用内核模块的机制	临时删除使用的机制。通常用于测试。	第 275 页中的 “如何禁止使用内核软件提供器”
卸载提供器	删除使用的内核软件提供器。	示例 14–22
列出可用硬件提供器	显示连接的硬件、硬件提供的机制以及为使用而启用的机制。	第 278 页中的 “如何列出硬件提供器”
禁用硬件提供器的机制	请确保未使用所选择的硬件加速器机制。	第 279 页中的 “如何禁用硬件提供器机制和功能”
重新启动或刷新加密服务	请确保加密服务可用。	第 281 页中的 “如何刷新或重新启动所有加密服务”

管理加密框架

本节介绍如何在 Solaris 加密框架中管理软件提供器和硬件提供器。如果需要，可以删除使用的软件提供器和硬件提供器。例如，可以禁止实施某软件提供器的算法。然后，可以强制系统使用其他软件提供器的算法。

▼ 如何列出可用提供器

Solaris 加密框架可为多种类型的使用者提供算法：

- 用户级提供器为与 `libpkcs11` 库相链接的应用程序提供 PKCS #11 加密接口
- 内核软件提供器为 IPsec、Kerberos 和其他 Solaris 内核组件提供算法
- 内核硬件提供器通过 `pkcs11_kernel` 库提供内核使用者和应用程序可使用的算法

1 **以简要格式列出提供器。**

普通用户只能使用用户级机制。

```
% cryptoadm list

user-level providers:

    /usr/lib/security/$ISA/pkcs11_kernel.so

    /usr/lib/security/$ISA/pkcs11_softtoken.so


kernel software providers:

    des

    aes

    blowfish

    arcfour

    sha1

    md5

    rsa


kernel hardware providers:

    dca/0
```

2 **列出 Solaris 加密框架中的提供器及其机制。**

以下输出列出了所有的机制。但是，所列出的一些机制可能无法使用。要仅列出管理员已批准使用的机制，请参见示例 14–15。

为了便于显示，已重新设置输出格式。

```
% cryptoadm list -m

user-level providers:

====================

/usr/lib/security/$ISA/pkcs11_kernel.so: CKM_MD5,CKM_MD5_HMAC,
```

```
CKM_MD5_HMAC_GENERAL,CKM_SHA_1,CKM_SHA_1_HMAC,CKM_SHA_1_HMAC_GENERAL,

...

/usr/lib/security/$ISA/pkcs11_softtoken.so:

CKM_DES_CBC,CKM_DES_CBC_PAD,CKM_DES_ECB,CKM_DES_KEY_GEN,

CKM_DES3_CBC,CKM_DES3_CBC_PAD,CKM_DES3_ECB,CKM_DES3_KEY_GEN,

CKM_AES_CBC,CKM_AES_CBC_PAD,CKM_AES_ECB,CKM_AES_KEY_GEN,

...

kernel software providers:

==========================

des: CKM_DES_ECB,CKM_DES_CBC,CKM_DES3_ECB,CKM_DES3_CBC

aes: CKM_AES_ECB,CKM_AES_CBC

blowfish: CKM_BF_ECB,CKM_BF_CBC

arcfour: CKM_RC4

sha1: CKM_SHA_1,CKM_SHA_1_HMAC,CKM_SHA_1_HMAC_GENERAL

md5: CKM_MD5,CKM_MD5_HMAC,CKM_MD5_HMAC_GENERAL

rsa: CKM_RSA_PKCS,CKM_RSA_X_509,CKM_MD5_RSA_PKCS,CKM_SHA1_RSA_PKCS

swrand: No mechanisms presented.


kernel hardware providers:

==========================

dca/0: CKM_MD5,CKM_MD5_HMAC,CKM_MD5_HMAC_GENERAL,...
```

示例 14–14 查找现有的加密机制

在以下示例中，列出了用户级库 pkcs11_softtoken 提供的所有机制。

```
% cryptoadm list -m provider=/usr/lib/security/'$ISA'/pkcs11_softtoken.so
```

```
/usr/lib/security/$ISA/pkcs11_softtoken.so:

CKM_DES_CBC,CKM_DES_CBC_PAD,CKM_DES_ECB,CKM_DES_KEY_GEN,

CKM_DES3_CBC,CKM_DES3_CBC_PAD,CKM_DES3_ECB,CKM_DES3_KEY_GEN,

...

CKM_SSL3_KEY_AND_MAC_DERIVE,CKM_TLS_KEY_AND_MAC_DERIVE
```

示例 14–15 查找可用的加密机制

策略确定可使用的机制。管理员设置该策略。管理员可以选择禁用特定提供器的机制。-p 选项显示管理员设置的策略允许的机制列表。

```
% cryptoadm list -p

user-level providers:

=====================

/usr/lib/security/$ISA/pkcs11_kernel.so: all mechanisms are enabled.

random is enabled.

/usr/lib/security/$ISA/pkcs11_softtoken.so: all mechanisms are enabled.

random is enabled.



kernel software providers:

==========================

des: all mechanisms are enabled.

aes: all mechanisms are enabled.

blowfish: all mechanisms are enabled.

arcfour: all mechanisms are enabled.

sha1: all mechanisms are enabled.

md5: all mechanisms are enabled.

rsa: all mechanisms are enabled.
```

```
swrand: random is enabled.


kernel hardware providers:

=========================

dca/0: all mechanisms are enabled. random is enabled.
```

▼ 如何添加软件提供器

1 **承担主管理员角色，或成为超级用户。**

主管理员角色拥有主管理员配置文件。要创建角色并将其指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2 **列出系统可使用的软件提供器。**

```
# cryptoadm list

user-level providers:

        /usr/lib/security/$ISA/pkcs11_kernel.so

        /usr/lib/security/$ISA/pkcs11_softtoken.so


kernel software providers:

    des

    aes

    blowfish

    arcfour

    sha1

    md5

    rsa

    swrand
```

```
kernel hardware providers:

     dca/0
```

3 使用 pkgadd 命令添加提供器的软件包

```
# pkgadd -d /path/to/package  pkginst
```

该软件包必须包括由 Sun 提供的证书签名的软件。要请求 Sun 提供的证书并对提供器签名，请参见 《Solaris Security for Developers Guide》 中的附录 F，“Packaging and Signing Cryptographic Providers”。

该软件包应包含这样的脚本：通知加密框架具有一组机制的另外一个提供器可用。有关打包要求的信息，请参见 《Solaris Security for Developers Guide》 中的附录 F，“Packaging and Signing Cryptographic Providers”。

4 刷新提供器。

如果添加了软件提供器或者添加了硬件并为该硬件添加了指定的策略，则需要刷新提供器。

```
# svcadm refresh svc:/system/cryptosvc
```

5 查找列表中的新提供器。

在本例中，安装了新的内核软件提供器。

```
# cryptoadm list

...

kernel software providers:

    des

    aes

    blowfish

    arcfour

    sha1

    md5

    rsa

    swrand
```

```
    ecc       <-- 增加的提供器
...
```

示例 14–16 添加用户级软件提供器

在以下示例中，将安装签名的 PKCS #11 库。

```
# pkgadd -d /cdrom/cdrom0/SolarisNew

    应答提示

# svcadm refresh system/cryptosvc

# cryptoadm list

user-level providers:

==========================

    /usr/lib/security/$ISA/pkcs11_kernel.so

    /usr/lib/security/$ISA/pkcs11_softtoken.so

    /opt/SUNWconn/lib/$ISA/libpkcs11.so.1      <-- 增加的提供器
```

使用加密框架测试库的开发者可以手动安装该库。

```
# cryptoadm install provider=/opt/SUNWconn/lib/'$ISA'/libpkcs11.so.1
```

▼ 如何禁止使用用户级机制

如果不能使用库提供器的某些加密机制，则可以删除所选机制。此过程使用 pkcs11_softtoken 库中的 DES 机制为例。

1. **成为超级用户或承担包括加密管理权限配置文件的角色。**

   要创建包括加密管理权限配置文件的角色并将该角色指定给用户，请参见示例 9–7。

2. **列出特定用户级软件提供器提供的机制。**

```
% cryptoadm list -m provider=/usr/lib/security/'$ISA'/pkcs11_softtoken.so

/usr/lib/security/$ISA/pkcs11_softtoken.so:

CKM_DES_CBC,CKM_DES_CBC_PAD,CKM_DES_ECB,CKM_DES_KEY_GEN,
```

```
CKM_DES3_CBC,CKM_DES3_CBC_PAD,CKM_DES3_ECB,CKM_DES3_KEY_GEN,

CKM_AES_CBC,CKM_AES_CBC_PAD,CKM_AES_ECB,CKM_AES_KEY_GEN,

...
```

3 列出可用的机制。

```
$ cryptoadm list -p

user-level providers:

====================

...

/usr/lib/security/$ISA/pkcs11_softtoken.so: all mechanisms are enabled.

random is enabled.

...
```

4 禁用无法使用的机制。

```
$ cryptoadm disable provider=/usr/lib/security/'$ISA'/pkcs11_softtoken.so \

> mechanism=CKM_DES_CBC,CKM_DES_CBC_PAD,CKM_DES_ECB
```

5 列出可使用的机制。

```
$ cryptoadm list -p provider=/usr/lib/security/'$ISA'/pkcs11_softtoken.so

/usr/lib/security/$ISA/pkcs11_softtoken.so: all mechanisms are enabled,

except CKM_DES_ECB,CKM_DES_CBC_PAD,CKM_DES_CBC. random is enabled.
```

示例 14–17 启用用户级软件提供器机制

在以下示例中，将使禁用的 DES 机制再次可用。

```
$ cryptoadm list -m provider=/usr/lib/security/'$ISA'/pkcs11_softtoken.so

/usr/lib/security/$ISA/pkcs11_softtoken.so:

CKM_DES_CBC,CKM_DES_CBC_PAD,CKM_DES_ECB,CKM_DES_KEY_GEN,

CKM_DES3_CBC,CKM_DES3_CBC_PAD,CKM_DES3_ECB,CKM_DES3_KEY_GEN,

...
```

```
$ cryptoadm list -p provider=/usr/lib/security/'$ISA'/pkcs11_softtoken.so
/usr/lib/security/$ISA/pkcs11_softtoken.so: all mechanisms are enabled,
except CKM_DES_ECB,CKM_DES_CBC_PAD,CKM_DES_CBC. random is enabled.
$ cryptoadm enable provider=/usr/lib/security/'$ISA'/pkcs11_softtoken.so \
> mechanism=CKM_DES_ECB
$ cryptoadm list -p provider=/usr/lib/security/'$ISA'/pkcs11_softtoken.so
/usr/lib/security/$ISA/pkcs11_softtoken.so: all mechanisms are enabled,
except CKM_DES_CBC_PAD,CKM_DES_CBC. random is enabled.
```

示例 14–18 启用所有用户级软件提供器机制

在以下示例中，将启用用户级库的所有机制。

```
$ cryptoadm enable provider=/usr/lib/security/'$ISA'/pkcs11_softtoken.so all
$ cryptoadm list -p provider=/usr/lib/security/'$ISA'/pkcs11_softtoken.so
/usr/lib/security/$ISA/pkcs11_softtoken.so: all mechanisms are enabled.
random is enabled.
```

示例 14–19 永久删除用户级软件提供器可用性

在以下示例中，将删除 `libpkcs11.so.1` 库。

```
$ cryptoadm uninstall provider=/opt/SUNWconn/lib/'$ISA'/libpkcs11.so.1
$ cryptoadm list
user-level providers:
    /usr/lib/security/$ISA/pkcs11_kernel.so
    /usr/lib/security/$ISA/pkcs11_softtoken.so

kernel software providers:
...
```

▼ 如何禁止使用内核软件提供器

如果加密框架提供多种模式的提供器（如 AES），则可以删除所使用的速度较慢的机制或损坏的机制。此过程使用 AES 算法为例。

1. **成为超级用户或承担包括加密管理权限配置文件的角色。**
   要创建包括加密管理权限配置文件的角色并将该角色指定给用户，请参见示例 9–7。

2. **列出特定内核软件提供器提供的机制。**

```
$ cryptoadm list -m provider=aes

aes: CKM_AES_ECB,CKM_AES_CBC
```

3. **列出可用的机制。**

```
$ cryptoadm list -p provider=aes

aes: all mechanisms are enabled.
```

4. **禁用无法使用的机制。**

```
$ cryptoadm disable provider=aes mechanism=CKM_AES_ECB
```

5. **列出可用的机制。**

```
$ cryptoadm list -p provider=aes

aes: all mechanisms are enabled, except CKM_AES_ECB.
```

示例 14–20 启用内核软件提供器机制

在以下示例中，将使禁用的 AES 机制再次可用。

```
cryptoadm list -m provider=aes

aes: CKM_AES_ECB,CKM_AES_CBC

$ cryptoadm list -p provider=aes

aes: all mechanisms are enabled, except CKM_AES_ECB.

$ cryptoadm enable provider=aes mechanism=CKM_AES_ECB

$ cryptoadm list -p provider=aes

aes: all mechanisms are enabled.
```

示例 14–21 临时删除内核软件提供器可用性

在以下示例中，将临时删除所使用的 AES 提供器。unload 子命令用于禁止在卸载某提供器时自动装入该提供器。例如，安装影响提供器的修补程序时，将使用 unload 子命令。

```
$ cryptoadm unload provider=aes

$ cryptoadm list

...

kernel software providers:

         des

         aes (inactive)

         blowfish

         arcfour

         sha1

         md5

         rsa

         swrand
```

刷新加密框架之前，AES 提供器不可用。

```
$ svcadm refresh system/cryptosvc

$ cryptoadm list

...

kernel software providers:

         des

         aes

         blowfish

         arcfour

         sha1
```

```
        md5

        rsa

        swrand
```

如果内核使用者正在使用内核软件提供器，则不会卸载该软件。此时将显示错误消息，但可继续使用该提供器。

示例 14–22 永久删除软件提供器可用性

在以下示例中，将删除所使用的 AES 提供器。一旦删除，该 AES 提供器将不会在内核软件提供器的策略列表中显示。

```
$ cryptoadm uninstall provider=aes

$ cryptoadm list

...

kernel software providers:

        des

        blowfish

        arcfour

        sha1

        md5

        rsa

        swrand
```

如果内核使用者正在使用内核软件提供器，将显示错误消息，但可继续使用该提供器。

示例 14–23 重新安装已删除的内核软件提供器

在以下示例中，将重新安装 AES 内核软件提供器。

```
$ cryptoadm install provider=aes mechanism=CKM_AES_ECB,CKM_AES_CBC

$ cryptoadm list
```

```
...

kernel software providers:

         des

         aes

         blowfish

         arcfour

         sha1

         md5

         rsa

         swrand
```

▼ 如何列出硬件提供器

硬件提供器将自动定位和装入。有关更多信息，请参见 `driver.conf`(4) 手册页。

开始之前 添加期望在 Solaris 加密框架中使用的硬件时，该硬件将使用 SPI 在内核中注册。框架将检查是否已对硬件驱动程序签名。具体地说，框架将检查是否已使用 Sun 发布的证书对驱动程序的对象文件签名。

1 列出系统中可用的硬件提供器。

```
% cryptoadm list

...

kernel hardware providers:

   dca/0
```

2 列出该板提供的机制。

```
% cryptoadm list -m provider=dca/0

dca/0: CKM_MD5,CKM_MD5_HMAC,CKM_MD5_HMAC_GENERAL,...
```

3 列出板中可使用的机制。

```
% cryptoadm list -p provider=dca/0

dca/0: all mechanisms are enabled. random is enabled.
```

▼ 如何禁用硬件提供器机制和功能

可以有选择地禁用硬件提供器的机制和随机数功能。要再次启用它们，请参见示例 14–24。

1. **列出板中可用的机制和功能。**

```
% cryptoadm list -p provider=dca/0

dca/0: all mechanisms are enabled. random is enabled.
```

2. **成为超级用户或承担包括加密管理权限配置文件的角色。**
   要创建包括加密管理权限配置文件的角色并将该角色指定给用户，请参见示例 9–7。

3. **选择要禁用的机制或功能：**

   - **禁用选择的机制。**

```
# cryptoadm list -m provider=dca/0

dca/0: CKM_MD5,CKM_MD5_HMAC,CKM_MD5_HMAC_GENERAL,...

CKM_DES_ECB,CKM_DES3_ECB...

random is enabled.

# cryptoadm disable provider=dca/0 mechanism=CKM_DES_ECB,CKM_DES3_ECB

# cryptoadm list -p provider=dca/0

dca/0: all mechanisms are enabled except CKM_DES_ECB,CKM_DES3_ECB.

random is enabled.
```

   - **禁用随机数生成器。**

```
# cryptoadm list -p provider=dca/0

dca/0: all mechanisms are enabled. random is enabled.

# cryptoadm disable provider=dca/0 random

# cryptoadm list -p provider=dca/0

dca/0: all mechanisms are enabled. random is disabled.
```

   - **禁用所有机制。不禁用随机数生成器。**

```
# cryptoadm list -p provider=dca/0

dca/0: all mechanisms are enabled. random is enabled.
```

```
# cryptoadm disable provider=dca/0 mechanism=all

# cryptoadm list -p provider=dca/0

dca/0: all mechanisms are disabled. random is enabled.
```

- 禁用硬件的每种功能和机制。

```
# cryptoadm list -p provider=dca/0

dca/0: all mechanisms are enabled. random is enabled.

# cryptoadm disable provider=dca/0 all

# cryptoadm list -p provider=dca/0

dca/0: all mechanisms are disabled. random is disabled.
```

示例 14–24 启用硬件提供器的机制和功能

在以下示例中，将有选择地启用单个硬件的已禁用机制。

```
# cryptoadm list -p provider=dca/0

dca/0: all mechanisms are enabled except CKM_DES_ECB,CKM_DES3_ECB.

random is enabled.

# cryptoadm enable provider=dca/0 mechanism=CKM_DES3_ECB

# cryptoadm list -p provider=dca/0

dca/0: all mechanisms are enabled except CKM_DES_ECB. random is enabled.
```

在以下示例中，将仅启用随机数生成器。

```
# cryptoadm list -p provider=dca/0

dca/0: all mechanisms are enabled, except CKM_MD5,CKM_MD5_HMAC,....

random is disabled.

# cryptoadm enable provider=dca/0 random

# cryptoadm list -p provider=dca/0

dca/0: all mechanisms are enabled, except CKM_MD5,CKM_MD5_HMAC,....
```

```
random is enabled.
```

在以下示例中，将仅启用机制。将继续禁用随机生成器。

```
# cryptoadm list -p provider=dca/0

dca/0: all mechanisms are enabled, except CKM_MD5,CKM_MD5_HMAC,....

random is disabled.

# cryptoadm enable provider=dca/0 mechanism=all

# cryptoadm list -p provider=dca/0

dca/0: all mechanisms are enabled. random is disabled.
```

在以下示例中，将启用板中的所有功能和机制。

```
# cryptoadm list -p provider=dca/0

dca/0: all mechanisms are enabled, except CKM_DES_ECB,CKM_DES3_ECB.

random is disabled.

# cryptoadm enable provider=dca/0 all

# cryptoadm list -p provider=dca/0

dca/0: all mechanisms are enabled. random is enabled.
```

▼ 如何刷新或重新启动所有加密服务

缺省情况下，将启用 Solaris 加密框架。当由于任何原因 `kcfd` 守护进程失败时，可以使用服务管理工具重新启动加密服务。有关更多信息，请参见 `smf`(5) 和 `svcadm`(1M) 手册页。有关重新启用加密服务的区域的影响，请参见第 255 页中的 “加密服务和区域”。

1 检查加密服务的状态。

```
% svcs \*cryptosvc\*

 STATE          STIME    FMRI

offline         Dec_09   svc:/system/cryptosvc:default
```

2. **成为超级用户或承担启用加密服务的等效角色。**

   角色包含授权和具有一定权限的命令。有关角色的更多信息，请参见第 186 页中的 “配置 RBAC（任务列表）” 。

```
# svcadm enable svc:/system/cryptosvc
```

示例 14–25 刷新加密服务

在以下示例中，将在全局区域中刷新加密服务。因此，每个非全局区域中的内核级加密策略也将被刷新。

```
# svcadm refresh system/cryptosvc
```

第 5 部 分

验证服务和安全通信

本部分讨论可以在非联网系统上或两个系统之间配置的验证服务。要配置经过验证的用户和系统的网络，请参见第 6 部分。

◆ ◆ ◆ 第 15 章

使用验证服务（任务）

本章介绍有关如何使用安全 RPC 通过 NFS 挂载来验证主机和用户的信息。以下是本章中的主题列表：

- 第 285 页中的 “安全 RPC 概述”
- 第 289 页中的 “管理安全 RPC（任务列表）”

安全 RPC 概述

安全 RPC（Remote Procedure Call，远程过程调用）使用验证机制保护远程过程。Diffie-Hellman 验证机制可验证发出服务请求的主机和用户。此验证机制使用数据加密标准 (Data Encryption Standard, DES) 加密。使用安全 RPC 的应用程序包括 NFS 和名称服务（NIS 和 NIS+）。

NFS 服务和安全 RPC

NFS 可使多台主机通过网络共享文件。在 NFS 服务中，一台服务器可为多台客户机保存数据和资源。这些客户机具有访问服务器与客户机共享的文件系统的权限。登录到客户机系统的用户可以通过从服务器挂载文件系统来访问文件系统。对于客户机系统上的用户，就好像这些文件是客户机的本地文件。NFS 的一种最常见的用途是允许将系统安装在办公室中，同时将所有用户文件存储在中心位置。NFS 服务的某些功能（例如 `mount` 命令的 `-nosuid` 选项）可用于禁止未经授权的用户打开设备和文件系统。

NFS 服务使用安全 RPC 来验证通过网络发出请求的用户。此过程称为**安全** *NFS*。Diffie-Hellman 验证机制 `AUTH_DH` 使用 DES 加密来确保授权访问。 `AUTH_DH` 机制也称为 `AUTH_DES`。详细信息，请参见以下内容：

- 要设置和管理安全 NFS，请参见《System Administration Guide: Network Services》中的“Administering the Secure NFS System”。
- 要设置 NIS+ 表以及在 `cred` 表中输入名称，请参见《System Administration Guide: Naming and Directory Services (NIS+)》。

- 有关涉及 RPC 验证的事务的概述，请参见第 286 页中的 “Diffie-Hellman 验证的实现”。

使用安全 NFS 的 DES 加密

数据加密标准 (Data Encryption Standard, DES) 加密功能使用 56 位密钥来加密数据。如果两个凭证用户或主体知道同一 DES 密钥，则他们可以通过使用此密钥加密和解密文本来进行秘密通信。DES 是一种相对迅速的加密机制。DES 芯片可使加密更迅速。但是，如果不存在此芯片，则将替换软件实现。

仅使用 DES 密钥的风险是入侵者可以收集足够的使用相同密钥加密的加密文本消息，从而能够获取密钥并对这些消息进行解密。因此，安全系统（例如安全 NFS）需要经常更改密钥。

Kerberos 验证

Kerberos 是 MIT（麻省理工学院）开发的验证系统。Kerberos 中的某些加密基于 DES。Kerberos V4 支持不再作为安全 RPC 的一部分提供。但是，此发行版中包括使用 RPCSEC_GSS 的 Kerberos V5 客户端和服务器端实现。有关更多信息，请参见第 20 章。

Diffie-Hellman 验证

入侵者很难破解用于验证用户的 Diffie-Hellman (DH) 方法。客户机和服务器都有自己的私钥，它们将此私钥与公钥一起使用以设计公用密钥。私钥也称为**密钥**。客户机和服务器使用公用密钥来相互通信。公用密钥使用公认的加密功能（例如 DES）进行加密。

验证基于发送系统使用公用密钥加密当前时间的能力。然后，接收系统可以进行解密，并根据其当前时间进行检查。客户机和服务器上的时间必须同步。有关更多信息，请参见《System Administration Guide: Network Services》中的“Managing Network Time Protocol (Tasks)”。

公钥和私钥存储在 NIS 或 NIS+ 数据库中。NIS 将密钥存储在 `publickey` 映射中。NIS+ 将密钥存储在 `cred` 表中。这些文件包含所有潜在用户的公钥和私钥。

系统管理员负责设置 NIS 映射或 NIS+ 表，以及为每个用户生成公钥和私钥。将使用用户口令以加密格式存储私钥。此过程使得私钥只对此用户公开。

Diffie-Hellman 验证的实现

本节介绍客户机-服务器会话中使用 Diffie-Hellman 验证 (`AUTH_DH`) 的系列事务。

生成公钥和密钥

执行事务之前，管理员会运行 `newkey` 或 `nisaddcred` 命令来生成公钥和密钥。每个用户都拥有唯一的公钥和密钥。公钥存储在公共数据库中。密钥以加密格式存储在同一数据库中。`chkey` 命令可更改密钥对。

运行 keylogin 命令

通常，登录口令与安全 RPC 口令相同。在这种情况下，不需要 keylogin 命令。但是，如果口令不同，则用户必须登录，然后运行 keylogin 命令。

keylogin 命令提示用户键入安全 RPC 口令。此命令然后使用口令对密钥进行解密。随后，keylogin 命令将解密的密钥传递到 *keyserver* 程序。keyserver 是一种在每台计算机上都有本地实例的 RPC 服务。keyserver 会保存解密的密钥，并等待用户使用服务器启动安全 RPC 事务。

如果登录口令与 RPC 口令相同，则登录进程会将密钥传递到 keyserver。如果要求不同的口令，则用户必须始终运行 keylogin 命令。如果 keylogin 命令包括在用户的环境配置文件（例如 ~/.login、~/.cshrc 或 ~/.profile 文件）中，则用户登录时便会自动运行 keylogin 命令。

生成对话密钥

用户使用服务器启动事务时，将发生以下情况：

1. keyserver 随机生成一个对话密钥。
2. 内核使用此对话密钥以及其他材料对客户机的时间标记进行加密。
3. keyserver 在公钥数据库中查找服务器的公钥。有关更多信息，请参见 publickey(4) 手册页。
4. keyserver 使用客户机的密钥以及服务器的公钥来创建一个公用密钥。
5. keyserver 使用此公用密钥对此对话密钥进行加密。

初始联系服务器

然后，会将包含加密的时间标记和对话密钥的传输内容发送到服务器。此传输内容包括凭证和检验器。此凭证包含三个组件：

- 客户机的网络名称
- 使用公用密钥加密的对话密钥
- 使用对话密钥加密的“窗口”

此窗口显示客户机允许服务器时钟与客户机时间标记之间存在的时间差异。如果服务器时钟与时间标记之间的差异大于此窗口显示的值，则服务器会拒绝客户机的请求。正常情况下，不会出现这种拒绝，因为客户机在启动 RPC 会话之前会先与服务器进行同步。

客户机的检验器包含以下内容：

- 加密的时间标记
- 按 1 递减的指定窗口的已加密检验器

如果某人要模拟用户，则会需要窗口检验器。模拟者可以编写一个程序，从而只需插入随机位，而无需填写凭证和检验器的已加密字段。服务器将对话密钥解密为某一随机密钥。然后，服务器使用此密钥尝试对窗口和时间标记进行解密。结果是随机数字。但是，经过几千次尝试之后，随机窗口/时间标记对可能会通过验证系统。窗口检验器可降低假凭证通过验证的可能性。

解密对话密钥

服务器从客户机接收传输内容时，将发生以下情况：

1. 服务器的本地 keyserver 在公钥数据库中查找客户机的公钥。
2. keyserver 使用客户机的公钥以及服务器的密钥来推导公用密钥。此公用密钥与客户机计算所得的公用密钥相同。只有服务器和客户机才能计算公用密钥，因为此计算过程需要知道其中一个密钥。
3. 内核使用公用密钥对此对话密钥进行解密。
4. 内核调用 keyserver 以使用解密的对话密钥对客户机的时间标记进行解密。

在服务器上存储信息

服务器对客户机的时间标记进行解密之后，将在凭证表中存储四个信息项：

- 客户机的计算机名称
- 对话密钥
- 窗口
- 客户机的时间标记

服务器存储前三项供将来使用，存储客户机的时间标记以防止重放。服务器只接受时间上晚于最新时间标记的时间标记。因此，可保证拒绝任何重放的事务。

注 – 调用方的名称将隐含在这些事务中，此调用方必须通过某种方式进行验证。keyserver 不能使用 DES 验证来验证调用方，因为 keyserver 使用 DES 时会导致死锁。为了避免死锁，keyserver 将通过用户 ID (user ID, UID) 来存储密钥，并且只将请求授予本地 `root` 进程。

将检验器返回到客户机

服务器将检验器返回到客户机，其中包括以下内容：

- 服务器在其凭证高速缓存中记录的索引 ID
- 减 1 的客户机时间标记（由对话密钥加密）

从客户机的时间标记中减 1 的原因是确保此时间标记已过时。过时的时间标记不能再用作客户机检验器。

验证服务器

客户机将接收检验器并验证服务器。客户机知道只有服务器才能发送检验器，因为只有服务器知道客户机发送的时间标记。

处理事务

对于第一个事务之后的每一个事务，客户机都在其下一个事务中将索引 ID 返回到服务器。客户机还发送另一个加密的时间标记。服务器会将使用对话密钥加密的减 1 的客户机时间标记发送回来。

管理安全 RPC（任务列表）

以下任务列表说明为 NIS、NIS+ 和 NFS 配置安全 RPC 的过程。

任务	说明	参考
1. 启动 keyserver。	确保可以创建密钥以对用户进行验证。	第 289 页中的 “如何重新启动安全 RPC Keyserver”
2. 在 NIS+ 主机上设置凭证。	确保主机上的 root 用户可以在 NIS+ 环境中进行验证。	第 290 页中的 “如何为 NIS+ 主机设置 Diffie-Hellman 密钥”
3. 为 NIS+ 用户提供密钥。	使用户可以在 NIS+ 环境中进行验证。	第 291 页中的 “如何为 NIS+ 用户设置 Diffie-Hellman 密钥”
4. 在 NIS 主机上设置凭证。	确保主机上的 root 用户可以在 NIS 环境中进行验证。	第 292 页中的 “如何为 NIS 主机设置 Diffie-Hellman 密钥”
5. 为 NIS 用户提供密钥。	使用户可以在 NIS 环境中进行验证。	第 293 页中的 “如何为 NIS 用户设置 Diffie-Hellman 密钥”
6. 通过验证共享 NFS 文件。	使 NFS 服务器可以使用验证来安全地保护共享的文件系统。	第 294 页中的 “如何通过 Diffie-Hellman 验证共享 NFS 文件”

使用安全 RPC 管理验证

通过要求对已挂载 NFS 文件系统的使用进行验证，可以增强网络的安全性。

▼ 如何重新启动安全 RPC Keyserver

1. **承担主管理员角色，或成为超级用户。**

   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2. **检验 keyserv 守护进程是否正在运行。**

```
# svcs \*keyserv\*

STATE    STIME   FMRI

disabled Dec_14  svc:/network/rpc/keyserv
```

3. **如果 keyserver 服务未联机，则启用此服务。**

```
# svcadm enable network/rpc/keyserv
```

▼ 如何为 NIS+ 主机设置 Diffie-Hellman 密钥

应该针对 NIS+ 域中的每个主机执行此过程。以 root 身份运行 keylogin 命令之后，服务器便会具有 mech_dh 的 GSS-API 接收器凭证，而客户机具有 GSS-API 启动器凭证。

有关 NIS+ 安全性的详细说明，请参见《System Administration Guide: Naming and Directory Services (NIS+)》。

1 **承担主管理员角色，或成为超级用户。**

主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2 **在名称服务中启用 publickey 表。**

将以下行添加到 /etc/nsswitch.conf 文件中：

```
publickey: nisplus
```

3 **初始化 NIS+ 客户机。**

```
# nisinit -cH hostname
```

其中，*hostname* 是其表中包含客户机系统项的受信任 NIS+ 服务器的名称。

4 **将客户机添加到 cred 表中。**

键入以下命令：

```
# nisaddcred local

# nisaddcred des
```

5 **使用 keylogin 命令检验此设置。**

如果系统提示键入口令，则表示此过程已经成功。

```
# keylogin

Password:
```

示例 15–1 在 NIS+ 客户机上为 root 设置新密钥

以下示例使用主机 pluto 将 earth 设置为 NIS+ 客户机。您可以忽略警告。系统将接受 keylogin 命令，检验 earth 是否已正确设置为安全 NIS+ 客户机。

```
# nisinit -cH pluto

NIS Server/Client setup utility.

This system is in the example.com. directory.
```

```
Setting up NIS+ client ...

All done.

# nisaddcred local

# nisaddcred des

DES principal name : unix.earth@example.com

Adding new key for unix.earth@example.com (earth.example.com.)

Network password:    <键入口令>

Warning, password differs from login password.

Retype password:     <重新键入口令>

# keylogin

Password:            <键入口令>

#
```

▼ 如何为 NIS+ 用户设置 Diffie-Hellman 密钥

应该针对 NIS+ 域中的每个用户执行此过程。

1. **承担主管理员角色，或成为超级用户。**
   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2. **在根主服务器上将用户添加到 `cred` 表中。**
   键入以下命令：
   # nisaddcred -p unix.*UID*@*domain-name* **-P** *username*.*domain-name*. **des**
   请注意，在这种情况下，*username.domain-name* 必须以点 (.) 结束。

3. **通过以客户身份登录并键入 `keylogin` 命令来检验此设置。**

示例 15–2 为 NIS+ 用户设置新密钥

在以下示例中，将为用户 `jdoe` 提供 Diffie-Hellman 验证密钥。

```
# nisaddcred -p unix.1234@example.com -P jdoe.example.com. des

DES principal name : unix.1234@example.com

Adding new key for unix.1234@example.com (jdoe.example.com.)

Password:           <键入口令>

Retype password:    <重新键入口令>

# rlogin rootmaster -l jdoe

% keylogin

Password:           <键入口令>

%
```

▼ 如何为 NIS 主机设置 Diffie-Hellman 密钥

应该针对 NIS 域中的每个主机执行此过程。

1. **承担主管理员角色，或成为超级用户。**
   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2. **在名称服务中启用 `publickey` 映射。**
   将以下行添加到 `/etc/nsswitch.conf` 文件中：

   ```
   publickey: nis
   ```

3. **使用 `newkey` 命令创建一个新的密钥对。**

   ```
   # newkey -h hostname
   ```

   其中，*hostname* 是客户机的名称。

示例 15–3 在 NIS 客户机上为 `root` 设置新密钥

在以下示例中，会将 `earth` 设置为安全 NIS 客户机。

```
# newkey -h earth

Adding new key for unix.earth@example.com

New Password:       <键入口令>
```

```
Retype password:     <重新键入口令>

Please wait for the database to get updated...

Your new key has been successfully stored away.

#
```

▼ 如何为 NIS 用户设置 Diffie-Hellman 密钥

应该针对 NIS 域中的每个用户执行此过程。

开始之前 只有系统管理员在登录到 NIS 主服务器时才能为用户生成新密钥。

1. **承担主管理员角色，或成为超级用户。**
   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2. **为用户创建新密钥。**
   # newkey -u *username*
   其中，*username* 是用户的名称。系统将提示键入口令。您可以键入通用口令。将使用此通用口令以加密格式存储私钥。

3. **通知用户登录并键入 chkey -p 命令。**
   此命令允许用户使用只有其自己知道的口令重新加密其私钥。

注 – 可以使用 chkey 命令为用户创建新密钥对。

示例 15–4 在 NIS 中设置并加密新用户密钥

在此示例中，超级用户将设置密钥。

```
# newkey -u jdoe

Adding new key for unix.12345@example.com

New Password:        <键入口令>

Retype password:     <重新键入口令>

Please wait for the database to get updated...
```

```
Your new key has been successfully stored away.

#
```

然后，用户 jdoe 将使用私人口令对此密钥进行重新加密。

```
% chkey -p

Updating nis publickey database.

Reencrypting key for unix.12345@example.com

Please enter the Secure-RPC password for jdoe:     <键入口令>

Please enter the login password for jdoe:          <键入口令>

Sending key change request to centralexample...
```

▼ 如何通过 Diffie-Hellman 验证共享 NFS 文件

此过程通过要求访问验证来保护 NFS 服务器上的共享文件系统。

开始之前 必须在网络中启用 Diffie-Hellman 公钥验证。要在网络中启用验证，请执行以下操作之一：

- 第 290 页中的 “如何为 NIS+ 主机设置 Diffie-Hellman 密钥”
- 第 292 页中的 “如何为 NIS 主机设置 Diffie-Hellman 密钥”

1. **成为超级用户或承担拥有系统管理配置文件的角色。**

   系统管理员角色拥有系统管理配置文件。要创建该角色并将其指定给用户，请参见第 186 页中的 “配置 RBAC（任务列表）”。

2. **在 NFS 服务器上，通过 Diffie-Hellman 验证共享文件系统。**

   # share -F nfs -o sec=dh */filesystem*

   其中，*filesystem* 是要共享的文件系统。

   -o sec=dh 选项意味着现在需要通过 AUTH_DH 验证来访问文件系统。

3. **在 NFS 客户机上，通过 Diffie-Hellman 验证挂载文件系统。**

   # mount -F nfs -o sec=dh *server:filesystem mount-point*

server	共享 *filesystem* 的系统的名称
filesystem	共享的文件系统的名称，例如 opt
mount-point	挂载点的名称，例如 /opt

-o sec=dh 选项通过 AUTH_DH 验证挂载文件系统。

◆ ◆ ◆ 第 16 章

使用 PAM

本章介绍可插拔验证模块 (Pluggable Authentication Module, PAM) 框架。PAM 提供了一种向 Solaris 操作系统 (Solaris Operating System, Solaris OS) 中“插入”验证服务的方法。PAM 会在访问系统时为多项验证服务提供支持。

- 第 297 页中的 “PAM（概述）”
- 第 299 页中的 “PAM（任务）”
- 第 302 页中的 “PAM 配置文件（参考）”

PAM（概述）

使用可插拔验证模块 (Pluggable Authentication Module, PAM) 框架，可以“插入”新的验证服务，而无需更改系统登录服务，例如 `login`、`ftp` 和 `telnet`。还可以使用 PAM 将 UNIX 登录与其他安全机制（如 Kerberos）进行集成。也可以使用此框架来“插入”帐户、凭证、会话以及口令管理的机制。

使用 PAM 的益处

使用 PAM 框架，可以为用户验证配置系统登录服务（如 `ftp`、`login`、`telnet` 或 `rsh`）。PAM 提供的一些益处如下所示：

- 灵活的配置策略
  - 按应用程序的验证策略
  - 选择缺省验证机制的功能
  - 在高安全性系统中要求提供多个口令的功能
- 易于最终用户使用
  - 如果对于不同的验证服务口令都相同，则无需重新键入口令
  - 可以提示用户输入多个验证服务的口令，而无需用户键入多个命令
- 可以将可选选项传送到用户验证服务
- 可以实现特定于站点的安全策略，而无需更改系统登录服务

PAM 组件

PAM 软件由一个库、各种服务模块以及一个配置文件组成。其中还包括可利用这些 PAM 接口的 Solaris 命令或守护进程。

下图说明了系统登录应用程序、PAM 库、pam.conf 文件和 PAM 服务模块之间的关系。

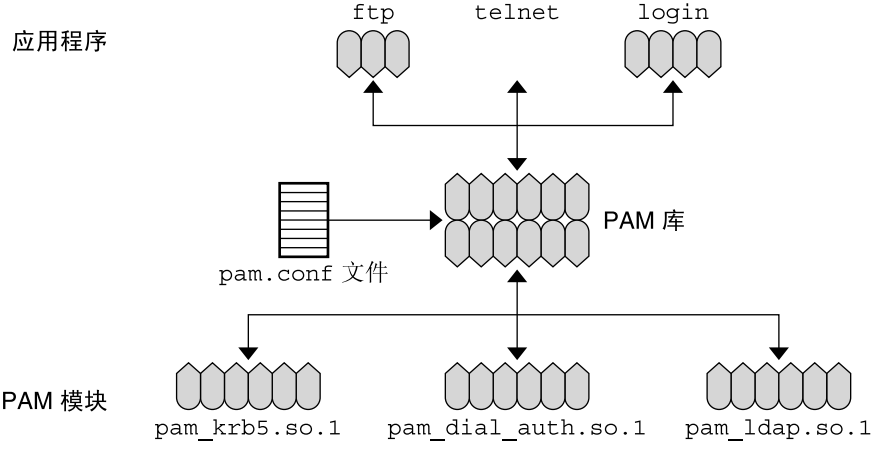


图 16–1 PAM 工作原理

系统登录应用程序（如 ftp、telnet 和 login）使用 PAM 库来调用配置策略。配置策略在 pam.conf 文件中定义。pam.conf 文件定义要使用的模块，以及每个应用程序使用这些模块的顺序。模块中的结果基于模块响应和已配置的控制标志。这些结果通过库传送回应用程序。

PAM 框架

PAM 框架提供了一种使用**堆叠**借助多项服务来验证用户的方法。根据配置，系统可以提示用户输入每种验证方法的口令。验证服务的使用顺序通过 PAM 配置文件确定。

PAM 库可提供框架以装入相应模块和管理堆叠过程。PAM 库提供了一种可在其中插入所有模块的通用结构。有关更多信息，请参见 pam_sm(3PAM) 手册页。

Solaris 10 发行版对 PAM 所做的更改

Solaris 10 发行版对可插拔验证模块 (Pluggable Authentication Module, PAM) 框架做了以下更改：

- 现在，pam_authtok_check 模块允许使用 /etc/default/passwd 文件中的新可调参数执行严格的口令检查。这些新参数定义以下各项：
  - 用逗号分隔的字典文件列表，这些字典文件用于检查口令中的常用字典字

  - 新口令与旧口令之间所需的最小差别
  - 新口令中必须用到的字母字符或非字母字符的最少个数
  - 新口令中必须用到的大写字母或小写字母的最少个数
  - 允许的连续重复字符的个数
- pam_unix_auth 模块可针对本地用户实现帐户锁定。帐户锁定通过 /etc/security/policy.conf 中的 LOCK_AFTER_RETRIES 参数以及 /etc/user_attr 中的 lock_after-retries 密钥启用。有关更多信息，请参见 policy.conf(4) 和 user_attr(4) 手册页。
- 定义了一个新的 binding 控制标志。此控制标志在 pam.conf(4) 手册页和第 303 页中的 “PAM 控制标志”中进行了介绍。
- pam_unix 模块已删除，并替换为一组等效或功能更强的服务模块。Solaris 9 发行版已引入其中的许多模块。替换模块列表如下：
  - pam_authtok_check
  - pam_authtok_get
  - pam_authtok_store
  - pam_dhkeys
  - pam_passwd_auth
  - pam_unix_account
  - pam_unix_auth
  - pam_unix_cred
  - pam_unix_session
- pam_unix_auth 模块的功能已分解为两个模块。现在，pam_unix_auth 模块可以检验用户的口令是否正确。新的 pam_unix_cred 模块可提供建立用户凭证信息的功能。
- 对 pam_krb5 模块进行扩充是为了使用 PAM 框架来管理 Kerberos 凭证高速缓存。
- 添加了新的 pam_deny 模块。此模块可用于拒绝对服务的访问。缺省情况下，不使用 pam_deny 模块。有关更多信息，请参见 pam_deny(5) 手册页。

PAM（任务）

本节介绍使 PAM 框架使用特定安全策略所需执行的一些任务。应注意与 PAM 配置文件关联的某些安全问题。有关安全问题的信息，请参见第 300 页中的 “规划 PAM 实现”。

PAM（任务列表）

任务	说明	参考
规划 PAM 安装。	开始软件配置过程之前，考虑配置问题并做出相关决定。	第 300 页中的 “规划 PAM 实现”

任务	说明	参考
添加新的 PAM 模块。	有时，必须写入并安装特定于站点的模块，以满足通用软件不包括的要求。此过程说明如何安装这些新的 PAM 模块。	第 300 页中的 “如何添加 PAM 模块”
阻止访问 `~/.rhosts`。	通过阻止访问 `~/.rhosts` 来进一步提高安全性。	第 301 页中的 “如何使用 PAM 防止从远程系统进行 Rhost 样式的访问”
启动错误日志。	通过 `syslog` 启动 PAM 错误消息日志。	第 301 页中的 “如何记录 PAM 错误报告”

规划 PAM 实现

所提供的 `pam.conf` 配置文件可实现标准的 Solaris 安全策略。此策略应适用于许多情况。如果需要实现其他安全策略，则应考虑以下问题：

- 确定需求，特别是应选择的 PAM 服务模块。
- 标识需要特殊配置选项的服务。使用 `other`（如果适用）。
- 决定运行模块的顺序。
- 选择每个模块的控制标志。有关所有控制标志的更多信息，请参见第 303 页中的 “PAM 控制标志”。
- 选择每个模块必需的所有选项。每个模块的手册页应列出所有的特殊选项。

以下是更改 PAM 配置文件之前要考虑的一些建议：

- 对每种模块类型使用 `other` 项，以便 `/etc/pam.conf` 中不必包括每个应用程序。
- 确保考虑 `binding`、`sufficient` 和 `optional` 控制标志所涉及的安全问题。
- 查看与模块关联的手册页。这些手册页有助于您了解每个模块的工作方式、可用的选项，以及堆叠模块之间的交互。

注意 – 如果 PAM 配置文件配置错误或者被损坏，则可能没有用户能登录。由于 `sulogin` 命令不使用 PAM，因此，需要超级用户口令才能将计算机引导至单用户模式并修复问题。

更改 `/etc/pam.conf` 文件之后，在您仍具有系统访问权限的情况下，尽可能多地检查此文件以更正问题。对更改可能影响到的所有命令进行测试。例如，向 `telnet` 服务中添加新模块。在此示例中，将使用 `telnet` 命令并检验所做更改是否使服务按预期方式运行。

▼ 如何添加 PAM 模块

此过程说明如何添加新的 PAM 模块。可以创建新模块以提供特定于站点的安全策略或支持第三方应用程序。

1. **成为超级用户或承担等效角色。**

   角色包含授权和具有一定权限的命令。有关角色的更多信息，请参见第 186 页中的 “配置 RBAC（任务列表）” 。

2. **确定应使用的控制标志和其他选项。**

   有关模块的信息，请参阅第 304 页中的 “PAM 模块” 。

3. **确保设置了拥有权和权限，以便模块文件由 root 拥有，并且权限为 555。**

4. **编辑 PAM 配置文件 /etc/pam.conf，并将此模块添加到相应的服务中。**

5. **检验是否正确添加了模块。**

   必须在重新引导系统**之前**进行测试，以防此配置文件配置错误。在重新引导系统之前，使用直接服务（例如 rlogin 或 telnet ）登录，并运行 su 命令。此服务可能是引导系统时仅产生一次的守护进程。因此，必须先重新引导系统，然后才能检验是否已添加模块。

▼ 如何使用 PAM 防止从远程系统进行 Rhost 样式的访问

1. **成为超级用户或承担等效角色。**

   角色包含授权和具有一定权限的命令。有关角色的更多信息，请参见第 186 页中的 “配置 RBAC（任务列表）” 。

2. **从 PAM 配置文件中删除所有包括 rhosts_auth.so.1 的行。**

   此步骤用于防止在 rlogin 会话期间读取 ~/.rhosts 文件。因此，此步骤可防止从远程系统对本地系统进行未经验证的访问。无论 ~/.rhosts 或 /etc/hosts.equiv 文件是否存在或包含什么内容，所有 rlogin 访问都需要口令。

3. **禁用 rsh 服务。**

   要防止对 ~/.rhosts 文件进行其他未经验证的访问，请记住要禁用 rsh 服务。

   ```
   # svcadm disable network/shell
   ```

▼ 如何记录 PAM 错误报告

1. **成为超级用户或承担等效角色。**

   角色包含授权和具有一定权限的命令。有关角色的更多信息，请参见第 186 页中的 “配置 RBAC（任务列表）” 。

2. **为所需的日志级别配置 /etc/syslog.conf 文件。**

   有关日志级别的更多信息，请参见 syslog.conf(4)。

3 **刷新 syslog 守护进程的配置信息。**

```
# svcadm refresh system/system-log
```

PAM 配置文件（参考）

PAM 配置文件 pam.conf 确定要使用的验证服务模块，以及这些模块的使用顺序。可以修改此文件来为每个系统登录应用程序选择验证模块。

PAM 配置文件语法

PAM 配置文件所包含项的语法如下：

service-name module-type control-flag module-path module-options

service-name	系统登录服务的名称，例如 ftp 、login、telnet。
module-type	服务的模块类型。有关更多信息，请参见第 302 页中的 “PAM 模块类型”。
control-flag	确定模块的延续行为或失败行为。
module-path	指定实现安全策略的库对象的路径。
module-options	指定传送到服务模块的选项。

通过以 #（井号）开始行，可以向 pam.conf 文件中添加注释。可以使用空格或制表符来分隔字段。

注 – 如果在 PAM 配置文件的项中发现错误，则会生成 syslog 错误消息。如果此错误是请求服务的项，则此服务可能会返回一个错误。

PAM 的服务名称

每项服务的特定服务名称应该在该服务的手册页中进行介绍。例如，sshd(1M) 手册页列出 sshd 命令的所有 PAM 服务名称。

PAM 模块类型

您需要了解 PAM 模块类型，因为这些类型定义模块的接口。以下是 PAM 模块的类型：

- **帐户模块**，检查口令生命期、帐户到期日期和访问限制。通过验证模块对用户身份进行验证之后，帐户模块会确定是否应授予该用户对系统的访问权限。
- **验证模块**，为用户提供验证。此类模块还允许设置、刷新或销毁凭证。

- **口令模块**，允许对用户口令进行更改。
- **会话模块**，管理登录会话的打开与关闭。这些模块还可以记录活动，或在会话结束后进行清除。

PAM 控制标志

使用 PAM 服务模块的请求返回以下三种状态之一：

- 成功－满足安全策略
- 失败－未满足安全策略
- 忽略－此请求未参与策略请求

栈中的每个模块都可以确定请求的成功或失败。要确定模块的延续行为或失败行为，必须为 PAM 配置文件中的每个项选择一个**控制标志**。

延续行为定义是否检查后面的所有模块。根据特定模块的响应，可以决定跳过所有其他模块。

失败行为定义如何记录或报告错误消息。失败信息既可以是可选信息，也可以是必需信息。**必需的失败信息**会导致此请求失败，即使其他模块成功也是如此。**可选的失败信息**不会始终导致此请求失败。

控制标志如下所示：

- `binding`－使用此控制标志，如果模块响应成功，并且先前带有 `required` 标志的模块都没有失败，则 PAM 会跳过其余模块并返回成功信息。如果返回失败信息，则 PAM 会记录必需的失败信息，然后继续处理栈。

  除模块响应成功情况下不再检查任何其他模块以外，`binding` 控制标志类似于 `required` 控制标志。无论其他模块如何响应，使用此标志的模块中的失败信息会阻止请求成功。如果先前的必需模块都响应成功，则使用此标志的模块中的成功信息会使请求成功。

- `required`－使用此控制标志，如果模块响应成功，则 PAM 会记录必需的成功信息并继续检查后面的所有模块。如果此模块响应失败，并且此失败信息是第一个必需的失败信息，则 PAM 会保存错误消息并继续检查栈。如果此失败信息不是第一个失败信息，则 PAM 只会继续检查栈。此标志允许处理整个序列，从而不会泄露可帮助攻击者进行攻击的信息。攻击者可找出的所有信息就是请求失败。

  如果某特定模块必须响应成功才能使请求成功，则应使用 `required` 控制标志。无论其他模块如何响应，使用此标志的模块中的失败信息会阻止请求成功。使用此标志的模块中的成功信息并不表示请求成功。栈中带有 `required`、`requisite` 或 `binding` 控制标志的其他模块必须都响应成功，请求才会成功。

- `requisite`－使用此控制标志，如果模块响应成功，则 PAM 会记录必需的成功信息并继续检查后面的所有模块。如果此模块响应失败，则 PAM 会记录必需的失败信息，返回第一个必需失败信息的错误消息，然后跳过任何其他检查。

除模块响应失败情况下不再检查任何其他模块以外，requisite 控制标志类似于 required 控制标志。无论其他模块如何响应，使用此标志的模块中的失败信息会阻止请求成功。使用此标志的模块中的成功信息并不表示请求成功。栈中带有 required、requisite 或 binding 控制标志的其他模块必须都响应成功，请求才会成功。

- optional－使用此控制标志，如果模块响应成功，则 PAM 会记录可选的成功信息并继续检查栈。如果此模块响应失败，则 PAM 会记录可选的失败信息并继续检查栈。

  当栈中的成功验证足以对用户进行验证时，应使用 optional 控制标志。仅当此特定服务无需成功执行时才应使用此标志。请求的成功或失败由必需的失败信息或成功信息确定。

  如果用户需要具有与特定服务关联的权限才能完成其工作，则不应将模块标记为 optional。

- sufficient－使用此控制标志，如果模块响应成功，并且先前带有 required 标志的模块都没有失败，则 PAM 会跳过其余模块并返回成功信息。如果此模块响应失败，则 PAM 会记录可选的失败信息并继续检查栈。

  除模块响应成功情况下不再检查任何其他模块以外，sufficient 控制标志类似于 optional 控制标志。如果先前的 required 模块都响应成功，则使用此标志的模块中的成功信息会使请求成功。如果其他模块都响应失败，则使用此标志的模块中的失败信息会导致请求失败。

有关这些控制标志的更多信息，请参见下一节，其中介绍了通用 /etc/pam.conf 文件。

PAM 模块

每个 PAM 模块都可实现一种特定的功能。设置 PAM 验证时，需要指定模块和模块类型，后者定义模块执行的操作。一个模块可以实现多个模块类型，例如 auth、account、session 或 password。

每个模块的路径由已安装的 Solaris 发行版中提供的指令集确定。对于 32 位模块，模块路径为 /usr/lib/security。对于 64 位模块，路径为 /usr/lib/security/*$ISA*。有关更多信息，请参见 isalist(5) 手册页。

Solaris PAM 模块的完整列表位于 /usr/lib/security/*$ISA* 中。每个模块都有关联的手册页，其中介绍了应用的模块类型以及所有的特殊选项。

出于安全原因，这些模块文件必须由 root 拥有，并且禁止使用 group 或 other 权限写入。如果文件并非由 root 拥有，则 PAM 不会装入模块。

通用 pam.conf 文件的示例

通用 /etc/pam.conf 文件包括以下项：

```
login   auth requisite          pam_authtok_get.so.1

login   auth required           pam_dhkeys.so.1

login   auth required           pam_unix_cred.so.1

login   auth required           pam_unix_auth.so.1

login   auth required           pam_dial_auth.so.1
```

运行 login 命令时，必须针对 pam_authtok_get、pam_dhkeys、pam_auth_cred、pam_auth_unix 和 pam_dial_auth 模块成功执行验证。pam_authtok_get 项中的 requisite 标志表示，如果此模块响应失败，则不再检查任何其他模块。但是，如果此模块响应成功，则会继续检查其余模块。如果针对所有模块的验证都失败，则验证请求也会失败。

```
rlogin  auth sufficient         pam_rhosts_auth.so.1

rlogin  auth requisite          pam_authtok_get.so.1

rlogin  auth required           pam_dhkeys.so.1

rlogin  auth required           pam_unix_cred.so.1

rlogin  auth required           pam_unix_auth.so.1
```

对于 rlogin 命令，sufficient 控制标志表示通过 pam_rhosts_auth 模块进行的验证足以使验证请求成功。无需执行任何其他检查。如果通过 pam_rhosts_auth 进行的验证失败，则通过 pam_authtok_get、pam_dhkeys、pam_auth_cred 和 pam_unix_auth 模块进行的验证必须成功。虽然其他模块中的失败信息会阻止成功验证，但是 pam_rhosts_auth 模块中的失败信息不会。此外，与 login 中的项相同，pam_authtok_get 项中的 requisite 控制标志表示如果此模块响应失败，则验证请求失败，并且不再检查任何其他模块。

```
other   session required        pam_unix_session.so.1
```

使用 other 服务名称，可以为 pam.conf 文件中未包括的任何其他命令设置缺省服务。other 服务名称简化了文件管理，因为只需一个项就可涵盖使用同一模块的许多服务。此外，other 服务名称用作“全面管理功能”时，可以确保每个访问都由一个模块来表示。

module-path 的项是“相对于根目录的”。如果为 *module-path* 指定的文件名未以斜杠 (/) 开始，则文件名前面为路径 /usr/lib/security/$ISA。必须对位于其他目录中的模块使用全路径名。可以在每个模块的手册页中找到 *module-options* 的值。

◆ ◆ ◆ 第 17 章

17 使用 SASL

本章介绍有关简单身份验证和安全层 (Simple Authentication and Security Layer, SASL) 的信息。

- 第 307 页中的 “SASL（概述）”
- 第 307 页中的 “SASL（参考）”

SASL（概述）

简单身份验证和安全层 (Simple Authentication and Security Layer, SASL) 是一种为网络协议提供验证和可选安全性服务的框架。应用程序将调用 SASL 库 `/usr/lib/libsasl.so`，此库提供应用程序与各种 SASL 机制之间的胶合层。验证过程及提供可选安全性服务时会使用 SASL 机制。Solaris 10 发行版提供的 SASL 版本由经过少量更改的 Cyrus SASL 派生而来。

SASL 提供以下服务：

- 装入任何插件
- 确定应用程序必需的安全选项以帮助选择安全机制
- 列出应用程序可使用的插件
- 为特定验证的尝试从可用机制列表中选择最佳机制
- 在应用程序和所选机制之间路由验证数据
- 将有关 SASL 协商的信息提供给应用程序

SASL（参考）

本节提供有关 Solaris 10 发行版的 SASL 实现的信息。

SASL 插件

SASL 插件提供对安全机制、用户标准化和辅助属性检索的支持。缺省情况下，动态装入的 32 位插件安装在 `/usr/lib/sasl` 中，64 位插件安装在 `/usr/lib/sasl/` *$ISA* 中。Solaris 10 发行版中提供了以下安全机制插件：

`crammd5.so.1`	CRAM-MD5，仅支持验证，不支持授权。
`digestmd5.so.1`	DIGEST-MD5，支持验证、完整性、保密性和授权。
`gssapi.so.1`	GSSAPI，支持验证、完整性、保密性和授权。GSSAPI 安全机制要求有效的 Kerberos 基础结构。
`plain.so.1`	PLAIN，支持验证和授权。

此外，EXTERNAL 安全机制插件和 INTERNAL 用户标准化插件内置在 `libsasl.so.1` 中。EXTERNAL 机制支持验证和授权。如果外部安全源提供该机制，则该机制支持完整性和保密性。如果用户名需要，INTERNAL 插件将添加领域名称。

目前，Solaris 10 发行版不提供任何 `auxprop` 插件。要使 CRAM-MD5 和 DIGEST-MD5 机制插件在服务器端能完全运行，用户必须提供 `auxprop` 插件以检索明文口令。PLAIN 插件还需要其他的支持，以对口令进行验证。支持口令验证的有：对服务器应用程序的回调、`auxprop` 插件、`saslauthd` 或 `pwcheck`。Solaris 发行版中未提供 `salauthd` 和 `pwcheck` 守护进程。要获取更好的互操作性，可使用 `mech_list` SASL 选项将服务器应用程序限制为可完全运行的那些机制。

SASL 环境变量

缺省情况下，客户机验证名称被设置为 `getenv("LOGNAME")`。客户机或插件可以重置此变量。

SASL 选项

使用可在 `/etc/sasl/`*app*`.conf` 文件中设置的选项，可以在服务器端修改 `libsasl` 和插件的行为。变量 *app* 是服务器定义的应用程序名称。服务器 *app* 的文档应指定该应用程序名称。

Solaris 10 发行版支持以下选项：

`auto_transition`	用户成功进行纯文本验证后自动将其转换到其他机制。
`auxprop_login`	列出要使用的辅助属性插件的名称。
`canon_user_plugin`	选择要使用的 canon_user 插件。
`mech_list`	列出允许服务器应用程序使用的机制。
`pwcheck_method`	列出用于验证口令的机制。目前，`auxprop` 是唯一允许的值。

`reauth_timeout` 设置缓存验证信息以便进行快速重新验证的时间（以分钟为单位）。此选项由 DIGEST-MD5 插件使用。将此选项设置为 0 将禁用重新验证。

Solaris 10 发行版不支持以下选项：

`plugin_list` 列出可用机制。不使用该选项是因为它会更改动态装入插件的行为。

`saslauthd_path` 定义用于与 `saslauthd` 守护进程通信的 saslauthd 门的位置。Solaris 10 发行版中不包括 `saslauthd` 守护进程。因此，也不包括此选项。

`keytab` 定义 GSSAPI 插件使用的 `keytab` 文件的位置。可使用 `KRB5_KTNAME` 环境变量代替此选项来设置缺省 `keytab` 位置。

Cyrus SASL 中不包括以下选项。但是，Solaris 10 发行版中添加了这些选项：

`use_authid` 创建 GSS 客户机安全性上下文时获取客户机凭证而不是使用缺省凭证。缺省情况下，将使用缺省客户机 Kerberos 身份。

`log_level` 为服务器设置需要的日志级别。

◆ ◆ ◆ 第 18 章

18 使用 Solaris 安全 Shell（任务）

使用 Solaris 安全 Shell，用户可以通过不安全的网络安全地访问远程主机。该 shell 提供了用于远程登录和远程文件传输的命令。本章包含以下主题：

- 第 311 页中的 “Solaris 安全 Shell（概述）”
- 第 314 页中的 “Solaris 10 发行版中 Solaris 安全 Shell 的增强功能”
- 第 315 页中的 “配置 Solaris 安全 Shell（任务列表）”
- 第 319 页中的 “使用 Solaris 安全 Shell（任务列表）”

有关参考信息，请参见第 19 章。

Solaris 安全 Shell（概述）

在 Solaris 安全 Shell 中，提供了使用口令、公钥，或同时使用二者的验证。所有网络通信都将被加密。因此，Solaris 安全 Shell 可防止可能的入侵者读取被拦截的通信。Solaris 安全 Shell 还可防止入侵者欺骗系统。

Solaris 安全 Shell 还可用作即时 virtual private network, VPN（虚拟专用网络）。VPN 可以转发 X 窗口系统通信，或通过加密的网络链路连接本地计算机和远程计算机之间的各个端口号。

使用 Solaris 安全 Shell，可以执行以下操作：

- 通过不安全的网络安全地登录到其他主机。
- 在两台主机之间安全地复制文件。
- 在远程主机上安全地运行命令。

Solaris 安全 Shell 支持两种版本的安全 Shell 协议。版本 1 是协议的原始版本。版本 2 更安全，该版本修正了版本 1 的一些基本安全设计缺陷。版本 1 仅提供用于协助用户迁移到版本 2。强烈建议用户不要使用版本 1。

注 – 在下文中，v1 用于表示版本 1，v2 用于表示版本 2。

Solaris 安全 Shell 验证

Solaris 安全 Shell 提供公钥和口令方法来验证与远程主机的连接。公钥验证是一种比口令验证更强大的验证机制，因为私钥从不通过网络传送。

请按以下顺序尝试这些验证方法。如果配置不满足验证方法的要求，请尝试下一种方法。

- **GSS-API** – 使用 `mech_krb5` (Kerberos V) 和 `mech_dh` (AUTH_DH) 等 GSS-API 机制的凭证来验证客户机和服务器。有关 GSS-API 的更多信息，请参见《Solaris Security for Developers Guide》中的“Introduction to GSS-API”。
- **基于主机的验证** – 使用主机密钥和 rhosts 文件。使用客户机的 RSA 和 DSA 公共/专用主机密钥验证客户机。使用 rhosts 文件向用户授权使用客户机。
- **公钥验证** – 验证用户的 RSA 和 DSA 公钥/私钥。
- **口令验证** – 使用 PAM 验证用户。 v2 中的键盘验证方法允许 PAM 的任意提示。有关更多信息，请参见 `sshd`(1M) 手册页中的 `SECURITY` 部分。

下表显示了验证正在尝试登录到远程主机的用户的要求。该用户位于本地主机（客户机）上。远程主机（服务器）正在运行 `sshd` 守护进程。下表显示了 Solaris 安全 Shell 验证方法、兼容的协议版本和主机要求。

表 18–1 Solaris 安全 Shell 的验证方法

验证方法（协议版本）	本地主机（客户机）要求	远程主机（服务器）要求
GSS-API (v2)	GSS 机制的启动器凭证。	GSS 机制的接收器凭证。有关更多信息，请参见第 332 页中的 “获取 Solaris 安全 Shell 中的 GSS 凭证” 。

基于主机 (v2)	用户帐户
	`/etc/ssh/ssh_host_rsa_key` 或 `/etc/ssh/ssh_host_dsa_key` 中的本地主机私钥
	`/etc/ssh/ssh_config` 中的 `HostbasedAuthentication yes`
		用户帐户
		`/etc/ssh/known_hosts` 或 `~/.ssh/known_hosts` 中的本地主机公钥
		`/etc/ssh/sshd_config` 中的 `HostbasedAuthentication yes`
		`/etc/ssh/sshd_config` 中的 `IgnoreRhosts no`
		`/etc/shosts.equiv`、`/etc/hosts.equiv`、`~/.rhosts` 或 `~/.shosts` 中的本地主机项

表 18–1 Solaris 安全 Shell 的验证方法 （续）

验证方法（协议版本）	本地主机（客户机）要求	远程主机（服务器）要求
RSA 或 DSA 公钥 (v2)	用户帐户 ~/.ssh/id_rsa 或 ~/.ssh/id_dsa 中的私钥 ~/.ssh/id_rsa.pub 或 ~/.ssh/id_dsa.pub 中的用户公钥	用户帐户 ~/.ssh/authorized_keys 中的用户公钥
RSA 公钥 (v1)	用户帐户 ~/.ssh/identity 中的私钥 ~/.ssh/identity.pub 中的用户公钥	用户帐户 ~/.ssh/authorized_keys 中的用户公钥
键盘交互 (v2)	用户帐户	用户帐户 支持 PAM，包括触发口令生命期后的任意提示和口令更改。
基于口令（v1 或 v2）	用户帐户	用户帐户 支持 PAM。
仅 .rhosts (v1)	用户帐户	用户帐户 /etc/ssh/sshd_config 中的 IgnoreRhosts no /etc/shosts.equiv、/etc/hosts.equiv、~/.shosts 或 ~/.rhosts 中的本地主机项
仅在服务器上使用 RSA (v1) 的 .rhosts	用户帐户 /etc/ssh/ssh_host_rsa1_key 中的本地主机公钥	用户帐户 /etc/ssh/ssh_known_hosts 或 ~/.ssh/known_hosts 中的本地主机公钥 /etc/ssh/sshd_config 中的 IgnoreRhosts no /etc/shosts.equiv、/etc/hosts.equiv、~/.shosts 或 ~/.rhosts 中的本地主机项

企业中的 Solaris 安全 Shell

有关 Solaris 系统中的安全 Shell 的全面介绍，请参见由 Jason Reid 编著的《Secure Shell in the Enterprise》，2003 年 6 月出版，ISBN 为 0-13-142900-0。该书是 Sun Microsystems Press 出版的 Sun BluePrints 丛书的一部分。

有关联机信息，请导航至 Sun 的 BigAdmin System Administration Portal 网站 http://www.sun.com/bigadmin。单击 docs.sun.com，然后在 Other documentation sites 下单击 Sun BluePrints。然后依次单击 Archives by Subject 和 Security。该文档集包括以下文章：

- 《Role Based Access Control and Secure Shell – A Closer Look At Two Solaris Operating Environment Security Features》
- 《Integrating the Secure Shell Software》

- 《Configuring the Secure Shell Software》

Solaris 10 发行版中 Solaris 安全 Shell 的增强功能

从 Solaris 9 发行版开始，Solaris 安全 Shell 中引入了以下变化：

- Solaris 安全 Shell 基于 OpenSSH 3.5p1。Solaris 实现还包括截至 OpenSSH 3.8p1 的各版本的功能和错误修复。
- 在 `/etc/ssh/sshd_config` 文件中，`X11Forwarding` 的缺省值为 `yes`。
- 以下是已引入的关键字：
  - `GSSAPIAuthentication`
  - `GSSAPIKeyExchange`
  - `GSSAPIDelegateCredentials`
  - `GSSAPIStoreDelegatedCredentials`
  - `KbdInteractiveAuthentication`

  通过 `GSSAPI` 关键字，Solaris 安全 Shell 可以使用 GSS 凭证进行验证。`KbdInteractiveAuthentication` 关键字支持 PAM 中的任意提示和口令更改。有关这些关键字及其缺省值的完整列表，请参见第 334 页中的 “Solaris 安全 Shell 中的关键字”。
- ARCFOUR 和 AES128-CTR 密码现在可用。ARCFOUR 也称为 RC4。AES 密码是计数器模式下的 AES。
- `sshd` 守护进程使用 `/etc/default/login` 和 `login` 命令中的变量。`/etc/default/login` 变量可以由 `sshd_config` 文件中的值覆盖。有关更多信息，请参见第 337 页中的 “Solaris 安全 Shell 和登录环境变量”和 `sshd_config`(4) 手册页。

Solaris 安全 Shell（任务列表）

以下任务列表介绍了用于配置 Solaris 安全 Shell 和使用 Solaris 安全 Shell 的任务列表。

任务	说明	参考
配置 Solaris 安全 Shell	指导管理员为用户配置 Solaris 安全 Shell。	第 315 页中的 “配置 Solaris 安全 Shell（任务列表）”
使用 Solaris 安全 Shell	指导用户使用 Solaris 安全 Shell。	第 319 页中的 “使用 Solaris 安全 Shell（任务列表）”

配置 Solaris 安全 Shell（任务列表）

以下任务列表介绍了配置 Solaris 安全 Shell 的过程。

任务	说明	参考
配置基于主机的验证	在客户机和服务器上配置基于主机的验证。	第 315 页中的 “如何为 Solaris 安全 Shell 设置基于主机的验证”
将主机配置为使用 v1 和 v2	为使用 v1 和 v2 协议的主机创建公钥文件。	第 318 页中的 “如何启用 Solaris 安全 Shell v1”
配置端口转发	允许用户使用端口转发。	第 318 页中的 “如何在 Solaris 安全 Shell 中配置端口转发”

配置 Solaris 安全 Shell

缺省情况下，Solaris 安全 Shell 中未启用基于主机的验证，并且不允许同时使用两种协议。更改这些缺省设置需要管理干预。另外，要端口转发正常工作，也需要管理干预。

▼ 如何为 Solaris 安全 Shell 设置基于主机的验证

以下过程将设置一个公钥系统，其中客户机的公钥用于服务器上的验证。用户还必须创建公钥/私钥对。

在此过程中，术语**客户机**和**本地主机**是指用户键入 `ssh` 命令的计算机。术语**服务器**和**远程主机**是指客户机要尝试访问的计算机。

1. **承担主管理员角色，或成为超级用户。**

   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2. **在客户机上，启用基于主机的验证。**

   在客户机配置文件 `/etc/ssh/ssh_config` 中，键入以下项：

   ```
   HostbasedAuthentication yes
   ```

3. **在服务器上，启用基于主机的验证。**

   在服务器配置文件 `/etc/ssh/sshd_config` 中，键入相同的项：

   ```
   HostbasedAuthentication yes
   ```

4 **在服务器上，配置文件以允许将客户机识别为受信任主机。**
有关更多信息，请参见 sshd(1M) 手册页中的 FILES 部分。

- **将客户机作为一项添加到服务器的 /etc/shosts.equiv 文件中。**
  client-host
- **或者，可以指示用户在服务器上将一个有关客户机的项添加到其 ~/.shosts 文件中。**
  client-host

5 **在服务器上，确保 sshd 守护进程可以访问受信任主机的列表。**
在 /etc/ssh/sshd_config 文件中，将 IgnoreRhosts 设置为 no。

```
# sshd_config

IgnoreRhosts no
```

6 **确保站点上的 Solaris 安全 Shell 用户在两台主机上都拥有帐户。**

7 **执行以下操作之一，以将客户机的公钥放到服务器上。**

- **修改服务器上的 sshd_config 文件，然后指示用户将客户机的公共主机密钥添加到其 ~/.ssh/known_hosts 文件中。**

  ```
  # sshd_config

  IgnoreUserKnownHosts no
  ```

  有关用户说明，请参见第 320 页中的 “如何生成用于 Solaris 安全 Shell 的公钥/私钥对”。

- **将客户机的公钥复制到服务器。**
  主机密钥存储在 /etc/ssh 目录中。这些密钥通常由 sshd 守护进程在首次引导时生成。

  a. **将密钥添加到服务器上的 /etc/ssh/ssh_known_hosts 文件中。**
  在客户机上，在某一行中键入该命令（不带反斜杠）。

  ```
  # cat /etc/ssh/ssh_host_dsa_key | ssh RemoteHost \

  'cat >> /etc/ssh/ssh_known_hosts && echo "Host key copied"'
  ```

  b. **出现提示时，提供登录口令。**
  复制文件时，将会显示消息 "Host key copied"。

示例 18–1 设置基于主机的验证

在以下示例中，每台主机同时配置为服务器和客户机。任一主机上的用户都可以启动与另一台主机的 ssh 连接。以下配置可使每台主机同时成为服务器和客户机：

- 在每台主机上，Solaris 安全 Shell 配置文件都包含以下项：

```
# /etc/ssh/ssh_config

HostBasedAuthentication yes

#

# /etc/ssh/sshd_config

HostBasedAuthentication yes

IgnoreRhosts no
```

- 在每台主机上，shosts.equiv 文件都包含对应于另一台主机的项：

```
# /etc/hosts.equiv on machine2

machine1

# /etc/hosts.equiv on machine1

machine2
```

- 每台主机的公钥位于另一台主机的 /etc/ssh/ssh_known_hosts 文件中：

```
# /etc/ssh/ssh_known_hosts on machine2

... machine1

# /etc/ssh/ssh_known_hosts on machine1

... machine2
```

- 用户在两台主机上都拥有帐户：

```
# /etc/passwd on machine1

jdoe:x:3111:10:J Doe:/home/jdoe:/bin/sh

# /etc/passwd on machine2

jdoe:x:3111:10:J Doe:/home/jdoe:/bin/sh
```

▼ 如何启用 Solaris 安全 Shell v1

此过程在一台主机与运行 v1 和 v2 的多台主机交互操作时非常有用。

1 承担主管理员角色，或成为超级用户。

主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2 将主机配置为同时使用两种 Solaris 安全 Shell 协议。

编辑 `/etc/ssh/sshd_config` 文件。

```
# Protocol 2

Protocol 2,1
```

3 为用于 v1 的主机密钥提供一个独立文件。

向 `/etc/ssh/sshd_config` 文件中添加 `HostKey` 项。

```
HostKey /etc/ssh/ssh_host_rsa_key

HostKey /etc/ssh/ssh_host_dsa_key

HostKey /etc/ssh/ssh_host_rsa1_key
```

4 生成用于 v1 的主机密钥。

```
# ssh-keygen -t rsa1 -f /etc/ssh/ssh_host_rsa1_key -N ''
```

`-t rsa1` 表示用于 v1 的 RSA 算法。

`-f` 表示保存主机密钥的文件。

`-N ''` 表示无需口令短语。

5 重新启动 `sshd` 守护进程。

```
# svcadm restart network/ssh:default
```

您也可以重新引导系统。

▼ 如何在 Solaris 安全 Shell 中配置端口转发

使用端口转发可以将本地端口转发到远程主机。实际上，分配了一个套接字用于侦听本地端的端口。同样，也可以在远程端指定端口。

注 – Solaris 安全 Shell 端口转发必须使用 TCP 连接。Solaris 安全 Shell 不支持使用 UDP 连接进行端口转发。

1. **承担主管理员角色，或成为超级用户。**

   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2. **将远程 Solaris 安全 Shell 服务器配置为允许端口转发。**

   在 `/etc/ssh/sshd_config` 文件中，将 `AllowTcpForwarding` 的值更改为 `yes`。

   ```
   # Port forwarding

   AllowTcpForwarding yes
   ```

3. **重新启动 Solaris 安全 Shell 服务。**

   ```
   remoteHost# svcadm restart network/ssh:default
   ```

   有关管理持久性服务的信息，请参见《System Administration Guide: Basic Administration》中的第 14 章，“Managing Services (Overview)” 和 `svcadm`(1M) 手册页。

4. **检验是否可以使用端口转发。**

   ```
   remoteHost# /usr/bin/pgrep -lf sshd

    1296 ssh -L 2001:remoteHost:23 remoteHost
   ```

使用 Solaris 安全 Shell（任务列表）

以下任务列表介绍了用户使用 Solaris 安全 Shell 的过程。

任务	说明	参考
创建公钥/私钥对。	针对要求公钥验证的站点启用对 Solaris 安全 Shell 的访问。	第 320 页中的 “如何生成用于 Solaris 安全 Shell 的公钥/私钥对”
更改口令短语	更改用于验证私钥的短语。	第 323 页中的 “如何更改 Solaris 安全 Shell 私钥的口令短语”
使用 Solaris 安全 Shell 登录	远程登录时提供加密的 Solaris 安全 Shell 通信。此过程与使用 `rsh` 命令类似。	第 323 页中的 “如何使用 Solaris 安全 Shell 登录到远程主机”

任务	说明	参考
登录到 Solaris 安全 Shell 而不会出现提示要求输入口令。	允许使用可向 Solaris 安全 Shell 提供口令的代理进行登录。	第 324 页中的 “如何减少 Solaris 安全 Shell 中的口令提示”
		第 326 页中的 “如何将 ssh-agent 命令设置为自动运行”
在 Solaris 安全 Shell 中使用端口转发	指定要在基于 TCP 的 Solaris 安全 Shell 连接中使用的本地端口或远程端口。	第 327 页中的 “如何在 Solaris 安全 Shell 中使用端口转发”
使用 Solaris 安全 Shell 复制文件	在主机之间安全地复制文件。	第 328 页中的 “如何使用 Solaris 安全 Shell 复制文件”
安全地从防火墙内的主机连接到防火墙外的主机。	使用与 HTTP 或 SOCKS5 兼容的 Solaris 安全 Shell 命令连接由防火墙隔离的主机。	第 329 页中的 “如何设置到防火墙外部主机的缺省连接”

使用 Solaris 安全 Shell

Solaris 安全 Shell 可提供本地 shell 和远程 shell 之间的安全访问。有关更多信息，请参见 ssh_config(4) 和 ssh(1) 手册页。

▼ 如何生成用于 Solaris 安全 Shell 的公钥/私钥对

如果用户的站点要实现基于主机的验证或用户公钥验证，则必须生成公钥/私钥对。有关其他选项，请参见 ssh-keygen(1) 手册页。

开始之前 通过系统管理员确认是否配置了基于主机的验证。

1 启动密钥生成程序。

```
myLocalHost% ssh-keygen -t rsa

Generating public/private rsa key pair.

...
```

其中，-t 是算法类型，可以是 rsa、dsa 或 rsa1 之一。

2 指定将保存密钥的文件的路径。

缺省情况下，文件名 id_rsa（表示 RSA v2 密钥）显示在括号中。可通过按回车键选择此文件。或者，可以键入替换的文件名。

```
Enter file in which to save the key (/home/jdoe/.ssh/id_rsa):     <按回车键>
```

通过将字符串 .pub 附加到私钥文件的名称后，可以自动创建公钥的文件名。

3 **键入使用密钥的口令短语。**

此口令短语用于加密私钥。**强烈建议不要**使用空项。请注意，键入口令短语时，它们不会显示。

```
Enter passphrase (empty for no passphrase):     <键入口令短语>
```

4 **重新键入口令短语以进行确认。**

```
Enter same passphrase again:     <键入口令短语>

Your identification has been saved in /home/jdoe/.ssh/id_rsa.

Your public key has been saved in /home/jdoe/.ssh/id_rsa.pub.

The key fingerprint is:

0e:fb:3d:57:71:73:bf:58:b8:eb:f3:a3:aa:df:e0:d1 jdoe@myLocalHost
```

5 **检查结果。**

检查密钥文件的路径是否正确。

```
% ls ~/.ssh

id_rsa

id_rsa.pub
```

此时，已创建公钥/私钥对。

6 **选择适当的选项：**

- **如果管理员已配置了基于主机的验证，则可能需要将本地主机的公钥复制到远程主机。**

  现在即可登录到远程主机。有关详细信息，请参见第 323 页中的 “如何使用 Solaris 安全 Shell 登录到远程主机”。

  a. **在一行中键入命令（不带反斜杠）。**

  ```
  % cat /etc/ssh/ssh_host_dsa_key | ssh RemoteHost \

  'cat >> ~./ssh/known_hosts && echo "Host key copied"'
  ```

  b. **出现提示时，提供登录口令。**

  ```
  Enter password:     <键入口令>

  Host key copied

  %
  ```

- **如果站点使用公钥来进行用户验证，请在远程主机上装载 `authorized_keys` 文件。**

  a. **将公钥复制到远程主机。**

     在一行中键入命令（不带反斜杠）。

     ```
     myLocalHost% cat $HOME/.ssh/id_rsa.pub | ssh myRemoteHost \

     'cat >> .ssh/authorized_keys && echo "Key copied"'
     ```

  b. **出现提示时，提供登录口令。**

     复制文件时，将会显示消息 "Key copied"。

     ```
     Enter password:      键入登录口令

     Key copied

     myLocalHost%
     ```

7 **（可选的）减少口令短语的提示。**

有关过程，请参见第 324 页中的 “如何减少 Solaris 安全 Shell 中的口令提示”。有关更多信息，请参见 `ssh-agent`(1) 和 `ssh-add`(1) 手册页。

示例 18–2 为用户建立 v1 RSA 密钥

在以下示例中，用户可以访问运行 Solaris 安全 Shell v1 协议的主机。要通过 v1 主机进行验证，用户应创建 v1 密钥，然后将公钥部分复制到远程主机。

```
myLocalHost% ssh-keygen -t rsa1 -f /home/jdoe/.ssh/identity

Generating public/private rsa key pair.

...

Enter passphrase (empty for no passphrase):     <键入口令短语>

Enter same passphrase again:     <键入口令短语>

Your identification has been saved in /home/jdoe/.ssh/identity.

Your public key has been saved in /home/jdoe/.ssh/identity.pub.

The key fingerprint is:

...

myLocalHost% ls ~/.ssh

id_rsa
```

```
id_rsa.pub

identity

identity.pub

myLocalHost% cat $HOME/.ssh/identity.pub | ssh myRemoteHost \

'cat >> .ssh/authorized_keys && echo "Key copied"'
```

▼ 如何更改 Solaris 安全 Shell 私钥的口令短语

以下过程不会更改私钥。此过程将更改私钥的验证机制，即口令短语。有关更多信息，请参见 ssh-keygen(1) 手册页。

◗ **更改口令短语。**

键入带有 -p 选项的 ssh-keygen 命令，并回答提示问题。

```
myLocalHost% ssh-keygen -p

Enter file which contains the private key (/home/jdoe/.ssh/id_rsa):    <按回车键>

Enter passphrase (empty for no passphrase):    <键入口令短语>

Enter same passphrase again:    <键入口令短语>
```

其中，-p 用于请求更改私钥文件的口令短语。

▼ 如何使用 Solaris 安全 Shell 登录到远程主机

1 **启动 Solaris 安全 Shell 会话。**

键入 ssh 命令，并指定远程主机的名称。

```
myLocalHost% ssh myRemoteHost
```

此时会出现提示，询问远程主机的真实性：

```
The authenticity of host 'myRemoteHost' can't be established.

RSA key fingerprint in md5 is: 04:9f:bd:fc:3d:3e:d2:e7:49:fd:6e:18:4f:9c:26

Are you sure you want to continue connecting(yes/no)?
```

初始连接到远程主机时，出现此提示为正常情况。

2 **如果出现提示，请检验远程主机密钥的真实性。**

- **如果无法确认远程主机的真实性，请键入 no 并与系统管理员联系。**

  ```
  Are you sure you want to continue connecting(yes/no)? no
  ```

  管理员负责更新全局 /etc/ssh/ssh_known_hosts 文件。更新的 ssh_known_hosts 文件可禁止出现此提示。

- **如果确认了远程主机的真实性，请回答提示问题，并继续下一步。**

  ```
  Are you sure you want to continue connecting(yes/no)? yes
  ```

3 **向 Solaris 安全 Shell 验证自身的身份。**

a. **出现提示时，键入口令短语。**

```
Enter passphrase for key '/home/jdoe/.ssh/id_rsa':     <键入口令短语>
```

b. **出现提示时，键入帐户口令。**

```
jdoe@myRemoteHost's password:     <键入口令>

Last login: Fri Jul 20 14:24:10 2001 from myLocalHost

myRemoteHost%
```

4 **执行远程主机上的事务。**

所发送的命令将会加密。所接收的任何响应都会加密。

5 **关闭 Solaris 安全 Shell 连接。**

完成后，键入 **exit** 或者使用常规方法退出 shell。

```
myRemoteHost% exit

myRemoteHost% logout

Connection to myRemoteHost closed

myLocalHost%
```

▼ 如何减少 Solaris 安全 Shell 中的口令提示

如果不想键入口令短语和口令来使用 Solaris 安全 Shell，则可以使用代理守护进程。请在会话开始时启动守护进程。然后，使用 ssh-add 命令将私钥存储到代理守护进程中。如果您在不同的主机上拥有不同的帐户，请添加需要用于会话的密钥。

可以根据需要手动启动代理守护进程，如以下过程中所述。或者，可以将代理守护进程设置为在每个会话开始时自动运行，如第 326 页中的 “如何将 ssh-agent 命令设置为自动运行”中所述。

1. **启动代理守护进程。**

```
myLocalHost% ssh-agent
```

2. **检验是否已启动代理守护进程。**

```
myLocalHost% eval ‘ssh-agent‘

Agent pid 9892
```

3. **将私钥添加到代理守护进程。**

键入 ssh-add 命令。

```
myLocalHost% ssh-add

Enter passphrase for /home/jdoe/.ssh/id_rsa:     <键入口令短语>

Identity added: /home/jdoe/.ssh/id_rsa(/home/jdoe/.ssh/id_rsa)

myLocalHost%
```

4. **启动 Solaris 安全 Shell 会话。**

```
myLocalHost% ssh myRemoteHost
```

此时不会提示您输入口令短语。

示例 18–3 使用 ssh-add 选项

在本示例中，jdoe 将向代理守护进程添加两个密钥。-l 选项用于列出守护进程中存储的所有密钥。在会话结束时，-D 选项用于删除代理守护进程中的所有密钥。

```
myLocalHost% ssh-agent

myLocalHost% ssh-add

Enter passphrase for /home/jdoe/.ssh/id_rsa:     <键入口令短语>

Identity added: /home/jdoe/.ssh/id_rsa(/home/jdoe/.ssh/id_rsa)

myLocalHost% ssh-add /home/jdoe/.ssh/id_dsa

Enter passphrase for /home/jdoe/.ssh/id_dsa:     <键入口令短语>

Identity added:

/home/jdoe/.ssh/id_dsa(/home/jdoe/.ssh/id_dsa)
```

```
myLocalHost% ssh-add -l

md5 1024 0e:fb:3d:53:71:77:bf:57:b8:eb:f7:a7:aa:df:e0:d1

/home/jdoe/.ssh/id_rsa(RSA)

md5 1024 c1:d3:21:5e:40:60:c5:73:d8:87:09:3a:fa:5f:32:53

/home/jdoe/.ssh/id_dsa(DSA)
```

用户管理 *Solaris* 安全 *Shell* 事务

```
myLocalHost% ssh-add -D

Identity removed:

/home/jdoe/.ssh/id_rsa(/home/jdoe/.ssh/id_rsa.pub)

/home/jdoe/.ssh/id_dsa(DSA)
```

▼ 如何将 ssh-agent 命令设置为自动运行

只要通过自动启动代理守护进程 `ssh-agent` 来使用 Solaris 安全 Shell，即可避免提供口令短语和口令。可以通过 `.dtprofile` 脚本启动代理守护进程。要将口令短语和口令添加到代理守护进程，请参见示例 18–3。

1 在用户启动脚本中自动启动代理守护进程。

将以下行添加到 `$HOME/.dtprofile` 脚本的末尾：

```
if [ "$SSH_AUTH_SOCK" = "" -a -x /usr/bin/ssh-agent ]; then

                eval ‘/usr/bin/ssh-agent‘

fi
```

2 退出 CDE 会话时，终止代理守护进程。

将以下行添加到 `$HOME/.dt/sessions/sessionexit` 脚本中：

```
if [ "$SSH_AGENT_PID" != "" -a -x /usr/bin/ssh-agent ]; then

                /usr/bin/ssh-agent -k

fi
```

此项可确保终止 CDE 会话后，任何用户都无法使用 Solaris 安全 Shell 代理。

▼ 如何在 Solaris 安全 Shell 中使用端口转发

可以指定将本地端口转发到远程主机。实际上，分配了一个套接字用于侦听本地端的端口。从此端口到远程主机的连接基于安全通道。例如，可以指定端口 143 以使用 IMAP4 远程获取电子邮件。同样，也可以在远程端指定端口。

开始之前 使用端口转发之前，管理员必须先在远程 Solaris 安全 Shell 服务器上启用端口转发。有关详细信息，请参见第 318 页中的 “如何在 Solaris 安全 Shell 中配置端口转发”。

◗ **要使用安全的端口转发，请选择以下选项之一：**

- **要将本地端口设置为接收来自远程端口的安全通信，请同时指定这两个端口。**
  指定用于侦听远程通信的本地端口。另外，指定用于转发通信的远程主机和远程端口。
  ```
  myLocalHost% ssh -L localPort:remoteHost:remotePort
  ```
- **要将远程端口设置为接收来自本地端口的安全通信，请同时指定这两个端口。**
  指定用于侦听远程通信的远程端口。另外，指定用于转发通信的本地主机和本地端口。
  ```
  myLocalHost% ssh -R remotePort:localhost:localPort
  ```

示例 18–4 使用本地端口转发接收邮件

以下示例说明如何使用本地端口转发来安全地接收来自远程服务器的邮件。

```
myLocalHost% ssh -L 9143:myRemoteHost:143 myRemoteHost
```

此命令可将连接从 myLocalHost 上的端口 9143 转发到端口 143。端口 143 是 myRemoteHost 上的 IMAP v2 服务器端口。用户启动邮件应用程序时，需要指定本地端口号，如以下对话框中所示。

请勿将对话框中的 localhost 与 myLocalHost 混淆。myLocalHost 是假设的主机名，而 localhost 则是用于标识本地系统的关键字。

示例 18–5 使用远程端口转发在防火墙外部进行通信

本示例说明企业环境中的用户如何将连接从外部网络中的主机转发到公司防火墙内的主机。

```
myLocalHost% ssh -R 9022:myLocalHost:22 myOutsideHost
```

此命令可将连接从 myOutsideHost 上的端口 9022 转发到本地主机上的端口 22（sshd 服务器）。

```
myOutsideHost% ssh -p 9022 localhost

myLocalHost%
```

▼ 如何使用 Solaris 安全 Shell 复制文件

以下过程说明如何使用 scp 命令在主机之间复制加密的文件。您可以在一台本地主机和一台远程主机之间，或者两台远程主机之间复制加密的文件。该命令的运行与 rcp 命令类似，不同的是 scp 命令会提示进行验证。有关更多信息，请参见 scp(1) 手册页。

也可以使用 sftp（一种形式更安全的 ftp 命令）。有关更多信息，请参见 sftp(1) 手册页。

1 **启动安全的复制程序。**

指定源文件、远程目标上的用户名和目标目录。

```
myLocalHost% scp myfile.1 jdoe@myRemoteHost:~
```

2 **出现提示时，提供口令短语。**

```
Enter passphrase for key '/home/jdoe/.ssh/id_rsa':     <键入口令短语>

myfile.1       25% |*******                      |    640 KB  0:20 ETA

myfile.1
```

键入口令短语后，将会显示进度计量器。请参见以上输出中的第二行。进度计量器显示以下内容：

- 文件名
- 已传输的文件百分比
- 表示已传输的文件百分比的一系列星号
- 已传输的数据量
- 完整文件的估计到达时间 (estimated time of arrival, ETA)，即剩余时间。

▼ 如何设置到防火墙外部主机的缺省连接

可以使用 Solaris 安全 Shell 建立从防火墙内的主机到防火墙外的主机的连接。通过在配置文件中指定 `ssh` 的代理命令或者在命令行中将该代理命令指定为选项，可以完成此任务。有关的命令行选项，请参见示例 18–6。

通常，可以通过配置文件自定义 `ssh` 交互。

- 可以在 `~/.ssh/config` 中自定义独立的个人文件。
- 或者，可以使用管理配置文件 `/etc/ssh/ssh_config` 中的设置。

可以使用两种类型的代理命令自定义这些文件。一个是用于 HTTP 连接的代理命令。另一个是用于 SOCKS5 连接的代理命令。有关更多信息，请参见 `ssh_config`(4) 手册页。

1 **在配置文件中指定代理命令和主机。**

使用以下语法添加所需数量的行：

```
[Host outside-host]

ProxyCommand proxy-command [-h proxy-server] \

[-p proxy-port] outside-host|%h outside-port|%p
```

`Host` *outside-host*
在命令行中指定远程主机名时，将代理命令规范限制为实例。如果对 *outside-host* 使用通配符，则可将代理命令规范应用于一组主机。

proxy-command
指定代理命令。该命令可以是以下之一：

- `/usr/lib/ssh/ssh-http-proxy-connect`，用于 HTTP 连接
- `/usr/lib/ssh/ssh-socks5-proxy-connect`，用于SOCKS5 连接

`-h` *proxy-server* 和 `-p` *proxy-port*
这些选项分别指定代理服务器和代理端口。如果存在，则代理将覆盖指定代理服务器和代理端口的任何环境变量，如 `HTTPPROXY`、`HTTPPROXYPORT`、`SOCKS5_PORT`、`SOCKS5_SERVER` 和 `http_proxy`。`http_proxy` 变量指定 URL。如果不使用这些选项，则必须设置相关的环境变量。有关更多信息，请参见 `ssh-socks5-proxy-connect`(1) 和 `ssh-http-proxy-connect`(1) 手册页。

outside-host
指定要连接到的特定主机。请在命令行中使用 `%h` 替换参数来指定主机。

outside-port
指定要连接到的特定端口。请在命令行中使用 `%p` 替换参数来指定端口。通过指定 `%h` 和 `%p` 而不使用 `Host` *outside-host* 选项，只要调用 `ssh` 命令即可将应用代理命令到主机参数。

2 运行 Solaris 安全 Shell，从而指定外部主机。

例如，键入以下命令：

```
myLocalHost% ssh myOutsideHost
```

此命令可在个人配置文件中查找 `myOutsideHost` 的代理命令规范。如果找不到规范，则该命令将在系统范围的配置文件 `/etc/ssh/ssh_config` 中查找。该代理命令将替换 `ssh` 命令。

示例 18–6 通过命令行连接到防火墙外部的主机

第 329 页中的 “如何设置到防火墙外部主机的缺省连接”说明如何在配置文件中指定代理命令。在本示例中，在 `ssh` 命令行中指定代理命令。

```
% ssh -o'Proxycommand=/usr/lib/ssh/ssh-http-proxy-connect \

-h myProxyServer -p 8080 myOutsideHost 22' myOutsideHost
```

`ssh` 命令的 `-o` 选项提供了指定代理命令的命令行方法。此示例命令可执行以下操作：

- 将 `ssh` 替换为 HTTP 代理命令
- 使用端口 `8080` 并将 `myProxyServer` 用作代理服务器
- 连接到 `myOutsideHost` 上的端口 `22`

◆ ◆ ◆ 第 19 章

19 Solaris 安全 Shell（参考）

本章介绍 Solaris 安全 Shell 中的配置选项。以下是本章中参考信息的列表：

- 第 331 页中的 “典型的 Solaris 安全 Shell 会话”
- 第 333 页中的 “Solaris 安全 Shell 中的客户机和服务器配置”
- 第 334 页中的 “Solaris 安全 Shell 中的关键字”
- 第 338 页中的 “维护 Solaris 安全 Shell 中的已知主机”
- 第 339 页中的 “Solaris 安全 Shell 软件包和初始化”
- 第 339 页中的 “Solaris 安全 Shell 文件”
- 第 341 页中的 “Solaris 安全 Shell 命令”

有关配置 Solaris 安全 Shell 的过程，请参见第 18 章。

典型的 Solaris 安全 Shell 会话

Solaris 安全 Shell 守护进程 (`sshd`) 通常在引导时（启动网络服务时）启动。该守护进程侦听来自客户机的连接。Solaris 安全 Shell 会话在用户运行 `ssh`、`scp` 或 `sftp` 命令时开始。系统将为每个传入连接派生一个新的 `sshd` 守护进程。派生的守护进程处理密钥交换、加密、验证、命令执行以及与客户机的数据交换。这些会话的特征由客户端配置文件和服务器端配置文件确定。命令行参数可以覆盖配置文件中的设置。

客户机和服务器必须相互进行认证。验证成功后，用户可以远程执行命令和在主机之间复制数据。

Solaris 安全 Shell 中会话的特征

`sshd` 守护进程的服务器端行为由 `/etc/ssh/sshd_config` 文件中的关键字设置控制。例如，`sshd_config` 文件控制访问服务器时可采用哪些类型的验证。启动 `sshd` 守护进程时，命令行选项也可控制服务器端行为。

客户端的行为由 Solaris 安全 Shell 关键字控制，其优先级顺序如下：

- 命令行选项

- 用户的配置文件 `~/.ssh/config`
- 系统范围的配置文件 `/etc/ssh/ssh_config`

例如，用户可以通过在命令行中指定 `-c 3des` 来覆盖在系统范围内配置的 `Cipher` 的设置值 `blowfish`。

Solaris 安全 Shell 中的验证和密钥交换

Solaris 安全 Shell 协议（v1 和 v2）均支持客户机用户/主机验证和服务器主机验证。这两种协议都涉及会话加密密钥（用于保护 Solaris 安全 Shell 会话）的交换。每种协议提供了用于验证和密钥交换的各种方法。一些方法是可选的。Solaris 安全 Shell 支持许多客户机验证机制，如表 18–1 中所示。通常使用已知的主机公钥对服务器进行验证。

对于 v1 协议，Solaris 安全 Shell 支持使用口令进行用户验证。该协议也支持用户公钥，以及使用受信任的主机公钥进行验证。服务器验证使用主机公钥完成。对于 v1 协议，所有公钥都是 RSA 密钥。会话密钥交换涉及对定期重新生成的暂时服务器密钥的使用。

对于 v2 协议，Solaris 安全 Shell 支持用户验证和一般交互验证，这通常涉及到口令。该协议也支持使用用户公钥和受信任的主机公钥进行验证。这些密钥可以是 RSA 或 DSA。会话密钥交换由在服务器验证步骤中签名的 Diffie-Hellman 暂时密钥交换组成。此外，Solaris 安全 Shell 可以使用 GSS 凭证进行验证。

获取 Solaris 安全 Shell 中的 GSS 凭证

要在 Solaris 安全 Shell 中使用 GSS-API 进行验证，服务器必须具有 GSS-API 接收器凭证，客户机必须具有 GSS-API 启动器凭证。支持 `mech_dh` 和 `mech_krb5`。

对于 `mech_dh`，如果 `root` 已运行 `keylogin` 命令，则服务器具有 GSS-API 接收器凭证。

对于 `mech_krb5`，如果 `/etc/krb5/krb5.keytab` 中包含与该服务器对应的主机主体的有效项，则服务器具有 GSS-API 接收器凭证。

如果执行了以下某项操作，则客户机具有 `mech_dh` 的启动器凭证：

- 已运行 `keylogin` 命令。
- `pam.conf` 文件中使用了 `pam_dhkeys` 模块。

如果执行了以下某项操作，则客户机具有 `mech_krb5` 的启动器凭证：

- 已运行 `kinit` 命令。
- `pam.conf` 文件中使用了 `pam_krb5` 模块。

有关 `mech_dh` 在安全 RPC 中的使用，请参见第 15 章。有关如何使用 `mech_krb5`，请参见第 20 章。有关机制的更多信息，请参见 `mech`(4) 和 `mech_spnego`(5) 手册页。

Solaris 安全 Shell 中的命令执行和数据转发

完成验证后，用户通常可通过请求 shell 或执行命令来使用 Solaris 安全 Shell。通过 ssh 命令选项可发出请求。请求可能包括分配伪 tty、转发 X11 连接或 TCP/IP 连接，或通过安全连接启用 ssh-agent 验证程序。用户会话的基本组成部分如下：

1. 用户请求 shell 或请求执行命令，以开始会话模式。

   在此模式下，数据通过客户端的终端进行发送或接收。在服务器端，数据通过 shell 或命令进行发送。
2. 数据传送完成后，用户程序将终止。
3. 除已存在的连接外，所有 X11 转发和 TCP/IP 转发均停止。现有 X11 连接和 TCP/IP 连接仍然处于打开状态。
4. 服务器向客户机发送退出状态消息。关闭所有连接后（如仍处于打开状态的转发端口），客户机将关闭到服务器的连接。然后，客户机退出。

Solaris 安全 Shell 中的客户机和服务器配置

Solaris 安全 Shell 会话的特征由配置文件控制。命令行中的选项可在一定程度上覆盖配置文件。

Solaris 安全 Shell 中的客户机配置

大多数情况下，Solaris 安全 Shell 会话的客户端特征由系统范围的配置文件 /etc/ssh/ssh_config 管理。用户配置文件 ~/.ssh/config 可覆盖 ssh_config 文件中的设置。此外，用户可在命令行中覆盖这两个配置文件。

服务器的 /etc/ssh/sshd_config 文件中的设置确定服务器允许哪些客户机请求。有关服务器配置设置的列表，请参见 第 334 页中的 “Solaris 安全 Shell 中的关键字” 。有关详细信息，请参见 sshd_config(4) 手册页。

第 334 页中的 “Solaris 安全 Shell 中的关键字” 中列出了客户机配置文件中的关键字；如果关键字具有缺省值，则会给出该值。ssh(1)、scp(1)、sftp(1) 和 ssh_config(4) 手册页中详细介绍了这些关键字。有关以字母顺序排列的关键字列表及其等效命令行覆盖，请参见表 19–8。

Solaris 安全 Shell 中的服务器配置

Solaris 安全 Shell 会话的服务器端特征由 /etc/ssh/sshd_config 文件管理。第 334 页中的 “Solaris 安全 Shell 中的关键字” 中列出了服务器配置文件中的关键字；如果关键字具有缺省值，则会给出该值。有关这些关键字的完整说明，请参见 sshd_config(4) 手册页。

Solaris 安全 Shell 中的关键字

下表列出了关键字及其缺省值（如果存在）。这些关键字按字母顺序排列。客户机上的关键字位于 ssh_config 文件中。应用于服务器的关键字位于 sshd_config 文件中。一些关键字在两个文件中均有设置。如果关键字仅应用于一种协议版本，则列出了该版本。

表 19–1 Solaris 安全 Shell 配置文件中的关键字（A 到 Escape）

关键字	缺省值	位置	协议
AllowGroups	无缺省值。	服务器	
AllowTcpForwarding	no	服务器	
AllowUsers	无缺省值。	服务器	
AuthorizedKeysFile	~/.ssh/authorized_keys	服务器	
Banner	/etc/issue	服务器	
Batchmode	no	客户机	
BindAddress	无缺省值。	客户机	
CheckHostIP	yes	客户机	
Cipher	blowfish, 3des	客户机	v1
Ciphers	aes128-ctr, aes128-cbc, 3des-cbc, blowfish-cbc, arcfour	两者	v2
ClearAllForwardings	无缺省值。	客户机	
ClientAliveInterval	0	服务器	v2
ClientAliveCountMax	3	服务器	v2
Compression	yes	两者	
CompressionLevel	无缺省值。	客户机	
ConnectionAttempts	1	客户机	
DenyGroups	无缺省值。	服务器	
DenyUsers	无缺省值。	服务器	
DynamicForward	无缺省值。	客户机	
EscapeChar	~	客户机	

表 19–2 Solaris 安全 Shell 配置文件中的关键字（Fall 到 Local）

关键字	缺省值	位置	协议
FallBackToRsh	no	客户机	
ForwardAgent	no	客户机	
ForwardX11	no	客户机	
GatewayPorts	no	两者	
GlobalKnownHostsFile	/etc/ssh/ssh_known_hosts	客户机	
GSSAPIAuthentication	yes	两者	v2
GSSAPIDelegateCredentials	no	客户机	v2
GSSAPIKeyExchange	yes	两者	v2
GSSAPIStoreDelegateCredentials	no	客户机	v2
Host	*有关更多信息，请参见 第 337 页中的 “Solaris 安全 Shell 中的主机特定参数” 。	客户机	
HostbasedAuthentication	no	两者	v2
HostbasedUsesNamesFromPacketOnly	no	服务器	v2
HostKey	/etc/ssh/ssh_host_key	服务器	v1
HostKey	/etc/ssh/host_rsa_key, /etc/ssh/host_dsa_key	服务器	v2
HostKeyAlgorithms	ssh-rsa, ssh-dss	客户机	v2
HostKeyAlias	无缺省值。	客户机	v2
IdentityFile	~/.ssh/identity	客户机	v1
IdentityFile	~/.ssh/id_dsa, ~/.ssh/id_rsa	客户机	v2
IgnoreRhosts	yes	服务器	
IgnoreUserKnownHosts	yes	服务器	
KbdInteractiveAuthentication	yes	两者	
KeepAlive	yes	两者	
KeyRegenerationInterval	3600（秒）	服务器	
ListenAddress	无缺省值。	服务器	
LocalForward	无缺省值。	客户机	

表 19–3 Solaris 安全 Shell 配置文件中的关键字（Login 到 R）

关键字	缺省值	位置	协议
LoginGraceTime	600（秒）	服务器	
LogLevel	info	两者	
LookupClientHostname	yes	服务器	
MACs	hmac-sha1,hmac-md5	两者	v2
MaxAuthTries	6	服务器	
MaxAuthTriesLog	无缺省值。	服务器	
MaxStartups	10:30:60	服务器	
NoHostAuthenticationForLocalHost	no	客户机	
NumberOfPasswordPrompts	3	客户机	
PAMAuthenticationViaKBDInt	yes	服务器	v2
PasswordAuthentication	yes	两者	
PermitEmptyPasswords	no	服务器	
PermitRootLogin	no	服务器	
PermitUserEnvironment	no	服务器	
PreferredAuthentications	gssapi-keyex, gssapi-with-mic, hostbased, publickey, keyboard-interactive, password	客户机	v2
Port	22	两者	
PrintMotd	no	服务器	
Protocol	2	两者	
ProxyCommand	无缺省值。	客户机	
PubkeyAuthentication	yes	两者	v2
RemoteForward	无缺省值。	客户机	
RhostsAuthentication	no	两者	v1
RhostsRSAAuthentication	no	两者	v1
RSAAuthentication	no	两者	v1

表 19–4 Solaris 安全 Shell 配置文件中的关键字（S 到 X）

关键字	缺省值	位置	协议
ServerKeyBits	768	服务器	
StrictHostKeyChecking	ask	客户机	
StrictModes	yes	服务器	
Subsystem	sftp /usr/lib/ssh/sftp-server	服务器	
SyslogFacility	auth	服务器	
UseLogin	no，已过时并被忽略。	服务器	
User	无缺省值。	客户机	
UserKnownHostsFile	~/.ssh/known_hosts	客户机	
VerifyReverseMapping	no	服务器	
X11Forwarding	yes	服务器	
X11DisplayOffset	10	服务器	
X11UseLocalHost	yes	服务器	
XAuthLocation	无缺省值。	两者	

Solaris 安全 Shell 中的主机特定参数

如果不同的本地主机具有不同 Solaris 安全 Shell 特征很有用，则管理员可以在 /etc/ssh/ssh_config 文件中定义单独的参数组，以根据主机或正则表达式进行应用。通过按 Host 关键字对文件中的项进行分组，可完成此任务。如果不使用 Host 关键字，则客户机配置文件中的项将应用于任一用户正在使用的本地主机。

Solaris 安全 Shell 和登录环境变量

如果 sshd_config 文件中未设置以下 Solaris 安全 Shell 关键字，则这些关键字将从 /etc/default/login 文件的等效项中获取各自的值：

/etc/default/login 中的项	sshd_config 中的关键字和值
CONSOLE=*	PermitRootLogin=without-password
#CONSOLE=*	PermitRootLogin=yes

/etc/default/login 中的项	sshd_config 中的关键字和值
PASSREQ=YES	PermitEmptyPasswords=no
PASSREQ=NO	PermitEmptyPasswords=yes
#PASSREQ	PermitEmptyPasswords=no
TIMEOUT=*secs*	LoginGraceTime=*secs*
#TIMEOUT	LoginGraceTime=300
RETRIES 和 SYSLOG_FAILED_LOGINS	仅应用于 password 和 keyboard-interactive 验证方法。

通过 login 命令设置以下变量后，sshd 守护进程将使用这些值。未设置这些变量时，守护进程将使用缺省值。

TIMEZONE　控制 TZ 环境变量的设置。如果未设置此值，在启动 sshd 守护进程时，守护进程将使用 TZ 的值。

ALTSHELL　控制 SHELL 环境变量的设置。缺省值是 ALTSHELL=YES，此时 sshd 守护进程使用用户 shell 的值。ALTSHELL=NO 时，不设置 SHELL 值。

PATH　控制 PATH 环境变量的设置。未设置此值时，缺省路径为 /usr/bin。

SUPATH　控制 root 的 PATH 环境变量的设置。未设置此值时，缺省路径为 /usr/sbin:/usr/bin。

有关更多信息，请参见 login(1) 和 sshd(1M) 手册页。

维护 Solaris 安全 Shell 中的已知主机

需要与其他主机安全通信的每台主机都必须将服务器的公钥存储在本地主机的 /etc/ssh/ssh_known_hosts 文件中。虽然脚本可用于更新 /etc/ssh/ssh_known_hosts 文件，但是强烈建议不要这样做，因为脚本会打开严重的安全漏洞。

/etc/ssh/ssh_known_hosts 文件应只按如下安全机制分发：

- 通过安全连接，如 Solaris 安全 Shell、IPsec 或已知和受信任计算机的基于 Kerberos 的 ftp
- 在系统安装时

要避免入侵者通过向 known_hosts 文件插入伪造公钥而获得访问权限的可能性，应使用 JumpStart™ 服务器作为 ssh_known_hosts 文件的已知和受信任源。ssh_known_hosts 文件可在安装过程中分发。然后，可将使用 scp 命令的脚本用于引入最新版本。由于每台主机都已具有 JumpStart 服务器的公钥，因此此方法是安全的。

Solaris 安全 Shell 软件包和初始化

Solaris 安全 Shell 依赖于核心 Solaris 软件包和以下软件包：

- SUNWgss－包含通用安全服务 (Generic Security Service, GSS) 软件
- SUNWtcpd－包含 TCP 包装
- SUNWopenssl-libraries－包含 OpenSSL 库
- SUNWzlib－包含 zip 压缩库

以下软件包安装 Solaris 安全 Shell：

- SUNWsshr－包含根 (/) 目录的客户机文件和实用程序
- SUNWsshdr－包含根 (/) 目录的服务器文件和实用程序
- SUNWsshcu－包含 /usr 目录的公用源文件
- SUNWsshdu－包含 /usr 目录的服务器文件
- SUNWsshu－包含 /usr 目录的客户机文件和实用程序

安装后重新引导时，sshd 守护进程将运行。该守护进程在系统中创建主机密钥。运行 sshd 守护进程的 Solaris 系统是 Solaris 安全 Shell 服务器。

Solaris 安全 Shell 文件

下表显示了重要的 Solaris 安全 Shell 文件和建议的文件权限。

表 19–5 Solaris 安全 Shell 文件

文件名	说明	建议的权限和属主
/etc/ssh/sshd_config	包含 sshd（Solaris 安全 Shell 守护进程）的配置数据。	-rw-r--r-- root
/etc/ssh/ssh_host_key	包含主机私钥 (v1)。	-rw-r--r-- root
/etc/ssh/ssh_host_dsa_key 或 /etc/ssh/ssh_host_rsa_key	包含主机私钥 (v2)。	-rw-r--r-- root
host-private-key.pub	包含主机公钥，如 /etc/ssh/ssh_host_rsa_key.pub。用于将主机密钥复制到本地 known_hosts 文件中。	-rw-r--r-- root
/var/run/sshd.pid	包含 Solaris 安全 Shell 守护进程 sshd 的进程 ID。如果正在运行多个守护进程，则文件包含最后启动的守护进程。	-rw-r--r-- root
~/.ssh/authorized_keys	存储允许登录到用户帐户的用户公钥。	-rw-rw-r-- *username*
/etc/ssh/ssh_known_hosts	包含客户机可安全与其通信的所有主机的主机公钥。此文件由管理员填写。	-rw-r--r-- root

表 19–5 Solaris 安全 Shell 文件 （续）

文件名	说明	建议的权限和属主
~/.ssh/known_hosts	包含客户机可安全与其通信的所有主机的主机公钥。该文件将会自动维护。只要用户与未知主机连接，就会向该文件中添加远程主机密钥。	-rw-r--r-- *username*
/etc/default/login	未设置对应的 sshd_config 参数时，将为 sshd 守护进程提供缺省值。	-r--r--r-- root
/etc/nologin	如果此文件存在，则 sshd 守护进程只允许 root 登录。此文件的内容将向尝试登录的用户显示。	-rw-r--r-- root
~/.rhosts	包含指定用户无需口令即可登录主机的主机-用户名称对。rlogind 和 rshd 守护进程也使用此文件。	-rw-r--r-- *username*
~/.shosts	包含指定用户无需口令即可登录主机的主机-用户名称对。其他实用程序不使用此文件。有关更多信息，请参见 sshd(1M) 手册页中的 FILES 部分。	-rw-r--r-- *username*
/etc/hosts.equiv	包含 .rhosts 验证中使用的主机。rlogind 和 rshd 守护进程也使用此文件。	-rw-r--r-- root
/etc/ssh/shosts.equiv	包含基于主机的验证中使用的主机。其他实用程序不使用此文件。	-rw-r--r-- root
~/.ssh/environment	包含登录时的初始赋值。缺省情况下，不会读取此文件。要读取此文件，必须将 sshd_config 文件中的 PermitUserEnvironment 关键字设置为 yes。	-rw------- *username*
~/.ssh/rc	包含启动用户 shell 前运行的初始化例程。有关初始化例程的样例，请参见 sshd 手册页。	-rw------- *username*
/etc/ssh/sshrc	包含管理员指定的主机特定的初始化例程。	-rw-r--r-- root
/etc/ssh/ssh_config	配置客户机系统上的系统设置。	-rw-r--r-- root
~/.ssh/config	配置用户设置。覆盖系统设置。	-rw------- *username*

下表列出了可被关键字或命令选项覆盖的 Solaris 安全 Shell 文件。

表 19–6 覆盖 Solaris 安全 Shell 文件的位置

文件名	关键字覆盖	命令行覆盖
/etc/ssh/ssh_config		ssh -F *config-file* scp -F *config-file*
~/.ssh/config		ssh -F *config-file*

表 19–6 覆盖 Solaris 安全 Shell 文件的位置 （续）

文件名	关键字覆盖	命令行覆盖
/etc/ssh/host_rsa_key	HostKey	
/etc/ssh/host_dsa_key		
~/.ssh/identity	IdentityFile	ssh -i *id-file*
~/.ssh/id_dsa ~/.ssh/id_rsa		scp -i *id-file*
~/.ssh/authorized_keys	AuthorizedKeysFile	
/etc/ssh/ssh_known_hosts	GlobalKnownHostsFile	
~/.ssh/known_hosts	UserKnownHostsFile	
	IgnoreUserKnownHosts	

Solaris 安全 Shell 命令

下表汇总了主要的 Solaris 安全 Shell 命令。

表 19–7 Solaris 安全 Shell 中的命令

命令	说明	手册页
ssh	使用户登录到远程计算机并在远程计算机上安全地执行命令。此命令是 Solaris 安全 Shell 中替代 rlogin 和 rsh 命令的命令。ssh 命令在不安全网络上两台不受信任主机之间启用安全的加密通信。X11 连接和任意 TCP/IP 端口也可通过安全通道进行转发。	ssh(1)
sshd	Solaris 安全 Shell 的守护进程。该守护进程侦听来自客户机的连接，并在不安全网络上两台不受信任主机之间启用安全的加密通信。	sshd(1M)
ssh-add	将 RSA 或 DSA 身份添加到验证代理 ssh-agent。这些身份也称为**密钥**。	ssh-add(1)
ssh-agent	存储用于公钥验证的私钥。ssh-agent 程序在 X 会话或登录会话开始时启动。所有其他窗口和其他程序均作为 ssh-agent 程序的客户机启动。通过使用环境变量，当用户使用 ssh 命令登录到其他系统时，可以找到代理并将其用于验证。	ssh-agent(1)
ssh-keygen	生成并管理用于 Solaris 安全 Shell 的验证密钥。	ssh-keygen(1)
ssh-keyscan	收集大量 Solaris 安全 Shell 主机的公钥。帮助构建并验证 ssh_known_hosts 文件。	ssh-keyscan(1)
ssh-keysign	由 ssh 命令用于访问本地主机上的主机密钥。生成使用 Solaris 安全 Shell v2 进行基于主机的验证时所需的数字签名。此命令由 ssh 命令（而不是用户）调用。	ssh-keysign(1M)
scp	通过加密的 ssh 传输在网络上的主机之间安全地复制文件。与 rcp 命令不同，如果验证需要口令信息，scp 命令会提示输入口令或口令短语。	scp(1)

表 19–7 Solaris 安全 Shell 中的命令 （续）

命令	说明	手册页
sftp	一个与 ftp 命令类似的交互式文件传输程序。与 ftp 命令不同，sftp 命令通过加密的 ssh 传输执行所有操作。此命令连接并登录到指定的主机名，然后进入交互式命令模式。	sftp(1)

下表列出了覆盖 Solaris 安全 Shell 关键字的命令选项。这些关键字在 ssh_config 和 sshd_config 文件中指定。

表 19–8 Solaris 安全 Shell 关键字的命令行等效关键字

关键字	ssh 命令行覆盖	scp 命令行覆盖
BatchMode		scp -B
BindAddress	ssh -b *bind-addr*	scp -a *bind-addr*
Cipher	ssh -c *cipher*	scp -c *cipher*
Ciphers	ssh -c *cipher-spec*	scp -c *cipher-spec*
Compression	ssh -C	scp -C
DynamicForward	ssh -D *SOCKS4-port*	
EscapeChar	ssh -e *escape-char*	
ForwardAgent	ssh -A（启用） ssh -a（禁用）	
ForwardX11	ssh -X（启用） ssh -x（禁用）	
GatewayPorts	ssh -g	
IPv4	ssh -4	scp -4
IPv6	ssh -6	scp -6
LocalForward	ssh -L *localport:remotehost:remoteport*	
MACS	ssh -m *mac-spec*	
Port	ssh -p *port*	scp -P *port*
Protocol	ssh -1（仅用于 v1） ssh -2（仅用于 v2）	
RemoteForward	ssh -R *remoteport:localhost:localport*	

第 6 部分

Kerberos服务

本部分提供有关 Kerberos 服务的配置、管理和使用方法的信息。

◆ ◆ ◆ 第 20 章

Kerberos 服务介绍

本章介绍 Kerberos 服务。以下是本章中概述信息的列表：

- 第 345 页中的 “什么是 Kerberos 服务？”
- 第 346 页中的 “Kerberos 服务的工作方式”
- 第 351 页中的 “Kerberos 安全服务”
- 第 351 页中的 “各种 Kerberos 发行版的组件”

什么是 Kerberos 服务？

Kerberos **服务**是一种通过网络提供安全事务处理的客户机/服务器体系结构。该服务可提供功能强大的用户验证以及完整性和保密性服务。通过**验证**，可保证网络事务的发送者和接收者的身份真实。该服务还可以检验来回传递的数据的有效性（**完整性**），并在传输过程中对数据进行加密（**保密性**）。使用 Kerberos 服务，可以安全登录到其他计算机、执行命令、交换数据以及传输文件。此外，该服务还提供**授权**服务，管理员可通过此服务限制对服务和计算机的访问。而且，作为 Kerberos 用户，您还可以控制其他用户对您帐户的访问。

Kerberos 服务是**单点登录**系统，这意味着您对于每个会话只需向服务进行一次自我验证，即可自动保护该会话过程中所有后续事务的安全。服务对您进行验证后，即无需在每次使用基于 Kerberos 的命令（如 `ftp` 或 `rsh`）或访问 NFS 文件系统上数据时都进行自我验证。因此，无需在每次使用这些服务时都在网络上发送口令（口令在网络上可能会被拦截）。

Solaris Kerberos 服务基于麻省理工学院 (Massachusetts Institute of Technology, MIT) 开发的 Kerberos V5 网络验证协议。因此，使用过 Kerberos V5 产品的用户会感觉对 Solaris 版本非常熟悉。因为 Kerberos V5 协议是网络安全性的**实际**行业标准，所以 Solaris 版本可提高与其他系统的互操作性。换句话说，因为可在使用 Kerberos V5 协议的系统中使用 Solaris Kerberos 服务，所以该服务甚至允许在异构网络上进行安全事务处理。此外，该服务还会在各个域之间以及单个域内提供验证和安全服务。

通过 Kerberos 服务可灵活运行 Solaris 应用程序。可以将该服务配置为允许同时向网络服务（如 NFS 服务、`telnet` 和 `ftp`）发出基于 Kerberos 的请求和非基于 Kerberos 的请求。因此，

即使当前 Solaris 应用程序运行的系统上未启用 Kerberos 服务，这些程序仍能工作。当然，也可以将 Kerberos 服务配置为仅允许基于 Kerberos 的网络请求。

Kerberos 服务提供了一种安全机制，通过该安全机制，在使用采用通用安全服务应用程序编程接口 (Generic Security Service Application Programming Interface, GSS-API) 的应用程序时，可使用 Kerberos 提供验证、完整性和保密性服务。但是，如果开发了其他安全机制，则应用程序无需继续承诺使用 Kerberos 服务。因为该服务设计为以模块形式集成到 GSS-API 中，所以使用 GSS-API 的应用程序可以利用最好地满足其需求的任何安全机制。

Kerberos 服务的工作方式

以下概述了 Kerberos 验证系统。有关更详细的说明，请参见第 508 页中的 “Kerberos 验证系统的工作方式”。

从用户的角度来看，启动 Kerberos 会话后，Kerberos 服务通常不可见。一些命令（如 `rsh` 或 `ftp`）也是如此。初始化 Kerberos 会话通常仅包括登录和提供 Kerberos 口令。

Kerberos 系统的工作围绕**票证**的概念展开。票证是一组标识用户或服务（如 NFS 服务）的电子信息。正如您的驾驶证可标识您的身份并表明您的驾驶级别一样，票证也可标识您的身份以及您的网络访问权限。执行基于 Kerberos 的事务时（例如，远程登录到另一台计算机），您将透明地向**密钥分发中心** (KDC) 发送票证请求。KDC 将访问数据库以验证您的身份，然后返回授予您访问其他计算机的权限的票证。“透明”意味着您无需显式请求票证。请求是在执行 `rlogin` 命令过程中进行的。因为只有通过验证的客户机可以获取特定服务的票证，所以其他客户机不能以虚假身份使用 `rlogin`。

票证具有一些与其关联的属性。例如，票证可以是**可转发的**，这意味着它可以在其他计算机上使用，而不必进行新的验证。票证也可以是**以后生效的**，这意味着它要到指定时间后才会生效。票证的使用方式（例如，如何指定允许哪些用户获取哪些类型的票证）由**策略**设置。策略在安装或管理 Kerberos 服务时确定。

注 – 您可能会经常看到术语**凭证**和**票证**。在更为广泛的 Kerberos 范围内，两者通常可互换使用。但是，从技术上讲，凭证指的是票证和会话的**会话密钥**。第 509 页中的 “使用 Kerberos 获取服务访问权限”中对此区别进行了更详细的说明。

以下各节将进一步说明 Kerberos 验证过程。

初始验证：票证授予票证

Kerberos 验证分为两个阶段：允许进行后续验证的初始验证以及所有后续验证自身。

下图显示了如何进行初始验证。

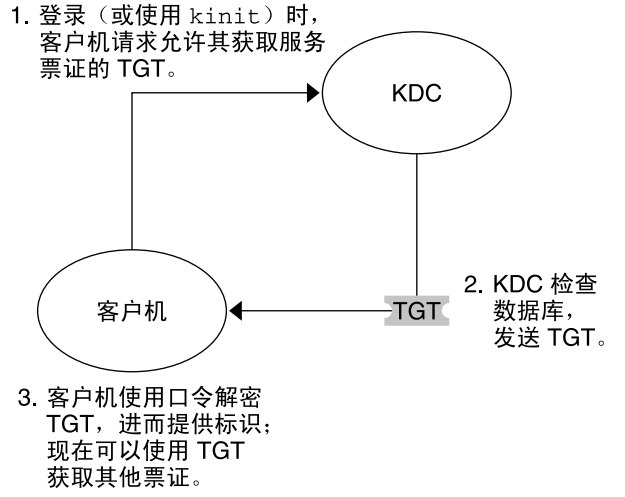


图 20–1 Kerberos 会话的初始验证

1. 客户机（用户或 NFS 等服务）通过从密钥分发中心 (Key Distribution Center, KDC) 请求**票证授予票证** (Ticket-Granting Ticket, TGT) 开始 Kerberos 会话。此请求通常在登录时自动完成。

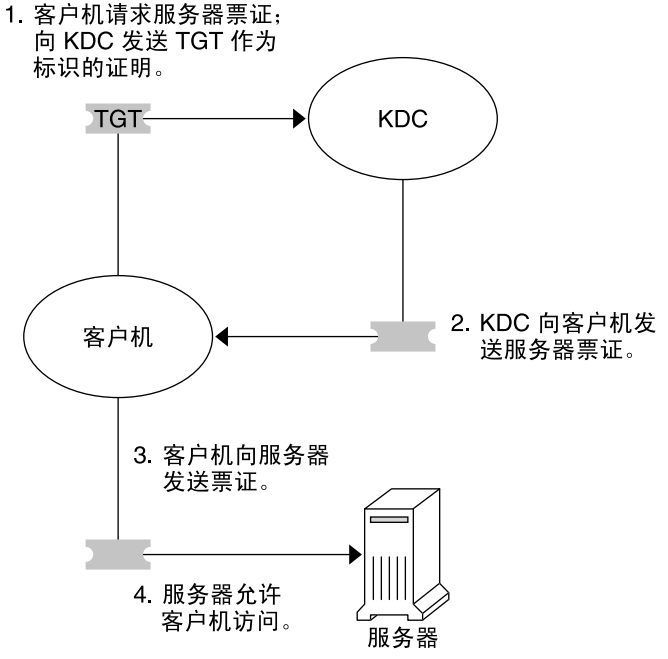
   要获取特定服务的其他票证，需要票证授予票证。票证授予票证类似于护照。与护照一样，票证授予票证可标识您的身份并允许您获取多个“签证”，此处的“签证”（票证）不是用于外国，而是用于远程计算机或网络服务。与护照和签证一样，票证授予票证和其他各种票证具有有限的生命周期。区别在于基于 Kerberos 的命令会通知您拥有护照并为您取得签证。您不必亲自执行该事务。

   与票证授予票证类似的另一种情况是可以在四个不同的滑雪场使用的三天滑雪入场卷。只要入场券未到期，您就可以在决定要去的任意一个滑雪场出示入场卷，并获取该滑雪场提供的缆车票。获取缆车票后，即可在该滑雪场随意滑雪。如果第二天去另一个滑雪场，您需要再次出示入场卷，并获取新滑雪场的另一张缆车票。区别在于基于 Kerberos 的命令会通知您拥有周末滑雪入场卷，并会为您取得缆车票。因此，您不必亲自执行该事务。

2. KDC 可创建票证授予票证，并采用加密形式将其发送回客户机。客户机使用其口令来解密票证授予票证。
3. 拥有有效的票证授予票证后，只要该票证授予票证未到期，客户机便可以请求所有类型的网络操作（如 `rlogin` 或 `telnet`）的票证。此票证的有效期通常为几个小时。每次客户机执行唯一的网络操作时，都将从 KDC 请求该操作的票证。

后续 Kerberos 验证

客户机收到初始验证后，每个后续验证都按下图所示的模式进行。



TGT = 票证授予票证
KDC = 密钥分发中心

图 20–2 使用 Kerberos 验证获取对服务的访问权

1. 客户机通过向 KDC 发送其票证授予票证作为其身份证明，从 KDC 请求特定服务（例如，远程登录到另一台计算机）的票证。
2. KDC 将该特定服务的票证发送到客户机。

   例如，假定用户 `joe` 要访问已通过要求的 `krb5` 验证共享的 NFS 文件系统。由于该用户已经通过了验证（即，该用户已经拥有票证授予票证），因此当其尝试访问文件时，NFS 客户机系统将自动透明地从 KDC 获取 NFS 服务的票证。

   例如，假定用户 `joe` 在服务器 `boston` 上使用 `rlogin`。由于该用户已经通过了验证（即，该用户已经拥有票证授予票证），所以在运行 `rlogin` 命令时，该用户将自动透明地获取票证。该用户使用此票证可随时远程登录到 `boston`，直到票证到期为止。如果 `joe` 要远程登录到计算机 `denver`，则需要按照步骤 1 获取另一个票证。
3. 客户机将票证发送到服务器。

   使用 NFS 服务时，NFS 客户机会自动透明地将 NFS 服务的票证发送到 NFS 服务器。
4. 服务器允许此客户机进行访问。

从这些步骤来看，服务器似乎并未与 KDC 通信。但服务器实际上与 KDC 进行了通信，并向 KDC 注册了其自身，正如第一台客户机所执行的操作。为简单起见，该部分已省略。

Kerberos 远程应用程序

用户（如 joe）可以使用的基于 Kerberos 的（即 "Kerberized"）命令包括：

- ftp
- rcp
- rdist
- rlogin
- rsh
- ssh
- telnet

这些应用程序与同名的 Solaris 应用程序相同。但是，它们已扩展为使用 Kerberos 主体来验证事务，因此会提供基于 Kerberos 的安全性。有关主体的信息，请参见第 349 页中的 “Kerberos 主体”。

第 494 页中的 “Kerberos 用户命令”中将进一步介绍这些命令。

Kerberos 主体

Kerberos 服务中的客户机由其**主体**标识。主体是 KDC 可以为其指定票证的唯一标识。主体可以是用户（如 joe）或服务（如 nfs 或 telnet）。

根据约定，主体名称分为三个部分：**主名称**、**实例**和**领域**。例如，典型的 Kerberos 主体可以是 joe/admin@ENG.EXAMPLE.COM。在本示例中：

- joe 是主名称。主名称可以是此处所示的用户名或 nfs 等服务。主名称还可以是单词 host，这表示此主体是设置用于提供各种网络服务（如 ftp、rcp、rlogin 等）的服务主体。
- admin 是实例。对于用户主体，实例是可选的；但对于服务主体，实例则是必需的。例如，如果用户 joe 有时充当系统管理员，则他可以使用 joe/admin 将其自身与平时的用户身份区分开来。同样，如果 joe 在两台不同的主机上拥有帐户，则他可以使用两个具有不同实例的主体名称，例如 joe/denver.example.com 和 joe/boston.example.com。请注意，Kerberos 服务会将 joe 和 joe/admin 视为两个完全不同的主体。

  对于服务主体，实例是全限定主机名。例如，bigmachine.eng.example.com 就是这种实例。此示例的主名称/实例可以为 ftp/bigmachine.eng.example.com 或 host/bigmachine.eng.example.com。
- ENG.EXAMPLE.COM 是 Kerberos 领域。领域将在第 350 页中的 “Kerberos 领域”中介绍。

以下都是有效的主体名称：

- joe

- joe/admin
- joe/admin@ENG.EXAMPLE.COM
- ftp/host.eng.example.com@ENG.EXAMPLE.COM
- host/eng.example.com@ENG.EXAMPLE.COM

Kerberos 领域

领域是定义属于同一**主** *KDC* 的一组系统的逻辑网络，类似于域。图 20–3 显示了各领域相互之间的关系。有些领域是分层的，其中，一个领域是另一个领域的超集。另外一些领域是不分层（或“直接”）的，必须定义两个领域之间的映射。Kerberos 服务的一种功能是它允许进行跨领域验证。每个领域只需在其 KDC 中有对应于另一个领域的主体项即可。此 Kerberos 功能称为**跨领域验证**。

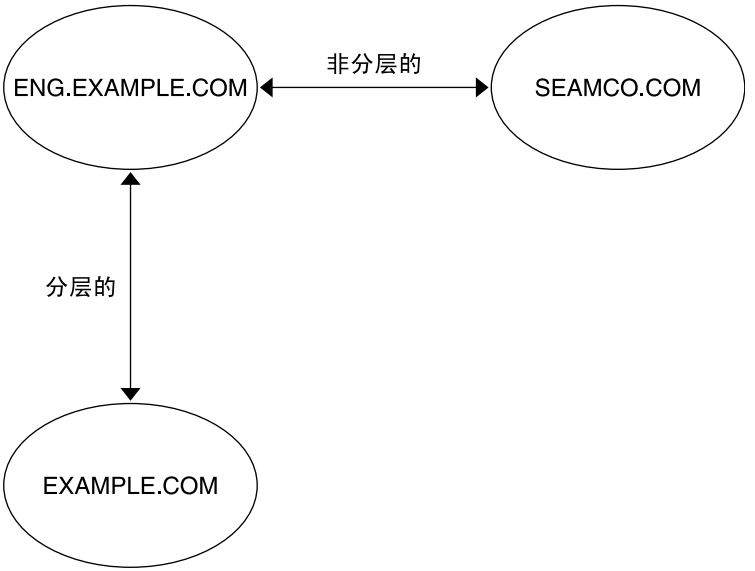


图 20–3 Kerberos 领域

Kerberos 领域和服务器

每个领域都必须包括一台用于维护主体数据库的主副本的服务器。此服务器称为**主** *KDC* **服务器**。此外，每个领域还应至少包含一台**从** *KDC* **服务器**，该服务器包含主体数据库的多个副本。主 KDC 服务器和从 KDC 服务器都可创建用于建立验证的票证。

领域还可以包含两种其他类型的 Kerberos 服务器。Kerberos 网络的**应用程序服务器**是用于提供对基于 Kerberos 的应用程序（如 `ftp`、`telnet` 和 `rsh`）的访问的服务器。领域还可以包括 *NFS* **服务器**，该服务器使用 Kerberos 验证来提供 NFS 服务。如果安装了 SEAM 1.0 或 1.0.1，则领域可能会包括 Kerberos 网络应用程序服务器。

下图显示了一个假设的领域可能包含的内容。

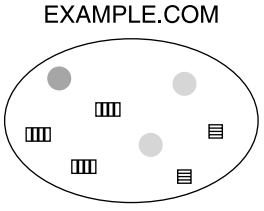


图 20–4 典型的 Kerberos 领域

Kerberos 安全服务

除了提供安全的用户验证外，Kerberos 服务还会提供两种安全服务：

- **完整性** – 正如验证服务可确保网络中客户机的身份真实可靠一样，完整性服务可确保客户机发送的数据有效并且在传输过程中未被篡改。完整性服务通过数据的加密校验和计算来实现。完整性还包括用户验证。
- **保密性** – 保密性服务可进一步增强安全性。保密性服务不仅包括对所传输数据的完整性进行验证，还会在传输之前加密数据，防止数据遭到窃听。保密性服务也对用户进行验证。

当前，在属于 Kerberos 服务的各种基于 Kerberos 的应用程序中，仅有 `ftp` 命令允许用户在运行时（“即时”）更改安全服务。开发者可以设计基于 RPC 的应用程序，以便使用 RPCSEC_GSS 编程接口来选择安全服务。

各种 Kerberos 发行版的组件

许多发行版中已包括 Kerberos 服务组件。Kerberos 服务以及对用于支持 Kerberos 服务的基本操作系统进行的更改最初都使用产品名 "Sun Enterprise Authentication Mechanism"（缩写为 SEAM）来发行。由于 Solaris 软件中包括的 SEAM 产品部件越来越多，因此 SEAM 发行版的内容会相应减少。Solaris 10 发行版中包括了 SEAM 产品的所有部件，因此不再需要 SEAM 产品。由于历史原因，文档中仍存在 SEAM 产品名。

下表说明了每个发行版中包括的组件。各个产品发行版按时间顺序列出。以下各节中对所有组件进行了说明。

表 20–1 Kerberos 发行版内容

发行版名称	内容
Solaris Easy Access Server 3.0 中的 SEAM 1.0	Solaris 2.6 和 7 发行版中 Kerberos 服务的完整发行版
Solaris 8 发行版中的 Kerberos 服务	仅 Kerberos 客户机软件
Solaris 8 Admin Pack 中的 SEAM 1.0.1	Solaris 8 发行版中的 Kerberos KDC 和远程应用程序
Solaris 9 发行版中的 Kerberos 服务	仅 Kerberos KDC 和客户机软件
SEAM 1.0.2	Solaris 9 发行版中的 Kerberos 远程应用程序
Solaris 10 发行版中的 Kerberos 服务	带有增强功能的 Kerberos 服务的完整发行版

Kerberos 组件

与 MIT 发布的 Kerberos V5 产品类似，Solaris Kerberos 服务也包括以下内容：

- 密钥分发中心 (Key Distribution Center, KDC)（主）：
  - Kerberos 数据库管理守护进程－`kadmind`。
  - Kerberos 票证处理守护进程－`krb5kdc`。
- 从 KDC。
- 数据库管理程序－`kadmin` 和 `kadmin.local`。
- 数据库传播软件－`kprop`。
- 用于获取、查看和销毁票证的用户程序－`kinit`、`klist` 和 `kdestroy`。
- 用于更改 Kerberos 口令的用户程序－`kpasswd`。
- 远程应用程序－`ftp`、`rcp`、`rdist`、`rlogin`、`rsh`、`ssh` 和 `telnet`。
- 远程应用程序守护进程－`ftpd`、`rlogind`、`rshd`、`sshd` 和 `telnetd`。
- 管理实用程序－`ktutil` 和 `kdb5_util`。
- 通用安全服务应用程序编程接口 (Generic Security Service Application Programming Interface, GSS-API)－允许应用程序使用多种安全机制，并且不需要在每次添加新机制时重新编译应用程序。因为 GSS-API 与计算机无关，所以适用于 Internet 上的各种应用程序。使用 GSS-API，应用程序可包括完整性和保密性的安全服务以及验证。
- RPCSEC_GSS 应用程序编程接口 (Application Programming Interface, API)－允许 NFS 服务使用 Kerberos 验证。RPCSEC_GSS 是一种安全特性，可提供与要使用的机制无关的安全服务。RPCSEC_GSS 位于 GSS-API 层的顶部。使用 RPCSEC_GSS 的应用程序可以使用所有基于可插拔 GSS_API 的安全机制。
- 多个库。

此外，Solaris Kerberos 服务还包括：

- SEAM 管理工具 (gkadmin) － 允许您管理 KDC。借助此基于 Java™ 技术的 GUI，管理员可以执行通常通过 kadmin 命令执行的任务。
- 可插拔验证模块 (Pluggable Authentication Module, PAM) － 允许应用程序使用各种验证机制。使用 PAM 可使登录和注销操作对用户而言透明化。
- 内核模块 － 为 NFS 提供 GSS-API 和 RPCSEC_GSS API 的内核实现。

Solaris 10 发行版中的 Kerberos 增强功能

Solaris 10 发行版中包括以下 Kerberos 增强功能。其中的一些增强功能已在先前的 Software Express 发行版中引入，并在 Solaris 10 Beta 版中进行了更新。

- 远程应用程序（如 ftp、rcp、rdist、rlogin、rsh、ssh 和 telnet）支持 Kerberos 协议。有关更多信息，请参见每个命令或守护进程的手册页和 krb5_auth_rules(5) 手册页。
- Kerberos 主体数据库现在可以通过增量更新进行传送，而不必每次传送整个数据库。增量传播有以下优点：
  - 增强了跨服务器数据库的一致性
  - 所需资源（网络、CPU 等）更少
  - 更新的传播更加及时
  - 是一种自动传播方法
- 可提供有助于自动配置 Kerberos 客户机的新脚本。此脚本可以帮助管理员迅速而轻松地安装 Kerberos 客户机。有关使用新脚本的过程，请参见第 388 页中的 “配置 Kerberos 客户机”。此外，有关更多信息，请参见 kclient(1M) 手册页。
- 在 Kerberos 服务中添加了几种新的加密类型。这几种加密类型提高了安全性，并增强了与支持这几种类型的其他 Kerberos 实现的兼容性。有关更多信息，请参见第 512 页中的 “使用 Kerberos 加密类型”。新的加密类型包括：
  - AES，该加密类型可用于高速、高安全性的 Kerberos 会话加密。通过加密框架启用 AES。
  - ARCFOUR-HMAC，该加密类型可提供与其他 Kerberos 实现的更好兼容性。
  - 带有 SHA1 的三重 DES (3DES)，该加密类型提高了安全性，还增强了与支持此种加密类型的其他 Kerberos 实现的互操作性。
- KDC 软件、用户命令和用户应用程序现在支持使用 TCP 网络协议。这种增强功能可提供更强健的操作，以及与其他 Kerberos 实现（包括 Microsoft 的 Active Directory）之间更好的互操作性。现在，KDC 可在传统的 UDP 端口和 TCP 端口进行侦听，因此可响应使用 UDP 或 TCP 协议的请求。用户命令和应用程序在将请求发送到 KDC 时，首先尝试使用 UDP，如果该操作失败，则尝试使用 TCP。
- KDC 软件（包括 kinit、klist 和 kprop 命令）中增加了对 IPv6 的支持。缺省情况下，提供对 IPv6 地址的支持。无需更改任何配置参数即可启用 IPv6 支持。kadmin 和 kadmind 命令不支持 IPv6。
- kadmin 命令的多个子命令中添加了新的 -e 选项。使用此新选项可以在创建主体过程中选择加密类型。有关更多信息，请参见 kadmin(1M) 手册页。

- 对 pam_krb5 模块进行扩充是为了使用 PAM 框架来管理 Kerberos 凭证高速缓存。有关更多信息，请参见 pam_krb5(5) 手册页。
- 支持自动搜索以下各项：Kerberos KDC、管理服务器、kpasswd 服务器以及使用 DNS 查找的主机（或域名）到领域的映射。此增强功能减少了安装 Kerberos 客户机所需的某些步骤。客户机可通过使用 DNS 而不是通过读取配置文件来找到 KDC 服务器。有关更多信息，请参见 krb5.conf(4) 手册页。
- 引入了称为 pam_krb5_migrate 的新 PAM 模块。该新模块可以帮助那些尚未有 Kerberos 帐户的用户自动向本地 Kerberos 领域迁移。有关更多信息，请参见 pam_krb5_migrate(5) 手册页。
- 现在，~/.k5login 文件可以用于 GSS 应用程序 ftp 和 ssh。有关更多信息，请参见 gss_auth_rules(5) 手册页。
- kproplog 实用程序已更新，可输出每个日志项的所有属性名。有关更多信息，请参见 kproplog (1M) 手册页。
- 使用新的配置文件选项，可以基于每个领域对严格的 TGT 验证功能进行选择性配置。有关更多信息，请参见 krb5.conf (4) 手册页。
- 通过扩充更改口令实用程序，Solaris Kerberos V5 管理服务器可接受未运行 Solaris 软件的客户机的口令更改请求。有关更多信息，请参见 kadmind(1M) 手册页。
- 重放高速缓存的缺省位置已从基于 RAM 的文件系统移动到 /var/krb5/rcache/ 中的持久性存储器。新位置在系统重新引导时可以避免重放。rcache 代码的性能得到增强。但是，由于使用了持久性存储器，因此整个重放高速缓存的性能有可能降低。
- 现在，可以将重放高速缓存配置为使用文件存储器或仅限于内存的存储器。 有关可为密钥表和凭证高速缓存类型或位置配置的环境变量的更多信息，请参阅 krb5envvar(5) 手册页。
- 对 Kerberos GSS 机制来说，GSS 凭证表不再是必需的。有关更多信息，请参见第 360 页中的 “将 GSS 凭证映射到 UNIX 凭证” 或 gsscred(1M)、gssd(1M) 和 gsscred.conf(4) 手册页。
- Kerberos 实用程序 kinit 和 ktutil 现在都基于 MIT Kerberos 版本 1.2.1。此更改为 kinit 命令添加了新的选项，并为 ktutil 命令添加了新的子命令。有关更多信息，请参见 kinit(1) 和 ktutil(1) 手册页。
- Solaris Kerberos 密钥分发中心 (KDC) 和 kadmind 现在都基于 MIT Kerberos 版本 1.2.1。现在，KDC 在缺省情况下是一个基于二叉树的数据库，这比当前基于散列的数据库更可靠。有关更多信息，请参见 kdb5_util(1M) 手册页。
- kpropd、kadmind、krb5kdc 和 ktkt_warnd 守护进程由服务管理工具管理。 可以使用 svcadm 命令对此服务执行管理操作，如启用、禁用或重新启动。 可使用 svcs 命令来查询此服务对应的所有守护进程的状态。有关服务管理工具的概述，请参阅《System Administration Guide: Basic Administration》中的第 14 章，“Managing Services (Overview)”。

Solaris 10 6/06 发行版的 Kerberos 新增功能

在 Solaris 10 6/06 发行版中，`ktkt_warnd` 守护进程可以自动更新凭证，而不是在凭证即将到期时才向用户发出警告。用户必须登录才能自动更新凭证。

Solaris 9 发行版中的 Kerberos 组件

Solaris 9 发行版包括第 352 页中的 “Kerberos 组件”中除远程应用程序外的所有组件。

SEAM 1.0.2 组件

SEAM 1.0.2 发行版包括远程应用程序。这些应用程序是 Solaris 9 发行版唯一未包括的部分 SEAM 1.0。远程应用程序的组件如下：

- 客户机应用程序－`ftp`、`rcp`、`rlogin`、`rsh` 和 `telnet`
- 服务器守护进程－`ftpd`、`rlogind`、`rshd` 和 `telnetd`

Solaris 8 发行版中的 Kerberos 组件

Solaris 8 发行版仅包括 Kerberos 服务的客户端部分，因此有许多组件未包括在内。借助此产品，运行 Solaris 8 发行版的系统可成为 Kerberos 客户机，而不需要单独安装 SEAM 1.0.1。要使用这些功能，必须安装使用 Solaris Easy Access Server 3.0 或 Solaris 8 Admin Pack 的 KDC、MIT 分发或 Windows 2000。如果没有配置 KDC 来分发票证，则客户端组件将不起作用。此发行版中包括以下组件：

- 用于获取、查看和销毁票证的用户程序－`kinit`、`klist` 和 `kdestroy`。
- 用于更改 Kerberos 口令的用户程序－`kpasswd`。
- 密钥表管理实用程序－`ktutil`。
- 可插拔验证模块 (Pluggable Authentication Module, PAM) 的新增功能－允许应用程序使用各种验证机制。使用 PAM 可使登录和注销操作对用户而言透明化。
- GSS_API 插件－提供 Kerberos 协议和加密支持。
- NFS 客户机和服务器支持。

SEAM 1.0.1 组件

SEAM 1.0.1 发行版包括 Solaris 8 发行版中未包括的所有 SEAM 1.0 发行版组件。这些组件包括：

- 密钥分发中心 (Key Distribution Center, KDC)（主）：
  - Kerberos 数据库管理守护进程－`kadmind`

  - Kerberos 票证处理守护进程－`krb5kdc`
- 从 KDC。
- 数据库管理程序－`kadmin` 和 `kadmin.local`。
- 数据库传播软件－`kprop`。
- 远程应用程序－`ftp`、`rcp`、`rlogin`、`rsh` 和 `telnet`。
- 远程应用程序守护进程－`ftpd`、`rlogind`、`rshd` 和 `telnetd`。
- 管理实用程序－`kdb5_util`。
- SEAM 管理工具 (`gkadmin`)－允许您管理 KDC。借助此基于 Java 技术的 GUI，管理员可以执行通常通过 `kadmin` 命令执行的任务。
- 预配置过程－允许您设置用于安装和配置 SEAM 1.0.1 的参数，使用这些参数可以自动进行 SEAM 安装。此过程对于批量安装尤其有用。
- 多个库。

SEAM 1.0 组件

SEAM 1.0 发行版包括第 352 页中的 “Kerberos 组件”中的所有项以及以下各项：

- 实用程序 (`gsscred`) 和守护进程 (`gssd`)－这些程序有助于将 UNIX 用户 ID (user ID, UID) 映射到主体名称。需要使用这些程序是因为 NFS 服务器使用 UNIX UID 来标识用户，而不是使用以不同格式存储的主体名称来标识用户。
- 通用安全服务应用程序编程接口 (Generic Security Service Application Programming Interface, GSS-API)－允许应用程序使用多种安全机制，并且不需要在每次添加新机制时重新编译应用程序。因为 GSS-API 与计算机无关，所以适用于 Internet 上的各种应用程序。使用 GSS-API，应用程序可包括完整性和保密性的安全服务以及验证。
- RPCSEC_GSS 应用程序编程接口 (Application Programming Interface, API)－允许 NFS 服务使用 Kerberos 验证。RPCSEC_GSS 是一种安全特性，可提供与要使用的机制无关的安全服务。RPCSEC_GSS 位于 GSS-API 层的顶部。使用 RPCSEC_GSS 的应用程序可以使用所有基于可插拔 GSS_API 的安全机制。
- 预配置过程－允许您设置用于安装和配置 SEAM 1.0.1 的参数，使用这些参数可以自动进行 SEAM 安装。此过程对于批量安装尤其有用。

◆ ◆ ◆ 第 21 章

规划 Kerberos 服务

参与安装和维护 Kerberos 服务的管理员应学习本章。本章介绍管理员在安装或配置服务之前必须确定的一些安装和配置选项。

以下是系统管理员或其他技术支持人员应学习的主题的列表：

- 第 357 页中的 “为什么要规划 Kerberos 部署？”
- 第 358 页中的 “Kerberos 领域”
- 第 359 页中的 “将主机名映射到领域”
- 第 359 页中的 “客户机名称和服务主体名称”
- 第 359 页中的 “KDC 端口和管理服务端口”
- 第 360 页中的 “从 KDC 数”
- 第 361 页中的 “要使用的数据库传播系统”
- 第 361 页中的 “领域内的时钟同步”
- 第 361 页中的 “客户机安装选项”
- 第 361 页中的 “Kerberos 加密类型”
- 第 362 页中的 “SEAM Administration Tool 中的联机帮助 URL”

为什么要规划 Kerberos 部署？

在安装 Kerberos 服务之前，必须解决几个配置问题。虽然在初始安装后可以更改配置，但每向系统中添加一台新客户机便会增加执行此操作的难度。而且某些更改可能需要进行完全重新安装，所以在规划 Kerberos 配置时最好应考虑长期目标。

部署 Kerberos 基础结构涉及以下任务：安装 KDC、为主机创建密钥以及迁移用户。重新配置 Kerberos 部署与执行初始部署一样困难，因此要认真规划部署以避免必须进行重新配置。

Kerberos 领域

领域是一个类似于域的逻辑网络，用于定义一组系统，这些系统位于同一主 KDC 下。与建立 DNS 域名一样，在配置 Kerberos 服务之前，应解决以下问题以便进行跨领域验证：领域名称、领域数和每个领域的大小以及各领域之间的关系。

领域名称

领域名称可以由任何 ASCII 字符串组成。通常，领域名称与 DNS 域名相同，只不过领域名称采用大写。使用常见的名称时，这种约定有助于将 Kerberos 服务问题与 DNS 名称空间问题区分开来。如果不使用 DNS 或选择使用其他字符串，则可以使用任何字符串。但是，配置过程需要更多工作。采用符合标准 Internet 名称结构的领域名称是明智之举。

领域数

安装需要的领域数取决于下列因素：

- 要支持的客户机数。一个领域中具有太多客户机会增加管理难度，最终会要求对领域进行分割。确定可以支持的客户机数的主要因素如下：
  - 每台客户机产生的 Kerberos 通信量
  - 物理网络的带宽
  - 主机的速度

  由于每种安装都有不同的限制，所以不存在确定最大客户机数的原则。
- 客户机间相隔的距离。如果客户机位于不同的地理区域中，则可以设置几个较小的领域。
- 可作为 KDC 安装的主机数。每个领域中应至少有两台 KDC 服务器：一台主服务器和一台从服务器。

建议将 Kerberos 领域与管理域结合使用。请注意，Kerberos V 领域可以跨与该领域相对应的 DNS 域的多个子域。

领域分层结构

为进行跨领域验证而配置多个领域时，需要决定如何将这些领域绑定在一起。可以在这些领域之间建立分层关系，以便提供到相关域的自动路径。当然，必须正确配置分层链中的所有领域。自动路径可以减轻管理负担。但是，如果域有许多层，您可能不想使用缺省路径，因为它需要太多事务。

您也可以选择直接建立连接。当两个分层领域之间存在的层太多或不存在分层关系时，直接连接最有用。必须在使用连接的所有主机上的 `/etc/krb5/krb5.conf` 文件中定义连接。因此，还需要执行一些其他工作。有关介绍，请参见第 350 页中的 “Kerberos 领域”。有关多个领域的配置过程，请参见第 377 页中的 “配置跨领域验证”。

将主机名映射到领域

主机名到领域名称的映射在 krb5.conf 文件的 domain_realm 部分中定义。可以根据需要对整个域和个别的主机定义这些映射。

DNS 还可用于查找有关 KDC 的信息。如果使用 DNS，则会使信息更改变得更加容易，因为每次执行更改时，无需编辑所有客户机上的 krb5.conf 文件。有关更多信息，请参见 krb5.conf(4) 手册页。

客户机名称和服务主体名称

使用 Kerberos 服务时，强烈建议已在所有主机上配置并运行 DNS 服务。如果使用 DNS，则必须在所有主机或未在任何主机上启用它。如果 DNS 可用，则主体应包含每台主机的全限定域名 (Fully Qualified Domain Name, FQDN)。例如，如果主机名是 boston，DNS 域名是 example.com，领域名称是 EXAMPLE.COM，则该主机的主体名称应为 host/boston.example.com@EXAMPLE.COM。本书中的示例要求对每台主机配置 DNS 并且使用 FQDN。

包含主机的 FQDN 的主体名称应与 /etc/resolv.conf 文件中说明 DNS 域名的字符串匹配。指定主体的 FQDN 时，Kerberos 服务要求 DNS 域名必须为小写字母。DNS 域名可以包含大小写字母，但在创建主机主体时只可以使用小写字母。例如，DNS 域名为 example.com、Example.COM 还是任何其他变体并不重要。主机的主体名称仍为 host/boston.example.com@EXAMPLE.COM。

Kerberos 服务可以在没有运行 DNS 服务的情况下运行。但是，一些主要功能（例如，与其他领域通信的功能）将不能工作。如果未配置 DNS，则可以将简单的主机名用作实例名称。在此情况下，主体将为 host/boston@EXAMPLE.COM。如果稍后启用 DNS，则必须删除并替换 KDC 数据库中的所有主机主体。

此外，还配置了服务管理工具，以便未运行 DNS 服务时，不会启动许多守护进程或命令。已将 kdb5_util、kadmind 和 kpropd 守护进程以及 kprop 命令配置为依赖于 DNS 服务。要充分利用使用 Kerberos 服务和 SMF 时可用的功能，必须在所有主机上配置 DNS。

KDC 端口和管理服务端口

缺省情况下，端口 88 和端口 750 用于 KDC，而端口 749 用于 KDC 管理守护进程。可以使用不同的端口号。但是，如果更改端口号，则必须在每台客户机上更改 /etc/services 文件和 /etc/krb5/krb5.conf 文件。此外，还必须更新每个 KDC 上的 /etc/krb5/kdc.conf 文件。

从 KDC 数

与主 KDC 一样，从 KDC 也会为客户机生成凭证。如果主 KDC 不可用，则从 KDC 将提供备份。每个领域应至少有一个从 KDC。可能会需要其他从 KDC，这取决于以下因素：

- 领域中的物理段的个数。通常，应将网络设置为至少每个段都可以独立于领域的其他部分而单独工作。为此，必须能够从每个段访问 KDC。此实例中的 KDC 可以是主 KDC 或从 KDC。
- 领域中的客户机数。通过添加更多从 KDC 服务器，可以减少当前服务器的负荷。

可能会添加太多从 KDC。请记住，必须将 KDC 数据库传播到每台服务器，因此安装的 KDC 服务器越多，更新领域中的数据所用的时间就越长。此外，因为每个从 KDC 都会保存一份 KDC 数据库的副本，所以较多的从 KDC 会增加破坏安全性的风险。

另外，可以很容易地将一个或多个从 KDC 配置为与主 KDC 交换。采用这种方式配置至少一个从 KDC 的优点是：如果主 KDC 由于任何原因出现故障，则可以使用很容易交换为主 KDC 的预配置系统。有关如何配置可交换的从 KDC 的说明，请参见第 405 页中的 “交换主 KDC 和从 KDC”。

将 GSS 凭证映射到 UNIX 凭证

对于需要 GSS 凭证名称到 UNIX 用户 ID (user ID, UID) 的映射的 GSS 应用程序（例如 NFS），Kerberos 服务提供了该缺省映射。使用 Kerberos 服务时，GSS 凭证名称相当于 Kerberos 主体名称。缺省映射算法是采用具有一个组成部分的 Kerberos 主体名称，并使用该组成部分（即主体的主名称）来查找 UID。可以使用 `/etc/krb5/krb5.conf` 中的 `auth_to_local_realm` 参数在缺省领域或允许的任何领域中进行查找。例如，可以使用口令表将用户主体名称 `bob@EXAMPLE.COM` 映射到名为 `bob` 的 UNIX 用户的 UID。但不会映射用户主体名称 `bob/admin@EXAMPLE.COM`，因为该主体名称包括 `admin` 的实例部分。如果用户凭证的缺省映射满足要求，则无需填充 GSS 凭证表。在先前的发行版中，需要填充 GSS 凭证表才能使 NFS 服务工作。如果缺省映射不满足要求（例如，如果要映射包含实例部分的主体名称），应使用其他方法。有关更多信息，请参见：

- 第 384 页中的 “如何创建凭证表”
- 第 384 页中的 “如何向凭证表中添加单个项”
- 第 385 页中的 “如何提供各领域之间的凭证映射”
- 第 443 页中的 “观察从 GSS 凭证到 UNIX 凭证的映射”

自动将用户迁移到 Kerberos 领域

可以使用 PAM 框架自动迁移在缺省 Kerberos 领域中没有有效用户帐户的 UNIX 用户。具体而言，将在 PAM 服务的验证栈中使用 `pam_krb5_migrate` 模块。该模块将对服务进行设置，以便每当没有 Kerberos 主体的用户使用其口令成功登录系统时，便会自动为该用户创建 Kerberos 主体。新的主体将使用相同的口令。有关如何使用 `pam_krb5_migrate` 模块的说明，请参见第 401 页中的 “在 Kerberos 领域中配置用户自动迁移”。

要使用的数据库传播系统

存储在主 KDC 上的数据库必须定期传播到从 KDC。可以将数据库的传播配置为增量。增量过程只将更新的信息传播到从 KDC，而不是整个数据库。 有关数据库传播的更多信息，请参见第 412 页中的 “管理 Kerberos 数据库”。

如果不使用增量传播，则要解决的首要问题之一是确定更新从 KDC 的频率。需要获取所有客户机可用的最新信息时，必须权衡完成更新所需的时间。

在一个领域中有许多 KDC 的大型安装中，一个或多个从 KDC 可以传播数据，以便以并行方式完成该过程。此策略减少了更新所用的时间，但同时增加了管理领域的复杂性。有关此策略的完整说明，请参见第 426 页中的 “设置并行传播”。

领域内的时钟同步

所有参与 Kerberos 验证系统的主机都必须在指定的最长时间内同步其内部时钟。称为**时钟相位差**的此功能提供了另一种 Kerberos 安全检查方法。如果任意两台参与主机之间的时间偏差超过了时钟相位差，则请求会被拒绝。

一种同步所有时钟的方法是使用网络时间协议 (Network Time Protocol, NTP) 软件。有关更多信息，请参见第 404 页中的 “同步 KDC 和 Kerberos 客户机的时钟”。还存在其他同步时钟的方法，因此并非一定要使用 NTP。但是，由于存在时钟相位差，所以应使用某种形式的同步来防止访问失败。

客户机安装选项

Solaris 10 发行版中的一个新功能是 `kclient` 安装实用程序。该实用程序可以交互模式或非交互模式运行。在交互模式下，将提示用户输入特定于 Kerberos 的参数值，从而允许用户在安装客户机时更改现有安装。在非交互模式下，将使用先前设置了参数值的文件。此外，可以在非交互模式下使用命令行选项。交互模式和非交互模式所需的步骤都比手动过程所需的步骤少，因此该过程可以更快完成并且不容易出错。有关所有客户机安装过程的说明，请参见第 388 页中的 “配置 Kerberos 客户机”。

Kerberos 加密类型

加密类型是指定 Kerberos 服务中使用的加密算法、加密模式和散列算法的标识符。Kerberos 服务中的密钥具有关联的加密类型，用于标识服务使用该密钥执行加密操作时要使用的加密算法和模式。以下是 Solaris 10 发行版中支持的加密类型：

- `des-cbc-md5`
- `des-cbc-crc`
- `des3-cbc-sha1`
- `arcfour-hmac-md5`

- arcfour-hmac-md5-exp
- aes128-cts-hmac-sha1-96

注 – 此外，如果安装了非捆绑强加密软件包，则可以将 aes256-cts-hmac-sha1-96 加密类型用于 Kerberos 服务。

如果要更改加密类型，则应在创建新的主体数据库时进行更改。由于 KDC 服务器与 KDC 客户机之间存在交互，所以在现有数据库上更改加密类型很困难。除非您要重新创建数据库，否则不要设置这些参数。有关更多信息，请参阅第 512 页中的 “使用 Kerberos 加密类型” 。

注 – 如果安装了未运行 Solaris 10 发行版的主 KDC，则在升级主 KDC 之前，必须将从 KDC 升级到 Solaris 10 发行版。Solaris 10 主 KDC 将使用新的加密类型，而较早版本的从 KDC 将无法处理这些加密类型。

SEAM Administration Tool 中的联机帮助 URL

SEAM Administration Tool 将使用联机帮助 URL，因此应正确定义该 URL，以使“帮助内容“菜单正常工作。可以在任何合适的服务器上安装本手册的 HTML 版本。或者，可以决定使用 http://docs.sun.com 中的文档集。

该 URL 是在配置主机以使用 Kerberos 服务时在 krb5.conf 文件中指定的。该 URL 应指向本书的“管理主体和策略（任务）”章中标题为 "SEAM Administration Tool" 的一节。如果存在更适合的位置，可以选择另一个 HTML 页面。

◆ ◆ ◆ 第 22 章

配置 Kerberos 服务（任务）

本章介绍 KDC 服务器、网络应用程序服务器、NFS 服务器和 Kerberos 客户机的配置过程。其中许多过程都要求超级用户访问权限，因此这些过程应由系统管理员或高级用户来执行。本章还将介绍跨领域配置过程以及与 KDC 服务器相关的其他主题。

本章包含以下主题：

- 第 363 页中的 “配置 Kerberos 服务（任务列表）”
- 第 364 页中的 “配置 KDC 服务器”
- 第 377 页中的 “配置跨领域验证”
- 第 380 页中的 “配置 Kerberos 网络应用程序服务器”
- 第 382 页中的 “配置 Kerberos NFS 服务器”
- 第 388 页中的 “配置 Kerberos 客户机”
- 第 404 页中的 “同步 KDC 和 Kerberos 客户机的时钟”
- 第 405 页中的 “交换主 KDC 和从 KDC”
- 第 412 页中的 “管理 Kerberos 数据库”
- 第 428 页中的 “增强 Kerberos 服务器的安全性”

配置 Kerberos 服务（任务列表）

部分配置过程依赖于其他配置过程，且必须按特定顺序执行。这些过程通常会建立使用 Kerberos 服务所需的服务。其他过程不依赖于任何顺序，且可以在适当的情况下执行。以下任务列表给出了 Kerberos 安装的建议顺序。

任务	说明	参考
1. 规划 Kerberos 安装。	在开始软件配置过程之前先解决配置问题。从长远来看，提前规划可以节省时间和其他资源。	第 21 章

任务	说明	参考
2. （可选）安装 NTP。	配置网络时间协议 (Network Time Protocol, NTP) 软件或其他时钟同步协议。要使 Kerberos 服务正常工作，必须同步领域中所有系统的时钟。	第 404 页中的 “同步 KDC 和 Kerberos 客户机的时钟”
3. 配置主 KDC 服务器。	配置并构建领域的主 KDC 服务器和数据库。	第 365 页中的 “如何配置主 KDC”
4. 配置从 KDC 服务器。	配置并构建领域的从 KDC 服务器。	第 372 页中的 “如何配置从 KDC”
5. （可选）增强 KDC 服务器的安全性。	阻止对 KDC 服务器的安全性破坏。	第 428 页中的 “如何限制对 KDC 服务器的访问”
6. （可选）配置可交换的 KDC 服务器。	使交换主 KDC 和从 KDC 的任务更容易执行。	第 405 页中的 “如何配置可交换的从 KDC”

配置其他 Kerberos 服务（任务列表）

完成所需步骤之后，可以在适当的情况下执行以下过程。

任务	说明	参考
配置跨领域验证。	启用领域之间的通信。	第 377 页中的 “配置跨领域验证”
配置 Kerberos 应用程序服务器。	使服务器支持使用 Kerberos 验证的服务，例如 `ftp`、`telnet` 和 `rsh`。	第 380 页中的 “配置 Kerberos 网络应用程序服务器”
配置 Kerberos 客户机。	使客户机使用 Kerberos 服务。	第 388 页中的 “配置 Kerberos 客户机”
配置 Kerberos NFS 服务器。	使服务器共享要求 Kerberos 验证的文件系统。	第 382 页中的 “配置 Kerberos NFS 服务器”
增强应用程序服务器的安全性。	通过只允许访问经过验证的事务来增强应用程序服务器的安全性。	第 428 页中的 “如何仅启用基于 Kerberos 的应用程序”

配置 KDC 服务器

安装 Kerberos 软件后，必须配置 KDC 服务器。配置一个主 KDC 和至少一个从 KDC 以提供颁发凭证的服务。这些凭证是 Kerberos 服务的基础，因此在尝试其他任务之前必须安装 KDC。

主 KDC 和从 KDC 之间的最大差别是，只有主 KDC 可以处理数据库管理请求。例如，更改口令或添加新的主体必须在主 KDC 上完成。然后可以将这些更改传播到从 KDC。从 KDC 和主 KDC 都可生成凭证。此功能可在主 KDC 无法响应时提供冗余性。

▼ 如何配置主 KDC

在此过程中，将配置增量传播。此外，还将使用以下配置参数：

- 领域名称 = EXAMPLE.COM
- DNS 域名 = example.com
- 主 KDC = kdc1.example.com
- admin 主体 = kws/admin
- 联机帮助 URL = http://denver:8888/ab2/coll.384.1/SEAM/@AB2PageView/6956

注 – 调整该 URL 以指向“SEAM Administration Tool”部分，如第 362 页中的 “SEAM Administration Tool 中的联机帮助 URL”中所述。

开始之前 此过程要求 DNS 必须正在运行。有关此主 KDC 是否可交换的特定命名说明，请参见第 405 页中的 “交换主 KDC 和从 KDC”。

1. **成为主 KDC 的超级用户。**

2. **编辑 Kerberos 配置文件 (krb5.conf)。**
   需要更改领域名称和服务器名称。有关此文件的完整说明，请参见 krb5.conf(4) 手册页。

```
kdc1 # cat /etc/krb5/krb5.conf

[libdefaults]

        default_realm = EXAMPLE.COM


[realms]

                EXAMPLE.COM = {

                kdc = kdc1.example.com

                admin_server = kdc1.example.com


        }


[domain_realm]
```

```
        .example.com = EXAMPLE.COM

#

# if the domain name and realm name are equivalent,

# this entry is not needed

#

[logging]

        default = FILE:/var/krb5/kdc.log

        kdc = FILE:/var/krb5/kdc.log



[appdefaults]

    gkadmin = {

        help_url = http://denver:8888/ab2/coll.384.1/SEAM/@AB2PageView/6956

        }
```

在此示例中，更改了 `default_realm`、`kdc` 和 `admin_server` 行以及所有 `domain_realm` 项。此外，还编辑了定义 `help_url` 的行。

注 – 如果要限制加密类型，可以设置 `default_tkt_enctypes` 或 `default_tgs_enctypes` 行。有关限制加密类型涉及的问题的说明，请参阅第 512 页中的 “使用 Kerberos 加密类型” 。

---

3 编辑 KDC 配置文件 (`kdc.conf`)。

需要更改领域名称。有关此文件的完整说明，请参见 `kdc.conf`(4) 手册页。

```
kdc1 # cat /etc/krb5/kdc.conf

[kdcdefaults]

        kdc_ports = 88,750



[realms]

        EXAMPLE.COM= {

                profile = /etc/krb5/krb5.conf
```

```
database_name = /var/krb5/principal

admin_keytab = /etc/krb5/kadm5.keytab

acl_file = /etc/krb5/kadm5.acl

kadmind_port = 749

max_life = 8h 0m 0s

max_renewable_life = 7d 0h 0m 0s

sunw_dbprop_enable = true

sunw_dbprop_master_ulogsize = 1000        }
```

在此示例中，更改了 realms 部分中的领域名称定义。此外，在 realms 部分中，添加了用于启用增量传播和选择主 KDC 将保留在日志中的更新数的行。

注 – 如果需要限制加密类型，可以设置 permitted_enctypes、supported_enctypes 或 master_key_type 行。有关限制加密类型涉及的问题的说明，请参阅第 512 页中的 “使用 Kerberos 加密类型”。

---

4 使用 kdb5_util 命令创建 KDC 数据库。

kdb5_util 命令创建 KDC 数据库。此外，与 -s 选项一起使用时，此命令会在启动 kadmind 和 krb5kdc 守护进程之前，创建一个用于向自己验证 KDC 的存储文件。

```
kdc1 # /usr/sbin/kdb5_util create -r EXAMPLE.COM -s

Initializing database ’/var/krb5/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:      <Type the key>

Re-enter KDC database master key to verify:      <Type it again>
```

如果领域名称与服务器的名称空间中的域名相同，则无需后跟领域名称的 -r 选项。

5 编辑 Kerberos 访问控制列表文件 (kadm5.acl)。

填充后，/etc/krb5/kadm5.acl 文件应包含允许管理 KDC 的所有主体名称。

```
kws/admin@EXAMPLE.COM   *
```

通过该项，EXAMPLE.COM 领域中的 kws/admin 主体可以修改 KDC 中的主体或策略。缺省安装包括用于匹配所有 admin 主体的星号 (*)。此缺省安装可能会存在安全风险，因此更安全的方法是包括所有 admin 主体的列表。有关更多信息，请参见 kadm5.acl(4) 手册页。

6 启动 kadmin.local 命令并添加主体。

接下来的子步骤创建 Kerberos 服务使用的主体。

```
kdc1 # /usr/sbin/kadmin.local

kadmin.local:
```

a. 向数据库中添加管理主体。

可以根据需要添加任意数量的 admin 主体。要完成 KDC 配置过程，必须至少添加一个 admin 主体。对于此示例，将添加一个 kws/admin 主体。可以将 "kws" 替换为相应的主体名称。

```
kadmin.local: addprinc kws/admin

Enter password for principal kws/admin@EXAMPLE.COM:     <Type the password>

Re-enter password for principal kws/admin@EXAMPLE.COM:      <Type it again>

Principal "kws/admin@EXAMPLE.COM" created.

kadmin.local:
```

b. 创建 kiprop 主体。

kiprop 主体用于授权来自主 KDC 的更新。

```
kadmin.local: addprinc -randkey kiprop/kdc1.example.com

Principal "kiprop/kdc1.example.com@EXAMPLE.COM" created.

kadmin.local:
```

c. 为 kadmind 服务创建密钥表文件。

此命令序列创建包含 kadmin 和 changepw 的主体项的特殊密钥表文件。kadmind 服务需要使用这些主体。请注意，主体实例为主机名时，无论 /etc/resolv.conf 文件中的域名是大写还是小写，都必须以小写字母指定 FQDN。

```
kadmin.local: ktadd -k /etc/krb5/kadm5.keytab kadmin/kdc1.example.com

Entry for principal kadmin/kdc1.example.com with kvno 3, encryption type AES-128 CTS mode

          with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/kadm5.keytab.

Entry for principal kadmin/kdc1.example.com with kvno 3, encryption type Triple DES cbc

          mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/kadm5.keytab.
```

```
Entry for principal kadmin/kdc1.example.com with kvno 3, encryption type ARCFOUR
          with HMAC/md5 added to keytab WRFILE:/etc/krb5/kadm5.keytab.
Entry for principal kadmin/kdc1.example.com with kvno 3, encryption type DES cbc mode
          with RSA-MD5 added to keytab WRFILE:/etc/krb5/kadm5.keytab.
kadmin.local: ktadd -k /etc/krb5/kadm5.keytab changepw/kdc1.example.com
EEntry for principal changepw/kdc1.example.com with kvno 3, encryption type AES-128 CTS mode
          with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/kadm5.keytab.
Entry for principal changepw/kdc1.example.com with kvno 3, encryption type Triple DES cbc
          mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/kadm5.keytab.
Entry for principal changepw/kdc1.example.com with kvno 3, encryption type ARCFOUR
          with HMAC/md5 added to keytab WRFILE:/etc/krb5/kadm5.keytab.
Entry for principal changepw/kdc1.example.com with kvno 3, encryption type DES cbc mode
          with RSA-MD5 added to keytab WRFILE:/etc/krb5/kadm5.keytab.
kadmin.local: ktadd -k /etc/krb5/kadm5.keytab kadmin/changepw
Entry for principal kadmin/changepw with kvno 3, encryption type AES-128 CTS mode
          with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/kadm5.keytab.
Entry for principal kadmin/changepw with kvno 3, encryption type Triple DES cbc
          mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/kadm5.keytab.
Entry for principal kadmin/changepw with kvno 3, encryption type ARCFOUR
          with HMAC/md5 added to keytab WRFILE:/etc/krb5/kadm5.keytab.
Entry for principal kadmin/changepw with kvno 3, encryption type DES cbc mode
          with RSA-MD5 added to keytab WRFILE:/etc/krb5/kadm5.keytab.
kadmin.local:
```

d. 将主 KDC 服务器的 kiprop 主体添加到 kadmind 密钥表文件中。

通过将 kiprop 主体添加到 kadm5.keytab 文件中，kadmind 命令可以在启动增量传播时对其自身进行验证。

```
kadmin.local: ktadd -k /etc/krb5/kadm5.keytab kiprop/kdc1.example.com

Entry for principal kiprop/kdc1.example.com with kvno 3, encryption type AES-128 CTS mode

          with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/kadm5.keytab.

Entry for principal kiprop/kdc1.example.com with kvno 3, encryption type Triple DES cbc

          mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/kadm5.keytab.

Entry for principal kiprop/kdc1.example.com with kvno 3, encryption type ARCFOUR

          with HMAC/md5 added to keytab WRFILE:/etc/krb5/kadm5.keytab.

Entry for principal kiprop/kdc1.example.com with kvno 3, encryption type DES cbc mode

          with RSA-MD5 added to keytab WRFILE:/etc/krb5/kadm5.keytab.

kadmin.local:
```

e. 退出 kadmin.local。

已经添加了接下来的步骤所需的所有主体。

```
kadmin.local: quit
```

7 启动 Kerberos 守护进程。

```
kdc1 # svcadm enable -r network/security/krb5kdc

kdc1 # svcadm enable -r network/security/kadmin
```

8 启动 kadmin 并添加更多主体。

此时，可以使用 SEAM Administration Tool 添加主体。为此，必须使用此过程前面创建的一个 admin 主体名称登录。但是，为简单起见，给出了以下命令行示例。

```
kdc1 # /usr/sbin/kadmin -p kws/admin

Enter password:      <Type kws/admin password>
```

```
kadmin:
```

a. 创建主 KDC `host` 主体。

基于 Kerberos 的应用程序（例如 `klist` 和 `kprop`）将使用主机主体。Solaris 10 客户机在挂载经过验证的 NFS 文件系统时将使用此主体。请注意，主体实例为主机名时，无论 `/etc/resolv.conf` 文件中的域名是大写还是小写，都必须以小写字母指定 FQDN。

```
kadmin: addprinc -randkey host/kdc1.example.com

Principal "host/kdc1.example.com@EXAMPLE.COM" created.

kadmin:
```

b. （可选的）创建 `kclient` 主体。

安装 Kerberos 客户机过程中 `kclient` 实用程序将使用此主体。如果不打算使用此实用程序，则无需添加该主体。`kclient` 实用程序的用户需要使用此口令。

```
kadmin: addprinc clntconfig/admin

Enter password for principal clntconfig/admin@EXAMPLE.COM:     <Type the password>

Re-enter password for principal clntconfig/admin@EXAMPLE.COM:      <Type it again>

Principal "clntconfig/admin@EXAMPLE.COM" created.

kadmin:
```

c. 将主 KDC 的 `host` 主体添加到主 KDC 的密钥表文件中。

通过将主机主体添加到密钥表文件中，可以自动使用此主体。

```
kadmin: ktadd host/kdc1.example.com

Entry for principal host/kdc1.example.com with kvno 3, encryption type AES-128 CTS mode

          with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal host/kdc1.example.com with kvno 3, encryption type Triple DES cbc

          mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal host/kdc1.example.com with kvno 3, encryption type ARCFOUR

          with HMAC/md5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal host/kdc1.example.com with kvno 3, encryption type DES cbc mode

          with RSA-MD5 added to keytab WRFILE:/etc/krb5/krb5.keytab.
```

```
kadmin:
```

d. 退出 kadmin。

```
kadmin: quit
```

9 （可选的）使用 NTP 或其他时钟同步机制同步主 KDC 时钟。

安装和使用网络时间协议 (Network Time Protocol, NTP) 并非必需。但是，要成功验证，每个时钟都必须处于 krb5.conf 文件的 libdefaults 部分中定义的缺省时间内。有关 NTP 的信息，请参见第 404 页中的 “同步 KDC 和 Kerberos 客户机的时钟”。

10 配置从 KDC。

要提供冗余性，请确保至少安装一个从 KDC。有关特定说明，请参见第 372 页中的 “如何配置从 KDC”。

▼ 如何配置从 KDC

在此过程中，将配置一个名为 kdc2 的新从 KDC。此外，还将配置增量传播。此过程使用以下配置参数：

- 领域名称 = EXAMPLE.COM
- DNS 域名 = example.com
- 主 KDC = kdc1.example.com
- 从 KDC = kdc2.example.com
- admin 主体 = kws/admin
- 联机帮助 URL = http://denver:8888/ab2/coll.384.1/SEAM/@AB2PageView/6956

注 – 调整该 URL 以指向“SEAM Administration Tool”部分，如第 362 页中的 “SEAM Administration Tool 中的联机帮助 URL”中所述。

开始之前 必须配置主 KDC。有关此从 KDC 是否可交换的特定说明，请参见第 405 页中的 “交换主 KDC 和从 KDC”。

1 在主 KDC 上，成为超级用户。

2 在主 KDC 上，启动 kadmin。

必须使用在配置主 KDC 时创建的一个 admin 主体名称登录。

```
kdc1 # /usr/sbin/kadmin -p kws/admin

Enter password:      <Type kws/admin password>
```

```
kadmin:
```

a. 在主 KDC 上，将从主机主体添加到数据库中（如果尚未执行此操作）。

要使从 KDC 正常工作，该从 KDC 必须具有主机主体。请注意，主体实例为主机名时，无论 /etc/resolv.conf 文件中的域名是大写还是小写，都必须以小写字母指定 FQDN。

```
kadmin: addprinc -randkey host/kdc2.example.com

Principal "host/kdc2@EXAMPLE.COM" created.

kadmin:
```

b. 在主 KDC 上，创建 kiprop 主体。

kiprop 主体用于授权来自主 KDC 的增量传播。

```
kadmin: addprinc -randkey kiprop/kdc2.example.com

Principal "kiprop/kdc2.example.com@EXAMPLE.COM" created.

kadmin:
```

c. 退出 kadmin。

```
kadmin: quit
```

3 在主 KDC 上，编辑 Kerberos 配置文件 (krb5.conf)。

需要添加每个从 KDC 的项。有关此文件的完整说明，请参见 krb5.conf(4) 手册页。

```
kdc1 # cat /etc/krb5/krb5.conf

 .

 .

[realms]

                EXAMPLE.COM = {

                kdc = kdc1.example.com

                kdc = kdc2.example.com

                admin_server = kdc1.example.com

        }
```

4 **在主 KDC 上，将 kiprop 项添加到 kadm5.acl 中。**

通过此项，主 KDC 可以接收对 kdc2 服务器的增量传播请求。

```
kdc1 # cat /etc/krb5/kadm5.acl

*/admin@EXAMPLE.COM *

kiprop/kdc2.example.com@EXAMPLE.COM p
```

5 **在主 KDC 上，重新启动 kadmind 以使用 kadm5.acl 文件中的新项。**

```
kdc1 # svcadm restart network/security/kadmin
```

6 **在所有从 KDC 上，复制主 KDC 服务器的 KDC 管理文件。**

由于主 KDC 服务器已更新每台 KDC 服务器所需的信息，因此需要在所有从 KDC 上执行此步骤。可以使用 ftp 或类似的传送机制从主 KDC 获取以下文件的副本：

- /etc/krb5/krb5.conf
- /etc/krb5/kdc.conf

7 **在所有从 KDC 上，将主 KDC 和每个从 KDC 的项添加到数据库传播配置文件 kpropd.acl 中。**

需要更新所有从 KDC 服务器上的此信息。

```
kdc2 # cat /etc/krb5/kpropd.acl

host/kdc1.example.com@EXAMPLE.COM

host/kdc2.example.com@EXAMPLE.COM
```

8 **在所有从 KDC 上，请确保未填充 Kerberos 访问控制列表文件 kadm5.acl。**

未修改的 kadm5.acl 文件如下所示：

```
kdc2 # cat /etc/krb5/kadm5.acl

*/admin@___default_realm___ *
```

如果此文件中包含 kiprop 项，请删除它们。

9 **在新的从 KDC 上，更改 kdc.conf 中的项。**

将 sunw_dbprop_master_ulogsize 项替换为定义 sunw_dbprop_slave_poll 的项。该项将轮询时间设置为 2 分钟。

```
kdc1 # cat /etc/krb5/kdc.conf

[kdcdefaults]

        kdc_ports = 88,750
```

```
[realms]

        EXAMPLE.COM= {

                profile = /etc/krb5/krb5.conf

                database_name = /var/krb5/principal

                admin_keytab = /etc/krb5/kadm5.keytab

                acl_file = /etc/krb5/kadm5.acl

                kadmind_port = 749

                max_life = 8h 0m 0s

                max_renewable_life = 7d 0h 0m 0s

                sunw_dbprop_enable = true

                sunw_dbprop_slave_poll = 2m

        }
```

10 **在新的从 KDC 上，启动 kadmin 命令。**

必须使用在配置主 KDC 时创建的一个 admin 主体名称登录。

```
kdc2 # /usr/sbin/kadmin -p kws/admin

Enter password:      <Type kws/admin password>

kadmin:
```

a. **使用 kadmin 将从 KDC 的主机主体添加到从 KDC 的密钥表文件中。**

此项可使 kprop 和其他基于 Kerberos 的应用程序正常工作。请注意，主体实例为主机名时，无论 /etc/resolv.conf 文件中的域名是大写还是小写，都必须以小写字母指定 FQDN。

```
kadmin: ktadd host/kdc2.example.com

Entry for principal host/kdc2.example.com with kvno 3, encryption type AES-128 CTS mode

          with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal host/kdc2.example.com with kvno 3, encryption type Triple DES cbc

          mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal host/kdc2.example.com with kvno 3, encryption type ARCFOUR
```

```
          with HMAC/md5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal host/kdc2.example.com with kvno 3, encryption type DES cbc mode

          with RSA-MD5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

kadmin:
```

b. **将 kiprop 主体添加到从 KDC 的密钥表文件中。**

通过将 kiprop 主体添加到 krb5.keytab 文件中，kpropd 命令可以在启动增量传播时对其自身进行验证。

```
kadmin: ktadd kiprop/kdc2.example.com

Entry for principal kiprop/kdc2.example.com with kvno 3, encryption type AES-128 CTS mode

          with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal kiprop/kdc2.example.com with kvno 3, encryption type Triple DES cbc

          mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal kiprop/kdc2.example.com with kvno 3, encryption type ARCFOUR

          with HMAC/md5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal kiprop/kdc2.example.com with kvno 3, encryption type DES cbc mode

          with RSA-MD5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

kadmin:
```

c. **退出 kadmin。**

```
kadmin: quit
```

11 **在新的从 KDC 上，启动 Kerberos 传播守护进程。**

```
kdc2 # /usr/lib/krb5/kpropd
```

12 **在新的从 KDC 上，使用 kdb5_util 创建一个存储文件。**

```
kdc2 # /usr/sbin/kdb5_util stash

kdb5_util: Cannot find/read stored master key while reading master key

kdb5_util: Warning: proceeding without master key
```

```
Enter KDC database master key:      <Type the key>
```

13 **中止 Kerberos 传播守护进程。**

```
kdc2 # pkill kpropd
```

14 **（可选的）在新的从 KDC 上，使用 NTP 或其他时钟同步机制同步主 KDC 时钟。**

安装和使用网络时间协议 (Network Time Protocol, NTP) 并非必需。但是，要成功验证，每个时钟必须处于 `krb5.conf` 文件的 `libdefaults` 部分中定义的缺省时间内。有关 NTP 的信息，请参见第 404 页中的 “同步 KDC 和 Kerberos 客户机的时钟”。

15 **在新的从 KDC 上，启动 KDC 守护进程 (`krb5kdc`)。**

启用 `krb5kdc` 服务时，如果系统配置为从 KDC，则还将启动 `kpropd`。

```
kdc2 # svcadm enable network/security/krb5kdc
```

配置跨领域验证

有几种方法可以将各个领域链接在一起，从而可以在一个领域中验证另一个领域中的用户。通常，跨领域验证通过在两个领域之间建立共享私钥来实现。领域之间的关系可以是分层关系或直接关系（请参见第 358 页中的 “领域分层结构”）。

▼ 如何建立分层跨领域验证

此过程中的示例使用 `ENG.EAST.EXAMPLE.COM` 和 `EAST.EXAMPLE.COM` 两个领域。将按两个方向建立跨领域验证。必须在两个领域的主 KDC 上完成此过程。

开始之前 必须配置每个领域的主 KDC。要完全测试验证过程，必须安装多个客户机或从 KDC。

1 **成为第一个主 KDC 的超级用户。**

2 **为两个领域创建票证授予票证服务主体。**

必须使用在配置主 KDC 时创建的一个 `admin` 主体名称登录。

```
# /usr/sbin/kadmin -p kws/admin

Enter password:      <Type kws/admin password>

kadmin: addprinc krbtgt/ENG.EAST.EXAMPLE.COM@EAST.EXAMPLE.COM

Enter password for principal krgtgt/ENG.EAST.EXAMPLE.COM@EAST.EXAMPLE.COM:      <Type password>

kadmin: addprinc krbtgt/EAST.EXAMPLE.COM@ENG.EAST.EXAMPLE.COM
```

```
Enter password for principal krgtgt/EAST.EXAMPLE.COM@ENG.EAST.EXAMPLE.COM:      <Type password>

kadmin: quit
```

注 – 在两个 KDC 中为每个服务主体指定的口令必须相同。因此，服务主体 krbtgt/ENG.EAST.EXAMPLE.COM@EAST.EXAMPLE.COM 的口令在两个领域中一定相同。

3 **将相应项添加到 Kerberos 配置文件 (krb5.conf) 中以定义每个领域的域名。**

```
# cat /etc/krb5/krb5.conf

[libdefaults]

 .

 .

[domain_realm]

        .eng.east.example.com = ENG.EAST.EXAMPLE.COM

        .east.example.com = EAST.EXAMPLE.COM
```

在此示例中，定义了 ENG.EAST.EXAMPLE.COM 和 EAST.EXAMPLE.COM 领域的域名。由于会从上向下搜索文件，因此先包含子域非常重要。

4 **将 Kerberos 配置文件复制到此领域中的所有客户机。**

要使跨领域验证正常工作，所有系统（包括从 KDC 和其他服务器）必须安装 Kerberos 配置文件 (/etc/krb5/krb5.conf) 的新版本。

5 **在第二个领域中重复以上所有步骤。**

▼ 如何建立直接跨领域验证

此过程中的示例使用 ENG.EAST.EXAMPLE.COM 和 SALES.WEST.EXAMPLE.COM 两个领域。将按两个方向建立跨领域验证。必须在两个领域的主 KDC 上完成此过程。

开始之前 必须配置每个领域的主 KDC。要完全测试验证过程，必须安装多个客户机或从 KDC。

1 **成为一台主 KDC 服务器的超级用户。**

2 **为两个领域创建票证授予票证服务主体。**

必须使用在配置主 KDC 时创建的一个 admin 主体名称登录。

```
# /usr/sbin/kadmin -p kws/admin
```

```
Enter password:       <Type kws/admin password>

kadmin: addprinc krbtgt/ENG.EAST.EXAMPLE.COM@SALES.WEST.EXAMPLE.COM

Enter password for principal

  krgtgt/ENG.EAST.EXAMPLE.COM@SALES.WEST.EXAMPLE.COM:       <Type the password>

kadmin: addprinc krbtgt/SALES.WEST.EXAMPLE.COM@ENG.EAST.EXAMPLE.COM

Enter password for principal

  krgtgt/SALES.WEST.EXAMPLE.COM@ENG.EAST.EXAMPLE.COM:       <Type the password>

kadmin: quit
```

注 – 在两个 KDC 中为每个服务主体指定的口令必须相同。因此，服务主体 krbtgt/ENG.EAST.EXAMPLE.COM@SALES.WEST.EXAMPLE.COM 的口令在两个领域中一定相同。

3 在 Kerberos 配置文件中添加相应的项以定义指向远程领域的直接路径。

此示例显示了 ENG.EAST.EXAMPLE.COM 领域中的客户机。可能需要交换领域名称以获取 SALES.WEST.EXAMPLE.COM 领域中相应的定义。

```
# cat /etc/krb5/krb5.conf

[libdefaults]

 .

 .

[capaths]

    ENG.EAST.EXAMPLE.COM = {

        SALES.WEST.EXAMPLE.COM = .

    }


    SALES.WEST.EXAMPLE.COM = {

         ENG.EAST.EXAMPLE.COM = .

    }
```

4 **将 Kerberos 配置文件复制到当前领域中的所有客户机。**
要使跨领域验证正常工作，所有系统（包括从 KDC 和其他服务器）必须安装 Kerberos 配置文件 (`/etc/krb5/krb5.conf`) 的新版本。

5 **对第二个领域重复以上所有步骤。**

配置 Kerberos 网络应用程序服务器

网络应用程序服务器是使用以下一个或多个网络应用程序提供访问的主机：`ftp`、`rcp`、`rlogin`、`rsh` 和 `telnet`。要在服务器上启用这些命令的 Kerberos 版本，只需执行几个步骤。

▼ 如何配置 Kerberos 网络应用程序服务器

此过程使用以下配置参数：

- 应用程序服务器 = `boston`
- `admin` 主体 = `kws/admin`
- DNS 域名 = `example.com`
- 领域名称 = `EXAMPLE.COM`

开始之前 此过程要求已配置主 KDC。要完全测试该过程，必须安装多个客户机。

1 **安装 Kerberos 客户机软件。**

2 **（可选的）安装 NTP 客户机或其他时钟同步机制。**
有关 NTP 的信息，请参见第 404 页中的 “同步 KDC 和 Kerberos 客户机的时钟”。

3 **为新服务器添加主体并更新该服务器的密钥表。**
以下命令报告是否存在主机主体：

```
boston # klist -k |grep host

4 host/boston.example.com@EXAMPLE.COM

4 host/boston.example.com@EXAMPLE.COM

4 host/boston.example.com@EXAMPLE.COM

4 host/boston.example.com@EXAMPLE.COM
```

如果此命令未返回主体，则可以使用以下步骤创建新主体。

有关如何使用 SEAM Administration Tool 添加主体的说明将在第 456 页中的 “如何创建新的 Kerberos 主体”中介绍。以下步骤中的示例说明如何使用命令行添加所需的主体。必须使用在配置主 KDC 时创建的一个 admin 主体名称登录。

```
boston # /usr/sbin/kadmin -p kws/admin

Enter password:      <Type kws/admin password>

kadmin:
```

a. 创建服务器的 host 主体。

```
kadmin: addprinc -randkey host/boston.example.com

Principal "host/boston.example.com" created.

kadmin:
```

b. 将服务器的 host 主体添加到服务器的密钥表中。

如果未运行 kadmin 命令，请使用以下类似命令重新启动该命令： /usr/sbin/kadmin -p *kws*/admin

```
kadmin: ktadd host/boston.example.com

Entry for principal host/boston.example.com with kvno 3, encryption type AES-128 CTS mode

          with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal host/boston.example.com with kvno 3, encryption type Triple DES cbc

          mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal host/boston.example.com with kvno 3, encryption type ARCFOUR

          with HMAC/md5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal host/boston.example.com with kvno 3, encryption type DES cbc mode

          with RSA-MD5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

kadmin:
```

c. 退出 kadmin。

```
kadmin: quit
```

配置 Kerberos NFS 服务器

NFS 服务可以使用 UNIX 用户 ID (user ID, UID) 标识用户，但不能直接使用 GSS 凭证。要将凭证转换为 UID，可能需要创建将用户凭证映射到 UNIX UID 的凭证表。有关缺省凭证映射的更多信息，请参见第 360 页中的 “将 GSS 凭证映射到 UNIX 凭证” 。本节中的过程重点介绍配置 Kerberos NFS 服务器、管理凭证表以及对已挂载 NFS 的文件系统启动 Kerberos 安全模式所需的任务。以下任务列表说明了本节中所包含的任务。

表 22–1 配置 Kerberos NFS 服务器（任务列表）

任务	说明	参考
配置 Kerberos NFS 服务器。	使服务器共享要求 Kerberos 验证的文件系统。	第 382 页中的 “如何配置 Kerberos NFS 服务器”
创建凭证表。	在缺省映射不满足要求的情况下，生成可用于提供从 GSS 凭证到 UNIX 用户 ID 的映射的凭证表。	第 384 页中的 “如何创建凭证表”
更改将用户凭证映射到 UNIX UID 的凭证表。	更新凭证表中的信息。	第 384 页中的 “如何向凭证表中添加单个项”
在两个类似领域之间创建凭证映射。	在多个领域共享同一个口令文件的情况下，提供有关如何将 UID 从一个领域映射到另一个领域的说明。	第 385 页中的 “如何提供各领域之间的凭证映射”
使用 Kerberos 验证共享文件系统。	使用安全模式共享文件系统，以便要求 Kerberos 验证。	第 386 页中的 “如何使用多种 Kerberos 安全模式设置安全的 NFS 环境”

▼ 如何配置 Kerberos NFS 服务器

在此过程中，将使用以下配置参数：

- 领域名称 = `EXAMPLE.COM`
- DNS 域名 = `example.com`
- NFS 服务器 = `denver.example.com`
- `admin` 主体 = `kws/admin`

1 完成配置 Kerberos NFS 服务器的先决条件。

必须配置主 KDC。要完全测试此过程，需要多个客户机。

2 （可选的）安装 NTP 客户机或其他时钟同步机制。

安装和使用网络时间协议 (Network Time Protocol, NTP) 并非必需。但是，要成功验证，每个时钟必须处于 `krb5.conf` 文件的 `libdefaults` 部分中定义的缺省时间内。有关 NTP 的信息，请参见第 404 页中的 “同步 KDC 和 Kerberos 客户机的时钟” 。

3 **启动** kadmin。

可以使用 SEAM Administration Tool 添加主体，如第 456 页中的 “如何创建新的 Kerberos 主体”中所述。为此，必须使用在配置主 KDC 时创建的一个 admin 主体名称登录。不过，以下示例说明如何使用命令行添加所需的主体。

```
denver # /usr/sbin/kadmin -p kws/admin

Enter password:      <Type kws/admin password>

kadmin:
```

a. **创建服务器的 NFS 服务主体。**

请注意，主体实例为主机名时，无论 /etc/resolv.conf 文件中的域名是大写还是小写，都必须以小写字母指定 FQDN。

对系统上可能用于访问 NFS 数据的每个唯一接口重复此步骤。如果主机有多个接口具有唯一名称，则每个唯一名称必须具有自己的 NFS 服务主体。

```
kadmin: addprinc -randkey nfs/denver.example.com

Principal "nfs/denver.example.com" created.

kadmin:
```

b. **将服务器的 NFS 服务主体添加到服务器的密钥表文件中。**

对步骤 a 中创建的每个唯一服务主体重复此步骤。

```
kadmin: ktadd nfs/denver.example.com

Entry for principal nfs/denver.example.com with kvno 3, encryption type AES-128 CTS mode

          with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal nfs/denver.example.com with kvno 3, encryption type Triple DES cbc

          mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal nfs denver.example.com with kvno 3, encryption type ARCFOUR

          with HMAC/md5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal nfs/denver.example.com with kvno 3, encryption type DES cbc mode

          with RSA-MD5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

kadmin:
```

c. **退出** kadmin。

```
kadmin: quit
```

4 **（可选的）如果需要，可创建特殊 GSS 凭证映射。**

通常，Kerberos 服务在 GSS 凭证和 UNIX UID 之间生成相应的映射。缺省映射在第 360 页中的 “将 GSS 凭证映射到 UNIX 凭证”中介绍。如果缺省映射不满足要求，请参见第 384 页中的 “如何创建凭证表”以获取更多信息。

5 **使用 Kerberos 安全模式共享 NFS 文件系统。**

有关更多信息，请参见第 386 页中的 “如何使用多种 Kerberos 安全模式设置安全的 NFS 环境”。

▼ 如何创建凭证表

NFS 服务器使用 `gsscred` 凭证表将 Kerberos 凭证映射到 UID。对于从使用 Kerberos 验证的 NFS 服务器挂载文件系统的 NFS 客户机，如果缺省映射不满足要求，则必须创建此表。

1 **编辑 `/etc/gss/gsscred.conf` 并更改安全机制。**

将机制更改为 `files`。

2 **使用 `gsscred` 命令创建凭证表。**

```
# gsscred -m kerberos_v5 -a
```

`gsscred` 命令从 `/etc/nsswitch.conf` 文件的 `passwd` 项列出的所有源中收集信息。如果希望凭证表中不包括本地口令项，则可能需要临时删除 `files` 项。有关更多信息，请参见 `gsscred`(1M) 手册页。

▼ 如何向凭证表中添加单个项

开始之前 此过程要求已在 NFS 服务器上创建 `gsscred` 表。有关说明，请参见第 384 页中的 “如何创建凭证表”。

1 **成为 NFS 服务器上的超级用户。**

2 **使用 `gsscred` 命令向凭证表中添加项。**

```
# gsscred -m mech [ -n name [ -u uid ]] -a
```

mech 定义要使用的安全机制。

name 定义用户的主体名称，如 KDC 中所定义。

uid 定义用户的 UID，如口令数据库中所定义。

`-a` 向主体名称映射中添加 UID。

示例 22–1 向凭证表中添加多组成部分主体

在以下示例中，将添加名为 sandy/admin 的主体的项，该主体映射到 UID 3736。

```
# gsscred -m kerberos_v5 -n sandy/admin -u 3736 -a
```

示例 22–2 向凭证表中添加其他域中的主体

在以下示例中，将添加名为 sandy/admin@EXAMPLE.COM 的主体的项，该主体映射到 UID 3736。

```
# gsscred -m kerberos_v5 -n sandy/admin@EXAMPLE.COM -u 3736 -a
```

▼ 如何提供各领域之间的凭证映射

此过程在使用相同口令文件的领域之间提供相应的凭证映射。在此示例中，领域 CORP.EXAMPLE.COM 和 SALES.EXAMPLE.COM 使用相同的口令文件。bob@CORP.EXAMPLE.COM 和 bob@SALES.EXAMPLE.COM 的凭证映射到相同的 UID。

1. **成为超级用户。**

2. **在客户机系统上，向 krb5.conf 文件中添加项。**

```
# cat /etc/krb5/krb5.conf

[libdefaults]

        default_realm = CORP.EXAMPLE.COM

 .

[realms]

    CORP.EXAMPLE.COM = {

        .

        auth_to_local_realm = SALES.EXAMPLE.COM

        .

    }
```

故障排除 有关对凭证映射问题进行疑难解答的过程的帮助，请参见第 443 页中的 “观察从 GSS 凭证到 UNIX 凭证的映射”。

▼ 如何使用多种 Kerberos 安全模式设置安全的 NFS 环境

通过此过程，NFS 服务器可以使用不同的安全模式或特性提供安全的 NFS 访问。客户机与 NFS 服务器协商安全特性时，将使用该客户机有权访问的服务器所提供的第一种特性。此特性用于 NFS 服务器共享的文件系统的所有后续客户机请求。

1 **成为 NFS 服务器上的超级用户。**

2 **验证在密钥表文件中是否存在 NFS 服务主体。**

`klist` 命令报告是否存在密钥表文件并显示主体。如果结果显示不存在密钥表文件或者不存在 NFS 服务主体，则需要验证是否已完成第 382 页中的 “如何配置 Kerberos NFS 服务器”中的所有步骤。

```
# klist -k

Keytab name: FILE:/etc/krb5/krb5.keytab

KVNO Principal

---- -------------------------------------------------------

   3 nfs/denver.example.com@EXAMPLE.COM

   3 nfs/denver.example.com@EXAMPLE.COM

   3 nfs/denver.example.com@EXAMPLE.COM

   3 nfs/denver.example.com@EXAMPLE.COM
```

3 **在 `/etc/nfssec.conf` 文件中启用 Kerberos 安全模式。**

编辑 `/etc/nfssec.conf` 文件并删除位于 Kerberos 安全模式前面的 "#"。

```
# cat /etc/nfssec.conf

 .

 .

#

# Uncomment the following lines to use Kerberos V5 with NFS

#

krb5            390003  kerberos_v5     default -               # RPCSEC_GSS
```

```
krb5i           390004  kerberos_v5     default integrity       # RPCSEC_GSS

krb5p           390005  kerberos_v5     default privacy         # RPCSEC_GSS
```

4 **编辑 /etc/dfs/dfstab 文件，并将带有所需安全模式的 sec= 选项添加到相应的项中。**

share -F nfs -o sec=*mode file_system*

mode 指定共享文件系统时要使用的安全模式。使用多种安全模式时，会将列表中的第一种模式用作缺省模式。

file_system 定义要共享的文件系统的路径。

尝试从指定的文件系统访问文件的所有客户机都要求 Kerberos 验证。要访问文件，应验证 NFS 客户机上的用户主体。

5 **请确保服务器上正在运行 NFS 服务。**

如果此命令是您所启动的第一个 share 命令或 share 命令集，则 NFS 守护进程可能未运行。以下命令将重新启动该守护进程：

```
# svcadm restart network/nfs/server
```

6 **（可选的）如果使用的是自动挂载程序，请编辑 auto_master 数据库以选择非缺省安全模式。**

如果不使用自动挂载程序访问文件系统或者安全模式的缺省选择可接受，则无需执行此过程。

file_system auto_home -nosuid,sec=*mode*

7 **（可选的）使用非缺省模式手动发布用于访问文件系统的 mount 命令。**

或者，可以使用 mount 命令指定安全模式，但此替代方法不会利用自动挂载程序。

 # mount -F nfs -o sec=*mode file_system*

示例 22–3 使用一种 Kerberos 安全模式共享文件系统

在此示例中，dfstab 文件行表明：在通过 NFS 服务访问任何文件之前，必须先成功完成 Kerberos 验证。

```
# grep krb /etc/dfs/dfstab

share -F nfs -o sec=krb5 /export/home
```

示例 22–4 使用多种 Kerberos 安全模式共享文件系统

在此示例中，选择了所有三种 Kerberos 安全模式。如果发出挂载请求时未指定任何安全模式，则将在所有 NFS V3 客户机中使用列出的第一种模式（在此例中为 krb5）。有关更多信息，请参见 nfssec(5) 手册页。

```
# grep krb /etc/dfs/dfstab

share -F nfs -o sec=krb5:krb5i:krb5p /export/home
```

配置 Kerberos 客户机

Kerberos 客户机是网络上需要使用 Kerberos 服务的任何主机（不是 KDC 服务器）。本节介绍有关安装 Kerberos 客户机的过程以及使用 root 验证以挂载 NFS 文件系统的特定信息。

配置 Kerberos 客户机（任务列表）

以下任务列表包括有关设置 Kerberos 客户机的所有过程。每行都包括任务说明（说明执行该项任务的原因）以及指向该任务的链接。

任务	说明	参考
建立 Kerberos 客户机安装配置文件。	生成可用于自动安装 Kerberos 客户机的客户机安装配置文件。	第 388 页中的 “如何创建 Kerberos 客户机安装配置文件”
配置 Kerberos 客户机。	手动安装 Kerberos 客户机。如果每台客户机安装要求唯一的安装参数，请使用此过程。 自动安装 Kerberos 客户机。如果每台客户机的安装参数都相同，请使用此过程。 交互式安装 Kerberos 客户机。如果仅需要更改一些安装参数，请使用此过程。	第 393 页中的 “如何手动配置 Kerberos 客户机” 第 389 页中的 “如何自动配置 Kerberos 客户机” 第 391 页中的 “如何交互配置 Kerberos 客户机”
允许客户机以 root 用户身份访问 NFS 文件系统。	在客户机上创建 root 主体，以便客户机可以挂载使用 root 访问权限共享的 NFS 文件系统。此外，允许为客户机设置对 NFS 文件系统的非交互 root 访问权限，以便可以运行 cron 作业。	第 399 页中的 “如何以 root 用户身份访问受 Kerberos 保护的 NFS 文件系统”

▼ 如何创建 Kerberos 客户机安装配置文件

此过程创建可在安装 Kerberos 客户机时使用的 kclient 配置文件。使用 kclient 配置文件，可降低出现键入错误的可能性。此外，与交互式过程相比，使用该配置文件可以减少用户干预。

1 成为超级用户。

2 **创建 kclient 安装配置文件。**

kclient 配置文件样例与以下内容类似：

```
client# cat /net/kdc1.example.com/export/install/profile

REALM EXAMPLE.COM

KDC kdc1.example.com

ADMIN clntconfig

FILEPATH /net/kdc1.example.com/export/install/krb5.conf

NFS 1

DNSLOOKUP none
```

▼ 如何自动配置 Kerberos 客户机

开始之前 此过程使用安装配置文件。请参见第 388 页中的 “如何创建 Kerberos 客户机安装配置文件”。

1 **成为超级用户。**

2 **运行 `kclient` 安装脚本。**

要完成此过程，需要提供 `clntconfig` 主体的口令。

```
client# /usr/sbin/kclient -p /net/kdc1.example.com/export/install/krb5.conf


Starting client setup

-------------------------------------------------


kdc1.example.com


Setting up /etc/krb5/krb5.conf.


Obtaining TGT for clntconfig/admin ...
```

```
Password for clntconfig/admin@EXAMPLE.COM:      <Type the password>


nfs/client.example.com entry ADDED to KDC database.

nfs/client.example.com entry ADDED to keytab.


host/client.example.com entry ADDED to KDC database.

host/client.example.com entry ADDED to keytab.


Copied /net/kdc1.example.com/export/clientinstall/krb5.conf.


-------------------------------------------------

Setup COMPLETE.


client#
```

示例 22–5 使用命令行覆盖项自动配置 Kerberos 客户机

以下示例将覆盖在安装配置文件中设置的 DNSARG 和 KDC 参数。

```
# /usr/sbin/kclient -p /net/kdc1.example.com/export/install/krb5.conf\
-d dns_fallback -k kdc2.example.com


Starting client setup

-------------------------------------------------


kdc1.example.com
```

```
Setting up /etc/krb5/krb5.conf.


Obtaining TGT for clntconfig/admin ...

Password for clntconfig/admin@EXAMPLE.COM:      <Type the password>


nfs/client.example.com entry ADDED to KDC database.

nfs/client.example.com entry ADDED to keytab.


host/client.example.com entry ADDED to KDC database.

host/client.example.com entry ADDED to keytab.


Copied /net/kdc1.example.com/export/install/krb5.conf.


-------------------------------------------------

Setup COMPLETE.


client#
```

▼ 如何交互配置 Kerberos 客户机

此过程使用 `kclient` 安装实用程序而不是使用安装配置文件。

1. **成为超级用户。**
2. **运行 `kclient` 安装脚本。**
   需要提供以下信息：
   - Kerberos 领域名称
   - 主 KDC 主机名
   - 管理主体名称
   - 管理主体的口令

示例 22–6 运行 kclient 安装实用程序

以下输出给出了运行 kclient 命令的结果。

```
client# /usr/sbin/kclient


Starting client setup

------------------------------------------------


Do you want to use DNS for kerberos lookups ? [y/n]: n

        No action performed.
Enter the Kerberos realm: EXAMPLE.COM
Specify the KDC hostname for the above realm: kdc1.example.com


Setting up /etc/krb5/krb5.conf.


Enter the krb5 administrative principal to be used: clntconfig/admin
Obtaining TGT for clntconfig/admin ...
Password for clntconfig/admin@EXAMPLE.COM:      <Type the password>
Do you plan on doing Kerberized nfs ? [y/n]: n


host/client.example.com entry ADDED to KDC database.
host/client.example.com entry ADDED to keytab.


Do you want to copy over the master krb5.conf file ? [y/n]: y
Enter the pathname of the file to be copied: \
```

```
/net/kdc1.example.com/export/install/krb5.conf


Copied /net/kdc1.example.com/export/install/krb5.conf.


-----------------------------------------------

Setup COMPLETE !

#
```

▼ 如何手动配置 Kerberos 客户机

在此过程中，将使用以下配置参数：

- 领域名称 = `EXAMPLE.COM`
- DNS 域名 = `example.com`
- 主 KDC = `kdc1.example.com`
- 从 KDC = `kdc2.example.com`
- 客户机 = `client.example.com`
- `admin` 主体 = `kws/admin`
- 用户主体 = `mre`
- 联机帮助 URL = `http://denver:8888/ab2/coll.384.1/SEAM/@AB2PageView/6956`

注 – 调整该 URL 以指向“SEAM Administration Tool”部分，如第 362 页中的 “SEAM Administration Tool 中的联机帮助 URL”中所述。

---

1. **成为超级用户。**

2. **编辑 Kerberos 配置文件 (`krb5.conf`)。**

   要从 Kerberos 缺省版本更改该文件，需要更改领域名称和服务器名称。您还需要标识 `gkadmin` 帮助文件的路径。

```
kdc1 # cat /etc/krb5/krb5.conf

[libdefaults]

        default_realm = EXAMPLE.COM
```

```
[realms]
                EXAMPLE.COM = {
                kdc = kdc1.example.com
                kdc = kdc2.example.com
                admin_server = kdc1.example.com
        }

[domain_realm]
        .example.com = EXAMPLE.COM
#
# if the domain name and realm name are equivalent,
# this entry is not needed
#
[logging]
        default = FILE:/var/krb5/kdc.log
        kdc = FILE:/var/krb5/kdc.log

[appdefaults]
    gkadmin = {
        help_url = http://denver:8888/ab2/coll.384.1/SEAM/@AB2PageView/6956
```

注 – 如果要限制加密类型，可以设置 `default_tkt_enctypes` 或 `default_tgs_enctypes` 行。有关限制加密类型涉及的问题的说明，请参阅第 512 页中的 “使用 Kerberos 加密类型”。

3 （可选的）更改用于定位 KDC 的过程。

缺省情况下，使用主机和域名到 kerberos 领域的映射定位 KDC。可以通过将 `dns_lookup_kdc`、`dns_lookup_realm` 或 `dns_fallback` 添加到 `krb5.conf` 文件的 `libdefaults` 部分来更改此行为。有关更多信息，请参见 `krb5.conf`(4) 手册页。

4 （可选的）使用 NTP 或其他时钟同步机制将客户机时钟与主 KDC 时钟同步。

安装和使用网络时间协议 (Network Time Protocol, NTP) 并非必需。但是，要成功验证，每个时钟必须处于 `krb5.conf` 文件的 `libdefaults` 部分中定义的缺省时间内。有关 NTP 的信息，请参见第 404 页中的 “同步 KDC 和 Kerberos 客户机的时钟”。

5 启动 `kadmin`。

可以使用 SEAM Administration Tool 添加主体，如第 456 页中的 “如何创建新的 Kerberos 主体”中所述。为此，必须使用在配置主 KDC 时创建的一个 `admin` 主体名称登录。不过，以下示例说明如何使用命令行添加所需的主体。

```
denver # /usr/sbin/kadmin -p kws/admin

Enter password:      <Type kws/admin password>

kadmin:
```

a. （可选的）如果不存在用户主体，请创建用户主体。

仅当尚未对与此主机关联的用户指定主体时，才需要创建用户主体。

```
kadmin: addprinc mre

Enter password for principal mre@EXAMPLE.COM:      <Type the password>

Re-enter password for principal mre@EXAMPLE.COM:      <Type it again>

kadmin:
```

b. （可选的）创建 `root` 主体。

如果客户机不要求对使用 NFS 服务挂载的远程文件系统拥有 `root` 访问权限，则可以跳过此步骤。为了避免创建领域范围的 `root` 主体，`root` 主体应是由两个部分组成的主体（第二个组成部分为 Kerberos 客户机系统的主机名）。请注意，主体实例为主机名时，无论 `/etc/resolv.conf` 文件中的域名是大写还是小写，都必须以小写字母指定 FQDN。

```
kadmin: addprinc -randkey root/client.example.com

Principal "root/client.example.com" created.

kadmin:
```

c. 创建 `host` 主体。

`host` 主体用于验证应用程序。

```
kadmin: addprinc -randkey host/denver.example.com
```

```
Principal "host/denver.example.com@EXAMPLE.COM" created.

kadmin:
```

d. （可选的）将服务器的 NFS 服务主体添加到服务器的密钥表文件中。

仅当客户机需要使用 Kerberos 验证访问 NFS 文件系统时，才需要执行此步骤。

```
kadmin: ktadd nfs/denver.example.com

Entry for principal nfs/denver.example.com with kvno 3, encryption type AES-128 CTS mode

          with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal nfs/denver.example.com with kvno 3, encryption type Triple DES cbc

          mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal nfs/denver.example.com with kvno 3, encryption type ARCFOUR

          with HMAC/md5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal nfs/denver.example.com with kvno 3, encryption type DES cbc mode

          with RSA-MD5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

kadmin:
```

e. （可选的）将 root 主体添加到服务器的密钥表文件中。

如果添加了 root 主体，则必须执行此步骤，以便客户机对使用 NFS 服务挂载的文件系统拥有 root 访问权限。如果需要非交互 root 访问权限（例如，以 root 身份运行 cron 作业），也必须执行此步骤。

```
kadmin: ktadd root/client.example.com

Entry for principal root/client.example.com with kvno 3, encryption type AES-128 CTS mode

          with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal root/client.example.com with kvno 3, encryption type Triple DES cbc

          mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal root/client.example.com with kvno 3, encryption type ARCFOUR

          with HMAC/md5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal root/client.example.com with kvno 3, encryption type DES cbc mode

          with RSA-MD5 added to keytab WRFILE:/etc/krb5/krb5.keytab.
```

```
kadmin:
```

f. 将 host 主体添加到服务器的密钥表文件中。

```
kadmin: ktadd host/denver.example.com

Entry for principal host/denver.example.com with kvno 3, encryption type AES-128 CTS mode

          with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal host/denver.example.com with kvno 3, encryption type Triple DES cbc

          mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal host/denver.example.com with kvno 3, encryption type ARCFOUR

          with HMAC/md5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal host/denver.example.com with kvno 3, encryption type DES cbc mode

          with RSA-MD5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

kadmin:
```

g. 退出 kadmin。

```
kadmin: quit
```

6 （可选的）要在 NFS 上使用 Kerberos，请在 /etc/nfssec.conf 文件中启用 Kerberos 安全模式。

编辑 /etc/nfssec.conf 文件并删除位于 Kerberos 安全模式前面的 "#"。

```
# cat /etc/nfssec.conf

 .

 .

#

# Uncomment the following lines to use Kerberos V5 with NFS

#

krb5            390003  kerberos_v5     default -               # RPCSEC_GSS

krb5i           390004  kerberos_v5     default integrity       # RPCSEC_GSS

krb5p           390005  kerberos_v5     default privacy         # RPCSEC_GSS
```

7 **如果希望客户机自动更新 TGT 或者向用户发出有关 Kerberos 票证失效的警告，请在 `/etc/krb5/warn.conf` 文件中创建相应的项。**

有关更多信息，请参见 `warn.conf`(4) 手册页。

示例 22–7 使用非 Kerberos KDC 设置 Kerberos 客户机

可以设置 Kerberos 客户机，使其与非 Kerberos KDC 协同工作。在此情况下，必须在 `realms` 部分的 `/etc/krb5/krb5.conf` 文件中包括一行。该行更改客户机与 Kerberos 口令更改服务器通信时要使用的协议。该行的格式如下：

```
[realms]

                EXAMPLE.COM = {

                kdc = kdc1.example.com

                kdc = kdc2.example.com

                admin_server = kdc1.example.com

                kpasswd_protocol = SET_CHANGE

        }
```

示例 22–8 主机和域名到 Kerberos 领域的映射的 DNS TXT 记录

```
@ IN SOA kdc1.example.com root.kdc1.example.com (

                                1989020501   ;serial

                                10800        ;refresh

                                3600         ;retry

                                3600000      ;expire

                                86400 )      ;minimum


                        IN      NS      kdc1.example.com.

kdc1                    IN      A       192.146.86.20

kdc2                    IN      A       192.146.86.21
```

```
_kerberos.example.com.             IN      TXT     "EXAMPLE.COM"

_kerberos.kdc1.example.com.        IN      TXT     "EXAMPLE.COM"

_kerberos.kdc2.example.com.        IN      TXT     "EXAMPLE.COM"
```

示例 22–9 Kerberos 服务器位置的 DNS SRV 记录

此示例定义主 KDC、admin 服务器和 kpasswd 服务器的位置记录。

```
@ IN SOA kdc1.example.com root.kdc1.example.com (

                                1989020501   ;serial

                                10800        ;refresh

                                3600         ;retry

                                3600000      ;expire

                                86400 )      ;minimum


                          IN      NS      kdc1.example.com.

kdc1                      IN      A       192.146.86.20

kdc2                      IN      A       192.146.86.21


_kerberos._udp.EXAMPLE.COM         IN      SRV 0 0 88  kdc1.example.com

_kerberos-adm._udp.EXAMPLE.COM     IN      SRV 0 0 749 kdc1.example.com

_kpasswd._udp.EXAMPLE.COM          IN      SRV 0 0 749 kdc1.example.com
```

▼ 如何以 root 用户身份访问受 Kerberos 保护的 NFS 文件系统

通过此过程，客户机可以使用 root ID 权限访问要求 Kerberos 验证的 NFS 文件系统。特别是，可以访问使用以下选项共享的 NFS 文件系统：-o sec=krb5,root=client1.sun.com。

1 成为超级用户。

2 **启动** `kadmin`。

可以使用 SEAM Administration Tool 添加主体，如第 456 页中的 “如何创建新的 Kerberos 主体”中所述。为此，必须使用在配置主 KDC 时创建的一个 `admin` 主体名称登录。不过，以下示例说明如何使用命令行添加所需的主体。

```
denver # /usr/sbin/kadmin -p kws/admin

Enter password:      <Type kws/admin password>

kadmin:
```

a. **为 NFS 客户机创建** `root` **主体**。

此主体用于对要求 Kerberos 验证的已挂载 NFS 的文件系统提供 root 等效访问权限。为了避免创建领域范围的 root 主体，`root` 主体应是由两个部分组成的主体（第二个组成部分为 Kerberos 客户机系统的主机名）。请注意，主体实例为主机名时，无论 `/etc/resolv.conf` 文件中的域名是大写还是小写，都必须以小写字母指定 FQDN。

```
kadmin: addprinc -randkey root/client.example.com

Principal "root/client.example.com" created.

kadmin:
```

b. **将** `root` **主体添加到服务器的密钥表文件中**。

如果添加了 `root` 主体，则必须执行此步骤，以便客户机对使用 NFS 服务挂载的文件系统拥有 `root` 访问权限。如果需要非交互 `root` 访问权限（例如，以 `root` 身份运行 cron 作业），也必须执行此步骤。

```
kadmin: ktadd root/client.example.com

Entry for principal root/client.example.com with kvno 3, encryption type AES-128 CTS mode

          with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal root/client.example.com with kvno 3, encryption type Triple DES cbc

          mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal root/client.example.com with kvno 3, encryption type ARCFOUR

          with HMAC/md5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal root/client.example.com with kvno 3, encryption type DES cbc mode

          with RSA-MD5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

kadmin:
```

c. **退出** kadmin。

```
kadmin: quit
```

▼ 在 Kerberos 领域中配置用户自动迁移

没有 Kerberos 主体的用户可以自动迁移到现有 Kerberos 领域。通过将 pam_krb5_migrate 模块堆叠在 /etc/pam.conf 的服务验证栈中，可以通过正在使用的服务的 PAM 框架实现迁移。

在此示例中，将配置 rlogin 和 other PAM 服务名称以使用自动迁移。将使用以下配置参数：

- 领域名称 = EXAMPLE.COM
- 主 KDC = kdc1.example.com
- 承载迁移服务的计算机 = server1.example.com
- 迁移服务主体 = host/server1.example.com

开始之前 将 server1 设置为 EXAMPLE.COM 领域的 Kerberos 客户机。有关更多信息，请参见第 388 页中的 “配置 Kerberos 客户机”。

1 **检查是否存在 server1 的主机服务主体。**

server1 的 keytab 文件中的主机服务主体用于向主 KDC 验证该服务器。

```
server1 # klist -k

Keytab name: FILE:/etc/krb5/krb5.keytab

    KVNO Principal

    ---- -------------------------------------------------

       3 host/server1.example.com@EXAMPLE.COM

       3 host/server1.example.com@EXAMPLE.COM

       3 host/server1.example.com@EXAMPLE.COM

       3 host/server1.example.com@EXAMPLE.COM
```

2 **对 PAM 配置文件进行更改。**

将 pam_krb5_migrate PAM 模块添加到 rlogin 和 other 服务名称的验证栈中。系统将自动为使用 rlogin、telnet 或 ssh 而不具有 Kerberos 主体的用户创建主体。

```
# cat /etc/pam.conf

 .
```

```
 .
#
# rlogin service (explicit because of pam_rhost_auth)
#
rlogin  auth sufficient         pam_rhosts_auth.so.1
rlogin  auth requisite          pam_authtok_get.so.1
rlogin  auth required           pam_dhkeys.so.1
rlogin  auth required           pam_unix_cred.so.1
rlogin  auth required           pam_unix_auth.so.1
rlogin  auth sufficient         pam_krb5.so.1
rlogin  auth optional           pam_krb5_migrate.so.1
#
 .
 .
#
# Default definitions for Authentication management
# Used when service name is not explicitly mentioned for authentication
#
other   auth requisite          pam_authtok_get.so.1
other   auth required           pam_dhkeys.so.1
other   auth required           pam_unix_cred.so.1
other   auth required           pam_unix_auth.so.1
other   auth sufficient         pam_krb5.so.1
other   auth optional           pam_krb5_migrate.so.1
```

3 （可选的）如果需要，可强制立即更改口令。

可以将新建 Kerberos 帐户的口令失效时间设置为当前时间（现在），以便强制立即更改 Kerberos 口令。要将失效时间设置为当前时间，请将 expire_pw 选项添加到使用 pam_krb5_migrate 模块的行中。有关更多信息，请参见 pam_krb5_migrate(5) 手册页。

```
# cat /etc/pam.conf

 .

 .

rlogin  auth optional           pam_krb5_migrate.so.1 expire_pw

#

 .

 .

other   auth optional           pam_krb5_migrate.so.1 expire_pw
```

4 在主 KDC 上，更新访问控制文件。

以下项将为所有用户（root 用户除外）授予对 host/server1.example.com 服务主体的迁移和查询权限。务必注意，不应使用 U 权限迁移 kadm5.acl 文件中列出的用户。这些项必须位于允许所有用户或 ui 项之前。有关更多信息，请参见 kadm5.acl(4) 手册页。

```
kdc1 # cat /etc/krb5/kadm5.acl

host/server1.example.com@EXAMPLE.COM U root

host/server1.example.com@EXAMPLE.COM ui *

*/admin@EXAMPLE.COM *
```

5 在主 KDC 上，重新启动 Kerberos 管理守护进程。

通过此步骤，kadmind 守护进程可以使用新的 kadm5.acl 项。

```
kdc1 # svcadm restart network/security/kadmin
```

6 在主 KDC 上，向 pam.conf 文件中添加项。

通过以下项，kadmind 守护进程可以使用 k5migrate PAM 服务来验证需要迁移的帐户的 UNIX 用户口令。

```
# grep k5migrate /etc/pam.conf

k5migrate       auth    required        pam_unix_auth.so.1

k5migrate       account required        pam_unix_account.so.1
```

同步 KDC 和 Kerberos 客户机的时钟

所有参与 Kerberos 验证系统的主机都必须在指定的最长时间（称为**时钟相位差**）内同步其内部时钟。针对这一要求，需要进行另一种 Kerberos 安全检查。如果任意两台参与主机之间的时间偏差超过了时钟相位差，则客户机请求会被拒绝。

时钟相位差还确定应用程序服务器必须跟踪所有 Kerberos 协议消息的时间长度，以便识别和拒绝重放的请求。因此，时钟相位差的值越大，应用程序服务器必须收集的信息就越多。

时钟相位差的最大缺省值为 300 秒（5 分钟）。可以在 `krb5.conf` 文件的 `libdefaults` 部分中更改此缺省值。

注 – 出于安全原因，不要将时钟相位差增大到超过 300 秒。

由于维护 KDC 和 Kerberos 客户机之间的同步时钟非常重要，因此应使用网络时间协议 (Network Time Protocol, NTP) 软件同步这些时钟。从 Solaris 2.6 发行版开始，Solaris 软件中提供了由美国特拉华大学开发的 NTP 公用软件。

注 – 同步时钟的另一种方法是使用 `rdate` 命令和 `cron` 作业（一种比使用 NTP 参与性更小的过程）。但是，本节重点介绍如何使用 NTP。并且，如果使用网络来同步时钟，时钟同步协议本身必须是安全的。

通过 NTP，可以在网络环境中管理准确时间或网络时钟同步，或者同时管理这两者。本质上，NTP 是一种服务器/客户机实现。可以选择一个系统（NTP 服务器）作为主时钟。然后，设置所有其他系统（NTP 客户机），使这些系统的时钟与主时钟同步。

为了同步时钟，NTP 使用 `xntpd` 守护进程，该守护进程设置并维护 UNIX 系统时间，使其与 Internet 标准时间服务器的时间保持一致。以下给出了此服务器/客户机 NTP 实现的示例。

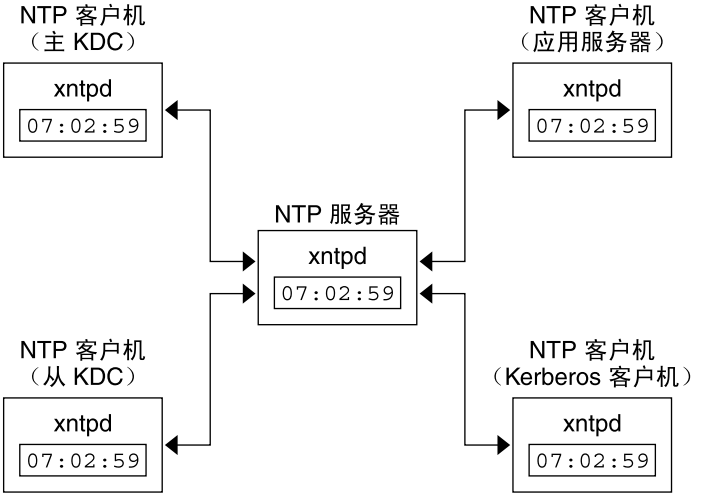


图 22–1 使用 NTP 同步时钟

确保 KDC 和 Kerberos 客户机保持时钟同步涉及以下步骤的实现：

1. 在网络上设置 NTP 服务器。此服务器可以是除主 KDC 之外的任何系统。要了解 NTP 服务器任务，请参见《System Administration Guide: Network Services》中的“Managing Network Time Protocol (Tasks)”。
2. 在网络上配置 KDC 和 Kerberos 客户机时，将它们设置为 NTP 服务器的 NTP 客户机。要了解 NTP 客户机任务，请参见《System Administration Guide: Network Services》中的“Managing Network Time Protocol (Tasks)”。

交换主 KDC 和从 KDC

使用本节中的过程可以更容易地将主 KDC 与从 KDC 进行交换。仅当主 KDC 服务器由于某种原因出现故障时，或者需要重新安装主 KDC（例如，由于安装了新硬件）时，才应将主 KDC 与从 KDC 进行交换。

▼ 如何配置可交换的从 KDC

在希望其可以成为主 KDC 的从 KDC 服务器上执行此过程。此过程假定将使用增量传播。

1 在安装 KDC 过程中使用主 KDC 和可交换从 KDC 的别名。

定义 KDC 的主机名时，请确保 DNS 中包括每个系统的别名。此外，在 `/etc/krb5/krb5.conf` 文件中定义主机时也应使用别名。

2 **逐步完成从 KDC 安装。**

在进行任何交换之前，在该领域中此服务器的作用应与任何其他从 KDC 相同。有关说明，请参见第 372 页中的 “如何配置从 KDC” 。

3 **移动主 KDC 命令**

要禁止从该从 KDC 运行主 KDC 命令，请将 `kprop`、`kadmind` 和 `kadmin.local` 命令移到一个保留位置。

```
kdc4 # mv /usr/lib/krb5/kprop /usr/lib/krb5/kprop.save

kdc4 # mv /usr/lib/krb5/kadmind /usr/lib/krb5/kadmind.save

kdc4 # mv /usr/sbin/kadmin.local /usr/sbin/kadmin.local.save
```

▼ 如何交换主 KDC 和从 KDC

在此过程中，要交换出的主 KDC 服务器名为 `kdc1`。将成为新的主 KDC 的从 KDC 名为 `kdc4`。此过程假定将使用增量传播。

开始之前 此过程要求已将该从 KDC 服务器设置为可交换的从 KDC。有关更多信息，请参见第 405 页中的 “如何配置可交换的从 KDC” 。

1 **在新的主 KDC 上，启动 `kadmin`。**

```
kdc4 # /usr/sbin/kadmin -p kws/admin

Enter password:      <Type kws/admin password>

kadmin:
```

a. **为 `kadmind` 服务创建新的主体。**

以下示例中的第一个 `addprinc` 命令显示为两行，但实际上该命令应在同一行中键入。

```
kadmin: addprinc -randkey -allow_tgs_req +password_changing_service -clearpolicy \
        changepw/kdc4.example.com

Principal "changepw/kdc4.example.com@ENG.SUN.COM" created.

kadmin: addprinc -randkey -allow_tgs_req -clearpolicy kadmin/kdc4.example.com

Principal "kadmin/kdc4.example.com@EXAMPLE.COM" created.

kadmin:
```

b. 创建密钥表文件。

```
kadmin: ktadd -k /etc/krb5/kadm5.keytab kadmin/kdc4.example.com

Entry for principal kadmin/kdc4.example.com with kvno 3, encryption type AES-128 CTS mode

          with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal kadmin/kdc4.example.com with kvno 3, encryption type Triple DES cbc

          mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal kadmin/kdc4.example.com with kvno 3, encryption type ARCFOUR

          with HMAC/md5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal kadmin/kdc4.example.com with kvno 3, encryption type DES cbc mode

          with RSA-MD5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

kadmin: ktadd -k /etc/krb5/kadm5.keytab changepw/kdc4.example.com

Entry for principal changepw/kdc4.example.com with kvno 3, encryption type AES-128 CTS mode

          with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal changepw/kdc4.example.com with kvno 3, encryption type Triple DES cbc

          mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal changepw/kdc4.example.com with kvno 3, encryption type ARCFOUR

          with HMAC/md5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal changepw/kdc4.example.com with kvno 3, encryption type DES cbc mode

          with RSA-MD5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

kadmin:
```

c. 退出 kadmin。

```
kadmin: quit
```

2 在新的主 KDC 上，强制执行同步。

以下步骤将在从服务器上强制执行完全 KDC 更新。

```
kdc4 # svcadm disable network/security/krb5kdc

kdc4 # rm /var/krb5/principal.ulog
```

```
kdc4 # svcadm enable network/security/krb5kdc
```

3 在新的主 KDC 上，清除更新日志。

以下步骤将重新初始化新的主 KDC 服务器的更新日志。

```
kdc4 # svcadm disable network/security/krb5kdc

kdc4 # rm /var/krb5/principal.ulog
```

4 在旧的主 KDC 上，中止 kadmind 和 krb5kdc 进程。

中止 kadmind 进程后，可防止对 KDC 数据库进行任何更改。

```
kdc1 # svcadm disable network/security/kadmin

kdc1 # svcadm disable network/security/krb5kdc
```

5 在旧的主 KDC 上，指定请求传播的轮询时间。

将 /etc/krb5/kdc.conf 中的 sunw_dbprop_master_ulogsize 项替换为定义 sunw_dbprop_slave_poll 的项。该项将轮询时间设置为 2 分钟。

```
kdc1 # cat /etc/krb5/kdc.conf

[kdcdefaults]

        kdc_ports = 88,750



[realms]

        EXAMPLE.COM= {

                profile = /etc/krb5/krb5.conf

                database_name = /var/krb5/principal

                admin_keytab = /etc/krb5/kadm5.keytab

                acl_file = /etc/krb5/kadm5.acl

                kadmind_port = 749

                max_life = 8h 0m 0s

                max_renewable_life = 7d 0h 0m 0s

                sunw_dbprop_enable = true
```

```
                sunw_dbprop_slave_poll = 2m

        }
```

6 **在旧的主 KDC 上，移动主 KDC 命令和 kadm5.acl 文件。**

要禁止运行主 KDC 命令，请将 kprop、kadmind 和 kadmin.local 命令移到一个保留位置。

```
kdc1 # mv /usr/lib/krb5/kprop /usr/lib/krb5/kprop.save

kdc1 # mv /usr/lib/krb5/kadmind /usr/lib/krb5/kadmind.save

kdc1 # mv /usr/sbin/kadmin.local /usr/sbin/kadmin.local.save

kdc1 # mv /etc/krb5/kadm5.acl /etc/krb5/kadm5.acl.save
```

7 **在 DNS 服务器上，更改主 KDC 的别名。**

要更改服务器，请编辑 example.com 区域文件并更改 masterkdc 的项。

```
masterkdc IN CNAME kdc4
```

8 **在 DNS 服务器上，重新启动 Internet 域名服务器。**

运行以下命令以重新装入新的别名信息：

```
# svcadm refresh network/dns/server
```

9 **在新的主 KDC 上，移动主 KDC 命令和从 kpropd.acl 文件。**

```
kdc4 # mv /usr/lib/krb5/kprop.save /usr/lib/krb5/kprop

kdc4 # mv /usr/lib/krb5/kadmind.save /usr/lib/krb5/kadmind

kdc4 # mv /usr/sbin/kadmin.local.save /usr/sbin/kadmin.local

kdc4 # mv /etc/krb5/kpropd.acl /etc/krb5/kpropd.acl.save
```

10 **在新的主 KDC 上，创建 Kerberos 访问控制列表文件 (kadm5.acl)。**

填充后，/etc/krb5/kadm5.acl 文件应包含允许管理 KDC 的所有主体名称。该文件还应列出请求增量传播的所有从 KDC。有关更多信息，请参见 kadm5.acl (4) 手册页。

```
kdc4 # cat /etc/krb5/krb5.acl

kws/admin@EXAMPLE.COM   *

kiprop/kdc1.example.com@EXAMPLE.COM p
```

11 **在新的主 KDC 上的 kdc.conf 文件中，指定更新日志大小。**

将 sunw_dbprop_slave_poll 项替换为定义 sunw_dbprop_master_ulogsize 的项。该项将日志大小设置为 1000 项。

```
kdc1 # cat /etc/krb5/kdc.conf
```

```
[kdcdefaults]

        kdc_ports = 88,750


[realms]

        EXAMPLE.COM= {

                profile = /etc/krb5/krb5.conf

                database_name = /var/krb5/principal

                admin_keytab = /etc/krb5/kadm5.keytab

                acl_file = /etc/krb5/kadm5.acl

                kadmind_port = 749

                max_life = 8h 0m 0s

                max_renewable_life = 7d 0h 0m 0s

                sunw_dbprop_enable = true

                sunw_dbprop_master_ulogsize = 1000

        }
```

12 在新的主 KDC 上，将 kiprop 主体添加到 kadmind 密钥表文件中。

```
kdc4 # kadmin.local

kadmin.local: ktadd -k /etc/krb5/kadm5.keytab kiprop/kdc4.example.com

Entry for principal kiprop/kdc4.example.com with kvno 3, encryption type AES-128 CTS mode

          with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/kadm5.keytab.

Entry for principal kiprop/kdc4.example.com with kvno 3, encryption type Triple DES cbc

          mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/kadm5.keytab.

Entry for principal kiprop/kdc4.example.com with kvno 3, encryption type ARCFOUR

          with HMAC/md5 added to keytab WRFILE:/etc/krb5/kadm5.keytab.

Entry for principal kiprop/kdc4.example.com with kvno 3, encryption type DES cbc mode
```

```
          with RSA-MD5 added to keytab WRFILE:/etc/krb5/kadm5.keytab.

kadmin.local: quit
```

13 在新的主 KDC 上，启动 kadmind 和 krb5kdc。

```
kdc4 # svcadm enable network/security/krb5kdc

kdc4 # svcadm enable network/security/kadmin
```

14 在旧的主 KDC 上，添加 kiprop 服务主体。

通过将 kiprop 主体添加到 krb5.keytab 文件中，kpropd 守护进程可以对其自身进行增量传播服务验证。

```
kdc1 # /usr/sbin/kadmin -p kws/admin

Authenticating as pricipal kws/admin@EXAMPLE.COM with password.

Enter password:      <Type kws/admin password>

kadmin: ktadd kiprop/kdc1.example.com

Entry for principal kiprop/kdc1.example.com with kvno 3, encryption type AES-128 CTS mode

          with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal kiprop/kdc1.example.com with kvno 3, encryption type Triple DES cbc

          mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal kiprop/kdc1.example.com with kvno 3, encryption type ARCFOUR

          with HMAC/md5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal kiprop/kdc1.example.com with kvno 3, encryption type DES cbc mode

          with RSA-MD5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

kadmin: quit
```

15 在旧的主 KDC 上，将 krb5.conf 中列出的每个 KDC 的项添加到传播配置文件 kpropd.acl 中。

```
kdc1 # cat /etc/krb5/kpropd.acl

host/kdc1.example.com@EXAMPLE.COM

host/kdc2.example.com@EXAMPLE.COM

host/kdc3.example.com@EXAMPLE.COM
```

```
host/kdc4.example.com@EXAMPLE.COM
```

16 **在旧的主 KDC 上，启动 kpropd 和 krb5kdc。**

启动 krb5kdc 守护进程时，如果将系统配置为从 KDC，则 kpropd 也将启动。

```
kdc1 # svcadm enable network/security/krb5kdc
```

管理 Kerberos 数据库

Kerberos 数据库是 Kerberos 的主干，必须正确维护。本节介绍有关如何管理 Kerberos 数据库的一些过程，例如备份和恢复数据库、设置增量或并行传播以及管理存储文件。第 365 页中的 “如何配置主 KDC” 中介绍了初始设置该数据库的步骤。

备份和传播 Kerberos 数据库

将 Kerberos 数据库从主 KDC 传播到从 KDC 是最重要的配置任务之一。如果传播频率不够高，则主 KDC 和从 KDC 将不能同步。因此，如果主 KDC 关闭，则从 KDC 将不能获取最新的数据库信息。此外，如果出于平衡负载目的将从 KDC 配置为主 KDC，则将该从 KDC 用作主 KDC 的客户机将不能获取最新的信息。所以，必须确保传播频率足够高，或者基于更改 Kerberos 数据库的频率配置服务器使其进行增量传播。增量传播优先于手动传播，因为手动传播数据库时需要更多的管理开销。此外，执行完全数据库传播时效率很低。

配置主 KDC 时，可以在 cron 作业中设置 kprop_script 命令以自动将 Kerberos 数据库备份到 /var/krb5/slave_datatrans 转储文件，并将该文件传播到从 KDC。不过，与其他文件一样，Kerberos 数据库可能会损坏。如果从 KDC 上的数据受损，您可能无法注意到，因为下一次数据库自动传播会安装一个新的副本。但是，如果主 KDC 上的数据受损，则下一次传播期间会将损坏的数据库传播到所有从 KDC。而且，损坏的备份会覆写主 KDC 上先前未损坏的备份文件。

由于在这种情况下不存在任何“安全”的备份副本，因此还应设置 cron 作业，以便定期将 slave_datatrans 转储文件复制到另一位置，或者使用 kdb5_util 的 dump 命令创建另一份单独的备份副本。这样，如果数据库受损，则可以使用 kdb5_util 的 load 命令在主 KDC 上恢复最新备份。

另一条重要注意事项是：由于数据库转储文件包含主体密钥，因此需要阻止未经授权的用户访问该文件。缺省情况下，只有 root 身份才具有读写数据库转储文件的权限。要阻止未经授权的访问，请仅使用 kprop 命令传播数据库转储文件，该命令会对要传送的数据进行加密。此外，kprop 仅将数据传播到从 KDC，这可以最大程度地降低将数据库转储文件意外发送到未经授权的主机的几率。

注意 – 如果传播 Kerberos 数据库之后对其进行了更新，并且在下一次传播之前该数据库受损，则从 KDC 将不包含这些更新。这些更新将丢失。因此，如果要在计划的定期传播之前向 Kerberos 数据库中添加重要的更新，应手动传播该数据库，以避免数据丢失。

kpropd.acl 文件

KDC 上的 kpropd.acl 文件提供主机主体名称的列表（一个名称占一行），用于指定 KDC 可以通过传播从其接收更新数据库的系统。如果使用主 KDC 传播所有从 KDC，则每个从 KDC 上的 kpropd.acl 文件仅需包含主 KDC 的主机主体名称。

但是，本书中的 Kerberos 安装和后续配置步骤将指导您如何将相同的 kpropd.acl 文件添加到主 KDC 和从 KDC 中。此文件包含所有 KDC 主机主体名称。通过此配置，在传播 KDC 临时不可用时，可以从任何 KDC 进行传播。而且，通过在所有 KDC 上保留相同副本，可以更容易地维护配置。

kprop_script 命令

kprop_script 命令使用 kprop 命令将 Kerberos 数据库传播到其他 KDC。如果在从 KDC 上运行 kprop_script 命令，则会将该从 KDC 的 Kerberos 数据库副本传播到其他 KDC。kprop_script 的参数接受主机名列表，该列表以空格分隔，表示要传播的 KDC。

运行 kprop_script 时，将在 /var/krb5/slave_datatrans 文件中创建 Kerberos 数据库的备份，并将该文件复制到指定的 KDC。在完成传播之前，Kerberos 数据库处于锁定状态。

▼ 如何备份 Kerberos 数据库

1. **成为主 KDC 的超级用户。**

2. **使用 kdb5_util 命令的 dump 命令备份 Kerberos 数据库。**

   # /usr/sbin/kdb5_util dump [-verbose] [-d *dbname*] [*filename* [*principals*...]]

   -verbose　列显要备份的每个主体和策略的名称。

   dbname　定义要备份的数据库的名称。请注意，可以指定文件的绝对路径。如果未指定 -d 选项，则缺省数据库名称为 /var/krb5/principal。

   filename　定义用于备份数据库的文件。可以指定文件的绝对路径。如果未指定文件，则数据库将转储到标准输出。

   principals　定义要备份的一个或多个主体的列表（以空格分隔）。必须使用全限定主体名称。如果未指定任何主体，则将备份整个数据库。

示例 22–10 备份 Kerberos 数据库

在以下示例中，Kerberos 数据库将备份到名为 dumpfile 的文件中。由于指定了 -verbose 选项，因此备份时会列显每个主体。

```
# kdb5_util dump -verbose dumpfile

kadmin/kdc1.eng.example.com@ENG.EXAMPLE.COM

krbtgt/eng.example.com@ENG.EXAMPLE.COM

kadmin/history@ENG.EXAMPLE.COM

pak/admin@ENG.EXAMPLE.COM

pak@ENG.EXAMPLE.COM

changepw/kdc1.eng.example.com@ENG.EXAMPLE.COM
```

在以下示例中，将备份 Kerberos 数据库中的 pak 和 pak/admin 主体。

```
# kdb5_util dump -verbose dumpfile pak/admin@ENG.EXAMPLE.COM pak@ENG.EXAMPLE.COM

pak/admin@ENG.EXAMPLE.COM

pak@ENG.EXAMPLE.COM
```

▼ 如何恢复 Kerberos 数据库

1. **成为主 KDC 的超级用户。**
2. **使用 kdb_util 命令的 load 命令恢复 Kerberos 数据库。**

   # /usr/sbin/kdb5_util load [-verbose] [-d *dbname*] [-update] [*filename*]

-verbose	列显要恢复的每个主体和策略的名称。
dbname	定义要恢复的数据库的名称。请注意，可以指定文件的绝对路径。如果未指定 -d 选项，则缺省数据库名称为 /var/krb5/principal。
-update	更新现有数据库。否则，会创建新数据库或覆写现有数据库。
filename	定义用于恢复数据库的文件。可以指定文件的绝对路径。

示例 22–11 恢复 Kerberos 数据库

在以下示例中，将从 dumpfile 文件将名为 database1 的数据库恢复到当前目录。由于未指定 -update 选项，恢复操作将创建一个新数据库。

```
# kdb5_util load -d database1 dumpfile
```

▼ 如何重新装入 Kerberos 数据库

如果未在运行 Solaris 10 发行版的服务器上创建 KDC 数据库，则通过重新装入该数据库，可以利用改进的数据库格式。

开始之前 请确保数据库使用的是旧格式。请参见特定说明。

1. **在主 KDC 上，停止 KDC 守护进程。**

```
kdc1 # svcadm disable network/security/krb5kdc

kdc1 # svcadm disable network/security/kadmin
```

2. **转储 KDC 数据库。**

```
kdc1 # kdb5_util dump /tmp/prdb.txt
```

3. **保存当前数据库文件的副本。**

```
kdc1 # cd /var/krb5

kdc1 # mkdir old

kdc1 # mv princ* old/
```

4. **装入数据库。**

```
kdc1 # kdb5_util load /tmp/prdb.txt
```

5. **启动 KDC 守护进程。**

```
kdc1 # svcadm enable network/security/krb5kdc

kdc1 # svcadm enable network/security/kadmin
```

▼ 如何重新配置主 KDC 以使用增量传播

此过程中的步骤可用于重新配置现有的主 KDC，以使用增量传播。在此过程中，将使用以下配置参数：

- 领域名称 = `EXAMPLE.COM`
- DNS 域名 = `example.com`
- 主 KDC = `kdc1.example.com`
- 从 KDC = `kdc2.example.com`
- `admin` 主体 = `kws/admin`

1 **向 kdc.conf 中添加项。**

需要启用增量传播，并选择主 KDC 将在日志中存储的更新数。有关更多信息，请参见 kdc.conf(4) 手册页。

```
kdc1 # cat /etc/krb5/kdc.conf

[kdcdefaults]

        kdc_ports = 88,750



[realms]

        EXAMPLE.COM= {

                profile = /etc/krb5/krb5.conf

                database_name = /var/krb5/principal

                admin_keytab = /etc/krb5/kadm5.keytab

                acl_file = /etc/krb5/kadm5.acl

                kadmind_port = 749

                max_life = 8h 0m 0s

                max_renewable_life = 7d 0h 0m 0s

                sunw_dbprop_enable = true

                sunw_dbprop_master_ulogsize = 1000

        }
```

2 **创建 kiprop 主体。**

kiprop 主体用于验证主 KDC 服务器和授权来自主 KDC 的更新。

```
kdc1 # /usr/sbin/kadmin -p kws/admin

Enter password:      <Type kws/admin password>

kadmin: addprinc -randkey kiprop/kdc1.example.com

Principal "kiprop/kdc1.example.com@EXAMPLE.COM" created.

kadmin: addprinc -randkey kiprop/kdc2.example.com
```

```
Principal "kiprop/kdc2.example.com@EXAMPLE.COM" created.

kadmin:
```

3 **将 kiprop 主体添加到 kadmind 密钥表文件中**

通过将 kiprop 主体添加到 kadm5.keytab 文件中，kadmind 命令可以在启动时对其自身进行验证。

```
kadmin: ktadd -k /etc/krb5/kadm5.keytab kiprop/kdc1.example.com

Entry for principal kiprop/kdc1.example.com with kvno 3, encryption type AES-128 CTS mode

          with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/kadm5.keytab.

Entry for principal kiprop/kdc1.example.com with kvno 3, encryption type Triple DES cbc

          mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/kadm5.keytab.

Entry for principal kiprop/kdc1.example.com with kvno 3, encryption type ARCFOUR

          with HMAC/md5 added to keytab WRFILE:/etc/krb5/kadm5.keytab.

Entry for principal kiprop/kdc1.example.com with kvno 3, encryption type DES cbc mode

          with RSA-MD5 added to keytab WRFILE:/etc/krb5/kadm5.keytab.

kadmin: quit
```

4 **（可选的）在主 KDC 上，将 kiprop 项添加到 kpropd.acl 中**

通过此项，主 KDC 可以接收对 kdc2 服务器的增量传播请求。

```
kdc1 # cat /etc/krb5/kpropd.acl

host/kdc1.example.com@EXAMPLE.COM

host/kdc2.example.com@EXAMPLE.COM

*/admin@EXAMPLE.COM *

kiprop/kdc2.example.com@EXAMPLE.COM p
```

5 **注释掉 root crontab 文件中的 kprop 行。**

此步骤禁止从 KDC 传播其 KDC 数据库副本。

```
kdc1 # crontab -e

#ident  "@(#)root       1.20    01/11/06 SMI"
```

```
#
# The root crontab should be used to perform accounting data collection.
#
# The rtc command is run to adjust the real time clock if and when
# daylight savings time changes.
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
#10 3 * * * /usr/lib/krb5kprop_script kdc2.example.sun.com #SUNWkr5ma
```

6 **重新启动** kadmind。

```
kdc1 # svcadm restart network/security/kadmin
```

7 **重新配置所有使用增量传播的从 KDC 服务器。**

▼ 如何重新配置从 KDC 以使用增量传播

1 **向** krb5.conf **中添加项。**

这些新项启用增量传播并将轮询时间设置为 2 分钟。

```
kdc2 # cat /etc/krb5/kdc.conf

[kdcdefaults]

        kdc_ports = 88,750


[realms]

        EXAMPLE.COM= {
```

```
                profile = /etc/krb5/krb5.conf

                database_name = /var/krb5/principal

                admin_keytab = /etc/krb5/kadm5.keytab

                acl_file = /etc/krb5/kadm5.acl

                kadmind_port = 749

                max_life = 8h 0m 0s

                max_renewable_life = 7d 0h 0m 0s

                sunw_dbprop_enable = true

                sunw_dbprop_slave_poll = 2m

        }
```

2 将 kiprop 主体添加到 krb5.keytab 文件中。

```
kdc2 # /usr/sbin/kadmin -p kws/admin

Enter password:      <Type kws/admin password>

kadmin: ktadd kiprop/kdc2.example.com

Entry for principal kiprop/kdc2.example.com with kvno 3, encryption type AES-128 CTS mode

          with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal kiprop/kdc2.example.com with kvno 3, encryption type Triple DES cbc

          mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal kiprop/kdc2.example.com with kvno 3, encryption type ARCFOUR

          with HMAC/md5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal kiprop/kdc2.example.com with kvno 3, encryption type DES cbc mode

          with RSA-MD5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

kadmin: quit
```

3 禁用 kpropd。

```
kdc2 # svcadm disable network/security/krb5_prop
```

4 **重新启动 KDC 服务器。**

```
kdc2 # svcadm restart network/security/krb5kdc
```

▼ 如何配置从 KDC 以使用完全传播

此过程说明如何重新配置运行 Solaris 10 发行版的从 KDC 服务器，以使用完全传播。通常，只有运行 Solaris 9 发行版或更早发行版的主 KDC 服务器才需要使用此过程。在这种情况下，主 KDC 服务器不支持增量传播，因此需要配置从 KDC 以允许进行传播。

在此过程中，将配置名为 `kdc3` 的从 KDC。此过程使用以下配置参数：

- 领域名称 = `EXAMPLE.COM`
- DNS 域名 = `example.com`
- 主 KDC = `kdc1.example.com`
- 从 KDC = `kdc2.example.com` 和 `kdc3.example.com`
- `admin` 主体 = `kws/admin`
- 联机帮助 URL = `http://denver:8888/ab2/coll.384.1/SEAM/@AB2PageView/6956`

注 – 调整该 URL 以指向“SEAM Administration Tool”部分，如第 362 页中的 “SEAM Administration Tool 中的联机帮助 URL”中所述。

开始之前 必须配置主 KDC。有关此从 KDC 是否可交换的特定说明，请参见第 405 页中的 “交换主 KDC 和从 KDC”。

1 **在主 KDC 上，成为超级用户。**

2 **在主 KDC 上，启动 `kadmin`。**

必须使用在配置主 KDC 时创建的一个 `admin` 主体名称登录。

```
kdc1 # /usr/sbin/kadmin -p kws/admin

Enter password:      <Type kws/admin password>

kadmin:
```

a. **在主 KDC 上，将从主机主体添加到数据库中（如果尚未执行此操作）。**

要使从 KDC 正常工作，该从 KDC 必须具有主机主体。请注意，主体实例为主机名时，无论 `/etc/resolv.conf` 文件中的域名是大写还是小写，都必须以小写字母指定 FQDN。

```
kadmin: addprinc -randkey host/kdc3.example.com

Principal "host/kdc3@EXAMPLE.COM" created.
```

```
kadmin:
```

b. **退出** `kadmin`。

```
kadmin: quit
```

3 **在主 KDC 上，编辑 Kerberos 配置文件 (`krb5.conf`)。**

需要添加每个从 KDC 的项。有关此文件的完整说明，请参见 `krb5.conf`(4) 手册页。

```
kdc1 # cat /etc/krb5/krb5.conf

 .

 .

[realms]

                EXAMPLE.COM = {

                kdc = kdc1.example.com

                kdc = kdc2.example.com

                kdc = kdc3.example.com

                admin_server = kdc1.example.com

        }
```

4 **在主 KDC 上，将主 KDC 和每个从 KDC 的项添加到 `kpropd.acl` 文件中。**

有关此文件的完整说明，请参见 `kprop`(1M) 手册页。

```
kdc1 # cat /etc/krb5/kpropd.acl

host/kdc1.example.com@EXAMPLE.COM

host/kdc2.example.com@EXAMPLE.COM

host/kdc3.example.com@EXAMPLE.COM
```

5 **在所有从 KDC 上，复制主 KDC 服务器的 KDC 管理文件。**

由于主 KDC 服务器已更新每台 KDC 服务器所需的信息，因此需要在所有从 KDC 上执行此步骤。可以使用 `ftp` 或类似的传送机制从主 KDC 获取以下文件的副本：

- `/etc/krb5/krb5.conf`
- `/etc/krb5/kdc.conf`
- `/etc/krb5/kpropd.acl`

6 **在所有从 KDC 上，请确保未填充 Kerberos 访问控制列表文件** kadm5.acl。

未修改的 kadm5.acl 文件如下所示：

```
kdc2 # cat /etc/krb5/kadm5.acl

*/admin@___default_realm___ *
```

如果此文件中包含 kiprop 项，请删除它们。

7 **在新的从 KDC 上，启动** kadmin **命令。**

必须使用在配置主 KDC 时创建的一个 admin 主体名称登录。

```
kdc2 # /usr/sbin/kadmin -p kws/admin

Enter password:      <Type kws/admin password>

kadmin:
```

a. **使用** kadmin **将从 KDC 的** host **主体添加到从 KDC 的密钥表文件中。**

此项可使 kprop 和其他基于 Kerberos 的应用程序正常工作。请注意，主体实例为主机名时，无论 /etc/resolv.conf 文件中的域名是大写还是小写，都必须以小写字母指定 FQDN。

```
kadmin: ktadd host/kdc3.example.com

Entry for principal host/kdc3.example.com with kvno 3, encryption type AES-128 CTS mode

          with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal host/kdc3.example.com with kvno 3, encryption type Triple DES cbc

          mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal host/kdc3.example.com with kvno 3, encryption type ARCFOUR

          with HMAC/md5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

Entry for principal host/kdc3.example.com with kvno 3, encryption type DES cbc mode

          with RSA-MD5 added to keytab WRFILE:/etc/krb5/krb5.keytab.

kadmin:
```

b. **退出** kadmin。

```
kadmin: quit
```

8 **在主 KDC 上，将从 KDC 名称添加到 cron 作业中，该作业通过运行 crontab -e 自动运行备份。**

在 kprop_script 行的结尾添加每个从 KDC 服务器的名称。

```
10 3 * * * /usr/lib/krb5/kprop_script kdc2.example.com kdc3.example.com
```

您可能还希望更改备份的时间。此项将在每天上午的 3:10 启动备份过程。

9 **在新的从 KDC 上，启动 Kerberos 传播守护进程。**

```
kdc3 # svcadm enable network/security/krb5_prop
```

10 **在主 KDC 上，使用 kprop_script 备份并传播数据库。**

如果已存在数据库的备份副本，则无需完成其他备份。有关进一步的说明，请参见第 425 页中的 “如何手动将 Kerberos 数据库传播到从 KDC”。

```
kdc1 # /usr/lib/krb5/kprop_script kdc3.example.com

Database propagation to kdc3.example.com: SUCCEEDED
```

11 **在新的从 KDC 上，使用 kdb5_util 创建一个存储文件。**

```
kdc3 # /usr/sbin/kdb5_util stash

kdb5_util: Cannot find/read stored master key while reading master key

kdb5_util: Warning: proceeding without master key


Enter KDC database master key:      <Type the key>
```

12 **（可选的）在新的从 KDC 上，使用 NTP 或其他时钟同步机制同步主 KDC 时钟。**

安装和使用网络时间协议 (Network Time Protocol, NTP) 并非必需。但是，要成功验证，每个时钟必须处于 krb5.conf 文件的 libdefaults 部分中定义的缺省时间内。有关 NTP 的信息，请参见第 404 页中的 “同步 KDC 和 Kerberos 客户机的时钟”。

13 **在新的从 KDC 上，启动 KDC 守护进程 (krb5kdc)。**

```
kdc3 # svcadm enable network/security/krb5kdc
```

▼ 如何验证 KDC 服务器是否已同步

如果配置了增量传播，则此过程可确保已更新从 KDC 上的信息。

1 **在 KDC 主服务器上，运行 kproplog 命令。**

```
kdc1 # /usr/sbin/kproplog -h
```

2 在从 KDC 服务器上，运行 kproplog 命令。

```
kdc2 # /usr/sbin/kproplog -h
```

3 检查最后一个序列号和最后一个时间标记的值是否匹配。

示例 22–12 验证 KDC 服务器是否已同步

以下是在主 KDC 服务器上运行 kproplog 命令的结果样例。

```
kdc1 # /usr/sbin/kproplog -h


Kerberos update log (/var/krb5/principal.ulog)

Update log dump:

    Log version #: 1

    Log state: Stable

    Entry block size: 2048

    Number of entries: 2500

    First serial #: 137966

    Last serial #: 140465

    First time stamp: Fri Nov 28 00:59:27 2004

    Last time stamp: Fri Nov 28 01:06:13 2004
```

以下是在从 KDC 服务器上运行 kproplog 命令的结果样例。

```
kdc2 # /usr/sbin/kproplog -h


Kerberos update log (/var/krb5/principal.ulog)

Update log dump:

    Log version #: 1

    Log state: Stable
```

```
Entry block size: 2048

Number of entries: 0

First serial #: None

Last serial #: 140465

First time stamp: None

Last time stamp: Fri Nov 28 01:06:13 2004
```

请注意，最后一个序列号和最后一个时间标记的值相同，这表示从 KDC 服务器与主 KDC 服务器同步。

请注意，在从 KDC 服务器的输出中，从 KDC 服务器的更新日志中不存在任何更新项。这是因为与主 KDC 服务器不同，从 KDC 服务器不保留更新。此外，由于第一个序列号或第一个时间标记不是相关信息，因此从 KDC 服务器也不包括这些信息。

▼ 如何手动将 Kerberos 数据库传播到从 KDC

此过程说明如何使用 `kprop` 命令传播 Kerberos 数据库。如果需要在定期的 `cron` 作业之外将从 KDC 与主 KDC 同步，可使用此过程。与 `kprop_script` 不同，可以使用 `kprop` 仅传播当前数据库备份，而无需先创建 Kerberos 数据库的新备份。

注 – 如果使用的是增量传播，则不要使用此过程。

---

1. **成为主 KDC 的超级用户。**
2. **（可选的）使用 `kdb5_util` 命令备份数据库。**

   ```
   # /usr/sbin/kdb5_util dump /var/krb5/slave_datatrans
   ```

3. **使用 `kprop` 命令将数据库传播到从 KDC。**

   ```
   # /usr/lib/krb5/kprop -f /var/krb5/slave_datatrans slave-KDC
   ```

示例 22–13 使用 `kprop_script` 手动将 Kerberos 数据库传播到从 KDC

如果要备份数据库，并在定期的 `cron` 作业之外将数据库传播到从 KDC，则还可以按如下所示使用 `kprop_script` 命令：

```
# /usr/lib/krb5/kprop_script slave-KDC
```

设置并行传播

在大多数情况下，会以独占的方式使用主 KDC 将其 Kerberos 数据库传播到从 KDC。但是，如果站点上有很多从 KDC，则可以考虑共享装入传播进程，即所谓的**并行传播**。

注 – 如果使用的是增量传播，则不要使用此过程。

通过并行传播，特定的从 KDC 可以与主 KDC 共享传播功能。通过共享此功能，可以更快地完成传播并减轻主 KDC 的工作。

例如，假设站点上有一个主 KDC 和六个从 KDC（如图 22–2 中所示），其中，`slave-1` 到 `slave-3` 组成一个逻辑组，`slave-4` 到 `slave-6` 组成另一个逻辑组。要设置并行传播，可以使主 KDC 将数据库传播到 `slave-1` 和 `slave-4`。而这些从 KDC 又可将数据库传播到其组中的从 KDC。

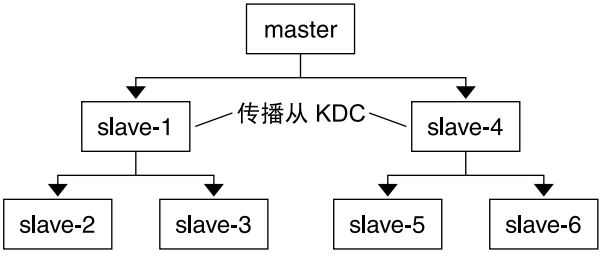


图 22–2 并行传播配置示例

设置并行传播的配置步骤

以下不是详细的逐步过程，而是用于启用并行传播的配置步骤的高级列表。这些步骤包括：

1. 在主 KDC 上，更改其 `cron` 作业的 `kprop_script` 项，以仅包括将执行后续传播的从 KDC（**传播从** *KDC*）的参数。
2. 在每个传播从 KDC 上，将 `kprop_script` 项添加到其 `cron` 作业中，其中必须包括要传播的从 KDC 的参数。要成功地以并行方式进行传播，应设置 `cron` 作业，使其在将新 Kerberos 数据库传播到从 KDC 本身之后再运行。

   注 – 对传播从 KDC 进行传播所需的时间取决于多种因素，例如网络带宽和 Kerberos 数据库的大小。

   ---

3. 在每个从 KDC 上，设置相应的传播权限。通过将从 KDC 传播的 KDC 的主机主体名称添加到其 `kpropd.acl` 文件中，可完成此步骤。

示例 22–14 设置并行传播

以图 22–2 为例，主 KDC 的 `kprop_script` 项与以下示例类似：

```
0 3 * * * /usr/lib/krb5/kprop_script slave-1.example.com slave-4.example.com
```

`slave-1` 的 `kprop_script` 项与以下示例类似：

```
0 4 * * * /usr/lib/krb5/kprop_script slave-2.example.com slave-3.example.com
```

请注意，从 KDC 上的传播在主 KDC 将数据库传播到它一小时后开始。

传播从 KDC 上的 `kpropd.acl` 文件将包含以下项：

```
host/master.example.com@EXAMPLE.COM
```

要由 `slave-1` 传播的从 KDC 上的 `kpropd.acl` 文件将包含以下项：

```
host/slave-1.example.com@EXAMPLE.COM
```

管理存储文件

存储文件包含 Kerberos 数据库的主密钥，该密钥在创建 Kerberos 数据库时自动创建。如果存储文件损坏，则可以使用 `kdb5_util` 实用程序的 `stash` 命令替换损坏的文件。仅在使用 `kdb5_util` 的 `destroy` 命令删除 Kerberos 数据库之后，才应删除存储文件。由于存储文件不会随数据库一起自动删除，所以必须删除存储文件以完成清除。

▼ 如何删除存储文件

1. **成为包含存储文件的 KDC 的超级用户。**

2. **删除存储文件。**

   `# rm` *stash-file*

   其中，*stash-file* 是存储文件的路径。缺省情况下，存储文件位于 `/var/krb5/.k5.`*realm* 中。

注 – 如果需要重新创建存储文件，则可以使用 `kdb5_util` 命令的 `-f` 选项。

增强 Kerberos 服务器的安全性

执行以下步骤以增强 Kerberos 应用程序服务器和 KDC 服务器的安全性。

▼ 如何仅启用基于 Kerberos 的应用程序

此过程限制对正在运行 `telnet`、`ftp`、`rcp`、`rsh` 和 `rlogin` 的服务器的网络访问，以便仅执行经过 Kerberos 验证的事务。

1 **更改 `telnet` 服务的 `exec` 属性。**

将 `-a user` 选项添加到 `telnet` 的 `exec` 属性，以将访问权限限制为可以提供有效验证信息的那些用户。

```
# inetadm -m svc:/network/telnet:default exec="/usr/sbin/in.telnetd -a user"
```

2 **（可选的）如果尚未配置，则更改 `telnet` 服务的 `exec` 属性。**

将 `-a` 选项添加到 `ftp` 的 `exec` 属性，以仅允许经过 Kerberos 验证的连接。

```
# inetadm -m svc:/network/ftp:default exec="/usr/sbin/in.ftpd -a"
```

3 **禁用其他服务。**

应禁用 `in.rshd` 和 `in.rlogind` 守护进程。

```
# svcadm disable network/shell

# svcadm disable network/login:rlogin
```

▼ 如何限制对 KDC 服务器的访问

主 KDC 服务器和从 KDC 服务器都包含存储在本地的 KDC 数据库副本。限制对这些服务器的访问（以便保证数据库安全）对于 Kerberos 安装的整体安全非常重要。

1 **根据需要，禁用远程服务。**

要提供安全的 KDC 服务器，应禁用所有不必要的网络服务。根据配置不同，可能已禁用其中某些服务。使用 `svcs` 命令检查服务状态。在大多数情况下，只需运行 `time` 和 `krdb5_kprop` 服务。此外，使用回送 TLI（`ticlts`、`ticotsord` 和 `ticots`）的任何服务可以保持启用状态。

```
# svcadm disable network/comsat

# svcadm disable network/dtspc/tcp

# svcadm disable network/finger

# svcadm disable network/login:rlogin
```

```
# svcadm disable network/rexec

# svcadm disable network/shell

# svcadm disable network/talk

# svcadm disable network/tname

# svcadm disable network/uucp

# svcadm disable network/rpc_100068_2-5/rpc_udp
```

2 **限制对支持 KDC 的硬件的访问。**

要限制物理访问，请确保 KDC 服务器及其监视器位于安全的设备中。用户应不能以任何方式访问此服务器。

3 **在本地磁盘或从 KDC 上存储 KDC 数据库备份。**

仅在可以安全存储磁带时创建 KDC 的磁带备份。该做法同样适用于创建密钥表文件的副本。最好在未与其他系统共享的本地文件系统上存储这些文件。存储文件系统可以位于主 KDC 服务器或任何从 KDC 上。

◆ ◆ ◆ 第 23 章

Kerberos 错误消息和疑难解答

本章将分析使用 Kerberos 服务时可能收到的错误消息，另外还针对各种问题提供一些疑难解答提示。以下是本章中错误消息和疑难解答信息的列表。

- 第 431 页中的 “SEAM Administration Tool 错误消息”
- 第 432 页中的 “常见的 Kerberos 错误消息 (A-M)”
- 第 438 页中的 “常见的 Kerberos 错误消息 (N-Z)”
- 第 441 页中的 “krb5.conf 文件的格式存在问题”
- 第 442 页中的 “传播 Kerberos 数据库时出现问题”
- 第 442 页中的 “挂载基于 Kerberos 的 NFS 文件系统时出现问题”
- 第 443 页中的 “以 root 身份进行验证时出现问题”
- 第 443 页中的 “观察从 GSS 凭证到 UNIX 凭证的映射”

Kerberos 错误消息

本节介绍有关 Kerberos 错误消息的信息，包括每个错误出现的原因以及解决此错误的方法。

SEAM Administration Tool 错误消息

`Unable to view the list of principals or policies; use the Name field.`

原因: 登录时使用的 `admin` 主体在 Kerberos ACL 文件 (`kadm5.acl`) 中没有列出权限 (`l`)。因此，无法查看主体列表或策略列表。

解决方法: 必须在 "Name" 字段中键入主体名称和策略名称才能对其进行处理，或者需要使用具有相应权限的主体登录。

```
JNI: Java array creation failed
JNI: Java class lookup failed
JNI: Java field lookup failed
JNI: Java method lookup failed
JNI: Java object lookup failed
JNI: Java object field lookup failed
JNI: Java string access failed
JNI: Java string creation failed
```

原因: SEAM Administration Tool (`gkadmin`) 使用的 Java 本机接口存在严重问题。

解决方法: 退出 `gkadmin` 然后重新启动。如果问题仍然存在，请报告错误。

常见的 Kerberos 错误消息 (A-M)

本节按字母顺序 (A-M) 列出了 Kerberos 命令、Kerberos 守护进程、PAM 框架、GSS 接口、NFS 服务和 Kerberos 库的常见错误消息。

`All authentication systems disabled; connection refused`

原因: 此版本的 `rlogind` 不支持任何验证机制。

解决方法: 请确保调用的 `rlogind` 带有 `-k` 选项。

`Another authentication mechanism must be used to access this host`

原因: 无法进行验证。

解决方法: 请确保客户机使用 Kerberos V5 机制进行验证。

`Authentication negotiation has failed, which is required for encryption. Good bye.`

原因: 无法与服务器协商验证。

解决方法: 请通过使用 `toggle authdebug` 命令调用 `telnet` 命令，启动验证调试并查看调试消息以获取更多线索。另外，请确保具有有效凭证。

`Bad krb5 admin server hostname while initializing kadmin interface`

原因: 在 `krb5.conf` 文件中为 `admin_server` 配置了无效的主机名。

解决方法: 请确保在 `krb5.conf` 文件的 `admin_server` 行中为主 KDC 指定了正确的主机名。

`Bad lifetime value`

原因: 提供的生命周期值无效或格式不正确。

解决方法: 请确保提供的值与 `kinit`(1) 手册页中的“时间格式”一节相符。

`Bad start time value`
原因: 提供的开始时间值无效或格式不正确。

解决方法: 请确保提供的值与 `kinit`(1) 手册页中的“时间格式”一节相符。

`Cannot contact any KDC for requested realm`
原因: 请求的领域中没有 KDC 响应。

解决方法: 请确保至少可访问一个 KDC（主 KDC 或从 KDC），或 `krb5kdc` 守护进程正在 KDC 上运行。有关已配置 KDC 的列表 (`kdc` = *kdc-name*)，请检查 `/etc/krb5/krb5.conf` 文件。

`Cannot determine realm for host`
原因: Kerberos 无法确定主机的领域名称。

解决方法: 请确保存在缺省领域名称，或在 Kerberos 配置文件 (`krb5.conf`) 中设置了域名映射。

`Cannot find KDC for requested realm`
原因: 在请求的领域中找不到 KDC。

解决方法: 请确保 Kerberos 配置文件 (`krb5.conf`) 在 `realm` 部分中指定了 KDC。

`cannot initialize realm` *realm_name*
原因: KDC 可能没有存储文件。

解决方法: 请确保 KDC 具有存储文件。否则，请使用 `kdb5_util` 命令创建一个存储文件，然后尝试重新启动 `krb5kdc` 命令。

`Cannot resolve KDC for requested realm`
原因: Kerberos 无法确定该领域的任何 KDC。

解决方法: 请确保 Kerberos 配置文件 (`krb5.conf`) 在 `realm` 部分中指定了 KDC。

`Cannot reuse password`
原因: 指定的口令之前已被此主体使用。

解决方法: 请选择一个以前尚未选用的口令，至少不要是 KDC 数据库中为每个主体保存的那些口令。此策略由该主体的策略强制执行。

`Can't get forwarded credentials`
原因: 无法建立凭证转发。

解决方法: 请确保主体具有可转发的凭证。

`Can't open/find Kerberos configuration file`
原因: Kerberos 配置文件 (`krb5.conf`) 不可用。

解决方法: 请确保 `krb5.conf` 文件在正确的位置中可用，并且具有正确的权限。此文件应可由 `root` 写入，并可由其他用户读取。

`Client did not supply required checksum--connection rejected`
原因: 未与客户机协商使用校验和进行验证。客户机使用的可能是不支持初始连接支持的早期 Kerberos V5 协议。

解决方法: 请确保客户机使用的是支持初始连接支持的 Kerberos V5 协议。

`Client/server realm mismatch in initial ticket request`
原因: 在初始票证请求中，客户机与服务器之间的领域不匹配。

解决方法: 请确保正在与您通信的服务器与客户机位于同一领域中，或确保领域配置正确。

`Client or server has a null key`
原因: 主体拥有空密钥。

解决方法: 请使用 `kadmin` 的 `cpw` 命令修改主体，使其拥有非空密钥。

`Communication failure with server while initializing kadmin interface`
原因: 为管理服务器指定的主机（也称为主 KDC）没有运行 `kadmind` 守护进程。

解决方法: 请确保为主 KDC 指定正确的主机名。如果指定了正确的主机名，请确保 `kadmind` 正在指定的主 KDC 上运行。

`Credentials cache file permissions incorrect`
原因: 您对凭证高速缓存 (`/tmp/krb5cc_` *uid*) 没有相应的读写权限。

解决方法: 请确保具有对凭证高速缓存的读写权限。

`Credentials cache I/O operation failed` *XXX*
原因: Kerberos 在向系统的凭证高速缓存 (`/tmp/krb5cc_`*uid*) 进行写入时出现问题。

解决方法: 请使用 `df` 命令确保尚未删除凭证高速缓存，并且设备中还有剩余空间。

`Decrypt integrity check failed`
原因: 您的票证可能无效。

解决方法: 请检验下列两种情况：

- 确保您的凭证有效。请使用 `kdestroy` 销毁票证，然后使用 `kinit` 创建新票证。
- 确保目标主机的密钥表文件的服务密钥版本正确。请使用 `kadmin` 查看 Kerberos 数据库中服务主体（例如 `host/`*FQDN-hostname*）的密钥版本号。另外，请在目标主机上使用 `klist -k`，以确保该主机具有相同的密钥版本号。

`Encryption could not be enabled. Goodbye.`
原因: 无法与服务器协商加密。

解决方法: 请通过使用 `toggle encdebug` 命令调用 `telnet` 命令，启动验证调试并查看调试消息以获取更多线索。

`failed to obtain credentials cache`
原因: 在 `kadmin` 初始化过程中，`kadmin` 尝试获取 `admin` 主体的凭证时失败。

解决方法: 请确保在执行 `kadmin` 时使用正确的主体和口令。

`Field is too long for this implementation`
原因: 基于 Kerberos 的应用程序所发送的消息太长。如果传输协议是 UDP，则可能会生成此错误。UDP 的缺省最大消息长度是 65535 字节。此外，对通过 Kerberos 服务发送的协议消息中的单个字段也有限制。

解决方法: 请检验是否已在 KDC 服务器的 `/etc/krb5/kdc.conf` 文件中将传输协议限制为 UDP。

`GSS-API (or Kerberos) error`
原因: 此消息是通用的 GSS-API 或 Kerberos 错误消息，可能由几种不同的问题所导致。

解决方法: 请检查 `/var/krb5/kdc.log` 文件，查找出现此错误时记录的更具体的错误消息。

`Hostname cannot be canonicalized`
原因: Kerberos 无法设置全限定主机名。

解决方法: 请确保在 DNS 中定义了该主机名，并且主机名至地址的映射和地址至主机名的映射保持一致。

`Illegal cross-realm ticket`
原因: 发送的票证所跨的领域不正确。领域可能未设置正确的信任关系。

解决方法: 请确保使用的领域具有正确的信任关系。

`Improper format of Kerberos configuration file`
原因: Kerberos 配置文件包含无效项。

解决方法: 请确保 `krb5.conf` 文件中的所有关系后面都跟有 "=" 符号和值。另外，请检验每个子段中的括号是否成对出现。

`Inappropriate type of checksum in message`
原因: 消息中包含无效的校验和类型。

解决方法: 请检查在 `krb5.conf` 和 `kdc.conf` 文件中指定的有效校验和类型。

`Incorrect net address`
原因: 网络地址不匹配。正在转发的票证中的网络地址与处理该票证的网络地址不同。转发票证时可能会出现此消息。

解决方法: 请确保网络地址正确。请使用 `kdestroy` 销毁票证，然后使用 `kinit` 创建新票证。

`Invalid credential was supplied`

`Service key not available`

原因: 凭证高速缓存中的服务票证可能不正确。

解决方法: 请在尝试使用此服务之前，销毁当前凭证高速缓存并重新运行 `kinit`。

`Invalid flag for file lock mode`

原因: 出现内部 Kerberos 错误。

解决方法: 请报告错误。

`Invalid message type specified for encoding`

原因: Kerberos 无法识别基于 Kerberos 的应用程序发送的消息类型。

解决方法: 如果使用的基于 Kerberos 的应用程序是由您的站点或供应商开发的，请确保此应用程序正确使用 Kerberos。

`Invalid number of character classes`

原因: 指定的主体口令没有按照主体策略的强制要求包含足够的口令类。

解决方法: 请确保指定的口令包含策略要求的最少口令类数。

`KADM err: Memory allocation failure`

原因: 用于运行 `kadmin` 的内存不足。

解决方法: 请释放内存，然后再次尝试运行 `kadmin`。

`KDC can't fulfill requested option`

原因: KDC 不允许请求的选项。一种可能是正在请求以后生效或可转发的选项，而 KDC 不允许这些选项。另一种可能是请求了 TGT 更新，但没有可更新的 TGT。

解决方法: 请确定是要请求 KDC 不允许的选项，还是请求不可用的票证类型。

`KDC policy rejects request`

原因: KDC 策略不允许该请求。例如，向 KDC 发出的请求中没有 IP 地址。或者请求了转发，但 KDC 不允许转发。

解决方法: 请确保使用带有正确选项的 `kinit`。如有必要，请修改与主体关联的策略或更改主体的属性以允许该请求。通过使用 `kadmin`，可以修改策略或主体。

`KDC reply did not match expectations`

原因: KDC 回复未包含期望的主体名称，或者响应中的其他值不正确。

解决方法: 请确保正在与您通信的 KDC 符合 RFC1510，正在发送的请求是 Kerberos V5 请求或该 KDC 可用。

`kdestroy: Could not obtain principal name from cache`
原因: 凭证高速缓存缺失或已损坏。

解决方法: 请检查提供的高速缓存位置是否正确。如有必要，请使用 `kinit` 删除 TGT 并获取新的 TGT。

`kdestroy: No credentials cache file found while destroying cache`
原因: 凭证高速缓存 (`/tmp/krb5c_` *uid*) 缺失或已损坏。

解决方法: 请检查提供的高速缓存位置是否正确。如有必要，请使用 `kinit` 删除 TGT 并获取新的 TGT。

`kdestroy: TGT expire warning NOT deleted`
原因: 凭证高速缓存缺失或已损坏。

解决方法: 请检查提供的高速缓存位置是否正确。如有必要，请使用 `kinit` 删除 TGT 并获取新的 TGT。

`Kerberos authentication failed`
原因: Kerberos 口令不正确，或该口令可能与 UNIX 口令不同步。

解决方法: 如果口令不同步，则必须指定其他口令才能完成 Kerberos 验证。用户可能会忘记其原始口令。

`Kerberos V5 refuses authentication`
原因: 无法与服务器协商验证。

解决方法: 请通过使用 `toggle authdebug` 命令调用 `telnet` 命令，启动验证调试并查看调试消息以获取更多线索。另外，请确保具有有效凭证。

`Key table entry not found`
原因: 网络应用程序服务器的密钥表文件中不存在服务主体项。

解决方法: 请将相应的服务主体添加到服务器的密钥表文件中，以便该文件可提供基于 Kerberos 的服务。

`Key version number for principal in key table is incorrect`
原因: 密钥表文件中主体的密钥版本与 Kerberos 数据库中的版本不同。可能已更改了服务密钥，也可能正在使用旧服务票证。

解决方法: 如果服务密钥已被更改（例如通过使用 `kadmin`），则需要提取新密钥并将其存储在正在运行该服务的主机的密钥表文件中。

或者，您也可能正在使用具有较旧密钥的旧服务票证。在这种情况下，可能需要运行 `kdestroy` 命令，然后再次运行 `kinit` 命令。

`kinit: gethostname failed`
原因: 本地网络配置中的错误导致 `kinit` 失败。

解决方法: 请确保主机配置正确。

`login: load_modules: can not open module /usr/lib/security/pam_krb5.so.1`
原因: Kerberos PAM 模块可能缺失，也可能该模块不是有效的可执行二进制文件。

解决方法: 请确保 Kerberos PAM 模块位于 `/usr/lib/security` 目录中，并且是有效的可执行二进制文件。另外，请确保 `/etc/pam.conf` 文件包含 `pam_krb5.so.1` 的正确路径。

`Looping detected inside krb5_get_in_tkt`
原因: Kerberos 多次尝试获取初始票证，但均失败。

解决方法: 请确保至少有一个 KDC 正在响应验证请求。

`Master key does not match database`
原因: 未从包含主密钥的数据库创建装入的数据库转储。主密钥位于 `/var/krb5/.k5.`*REALM* 中。

解决方法: 请确保装入的数据库转储中的主密钥与 `/var/krb5/.k5.` *REALM* 中的主密钥匹配。

`Matching credential not found`
原因: 未找到与请求匹配的凭证。凭证高速缓存中没有请求需要的凭证。

解决方法: 请使用 `kdestroy` 销毁票证，然后使用 `kinit` 创建新票证。

`Message out of order`
原因: 使用顺序保密性发送的消息到达时顺序混乱。某些消息可能已在传输过程中丢失。

解决方法: 应重新初始化 Kerberos 会话。

`Message stream modified`
原因: 计算出的校验和与消息校验和不匹配。消息在传输过程中可能已被修改，这表明存在安全泄露。

解决方法: 请确保消息在网络中正确发送。由于此消息还可表明消息在发送过程中被篡改，因此请使用 `kdestroy` 销毁票证，然后重新初始化所使用的 Kerberos 服务。

常见的 Kerberos 错误消息 (N-Z)

本节按字母顺序 (N-Z) 列出了 Kerberos 命令、Kerberos 守护进程、PAM 框架、GSS 接口、NFS 服务和 Kerberos 库的常见错误消息。

`No credentials cache file found`
原因: Kerberos 无法找到凭证高速缓存 (`/tmp/krb5cc_`*uid*)。

解决方法: 请确保该凭证文件存在并且可以读取。否则，请再次尝试执行 `kinit`。

`No credentials were supplied, or the credentials were unavailable or inaccessible`

`No credential cache found`
原因: 用户的凭证高速缓存不正确或不存在。

解决方法: 用户应在尝试启动服务之前运行 `kinit`。

`No credentials were supplied, or the credentials were unavailable or inaccessible`

`No principal in keytab matches desired name`
原因: 尝试验证服务器时出现错误。

解决方法: 请确保主机或服务主体位于服务器的密钥表文件中。

`Operation requires “`*privilege*`” privilege`
原因: 正在使用的 `admin` 主体未在 `kadm5.acl` 文件中配置相应的权限。

解决方法: 请使用具有相应权限的主体。或者，请通过修改 `kadm5.acl` 文件来配置所使用的主体，使其具有相应的权限。通常，名称中包含 `/admin` 的主体具有相应的权限。

`PAM-KRB5 (auth): krb5_verify_init_creds failed: Key table entry not found`
原因: 远程应用程序尝试在本地 `/etc/krb5/krb5.keytab` 文件中读取主机的服务主体，但不存在任何主体。

解决方法: 请将主机的服务主体添加到主机的密钥表文件中。

`Password is in the password dictionary`
原因: 指定的口令位于正在使用的口令字典中。您选择的口令不适合用作口令。

解决方法: 请选用包含混合口令类的口令。

`Permission denied in replay cache code`
原因: 无法打开系统的重放高速缓存。首次运行服务器时所使用的用户 ID 可能与当前的用户 ID 不同。

解决方法: 请确保重放高速缓存具有相应的权限。重放高速缓存存储在运行基于 Kerberos 的服务器应用程序的主机上。对于非 `root` 用户，重放高速缓存文件称为 `/var/krb5/rcache/rc_`*service_name*`_ uid`。对于 root 用户，重放高速缓存文件称为 `/var/krb5/rcache/root/rc_ `*service_name*。

`Protocol version mismatch`
原因: 很可能向 KDC 发送了 Kerberos V4 请求。Kerberos 服务仅支持 Kerberos V5 协议。

解决方法: 请确保应用程序使用的是 Kerberos V5 协议。

`Request is a replay`

原因: 请求已发送到此服务器并进行了处理。票证可能已被盗用，并且其他用户正在尝试重新使用这些票证。

解决方法: 请等待几分钟，然后重新发出请求。

`Requested principal and ticket don't match`

原因: 您正在连接的服务主体与您所拥有的服务票证不匹配。

解决方法: 请确保 DNS 正常运行。如果使用的是其他供应商的软件，请确保该软件使用的主体名称正确。

`Requested protocol version not supported`

原因: 很可能向 KDC 发送了 Kerberos V4 请求。Kerberos 服务仅支持 Kerberos V5 协议。

解决方法: 请确保应用程序使用的是 Kerberos V5 协议。

`Server refused to negotiate authentication, which is required for encryption. Good bye.`

原因: 远程应用程序无法接受或已配置为不接受来自客户机的 Kerberos 验证。

解决方法: 请提供可以协商验证的远程应用程序，或配置该应用程序以使用相应标志来打开验证。

`Server refused to negotiate encryption. Good bye.`

原因: 无法与服务器协商加密。

解决方法: 请通过使用 `toggle encdebug` 命令调用 `telnet` 命令，启动验证调试并查看调试消息以获取更多线索。

`Server rejected authentication (during sendauth exchange)`

原因: 您正在尝试与其通信的服务器拒绝验证。此错误通常出现在 Kerberos 数据库传播过程中。一些常见的原因可能是 `kpropd.acl` 文件、DNS 或密钥表文件存在问题。

解决方法: 如果在运行 `kprop` 以外的应用程序时收到此错误，请检查服务器的密钥表文件是否正确。

`The ticket isn't for us`

`Ticket/authenticator don't match`

原因: 票证与验证者不匹配。请求中的主体名称可能与服务主体的名称不匹配，因为发送票证使用的是主体的 FQDN 名称，而服务期望非 FQDN 名称，反之亦然。

解决方法: 如果在运行 `kprop` 以外的应用程序时收到此错误，请检查服务器的密钥表文件是否正确。

`Ticket expired`
原因: 票证时间已到期。

解决方法: 请使用 `kdestroy` 销毁票证，然后使用 `kinit` 创建新票证。

`Ticket is ineligible for postdating`
原因: 主体不允许其票证以后生效。

解决方法: 请使用 `kadmin` 修改主体以允许以后生效。

`Ticket not yet valid`
原因: 以后生效的票证仍然无效。

解决方法: 请使用正确的日期创建新票证，或等待当前票证生效。

`Truncated input file detected`
原因: 操作中使用的数据库转储文件不是完整的转储文件。

解决方法: 请重新创建转储文件，或使用其他数据库转储文件。

`Unable to securely authenticate user ... exit`
原因: 无法与服务器协商验证。

解决方法: 请通过使用 `toggle authdebug` 命令调用 `telnet` 命令，启动验证调试并查看调试消息以获取更多线索。另外，请确保具有有效凭证。

`Wrong principal in request`
原因: 票证中包含无效的主体名称。此错误可能表明 DNS 或 FQDN 存在问题。

解决方法: 请确保服务主体与票证中的主体匹配。

Kerberos 疑难解答

本节介绍有关 Kerberos 软件的疑难解答信息。

krb5.conf 文件的格式存在问题

如果 `krb5.conf` 文件的格式不正确，`telnet` 命令将会失败。但是，`dtlogin` 和 `login` 命令仍将成功，即使按这些命令的要求指定 `krb5.conf` 文件也是如此。如果出现此问题，则会显示以下错误消息：

```
Error initializing krb5: Improper format of Kerberos configuration
```

此外，格式不正确的 `krb5.conf` 文件还会阻止使用 GSSAPI 的应用程序使用 `krb5` 机制。

如果 `krb5.conf` 文件的格式存在问题，则安全性很容易受到破坏。您应首先解决该问题，然后再允许使用 Kerberos 功能。

传播 Kerberos 数据库时出现问题

如果传播 Kerberos 数据库失败，请在从 KDC 与主 KDC 之间尝试使用 `/usr/bin/rlogin -x`，反之亦然。

如果 KDC 已设置为限制访问，则会禁用 `rlogin`，因此无法使用它来解决此问题。要在 KDC 上启用 `rlogin`，必须启用 `eklogin` 服务。

```
# svcadm enable svc:/network/login:eklogin
```

解决此问题后，需要禁用 `eklogin` 服务。

如果 `rlogin` 无法运行，则可能是因为 KDC 上的密钥表文件存在问题。如果 `rlogin` 可以运行，则问题不在于密钥表文件或名称服务，因为 `rlogin` 和传播软件使用相同的 `host/`*host-name* 主体。在这种情况下，请确保 `kpropd.acl` 文件正确。

挂载基于 Kerberos 的 NFS 文件系统时出现问题

- 如果挂载基于 Kerberos 的 NFS 文件系统失败，请确保 NFS 服务器上存在 `/var/rcache/root` 文件。如果该文件系统不属于 `root`，请将其删除并再次尝试挂载。
- 如果访问基于 Kerberos 的 NFS 文件系统时出现问题，请确保系统和 NFS 服务器上启用了 `gssd` 服务。
- 如果在尝试访问基于 Kerberos 的 NFS 文件系统时出现 `invalid argument` 或 `bad directory` 错误消息，则可能是因为在尝试挂载 NFS 文件系统时未使用全限定 DNS 名称。正在挂载的主机与服务器的密钥表文件中的服务主体的主机名部分不相同。

  如果服务器有多个以太网接口，并且已将 DNS 设置为使用“每个接口一个名称”的方案，而不是“每台主机多条地址记录”的方案，则也可能会出现此问题。对于 Kerberos 服务，应为每台主机设置多条地址记录，如下所示[1]：

```
my.host.name.   A       1.2.3.4

                A       1.2.4.4

                A       1.2.5.4



my-en0.host.name.       A       1.2.3.4

my-en1.host.name.       A       1.2.4.4

my-en2.host.name.       A       1.2.5.4
```

1 Ken Hornstein，"Kerberos FAQ"，[http://www.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html]，访问时间：1998 年 12 月 11 日。

```
4.3.2.1         PTR     my.host.name.

4.4.2.1         PTR     my.host.name.

4.5.2.1         PTR     my.host.name.
```

在本示例中，此设置允许引用服务器的密钥表文件中的不同接口和一个服务主体（而非三个服务主体）一次。

以 root 身份进行验证时出现问题

如果在尝试成为系统超级用户时验证失败，并且已将 root 主体添加到主机的密钥表文件中，则需要检查两个可能的问题。首先，请确保密钥表文件中的 root 主体具有一个全限定主机名作为其实例。如果具有该名称，请检查 /etc/resolv.conf 文件，以确保系统已正确设置为 DNS 客户机。

观察从 GSS 凭证到 UNIX 凭证的映射

为了可以监视凭证映射，请首先在 /etc/gss/gsscred.conf 文件中取消对以下行的注释。

```
SYSLOG_UID_MAPPING=yes
```

然后，指示 gssd 服务从 /etc/gss/gsscred.conf 文件中获取信息。

```
# pkill -HUP gssd
```

现在，您应该可以在 gssd 请求凭证映射时对其进行监视。如果针对 auth 系统功能将 syslog.conf 文件设置为 debug 严重级别，则可通过 syslogd 记录这些映射。

◆ ◆ ◆ 第 24 章

管理 Kerberos 主体和策略（任务）

本章介绍有关管理主体及与其关联的策略的过程。本章还将说明如何管理主机的密钥表文件。

需要管理主体和策略的用户应该阅读本章。阅读本章之前，应熟悉主体和策略，包括所有规划注意事项。请分别参阅第 20 章和第 21 章。

以下是本章中信息的列表。

- 第 445 页中的 “管理 Kerberos 主体和策略的方法”
- 第 446 页中的 “SEAM Administration Tool”
- 第 450 页中的 “管理 Kerberos 主体”
- 第 463 页中的 “管理 Kerberos 策略”
- 第 472 页中的 “SEAM Tool 参考”
- 第 476 页中的 “管理密钥表文件”

管理 Kerberos 主体和策略的方法

主 KDC 上的 Kerberos 数据库包含您所在领域的所有 Kerberos 主体、主体口令、策略和其他管理信息。创建和删除主体以及修改其属性时，可以使用 `kadmin` 或 `gkadmin` 命令。

`kadmin` 命令提供一个交互式的命令行界面，用于维护 Kerberos 主体、策略和密钥表文件。`kadmin` 命令具有以下两个版本：

- `kadmin`－使用 Kerberos 验证从网络中的任何位置安全操作
- `kadmin.local`－必须直接在主 KDC 上运行

除 `kadmin` 使用 Kerberos 来验证用户外，这两个版本的功能完全相同。如果要设置足够的数据库以便使用远程版本，则必须使用本地版本。

另外，Solaris 发行版还提供了 SEAM Administration Tool (`gkadmin`)，这是一个交互式的图形用户界面 (graphical user interface, GUI)，其功能基本上与 `kadmin` 命令相同。有关更多信息，请参见第 446 页中的 “SEAM Administration Tool” 。

SEAM Administration Tool

SEAM Administration Tool (SEAM Tool) 是一个交互式的图形用户界面 (graphical user interface, GUI)，用于维护 Kerberos 主体和策略。此工具提供的功能与 kadmin 命令基本相同。但是，此工具不支持密钥表文件管理。必须使用 kadmin 命令来管理密钥表文件，如第 476 页中的 “管理密钥表文件”中所述。

与 kadmin 命令类似，SEAM Tool 使用 Kerberos 验证和加密 RPC 从网络中的任何位置安全操作。SEAM Tool 可以执行以下操作：

- 创建基于缺省值或现有主体的新主体。
- 创建基于现有策略的新策略。
- 为主体添加注释。
- 设置缺省值以创建新主体。
- 在不退出工具的情况下以其他主体身份登录。
- 打印或保存主体列表和策略列表。
- 查看和搜索主体列表和策略列表。

SEAM Tool 还会提供关联说明和一般联机帮助。

以下任务列表提供了指向可借助 SEAM Tool 完成的各种任务的链接：

- 第 451 页中的 “管理 Kerberos 主体（任务列表）”
- 第 464 页中的 “管理 Kerberos 策略（任务列表）”

此外，还可转至第 472 页中的 “SEAM Tool 面板说明”，了解可在 SEAM Tool 中指定或查看的所有主体属性和策略属性的说明。

SEAM Tool 的等效命令行

本节列出了一些 kadmin 命令，其提供的功能与 SEAM Tool 相同。无需运行 X 窗口系统，便可使用这些命令。尽管本章中的大多数过程使用 SEAM Tool，但其中许多过程还提供了使用等效命令行的对应示例。

表 24–1 SEAM Tool 的等效命令行

SEAM Tool 过程	等效的 kadmin 命令
查看主体列表。	list_principals 或 get_principals
查看主体属性。	get_principal
创建新主体。	add_principal
复制主体。	无等效命令行
修改主体。	modify_principal 或 change_password

表 24–1 SEAM Tool 的等效命令行 （续）

SEAM Tool 过程	等效的 kadmin 命令
删除主体。	delete_principal
设置缺省值以创建新主体。	无等效命令行
查看策略列表。	list_policies 或 get_policies
查看策略属性。	get_policy
创建新策略。	add_policy
复制策略。	无等效命令行
修改策略。	modify_policy
删除策略。	delete_policy

SEAM Tool 修改的唯一文件

SEAM Tool 修改的唯一文件是 $HOME/.gkadmin 文件。该文件包含用于创建新主体的缺省值。通过从 "Edit" 菜单中选择 "Properties"，可以更新该文件。

SEAM Tool 的打印和联机帮助功能

SEAM Tool 提供了打印功能和联机帮助功能。通过 "Print" 菜单，可将以下各项发送至打印机或文件：

- 指定主 KDC 上的可用主体列表
- 指定主 KDC 上的可用策略列表
- 当前选择的主体或装入的主体
- 当前选择的策略或装入的策略

通过 "Help" 菜单，可以访问关联说明和一般帮助。从 "Help" 菜单中选择 "Context-Sensitive Help" 时，将显示 "Context-Sensitive Help" 窗口并且工具会切换为帮助模式。在帮助模式下，如果单击该窗口中的任何字段、标签或按钮，将在 "Help" 窗口中显示有关该项的帮助。要切换回工具的一般模式，请在 "Help" 窗口中单击 "Dismiss"。

此外，还可选择 "Help Contents"，这将打开一个 HTML 浏览器，其中会提供指向本章中介绍的一般概述和任务信息的链接。

在 SEAM Tool 中处理大型列表

随着站点开始积累大量主体和策略，使用 SEAM Tool 装入并显示主体和策略列表所需的时间将会越来越长。因此，使用该工具时的工作效率会下降。解决此问题有多种办法。

首先，通过使 SEAM Tool 不装入列表，可以完全省去装入列表的时间。可以设置此选项，方法是从 "Edit" 菜单中选择 "Properties"，然后取消选中 "Show Lists" 字段。当然，如果该工具不装入列表，则不能显示这些列表，因此将无法再使用列表面板来选择主体或策略。而必须在提供的新 "Name" 字段中键入主体或策略名称，然后选择要对其执行的操作。键入名称与从列表中选择项的结果相同。

处理大型列表的另一种方法是对其进行高速缓存。实际上，已将 SEAM Tool 的缺省行为设置为将列表高速缓存一段时间。最初 SEAM Tool 还是必须将这些列表装入高速缓存。但在此后，该工具就可以使用高速缓存，而不必再次获取列表。这样便无需不断从服务器装入列表（正是此操作占用了大量时间）。

通过从 "Edit" 菜单中选择 "Properties"，可以设置列表高速缓存。有两种高速缓存设置。可以选择将列表永久高速缓存；也可以指定必须将列表从服务器重新装入高速缓存的时间限制。

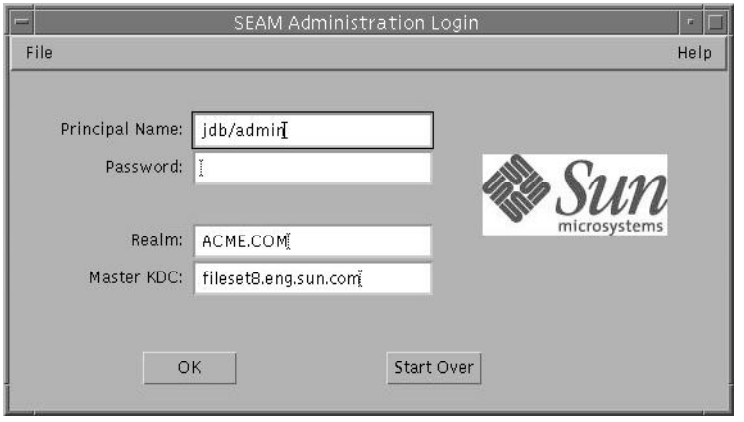
对列表进行高速缓存时，仍然可以使用列表面板来选择主体和策略，因此该方法不会像第一种方法那样影响 SEAM Tool 的使用方式。另外，尽管使用高速缓存使您无法查看其他用户所做的更改，但您仍可以根据自己所做的更改查看最新列表信息，因为您所做的更改会对服务器和高速缓存中的列表进行更新。而且，如果要更新高速缓存以查看其他更改并获取最新列表副本，可在需要从服务器刷新高速缓存时使用 "Refresh" 菜单。

▼ 如何启动 SEAM Tool

1 使用 `gkadmin` 命令启动 SEAM Tool。

```
$ /usr/sbin/gkadmin
```

此时会显示 "SEAM Administration Login" 窗口。



2 如果不想使用现有的缺省值，请指定新的缺省值。

该窗口会自动使用缺省值填充。缺省主体名称 (*username*/admin) 是通过从 USER 环境变量获取当前身份并在其后附加 /admin 确定的。缺省的 "Realm" 和 "Master KDC" 字段选自 /etc/krb5/krb5.conf 文件。如果要恢复这些缺省值，请单击 "Start Over"。

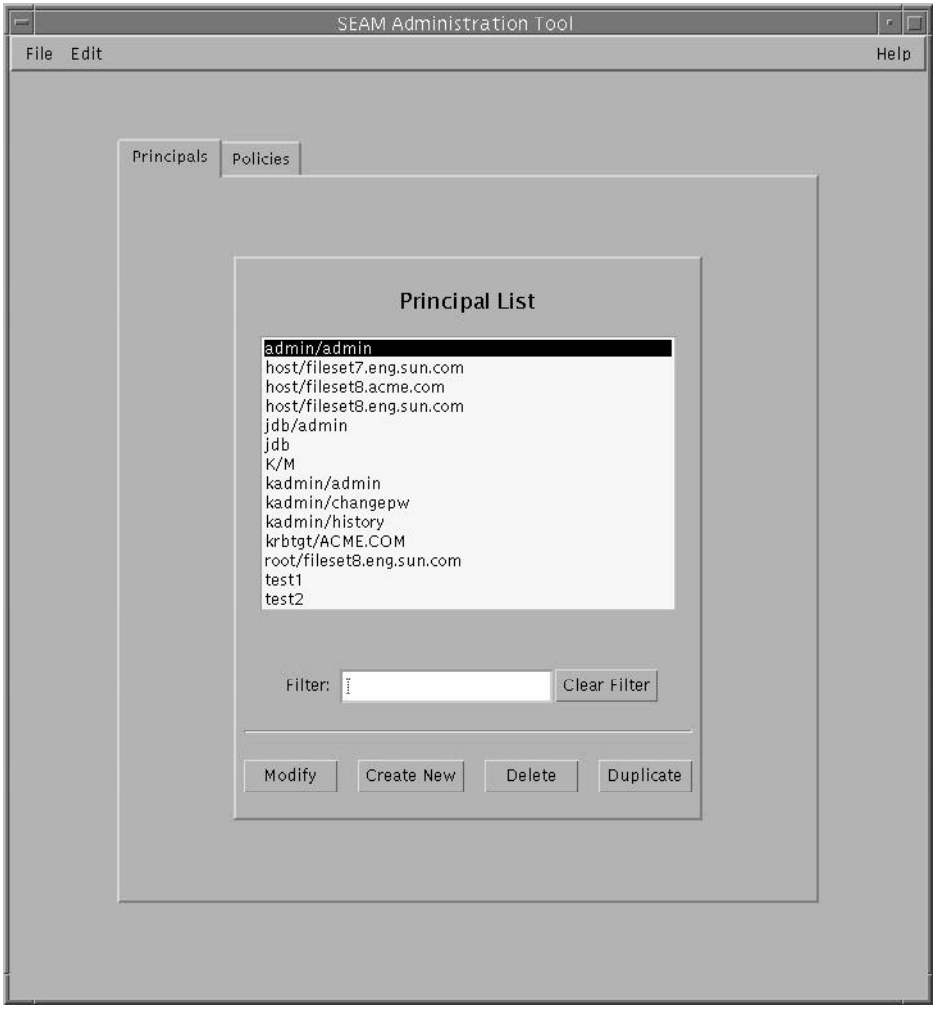
注 – 每个主体名称可以执行的管理操作在 Kerberos ACL 文件 /etc/krb5/kadm5.acl 中指定。有关受限权限的信息，请参见第 474 页中的 “以受限 Kerberos 管理权限使用 SEAM Tool” 。

---

3 键入指定主体名称的口令。

4 单击 "OK"。

此时会显示以下窗口：



管理 Kerberos 主体

本节提供使用 SEAM Tool 管理主体的逐步说明，还提供等效命令行示例（如果有）。

管理 Kerberos 主体（任务列表）

任务	说明	参考
查看主体列表。	通过单击 "Principals" 选项卡来查看主体列表。	第 452 页中的 “如何查看 Kerberos 主体列表”
查看主体属性。	通过在 "Principal List" 中选择 "Principal"，然后单击 "Modify" 按钮来查看主体的属性。	第 454 页中的 “如何查看 Kerberos 主体属性”
创建新主体。	通过单击 "Principal List" 面板中的 "Create New" 按钮来创建新主体。	第 456 页中的 “如何创建新的 Kerberos 主体”
复制主体。	通过在 "Principal List" 中选择要复制的主体，然后单击 "Duplicate" 按钮来复制主体。	第 459 页中的 “如何复制 Kerberos 主体”
修改主体。	通过在 "Principal List" 中选择要修改的主体，然后单击 "Modify" 按钮来修改主体。 请注意，不能修改主体的名称。要重命名主体，必须首先复制该主体，为其指定一个新名称并保存，然后删除旧主体。	第 459 页中的 “如何修改 Kerberos 主体”
删除主体。	通过在 "Principal List" 中选择要删除的主体，然后单击 "Delete" 按钮来删除主体。	第 460 页中的 “如何删除 Kerberos 主体”
设置缺省值以创建新主体。	通过从 "Edit" 菜单中选择 "Properties" 来设置缺省值以创建新主体。	第 461 页中的 “如何设置缺省值以创建新的 Kerberos 主体”
修改 Kerberos 管理权限（`kadm5.acl` 文件）。	**仅限命令行。** Kerberos 管理权限确定主体可对 Kerberos 数据库执行的操作，如添加和修改。 要修改每个主体的 Kerberos 管理权限，需要编辑 `/etc/krb5/kadm5.acl` 文件。	第 462 页中的 “如何修改 Kerberos 管理权限”

自动创建新的 Kerberos 主体

尽管 SEAM Tool 使用方便，但它不提供自动创建新主体的方法。如果需要在短时间内添加 10 个甚至 100 个新主体，则自动创建尤其有用。而在 Bourne shell 脚本中使用 `kadmin.local` 命令正好可满足这一需要。

以下 shell 脚本行示例说明了如何自动创建新主体：

```
awk '{ print "ank +needchange -pw", $2, $1 }' < /tmp/princnames |

        time /usr/sbin/kadmin.local> /dev/null
```

为了方便阅读，已将此示例拆分为两行。该脚本将读入一个称为 `princnames` 的文件（其中包含主体名称及其口令）然后将其添加到 Kerberos 数据库。您必须创建 `princnames` 文件，

并在每一行上包含一个主体名称及其口令，中间用一个或多个空格分隔。+needchange 选项用于配置主体，以便在用户第一次使用该主体登录时提示其输入新口令。此做法有助于确保 princnames 文件中的口令不会引入安全风险。

可以生成更详细的脚本。例如，脚本可使用名称服务中的信息来获取主体名称的用户名列表。所执行的操作和执行操作的方式取决于站点的需要以及脚本编制技术。

▼ 如何查看 Kerberos 主体列表

此过程后附等效命令行示例。

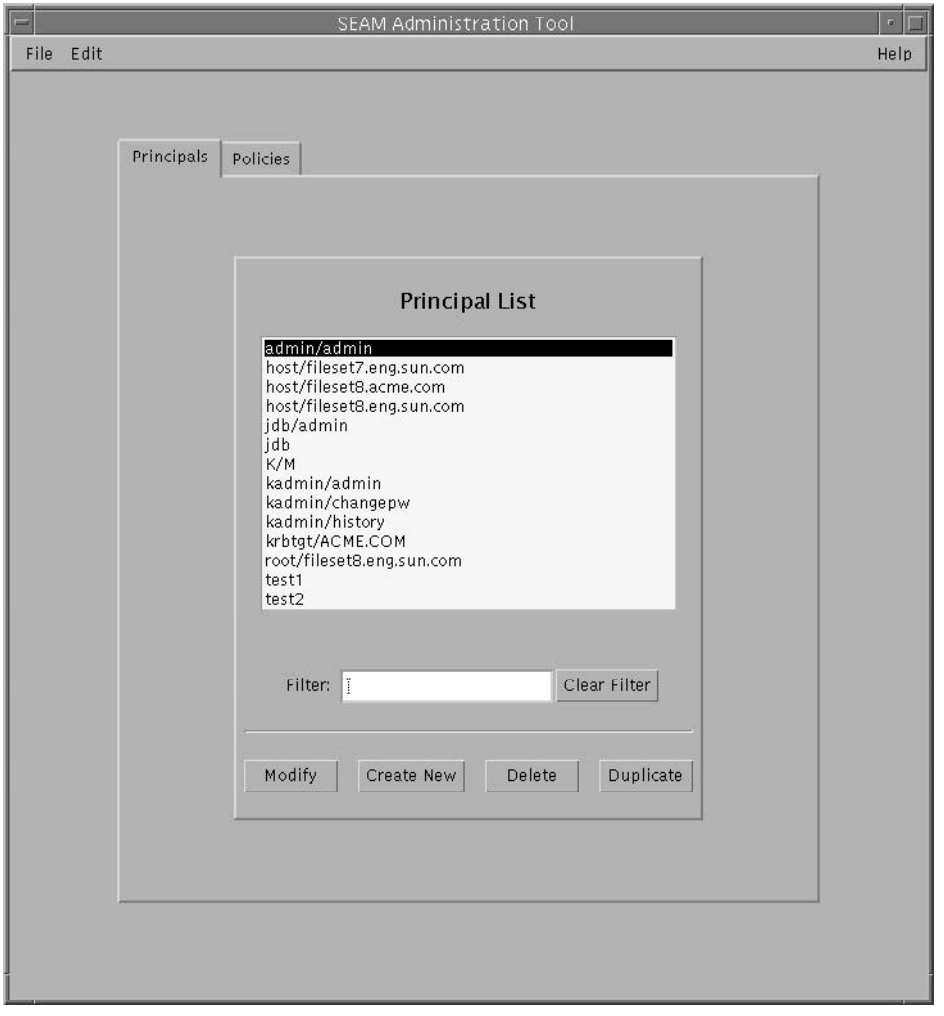
1. **如有必要，启动 SEAM Tool。**

   有关更多信息，请参见第 448 页中的 “如何启动 SEAM Tool”。

   ```
   $ /usr/sbin/gkadmin
   ```

2. **单击 "Principals" 选项卡。**

   此时会显示主体列表。



3 显示特定主体或主体子列表。

在 "Filter" 字段中键入过滤字符串，然后按 "Return"。如果过滤操作成功，则会显示与过滤器匹配的主体列表。

过滤字符串必须由一个或多个字符组成。由于过滤机制区分大小写，因此需要对过滤器使用正确的大小写字母。例如，如果键入过滤字符串 `ge`，则过滤机制仅显示包含 `ge` 字符串的主体（如 `george` 或 `edge`）。

如果要显示主体的完整列表，请单击 "Clear Filter"。

示例 24–1 查看 Kerberos 主体列表（命令行）

在以下示例中，kadmin 的 list_principals 命令用于列出与 test* 匹配的所有主体。通配符可与 list_principals 命令一起使用。

```
kadmin: list_principals test*

test1@EXAMPLE.COM

test2@EXAMPLE.COM

kadmin: quit
```

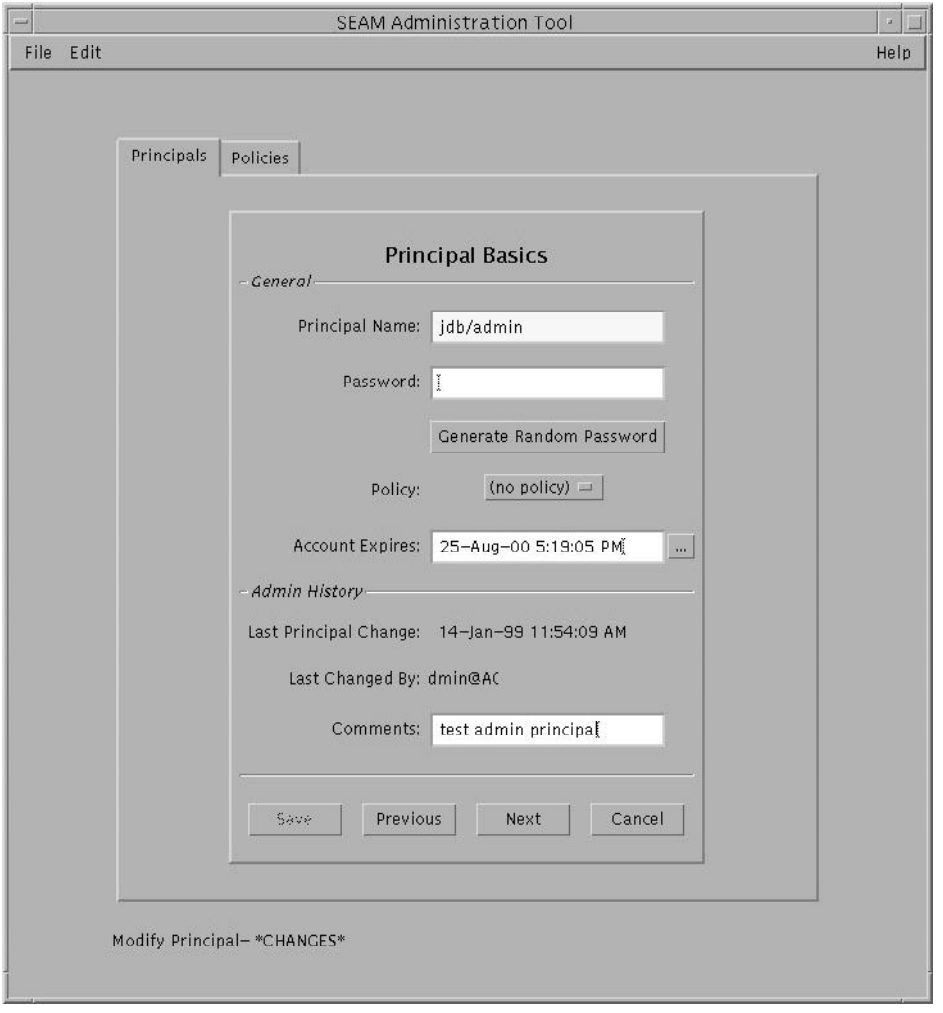
▼ 如何查看 Kerberos 主体属性

此过程后附等效命令行示例。

1. **如有必要，启动 SEAM Tool。**
   有关更多信息，请参见第 448 页中的 “如何启动 SEAM Tool”。
   ```
   $ /usr/sbin/gkadmin
   ```
2. **单击 "Principals" 选项卡。**
3. **在列表中选择要查看的主体，然后单击 "Modify"。**
   此时会显示包含该主体某些属性的 "Principal Basics" 面板。
4. **继续单击 "Next" 以查看该主体的所有属性。**
   有三个窗口包含属性信息。请从 "Help" 菜单中选择 "Context-Sensitive Help"，获取有关每个窗口中各种属性的信息。或者，转至第 472 页中的 “SEAM Tool 面板说明”，了解所有主体属性说明。
5. **查看完毕后，单击 "Cancel"。**

示例 24–2 查看 Kerberos 主体属性

以下示例显示了查看 jdb/admin 主体时的第一个窗口。



示例 24–3 查看 Kerberos 主体属性（命令行）

在以下示例中，kadmin 的 get_principal 命令用于查看 jdb/admin 主体的属性。

```
kadmin: getprinc jdb/admin

Principal: jdb/admin@EXAMPLE.COM

Expiration date: Fri Aug 25 17:19:05 PDT 2004

Last password change: [never]
```

```
Password expiration date: Wed Apr 14 11:53:10 PDT 2003

Maximum ticket life: 1 day 16:00:00

Maximum renewable life: 1 day 16:00:00

Last modified: Thu Jan 14 11:54:09 PST 2003 (admin/admin@EXAMPLE.COM)

Last successful authentication: [never]

Last failed authentication: [never]

Failed password attempts: 0

Number of keys: 1

Key: vno 1, DES cbc mode with CRC-32, no salt

Attributes: REQUIRES_HW_AUTH

Policy: [none]

kadmin: quit
```

▼ 如何创建新的 Kerberos 主体

此过程后附等效命令行示例。

1 **如有必要，启动 SEAM Tool。**
有关更多信息，请参见第 448 页中的 “如何启动 SEAM Tool”。

注 – 如果要创建一个可能需要新策略的新主体，则应在创建新主体之前创建新策略。请转至第 468 页中的 “如何创建新的 Kerberos 策略”。

```
$ /usr/sbin/gkadmin
```

2 **单击 "Principals" 选项卡。**

3 **单击 "New"。**
此时会显示包含某些主体属性的 "Principal Basics" 面板。

4 **指定主体名称和口令。**
必须提供主体名称和口令。

5 **指定该主体属性的值，然后继续单击 "Next" 以指定其他属性。**

有三个窗口包含属性信息。请从 "Help" 菜单中选择 "Context-Sensitive Help"，获取有关每个窗口中各种属性的信息。或者，转至第 472 页中的 “SEAM Tool 面板说明”，了解所有主体属性说明。

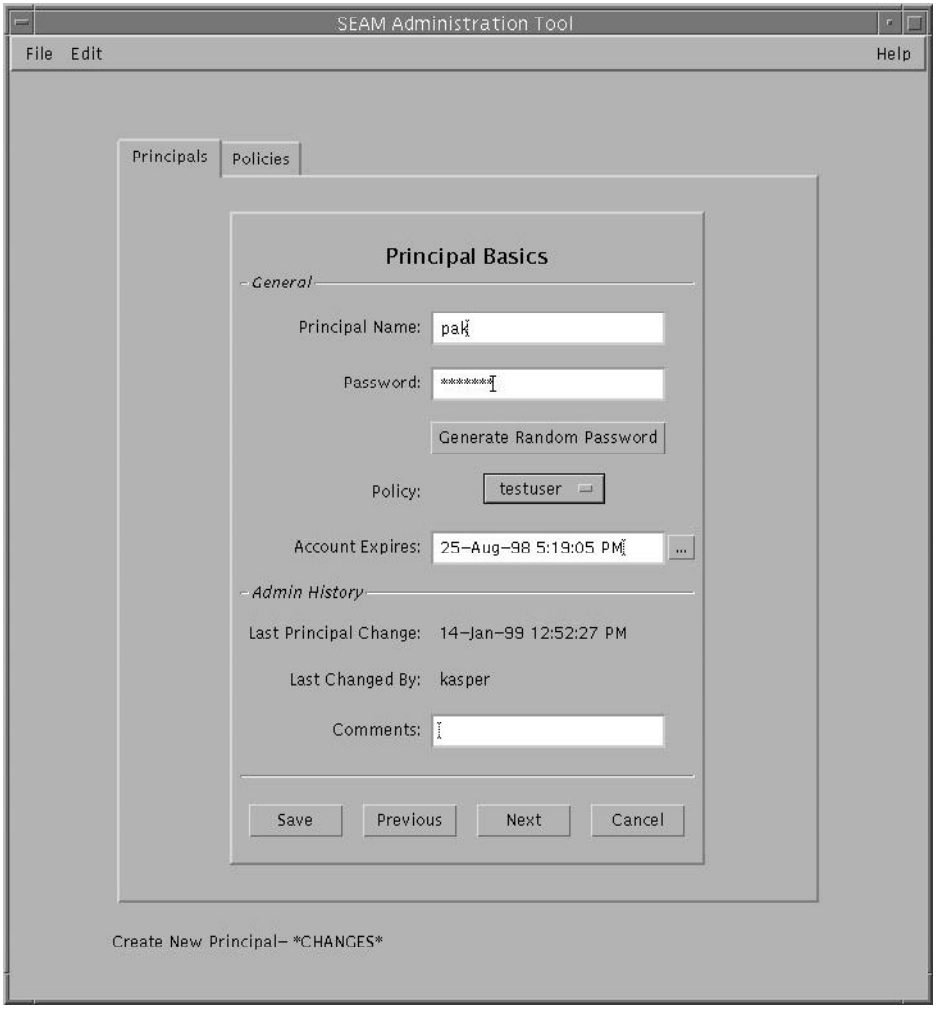
6 **单击 "Save" 以保存主体，或在最后一个面板上单击 "Done"。**

7 **如有必要，在 `/etc/krb5/kadm5.acl` 文件中为新主体设置 Kerberos 管理权限。**

有关更多详细信息，请参见第 462 页中的 “如何修改 Kerberos 管理权限”。

示例 24–4 创建新的 Kerberos 主体

以下示例显示了创建称为 `pak` 的新主体时的 "Principal Basics" 面板。该策略设置为 `testuser`。



示例 24–5 创建新的 Kerberos 主体（命令行）

在以下示例中，`kadmin` 的 `add_principal` 命令用于创建称为 `pak` 的新主体。该主体的策略设置为 `testuser`。

```
kadmin: add_principal -policy testuser pak

Enter password for principal "pak@EXAMPLE.COM": <Type the password>

Re-enter password for principal "pak@EXAMPLE.COM": <Type the password again>
```

```
Principal "pak@EXAMPLE.COM" created.

kadmin: quit
```

▼ 如何复制 Kerberos 主体

此过程说明如何使用某个现有主体的全部或部分属性来创建新主体。此过程没有等效命令行。

1. **如有必要，启动 SEAM Tool。**
   有关更多信息，请参见第 448 页中的 “如何启动 SEAM Tool”。
   ```
   $ /usr/sbin/gkadmin
   ```
2. **单击 "Principals" 选项卡。**
3. **在列表中选择要复制的主体，然后单击 "Duplicate"。**
   此时会显示 "Principal Basics" 面板。除空的 "Principal Name" 和 "Password" 字段之外，选定主体的其他属性都将被复制。
4. **指定主体名称和口令。**
   必须提供主体名称和口令。要完全复制选定主体，请单击 "Save" 并跳至步骤 7。
5. **指定该主体属性的其他值，然后继续单击 "Next" 以指定其他属性。**
   有三个窗口包含属性信息。请从 "Help" 菜单中选择 "Context-Sensitive Help"，获取有关每个窗口中各种属性的信息。或者，转至第 472 页中的 “SEAM Tool 面板说明”，了解所有主体属性说明。
6. **单击 "Save" 以保存主体，或在最后一个面板上单击 "Done"。**
7. **如有必要，在 `/etc/krb5/kadm5.acl` 文件中为主体设置 Kerberos 管理权限。**
   有关更多详细信息，请参见第 462 页中的 “如何修改 Kerberos 管理权限”。

▼ 如何修改 Kerberos 主体

此过程后附等效命令行示例。

1. **如有必要，启动 SEAM Tool。**
   有关更多信息，请参见第 448 页中的 “如何启动 SEAM Tool”。
   ```
   $ /usr/sbin/gkadmin
   ```
2. **单击 "Principals" 选项卡。**

3 **在列表中选择要修改的主体，然后单击 "Modify"。**

此时会显示包含该主体某些属性的 "Principal Basics" 面板。

4 **修改该主体的属性，然后继续单击 "Next" 以修改其他属性。**

有三个窗口包含属性信息。请从 "Help" 菜单中选择 "Context-Sensitive Help"，获取有关每个窗口中各种属性的信息。或者，转至第 472 页中的 “SEAM Tool 面板说明”，了解所有主体属性说明。

注 – 不能修改主体的名称。要重命名主体，必须首先复制该主体，为其指定一个新名称并保存，然后删除旧主体。

---

5 **单击 "Save" 按钮以保存主体，或在最后一个面板上单击 "Done"。**

6 **在 `/etc/krb5/kadm5.acl` 文件中，修改该主体的 Kerberos 管理权限。**

有关更多详细信息，请参见第 462 页中的 “如何修改 Kerberos 管理权限”。

示例 24–6 修改 Kerberos 主体口令（命令行）

在以下示例中，`kadmin` 的 `change_password` 命令用于修改 `jdb` 主体的口令。`change_password` 命令不允许将口令更改为主体口令历史记录中的口令。

```
kadmin: change_password jdb

Enter password for principal "jdb": <Type the new password>

Re-enter password for principal "jdb": <Type the password again>

Password for "jdb@EXAMPLE.COM" changed.

kadmin: quit
```

要修改主体的其他属性，必须使用 `kadmin` 的 `modify_principal` 命令。

▼ 如何删除 Kerberos 主体

此过程后附等效命令行示例。

1 **如有必要，启动 SEAM Tool。**

有关更多信息，请参见第 448 页中的 “如何启动 SEAM Tool”。

```
$ /usr/sbin/gkadmin
```

2 **单击 "Principals" 选项卡。**

3. **在列表中选择要删除的主体，然后单击 "delete"。**
   确认删除后，将删除该主体。

4. **从 Kerberos 访问控制列表 (access control list, ACL) 文件 `/etc/krb5/kadm5.acl` 中删除该主体。**
   有关更多详细信息，请参见第 462 页中的 “如何修改 Kerberos 管理权限”。

示例 24–7 删除 Kerberos 主体（命令行）

在以下示例中，`kadmin` 的 `delete_principal` 命令用于删除 `jdb` 主体。

```
kadmin: delete_principal pak

Are you sure you want to delete the principal "pak@EXAMPLE.COM"? (yes/no): yes

Principal "pak@EXAMPLE.COM" deleted.

Make sure that you have removed this principal from all ACLs before reusing.

kadmin: quit
```

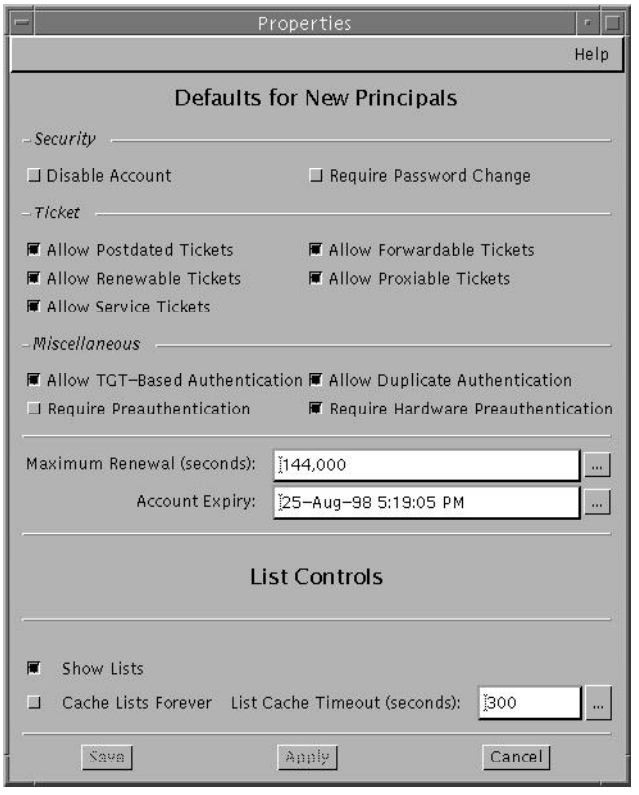
▼ 如何设置缺省值以创建新的 Kerberos 主体

此过程没有等效命令行。

1. **如有必要，启动 SEAM Tool。**
   有关更多信息，请参见第 448 页中的 “如何启动 SEAM Tool”。

   ```
   $ /usr/sbin/gkadmin
   ```

2. **从 "Edit" 菜单中选择 "Properties"。**
   此时会显示 "Properties" 窗口。



3 **选择要在创建新主体时使用的缺省值。**

请从 "Help" 菜单中选择 "Context-Sensitive Help"，以获取有关每个窗口中各种属性的信息。

4 **单击 "Save"。**

▼ 如何修改 Kerberos 管理权限

尽管您的站点可能有许多用户主体，但您通常只希望一小部分用户能够管理 Kerberos 数据库。管理 Kerberos 数据库的权限由 Kerberos 访问控制列表 (access control list, ACL) 文件 `kadm5.acl` 确定。通过 `kadm5.acl` 文件，可以允许或禁用各个主体的权限。或者，可在主体名称中使用 "*" 通配符来指定主体组的权限。

1 **成为主 KDC 的超级用户。**

2 **编辑 `/etc/krb5/kadm5.acl` 文件。**

`kadm5.acl` 文件中的项必须使用以下格式：

principal privileges [*principal-target*]

principal 指定要为其授予权限的主体。主体名称的任何部分都可以包含 "*" 通配符，这在为一组主体提供相同权限时很有用。例如，如果要指定包含 admin 实例的所有主体，则可使用 */admin@*realm*。

请注意，admin 实例常用于将单独的权限（如对 Kerberos 数据库的管理访问权限）授予单独的 Kerberos 主体。例如，用户 jdb 可能具有用于管理的主体 jdb/admin。这样，用户 jdb 便仅在实际需要使用这些权限时，才会获取 jdb/admin 票证。

privileges 指定主体能够执行和不能执行的操作。此字段由一个或多个下列字符或其对应大写形式的字符组成。如果字符为大写形式（或未指定），则不允许执行该操作。如果字符为小写形式，则允许执行该操作。

a	[不]允许添加主体或策略。
d	[不]允许删除主体或策略。
m	[不]允许修改主体或策略。
c	[不]允许更改主体的口令。
i	[不]允许查询 Kerberos 数据库。
l	[不]允许列出 Kerberos 数据库中的主体或策略。
x 或 *	允许所有权限 (admcil)。

principal-target 在此字段中指定主体时，*privileges* 仅在 *principal* 对 *principal-target* 进行操作时，才应用于 *principal*。主体名称的任何部分都可以包含 "*" 通配符，这在对主体分组时很有用。

示例 24–8 修改 Kerberos 管理权限

kadm5.acl 文件中的以下项授予 EXAMPLE.COM 领域中包含 admin 实例的任何主体对 Kerberos 数据库的所有权限：

```
*/admin@EXAMPLE.COM *
```

kadm5.acl 文件中的以下项授予 jdb@EXAMPLE.COM 主体添加、列出和查询包含 root 实例的任何主体的权限。

```
jdb@EXAMPLE.COM ali */root@EXAMPLE.COM
```

管理 Kerberos 策略

本节提供使用 SEAM Tool 管理策略的逐步说明，还提供等效命令行示例（如果有）。

管理 Kerberos 策略（任务列表）

任务	说明	参考
查看策略列表。	通过单击 "Policies" 选项卡来查看策略列表。	第 464 页中的 “如何查看 Kerberos 策略列表”
查看策略属性。	通过在 "Policy List" 中选择策略，然后单击 "Modify" 按钮来查看策略的属性。	第 466 页中的 “如何查看 Kerberos 策略属性”
创建新策略。	通过单击 "Policy List" 面板中的 "Create New" 按钮来创建新策略。	第 468 页中的 “如何创建新的 Kerberos 策略”
复制策略。	通过在 "Policy List" 中选择要复制的策略，然后单击 "Duplicate" 按钮来复制策略。	第 470 页中的 “如何复制 Kerberos 策略”
修改策略。	通过在 "Policy List" 中选择要修改的策略，然后单击 "Modify" 按钮来修改策略。 请注意，不能修改策略的名称。要重命名策略，必须首先复制该策略，为其指定一个新名称并保存，然后删除旧策略。	第 470 页中的 “如何修改 Kerberos 策略”
删除策略。	通过在 "Policy List" 中选择要删除的策略，然后单击 "Delete" 按钮来删除策略。	第 471 页中的 “如何删除 Kerberos 策略”

▼ 如何查看 Kerberos 策略列表

此过程后附等效命令行示例。

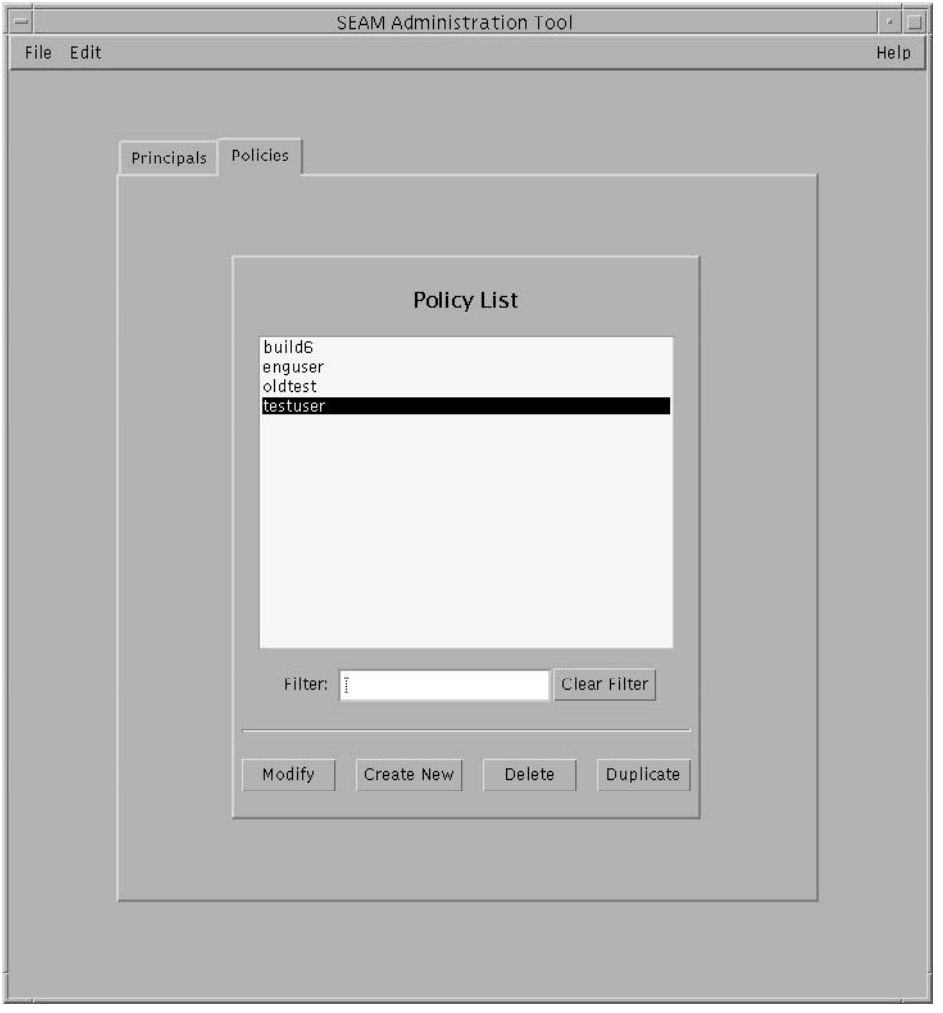
1 **如有必要，启动 SEAM Tool。**

有关更多信息，请参见第 448 页中的 “如何启动 SEAM Tool” 。

```
$ /usr/sbin/gkadmin
```

2 **单击 "Policies" 选项卡。**

此时会显示策略列表。



3 显示特定策略或策略子列表。

在 "Filter" 字段中键入过滤字符串，然后按 "Return"。如果过滤操作成功，则会显示与过滤器匹配的策略列表。

过滤字符串必须由一个或多个字符组成。由于过滤机制区分大小写，因此需要对过滤器使用正确的大小写字母。例如，如果键入过滤字符串 `ge`，则过滤机制仅显示包含 `ge` 字符串的策略（如 `george` 或 `edge`）。

如果要显示策略的完整列表，请单击 "Clear Filter"。

示例 24–9 查看 Kerberos 策略列表（命令行）

在以下示例中，kadmin 的 list_policies 命令用于列出与 *user* 匹配的所有策略。通配符可与 list_policies 命令一起使用。

```
kadmin: list_policies *user*

testuser

enguser

kadmin: quit
```

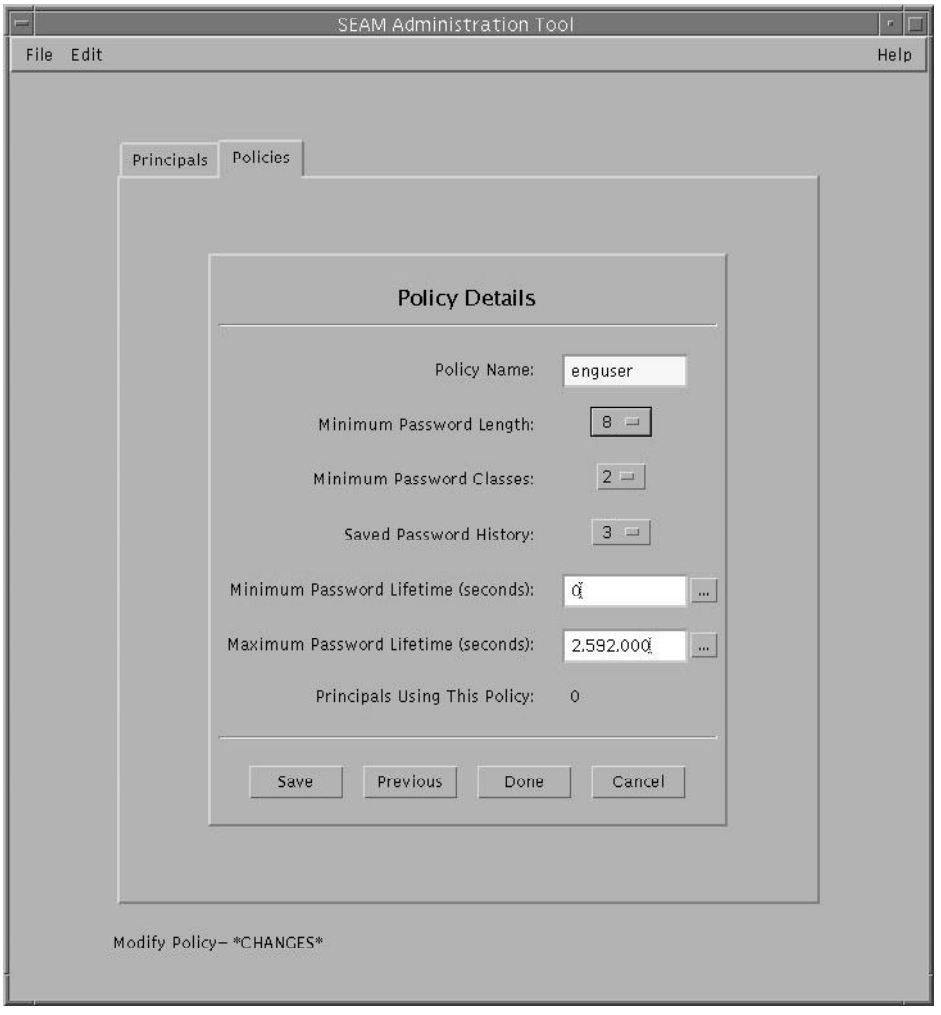
▼ 如何查看 Kerberos 策略属性

此过程后附等效命令行示例。

1. **如有必要，启动 SEAM Tool。**
   有关更多信息，请参见第 448 页中的 “如何启动 SEAM Tool”。
   ```
   $ /usr/sbin/gkadmin
   ```
2. **单击 "Policies" 选项卡。**
3. **在列表中选择要查看的策略，然后单击 "Modify"。**
   此时会显示 "Policy Details" 面板。
4. **查看完毕后，单击 "Cancel"。**

示例 24–10 查看 Kerberos 策略属性

以下示例显示了查看 test 策略时的 "Policy Details" 面板。



示例 24–11 查看 Kerberos 策略属性（命令行）

在以下示例中，`kadmin` 的 `get_policy` 命令用于查看 `enguser` 策略的属性。

```
kadmin: get_policy enguser

Policy: enguser

Maximum password life: 2592000

Minimum password life: 0
```

```
Minimum password length: 8

Minimum number of password character classes: 2

Number of old keys kept: 3

Reference count: 0

kadmin: quit
```

引用计数是使用此策略的主体数。

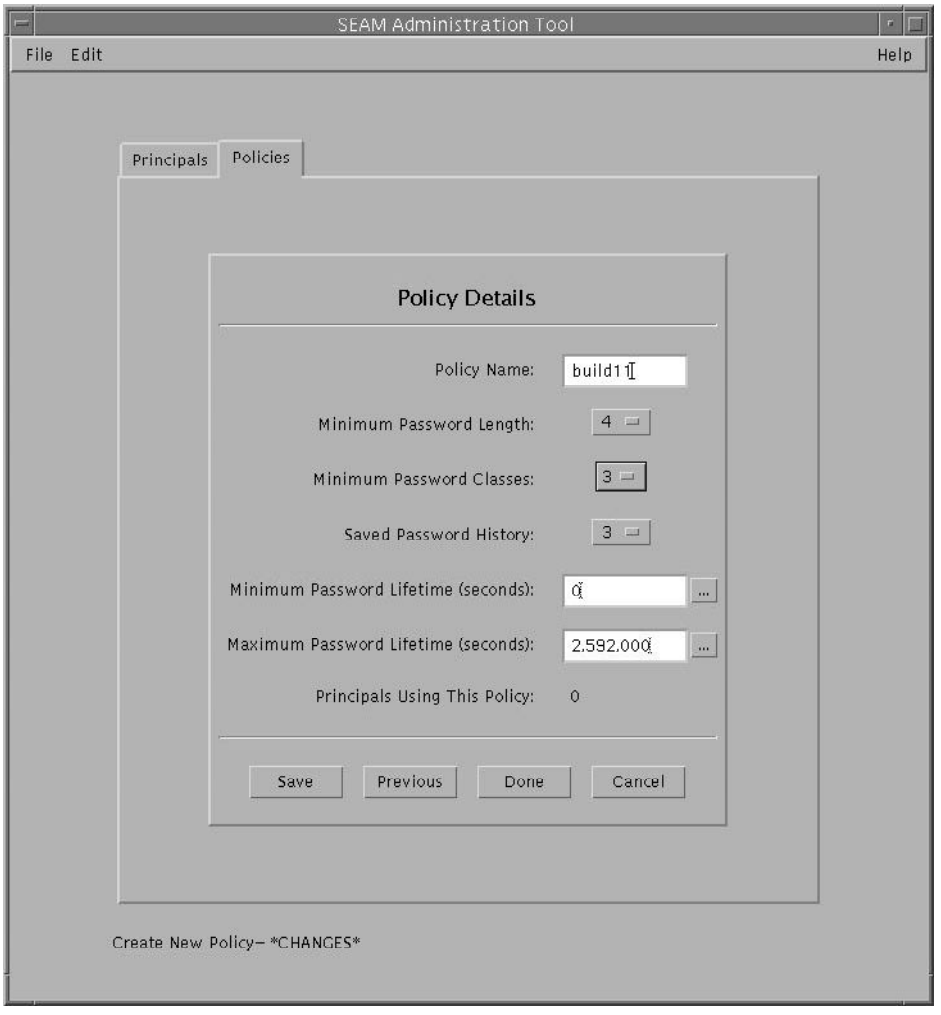
▼ 如何创建新的 Kerberos 策略

此过程后附等效命令行示例。

1. **如有必要，启动 SEAM Tool。**
   有关更多信息，请参见第 448 页中的 “如何启动 SEAM Tool”。
   ```
   $ /usr/sbin/gkadmin
   ```
2. **单击 "Policies" 选项卡。**
3. **单击 "New"。**
   此时会显示 "Policy Details" 面板。
4. **在 "Policy Name" 字段中指定策略的名称。**
   必须提供策略名称。
5. **指定策略属性的值。**
   请从 "Help" 菜单中选择 "Context-Sensitive Help"，获取有关此窗口中各种属性的信息。或者，转至表 24–5，了解所有策略属性说明。
6. **单击 "Save" 按钮以保存策略，或单击 "Done"。**

示例 24–12 创建新的 Kerberos 策略

在以下示例中，创建了一个称为 `build11` 的新策略。"Minimum Password Classes" 设置为 3。



示例 24–13 创建新的 Kerberos 策略（命令行）

在以下示例中，`kadmin` 的 `add_policy` 命令用于创建 `build11` 策略。此策略要求口令中至少有 3 类字符。

```
$ kadmin

kadmin: add_policy -minclasses 3 build11

kadmin: quit
```

▼ 如何复制 Kerberos 策略

此过程说明如何使用某个现有策略的全部或部分属性来创建新策略。此过程没有等效命令行。

1 **如有必要，启动 SEAM Tool。**

有关更多信息，请参见第 448 页中的 “如何启动 SEAM Tool”。

```
$ /usr/sbin/gkadmin
```

2 **单击 "Policies" 选项卡。**

3 **在列表中选择要复制的策略，然后单击 "Duplicate"。**

此时会显示 "Policy Details" 面板。除空的 "Policy Name" 字段以外，选定策略的其他所有属性都将被复制。

4 **在 "Policy Name" 字段中指定复制策略的名称。**

必须提供策略名称。要完全复制选定策略，请跳至步骤 6。

5 **指定策略属性的其他值。**

请从 "Help" 菜单中选择 "Context-Sensitive Help"，获取有关此窗口中各种属性的信息。或者，转至表 24–5，了解所有策略属性说明。

6 **单击 "Save" 按钮以保存策略，或单击 "Done"。**

▼ 如何修改 Kerberos 策略

此过程后附等效命令行示例。

1 **如有必要，启动 SEAM Tool。**

有关详细信息，请参见第 448 页中的 “如何启动 SEAM Tool”。

```
$ /usr/sbin/gkadmin
```

2 **单击 "Policies" 选项卡。**

3 **在列表中选择要修改的策略，然后单击 "Modify"。**

此时会显示 "Policy Details" 面板。

4 **修改该策略的属性。**

请从 "Help" 菜单中选择 "Context-Sensitive Help"，获取有关此窗口中各种属性的信息。或者，转至表 24–5，了解所有策略属性说明。

注 – 不能修改策略的名称。要重命名策略，必须首先复制该策略，为其指定一个新名称并保存，然后删除旧策略。

5 **单击 "Save" 按钮以保存策略，或单击 "Done"。**

示例 24–14 修改 Kerberos 策略（命令行）

在以下示例中，`kadmin` 的 `modify_policy` 命令用于将 `build11` 策略的最小口令长度修改为 5 个字符。

```
$ kadmin

kadmin: modify_policy -minlength 5 build11

kadmin: quit
```

▼ 如何删除 Kerberos 策略

此过程后附等效命令行示例。

注 – 删除某策略之前，必须从当前正在使用该策略的所有主体取消该策略。为此，需要修改这些主体的策略属性。如果有任何主体在使用该策略，则无法将其删除。

1 **如有必要，启动 SEAM Tool。**

有关更多信息，请参见第 448 页中的 “如何启动 SEAM Tool”。

```
$ /usr/sbin/gkadmin
```

2 **单击 "Policies" 选项卡。**

3 **在列表中选择要删除的策略，然后单击 "Delete"。**

确认删除后，将删除该策略。

示例 24–15 删除 Kerberos 策略（命令行）

在以下示例中，`kadmin` 的 `delete_policy` 命令用于删除 `build11` 策略。

```
kadmin: delete_policy build11

Are you sure you want to delete the policy "build11"? (yes/no): yes

kadmin: quit
```

删除某策略之前，必须从当前正在使用该策略的所有主体取消该策略。为此，需要对受影响的主体使用 kadmin 的 modify_principal - policy 命令。如果主体正在使用该策略，则 delete_policy 命令将会失败。

SEAM Tool 参考

本节为 SEAM Tool 中的每个面板提供说明。另外，还提供有关以受限权限使用 SEAM Tool 的信息。

SEAM Tool 面板说明

本节提供可在 SEAM Tool 中指定或查看的每个主体和策略属性的说明。这些属性按显示它们的面板进行组织。

表 24–2 SEAM Tool 的 "Principal Basics" 面板中的属性

属性	说明
Principal Name	主体的名称（全限定主体名称的 *primary*/*instance* 部分）。主体是 KDC 可以为其指定票证的唯一标识。 修改主体时不能编辑其名称。
Password	主体的口令。可使用 "Generate Random Password" 按钮为主体创建随机口令。
Policy	主体的可用策略菜单。
Account Expires	主体帐户的失效日期和时间。帐户失效后，主体就无法再获取票证授予票证 (Ticket-Granting Ticket, TGT)，并且可能无法登录。
Last Principal Change	上次修改主体信息的日期。（只读）
Last Changed By	上次更改此主体帐户的主体的名称。（只读）
注释	与主体有关的注释（如“临时帐户”）。

表 24–3 SEAM Tool 的 "Principal Details" 面板中的属性

属性	说明
Last Success	主体上次登录成功的日期和时间。（只读）
Last Failure	主体上次登录失败的日期和时间。（只读）
Failure Count	主体登录失败的次数。（只读）
Last Password Change	上次更改主体口令的日期和时间。（只读）
Password Expires	主体当前口令失效的日期和时间。

表 24–3 SEAM Tool 的 "Principal Details" 面板中的属性 （续）

属性	说明
Key Version	主体的密钥版本号。通常，只有在口令已泄漏的情况下才会更改此属性。
Maximum Lifetime (seconds)	可将票证授予主体的最长时间（不续用）。
Maximum Renewal (seconds)	主体可续用现有票证的最长时间。

表 24–4 SEAM Tool 的 "Principal Flags" 面板中的属性

属性（单选按钮）	说明
Disable Account	选中此项后，将禁止主体登录。此属性提供了一种临时冻结主体帐户的简便方法。
Require Password Change	选中此项后，将使主体的当前口令失效，这将会强制用户使用 `kpasswd` 命令来创建新口令。如果安全性被破坏，并且需要确保替换旧口令，则此属性很有用。
Allow Postdated Tickets	选中此项后，将允许主体获取以后生效的票证。 例如，如果 `cron` 作业必须在几小时后运行，但您又因为票证的生命周期不够长而无法提前获取票证，则可能需要对其使用以后生效的票证。
Allow Forwardable Tickets	选中此项后，将允许主体获取可转发的票证。 可转发的票证即可转发至远程主机以提供单点登录会话的票证。例如，如果使用可转发的票证并且通过 `ftp` 或 `rsh` 进行自我验证，则可使用其他服务（如 NFS 服务），而不会提示您输入其他口令。
Allow Renewable Tickets	选中此项后，将允许主体获取可续用的票证。 主体可以自动延长可续用票证的失效日期或时间，而不必在票证首次失效后获取新的票证。目前，NFS 服务是可以续用票证的票证服务。
Allow Proxiable Tickets	选中此项后，将允许主体获取可代理的票证。 可代理票证即可被服务以客户机名义执行客户机操作时使用的票证。借助可代理票证，服务可采用客户机的身份来获取其他服务的票证。但是，该服务不能获取票证授予票证 (Ticket-Granting Ticket, TGT)。
Allow Service Tickets	选中此项后，将允许为主体颁发服务票证。 不允许为 `kadmin/`*hostname* 和 `changepw/`*hostname* 主体颁发服务票证。此做法可确保只有这些主体才能更新 KDC 数据库。
Allow TGT-Based Authentication	选中此项后，将允许服务主体为其他主体提供服务。具体而言，此属性允许 KDC 为服务主体颁发服务票证。 此属性仅对服务主体有效。如果取消选中此项，将无法为服务主体颁发服务票证。
Allow Duplicate Authentication	选中此项后，将允许用户主体获取其他用户主体的服务票证。 此属性仅对用户主体有效。如果取消选中此项，用户主体将仍可获取服务主体的服务票证，但不能获取其他用户主体的服务票证。

表 24–4 SEAM Tool 的 "Principal Flags" 面板中的属性 （续）

属性（单选按钮）	说明
Required Preauthentication	选中此项后，KDC 在验证（通过软件）主体确为请求 TGT 的主体之前，不会将请求的票证授予票证 (Ticket-Granting Ticket, TGT) 发送给该主体。此预验证通常通过附加口令（如 DES 卡）完成。 如果取消选中此项，则 KDC 不必在向主体发送请求的 TGT 之前预先验证主体。
Required Hardware Authentication	选中此项后，KDC 在验证（通过硬件）主体确为请求 TGT 的主体之前，不会将请求的票证授予票证 (Ticket-Granting Ticket, TGT) 发送给该主体。例如，可对 Java 环形阅读器进行硬件预验证。 如果取消选中此项，则 KDC 不必在向主体发送请求的 TGT 之前预先验证主体。

表 24–5 SEAM Tool 的 "Policy Basics" 面板中的属性

属性	说明
Policy Name	策略的名称。策略是一组用于管理主体口令和票证的规则。 修改策略时不能编辑其名称。
Minimum Password Length	主体口令的最小长度。
Minimum Password Classes	主体口令中要求使用的最少不同字符类型数。 例如，最少类值为 2 表示口令必须至少使用两种不同的字符类型，如字母和数字 (hi2mom)。值为 3 表示口令必须至少使用三种不同的字符类型，如字母、数字和标点符号 (hi2mom!)。依此类推。 值为 1 则表示对口令字符类型数未设置任何限制。
Saved Password History	主体先前使用的口令数，以及无法重新使用的先前口令的列表。
Minimum Password Lifetime (seconds)	口令在可更改之前必须经历的最短时间。
Maximum Password Lifetime (seconds)	口令在必须更改之前可以经历的最长时间。
Principals Using This Policy	当前应用此策略的主体数。（只读）

以受限 Kerberos 管理权限使用 SEAM Tool

如果 `admin` 主体拥有管理 Kerberos 数据库的所有权限，则可使用 SEAM Administration Tool 的所有功能。但是，您的权限可能受到限制，如仅允许查看主体列表或更改主体口令。借助受限 Kerberos 管理权限，仍然可以使用 SEAM Tool。但是，SEAM Tool 的各个部分会基于未拥有的 Kerberos 管理权限而变化。表 24–6 显示了 SEAM Tool 基于 Kerberos 管理权限变化的情况。

没有列表权限时，SEAM Tool 会发生最直观的变化。如果没有列表权限，列表面板便不会显示供您处理的主体和策略列表。相反，您必须使用列表面板中的 "Name" 字段来指定要处理的主体或策略。

如果您登录到 SEAM Tool，但却没有足够的权限来使用它执行任务，则会显示以下消息并且会返回到 "SEAM Administration Login" 窗口：

```
Insufficient privileges to use gkadmin: ADMCIL. Please try using another principal.
```

要更改主体的权限以便它可管理 Kerberos 数据库，请转至第 462 页中的 “如何修改 Kerberos 管理权限”。

表 24–6 以受限 Kerberos 管理权限使用 SEAM Tool

禁用的权限	SEAM Tool 如何变化
a（添加）	"Principal List" 和 "Policy List" 面板中的 "Create New" 和 "Duplicate" 按钮不可用。如果没有添加权限，则无法创建新主体或策略，也不能复制它们。
d（删除）	"Principal List" 和 "Policy List" 面板中的 "Delete" 按钮不可用。如果没有删除权限，则无法删除主体或策略。
m（修改）	"Principal List" 和 "Policy List" 面板中的 "Modify" 按钮不可用。如果没有修改权限，则无法修改主体或策略。 而且，如果 "Modify" 按钮不可用，则即使您拥有更改口令的权限，也不能修改主体的口令。
c（更改口令）	"Principal Basics" 面板中的 "Password" 字段处于只读状态，无法更改。如果没有更改口令的权限，则无法修改主体的口令。 请注意，即使您拥有更改口令的权限，还必须同时拥有修改权限才能更改主体的口令。
i（查询数据库）	"Principal List" 和 "Policy List" 面板中的 "Modify" 和 "Duplicate" 按钮不可用。如果没有查询权限，则无法修改或复制主体或策略。 而且，如果 "Modify" 按钮不可用，则即使您拥有更改口令的权限，也不能修改主体的口令。
l（列出）	列表面板中的主体和策略列表不可用。如果没有列表权限，则必须使用列表面板中的 "Name" 字段来指定要处理的主体或策略。

管理密钥表文件

提供服务的每台主机都必须包含称为 *keytab*（**密钥表**）的本地文件，keytab（密钥表）是“key table（密钥表）”的缩写。密钥表包含相应服务的主体，称为**服务密钥**。服务使用服务密钥向 KDC 进行自我验证，并且只有 Kerberos 和服务本身知道服务密钥。例如，如果您有基于 Kerberos 的 NFS 服务器，则该服务器必须具有包含其 `nfs` 服务主体的密钥表文件。

要将服务密钥添加至密钥表文件，应使用 `kadmin` 的 `ktadd` 命令，将相应的服务主体添加至主机的密钥表文件。由于要将服务主体添加至密钥表文件，因此该主体必须已存在于 Kerberos 数据库中，以便 `kadmin` 可验证其存在。在主 KDC 上，密钥表文件的缺省位置为：`/etc/krb5/kadm5.keytab`。在提供基于 Kerberos 的服务的应用程序服务器上，密钥表文件的缺省位置为：`/etc/krb5/krb5.keytab`。

密钥表类似于用户的口令。正如用户保护其口令很重要一样，应用程序服务器保护其密钥表文件同样也很重要。应始终将密钥表文件存储在本地磁盘上，并且只允许 `root` 用户读取这些文件。另外，绝不要通过不安全的网络发送密钥表文件。

还有一种特殊情况需要将 `root` 主体添加至主机的密钥表文件。如果希望 Kerberos 客户机用户挂载基于 Kerberos 的 NFS 文件系统（要求与超级用户等效的权限），则必须将客户机的 `root` 主体添加至客户机的密钥表文件。否则，每当用户要使用 `root` 权限挂载基于 Kerberos 的 NFS 文件系统时，即使正在使用自动挂载程序，也必须以 `root` 身份使用 `kinit` 命令来获取客户机 `root` 主体的凭证。

注 – 设置主 KDC 时，需要将 `kadmind` 和 `changepw` 主体添加至 `kadm5.keytab` 文件。

可用于管理密钥表文件的另一个命令是 `ktutil` 命令。使用此交互式命令，可在没有 Kerberos 管理权限的情况下管理本地主机的密钥表文件，因为 `ktutil` 不会像 `kadmin` 那样与 Kerberos 数据库交互，因此，将主体添加至密钥表文件后，可使用 `ktutil` 来查看密钥表文件中的密钥列表，或临时禁用对服务的验证。

注 – 使用 `kadmin` 中的 `ktadd` 命令更改密钥表文件中的主体时，将生成一个新的密钥并将其添加至密钥表文件。

管理密钥表文件（任务列表）

任务	说明	参考
将服务主体添加至密钥表文件。	使用 `kadmin` 的 `ktadd` 命令将服务主体添加至密钥表文件。	第 477 页中的 “如何将 Kerberos 服务主体添加至密钥表文件”
从密钥表文件中删除服务主体。	使用 `kadmin` 的 `ktremove` 命令从密钥表文件中删除服务主体。	第 479 页中的 “如何从密钥表文件中删除服务主体”

任务	说明	参考
显示密钥表文件中的密钥列表（主体列表）。	使用 ktutil 命令来显示密钥表文件中的密钥列表。	第 480 页中的 “如何显示密钥表文件中的密钥列表（主体）”
临时禁用对主机上的服务的验证。	此过程可以快速地临时禁用对主机上的服务的验证，而不需要 kadmin 权限。 使用 ktutil 从服务器的密钥表文件中删除服务主体之前，应将原始密钥表文件复制到一个临时位置。如果要再次启用该服务，请将原始密钥表文件复制回其相应的位置。	第 481 页中的 “如何临时禁用对主机上的服务的验证”

▼ 如何将 Kerberos 服务主体添加至密钥表文件

1 **确保 Kerberos 数据库中已存在该主体。**

有关更多信息，请参见第 452 页中的 “如何查看 Kerberos 主体列表” 。

2 **成为需要将主体添加至其密钥表文件的主机的超级用户。**

3 **启动 kadmin 命令。**

```
# /usr/sbin/kadmin
```

4 **使用 ktadd 命令将主体添加至密钥表文件。**

```
kadmin: ktadd [-e enctype] [-k keytab] [-q] [principal | -glob principal-exp]
```

-e *enctype*	覆盖 krb5.conf 文件中定义的加密类型列表。
-k *keytab*	指定密钥表文件。缺省情况下，使用 /etc/krb5/krb5.keytab。
-q	显示简要信息。
principal	指定要添加至密钥表文件的主体。可以添加以下服务主体：host、root、nfs 和 ftp。
-glob *principal-exp*	指定主体表达式。与 *principal-exp* 匹配的所有主体都将添加至密钥表文件。主体表达式的规则与 kadmin 的 list_principals 命令的规则相同。

5 **退出 kadmin 命令。**

```
kadmin: quit
```

示例 24–16 将服务主体添加至密钥表文件

在以下示例中，kadmin/admin 和 kadmin/changepw 主体被添加至主 KDC 的密钥表文件。对于该示例，密钥表文件必须是在 kdc.conf 文件中指定的文件。

```
kdc1 # /usr/sbin/kadmin.local

kadmin.local: ktadd -k /etc/krb5/kadm5.keytab kadmin/admin kadmin/changepw

EnEntry for principal kadmin/admin@example.com with kvno 3, encryption type AES-128 CTS mode

          with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/kadm5.keytab.

Entry for principal kadmin/admin@example.com with kvno 3, encryption type Triple DES cbc

          mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/kadm5.keytab.

Entry for principal kadmin/admin@example.com with kvno 3, encryption type ARCFOUR

          with HMAC/md5 added to keytab WRFILE:/etc/krb5/kadm5.keytab.

Entry for principal kadmin/admin@example.com with kvno 3, encryption type DES cbc mode

          with RSA-MD5 added to keytab WRFILE:/etc/krb5/kadm5.keytab.

Entry for principal kadmin/changepw@example.com with kvno 3, encryption type AES-128 CTS

          mode with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/kadm5.keytab.

Entry for principal kadmin/changepw@example.com with kvno 3, encryption type Triple DES cbc

          mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/kadm5.keytab.

Entry for principal kadmin/changepw@example.com with kvno 3, encryption type ARCFOUR

          with HMAC/md5 added to keytab WRFILE:/etc/krb5/kadm5.keytab.

Entry for principal kadmin/changepw@example.com with kvno 3, encryption type DES cbc mode

          with RSA-MD5 added to keytab WRFILE:/etc/krb5/kadm5.keytab.

kadmin.local: quit
```

在以下示例中，denver 的 host 主体被添加至 denver 的密钥表文件，以便 KDC 验证 denver 的网络服务。

```
denver # /usr/sbin/kadmin

kadmin: ktadd host/denver@example.com@EXAMPLE.COM

Entry for principal host/denver@example.com with kvno 3, encryption type AES-128 CTS mode
```

```
          with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal host/denver@example.com with kvno 3, encryption type Triple DES cbc mode
          with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal host/denver@example.com with kvno 3, encryption type ARCFOUR
          with HMAC/md5 added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal host/denver@example.com with kvno 3, encryption type DES cbc mode
          with RSA-MD5 added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin: quit
```

▼ 如何从密钥表文件中删除服务主体

1. **成为包含必须从其密钥表文件中删除的服务主体的主机的超级用户。**

2. **启动** `kadmin` **命令。**

   ```
   # /usr/sbin/kadmin
   ```

3. **（可选的）要显示密钥表文件中的当前主体（密钥）列表，请使用** `ktutil` **命令。**
   有关详细说明，请参见第 480 页中的 “如何显示密钥表文件中的密钥列表（主体）”。

4. **使用** `ktremove` **命令从密钥表文件中删除主体。**

   ```
   kadmin: ktremove [-k keytab] [-q] principal [kvno | all | old ]
   ```

`-k` *keytab*	指定密钥表文件。缺省情况下，使用 `/etc/krb5/krb5.keytab`。
`-q`	显示简要信息。
principal	指定要从密钥表文件中删除的主体。
kvno	删除密钥版本号与 *kvno* 匹配的指定主体的所有项。
`all`	删除指定主体的所有项。
`old`	删除指定主体（具有最高密钥版本号的主体除外）的所有项。

5. **退出** `kadmin` **命令。**

   ```
   kadmin: quit
   ```

示例 24–17 从密钥表文件中删除服务主体

在以下示例中，从 denver 的密钥表文件中删除了 denver 的 host 主体。

```
denver # /usr/sbin/kadmin

kadmin: ktremove host/denver.example.com@EXAMPLE.COM

kadmin: Entry for principal host/denver.example.com@EXAMPLE.COM with kvno 3

  removed from keytab WRFILE:/etc/krb5/krb5.keytab.

kadmin: quit
```

▼ 如何显示密钥表文件中的密钥列表（主体）

1 成为包含密钥表文件的主机的超级用户。

注 – 尽管可以创建由其他用户拥有的密钥表文件，但使用密钥表文件的缺省位置需要 root 拥有权。

---

2 启动 ktutil 命令。

```
# /usr/bin/ktutil
```

3 使用 read_kt 命令将密钥表文件读入密钥列表缓冲区。

```
ktutil: read_kt keytab
```

4 使用 list 命令显示密钥列表缓冲区。

```
ktutil: list
```

此时会显示当前的密钥列表缓冲区。

5 退出 ktutil 命令。

```
ktutil: quit
```

示例 24–18 显示密钥表文件中的密钥列表（主体）

以下示例显示了 denver 主机的 /etc/krb5/krb5.keytab 文件中的密钥列表。

```
denver # /usr/bin/ktutil

    ktutil: read_kt /etc/krb5/krb5.keytab
```

```
    ktutil: list

slot KVNO Principal

---- ---- -------------------------------------

   1    5 host/denver@EXAMPLE.COM

    ktutil: quit
```

▼ 如何临时禁用对主机上的服务的验证

有时可能需要在网络应用程序服务器上，临时禁用对服务（如 rlogin 或 ftp）的验证机制。例如，可能希望在执行维护过程时禁止用户登录到系统。使用 ktutil 命令，可以通过从服务器的密钥表文件中删除服务主体来完成此任务，而不需要 kadmin 权限。要再次启用验证，只需要将保存的原始密钥表文件复制回其原始位置。

注 – 缺省情况下，大多数服务都被设置为要求验证。如果某服务未设置为要求验证，则即使对该服务禁用验证，该服务仍然会运行。

---

1. **成为包含密钥表文件的主机的超级用户。**

注 – 尽管可以创建由其他用户拥有的密钥表文件，但使用密钥表文件的缺省位置需要 root 拥有权。

---

2. **将当前密钥表文件保存到临时文件。**

3. **启动 ktutil 命令。**

   ```
   # /usr/bin/ktutil
   ```

4. **使用 read_kt 命令将密钥表文件读入密钥列表缓冲区。**

   ```
   ktutil: read_kt keytab
   ```

5. **使用 list 命令显示密钥列表缓冲区。**

   ```
   ktutil: list
   ```

   此时会显示当前的密钥列表缓冲区。请注意要禁用的服务的槽号。

6. **要临时禁用主机的服务，请使用 delete_entry 命令从密钥列表缓冲区中删除特定的服务主体。**

   ```
   ktutil: delete_entry slot-number
   ```

   其中，*slot-number* 指定要删除的服务主体的槽号，可使用 list 命令来显示它。

7. **使用 `write_kt` 命令，将密钥列表缓冲区写入新的密钥表文件。**

   ```
   ktutil: write_kt new-keytab
   ```

8. **退出 `ktutil` 命令。**

   ```
   ktutil: quit
   ```

9. **移动新的密钥表文件。**

   ```
   # mv new-keytab keytab
   ```

10. **如果要再次启用该服务，请将临时（原始）密钥表文件复制回其原始位置。**

示例 24–19 临时禁用主机上的服务

在以下示例中，临时禁用了 `denver` 主机上的 `host` 服务。要重新启用 `denver` 上的主机服务，应将 `krb5.keytab.temp` 文件复制到 `/etc/krb5/krb5.keytab` 文件中。

```
denver # cp /etc/krb5/krb5.keytab /etc/krb5/krb5.keytab.temp

denver # /usr/bin/ktutil

    ktutil:read_kt /etc/krb5/krb5.keytab

    ktutil:list

slot KVNO Principal

---- ---- ---------------------------------------

   1    8 root/denver@EXAMPLE.COM

   2    5 host/denver@EXAMPLE.COM

    ktutil:delete_entry 2

    ktutil:list

slot KVNO Principal

---- ---- ---------------------------------------

   1    8 root/denver@EXAMPLE.COM

    ktutil:write_kt /etc/krb5/new.krb5.keytab

    ktutil: quit
```

```
denver # cp /etc/krb5/new.krb5.keytab /etc/krb5/krb5.keytab
```

◆ ◆ ◆ 第 25 章

使用 Kerberos 应用程序（任务）

本章适用于其系统中配置了 Kerberos 服务的人员。本章介绍如何使用提供的基于 Kerberos 的命令和服务。阅读本章中的这些命令之前，您应该对非基于 Kerberos 的版本中的这些命令比较熟悉。

由于本章适用于一般读者，因此其中包含有关票证的信息：获取、查看和销毁票证。本章还包含有关选择或更改 Kerberos 口令的信息。

以下是本章中信息的列表：

- 第 485 页中的 “Kerberos 票证管理”
- 第 489 页中的 “Kerberos 口令管理”
- 第 494 页中的 “Kerberos 用户命令”

有关 Solaris Kerberos 产品的概述，请参见第 20 章。

Kerberos 票证管理

本节介绍如何获取、查看和销毁票证。有关票证的介绍，请参见第 346 页中的 “Kerberos 服务的工作方式” 。

是否需要担心票证？

安装任何 SEAM 发行版或 Solaris 10 发行版后，Kerberos 便内置在 `login` 命令中，并且您将在登录时自动获取票证。通常，由于会将基于 Kerberos 的命令 `rsh`、`rcp`、`rdist`、`telnet` 和 `rlogin` 设置为将票证副本转发到其他计算机，因此您不必显式请求票证来访问这些计算机。配置中可能不包括此自动转发，但这是缺省行为。有关转发票证的更多信息，请参见第 495 页中的 “基于 Kerberos 的命令概述”和第 497 页中的 “转发 Kerberos 票证” 。

有关票证生命周期的信息，请参见第 506 页中的 “票证生命周期” 。

创建 Kerberos 票证

通常，如果 PAM 配置正确，则会在登录时自动创建票证，并且无需执行任何特殊操作即可获取票证。但是，如果票证到期，则可能需要创建票证。另外，可能还需要使用缺省主体以外的其他主体，例如，如果使用 `rlogin -l` 以其他人的身份登录到计算机。

要创建票证，请使用 `kinit` 命令。

```
% /usr/bin/kinit
```

`kinit` 命令将提示您输入口令。有关 `kinit` 命令的完整语法，请参见 `kinit`(1) 手册页。

示例－创建 Kerberos 票证

本示例说明用户 `jennifer` 如何在自己的系统上创建票证。

```
% kinit

Password for jennifer@ENG.EXAMPLE.COM:       <Type password>
```

在以下示例中，用户 `david` 使用 `-l` 选项创建了一个有效期为三个小时的票证。

```
% kinit -l 3h david@EXAMPLE.ORG

Password for david@EXAMPLE.ORG:       <Type password>
```

本示例说明用户 `david` 如何使用 `-f` 选项为其自身创建可转发票证。例如，该用户可以使用此可转发票证登录到第二个系统，然后 `telnet` 到第三个系统。

```
% kinit -f david@EXAMPLE.ORG

Password for david@EXAMPLE.ORG:          <Type password>
```

有关转发票证如何工作的更多信息，请参见第 497 页中的 “转发 Kerberos 票证”和第 505 页中的 “票证类型”。

查看 Kerberos 票证

并非所有票证都相同。例如，一个票证可能是**可转发**票证，另一个票证则可能是**以后生效的**票证，而第三个票证既可能是可转发票证，又可能是以后生效的票证。使用带有 `-f` 选项的 `klist` 命令，可以查看所拥有的票证以及这些票证的属性：

```
% /usr/bin/klist -f
```

以下符号表示与每个票证关联的属性，如 `klist` 输出所示：

A
　已预验证

D
　可以后生效

d
　以后生效

F
　可转发

f
　已转发

I
　初始

i
　无效

P
　可代理

p
　代理

R
　可更新

第 505 页中的 “票证类型”介绍了票证可以具有的各种属性。

示例－查看 Kerberos 票证

本示例说明用户 `jennifer` 拥有一个**初始**票证，该票证是**可转发** (F) 和**以后生效的** (d) 票证，但尚未经过验证 (i)。

```
% /usr/bin/klist -f

Ticket cache: /tmp/krb5cc_74287

Default principal: jennifer@ENG.EXAMPLE.COM
```

```
Valid starting                Expires                Service principal

09 Mar 04 15:09:51  09 Mar 04 21:09:51  nfs/EXAMPLE.SUN.COM@EXAMPLE.SUN.COM

        renew until 10 Mar 04 15:12:51, Flags: Fdi
```

以下示例说明用户 david 拥有两个从另一台主机**转发** (f) 到其主机的票证。这些票证也是**可转发** (F) 票证。

```
% klist -f

Ticket cache: /tmp/krb5cc_74287

Default principal: david@EXAMPLE.SUN.COM


Valid starting                Expires                Service principal

07 Mar 04 06:09:51  09 Mar 04 23:33:51  host/EXAMPLE.COM@EXAMPLE.COM

        renew until 10 Mar 04 17:09:51, Flags: fF


Valid starting                Expires                Service principal

08 Mar 04 08:09:51  09 Mar 04 12:54:51  nfs/EXAMPLE.COM@EXAMPLE.COM

        renew until 10 Mar 04 15:22:51, Flags: fF
```

以下示例说明如何使用 -e 选项显示会话密钥和票证的加密类型。如果名称服务可以执行转换操作，则可使用 -a 选项将主机地址映射至主机名。

```
% klist -fea

Ticket cache: /tmp/krb5cc_74287

Default principal: david@EXAMPLE.SUN.COM


Valid starting                Expires                Service principal
```

```
07 Mar 04 06:09:51  09 Mar 04 23:33:51  krbtgt/EXAMPLE.COM@EXAMPLE.COM

        renew until 10 Mar 04 17:09:51, Flags: FRIA

        Etype(skey, tkt): DES cbc mode with RSA-MD5, DES cbc mode with CRC-32

        Addresses: client.example.com
```

销毁 Kerberos 票证

如果要销毁在当前会话期间获取的所有 Kerberos 票证，请使用 `kdestroy` 命令。该命令可销毁凭证高速缓存，从而销毁所有凭证和票证。虽然通常不必销毁凭证高速缓存，但运行 `kdestroy` 可减少您未登录期间凭证高速缓存遭受破坏的机会。

要销毁票证，请使用 `kdestroy` 命令。

```
% /usr/bin/kdestroy
```

`kdestroy` 命令将销毁**所有**票证。不能使用此命令来有选择性地销毁特定票证。

如果要离开系统而又担心入侵者会使用您的权限，则应使用 `kdestroy` 或用于锁定屏幕的屏幕保护程序。

Kerberos 口令管理

配置 Kerberos 服务后，您即会拥有两个口令：常规 Solaris 口令和 Kerberos 口令。可以将这两个口令设置为相同，也可以不同。

口令选择建议

口令几乎可以包括能够键入的任何字符，但 Ctrl 键和回车键除外。好口令是易于记忆而其他人不容易猜到的口令。以下是一些不合适的口令示例：

- 可在字典中找到的单词
- 任何常见名称或通俗名称
- 名人的姓名或字符
- 您的姓名或用户名的任何形式（例如：反向拼写您的姓名、姓名重复两次等）
- 配偶姓名、子女姓名或宠物名称
- 您的生日或亲戚的生日
- 您的社会安全号、驾照号、护照号或其他类似的身份标识号
- 本手册或任何其他手册中出现的任何口令样例

一个好的口令的长度至少为八个字符。此外，口令还应包含混合字符，如大小写字母、数字和标点符号。以下是一些好的口令示例（如果未出现在本手册中）：

- 首字母缩略词，如 "I2LMHinSF"（全称为 "I too left my heart in San Francisco"）
- 发音容易的无意义单词，如 "WumpaBun" 或 "WangDangdoodle!"
- 故意拼错的短语，如 "6o'cluck" 或 "RrriotGrrrlsRrrule!"

注意 – 请勿使用这些示例。手册中出现的口令是入侵者将首先尝试的口令。

更改口令

如果 PAM 配置正确，则可以采用以下两种方法来更改 Kerberos 口令：

- 使用常见的 UNIX `passwd` 命令。配置 Kerberos 服务后，Solaris `passwd` 命令还会自动提示您输入新的 Kerberos 口令。

  使用 `passwd` 而非 `kpasswd` 的优点在于可以同时设置 UNIX 口令和 Kerberos 口令。但是，在一般情况下，**无需**使用 `passwd` 同时更改这两个口令。通常，只能更改 UNIX 口令并保持 Kerberos 口令不变，反之亦然。

  注 – `passwd` 的行为取决于 PAM 模块的配置方式。在某些配置中，可能会要求您同时更改这两个口令。对于一些站点，必须更改 UNIX 口令，而对于其他站点，则要求更改 Kerberos 口令。

- 使用 `kpasswd` 命令。`kpasswd` 与 `passwd` 非常类似。这两个命令的一个区别是 `kpasswd` 仅更改 Kerberos 口令。如果要更改 UNIX 口令，则必须使用 `passwd`。

  另一个区别是 `kpasswd` 可以更改非有效 UNIX 用户的 Kerberos 主体的口令。例如，`david/admin` 是 Kerberos 主体，但不是实际的 UNIX 用户，因此必须使用 `kpasswd` 而非 `passwd`。

更改口令后，所做更改在系统中传播需要一些时间（尤其是通过大型网络传播）。此延迟可能需要几分钟到一个小时或更长时间，具体取决于系统的设置方式。如果需要在更改口令后立刻获取新的 Kerberos 票证，请首先尝试新口令。如果新口令无效，请使用旧口令重试。

通过 Kerberos V5 协议，系统管理员可以设置允许每个用户使用的口令的条件。此类条件由为每个用户设置的**策略**（或缺省策略）定义。有关策略的更多信息，请参见第 463 页中的 "管理 Kerberos 策略"。

例如，假定用户 `jennifer` 的策略（称为 `jenpol`）要求口令长度至少为八个字母，并且至少由两种类型的字符混合组成。这样，`kpasswd` 便会拒绝尝试使用 "sloth" 作为口令。

```
% kpasswd

kpasswd: Changing password for jennifer@ENG.EXAMPLE.COM.

Old password:          <Jennifer types her existing password>
```

```
kpasswd: jennifer@ENG.EXAMPLE.COM's password is controlled by

the policy jenpol

which requires a minimum of 8 characters from at least 2 classes

(the five classes are lowercase, uppercase, numbers, punctuation,

and all other characters).

New password:      <Jennifer types 'sloth'>

New password (again):      <Jennifer re-types 'sloth'>

kpasswd: New password is too short.

Please choose a password which is at least 4 characters long.
```

在以下示例中，`jennifer` 使用 "slothrop49" 作为口令。由于 "slothrop49" 的长度超过八个字母，并且包含两种不同类型的字符（数字和小写字母），因此此口令符合条件。

```
% kpasswd

kpasswd: Changing password for jennifer@ENG.EXAMPLE.COM.

Old password:      <Jennifer types her existing password>

kpasswd: jennifer@ENG.EXAMPLE.COM's password is controlled by

the policy jenpol

which requires a minimum of 8 characters from at least 2 classes

(the five classes are lowercase, uppercase, numbers, punctuation,

and all other characters).

New password:      <Jennifer types 'slothrop49'>

New password (again):      <Jennifer re-types 'slothrop49'>

Kerberos password changed.
```

示例－更改口令

在以下示例中，用户 `david` 使用 `passwd` 同时更改其 UNIX 口令和 Kerberos 口令。

```
% passwd

    passwd:  Changing password for david

    Enter login (NIS+) password:            <Type the current UNIX password>

    New password:                           <Type the new UNIX password>

    Re-enter password:                      <Confirm the new UNIX password>

    Old KRB5 password:                      <Type the current Kerberos password>

    New KRB5 password:                      <Type the new Kerberos password>

    Re-enter new KRB5 password:             <Confirm the new Kerberos password>
```

请注意，passwd 需要 UNIX 口令和 Kerberos 口令。此行为由缺省配置确定。在这种情况下，用户 david 必须使用 kpasswd 将其 Kerberos 口令设置为其他内容，如下所示。

本示例说明用户 david 如何使用 kpasswd 仅更改其 Kerberos 口令。

```
% kpasswd

kpasswd: Changing password for david@ENG.EXAMPLE.COM.

Old password:               <Type the current Kerberos password>

New password:               <Type the new Kerberos password>

New password (again):       <Confirm the new Kerberos password>

Kerberos password changed.
```

在本示例中，用户 david 更改了 Kerberos 主体 david/admin（非有效的 UNIX 用户）的口令。该用户必须使用 kpasswd。

```
% kpasswd david/admin

kpasswd:  Changing password for david/admin.

Old password:               <Type the current Kerberos password>

New password:               <Type the new Kerberos password>

New password (again):       <Type the new Kerberos password>

Kerberos password changed.
```

授予对帐户的访问权限

如果需要授予某个用户访问权限以登录到您的帐户（以您的身份），则可以通过 Kerberos 执行此操作而不必显示您的口令，方法是将 .k5login 文件放置在起始目录中。.k5login 文件是一个列表，其中包含一个或多个与要为其授予访问权限的各用户对应的 Kerberos 主体。每个主体都必须单独占一行。

假定用户 david 在其起始目录中按如下所示保存了一个 .k5login 文件：

```
jennifer@ENG.EXAMPLE.COM

joe@EXAMPLE.ORG
```

如果用户 jennifer 和 joe 在其各自的领域中已经拥有 Kerberos 票证，则此文件允许这两个用户采用 david 的身份。例如，jennifer 可以使用 david 的身份远程登录到 david 的计算机 (boston)，而不必提供 david 的口令。

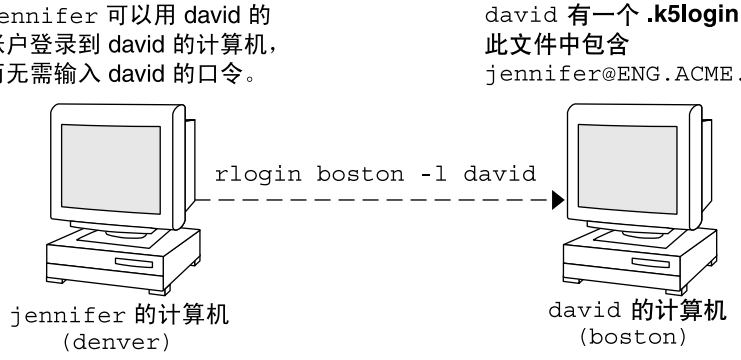


图 25–1 使用 .k5login 文件授予对帐户的访问权限

如果 david 的起始目录使用 Kerberos V5 协议从另一台（第三台）计算机挂载了 NFS，则 jennifer 必须具有可转发票证才能访问 david 的起始目录。有关使用可转发票证的示例，请参见第 486 页中的 “创建 Kerberos 票证”。

如果您要通过网络登录到其他计算机，则需要在这些计算机上的 .k5login 文件中包括您自己的 Kerberos 主体。

使用 .k5login 文件比公布口令安全得多，原因如下：

- 您可以随时通过从 .k5login 文件中删除主体来收回访问权限。

- 虽然在您的起始目录的 .k5login 文件中指定的用户主体对您在该计算机（或一组计算机，例如如果通过 NFS 共享 .k5login 文件）上的帐户拥有完全访问权限，但是，所有基于 Kerberos 的服务都将根据该用户的身份而不是您的身份来授权访问。因此，jennifer 可以登录到 joe 的计算机并在其中执行任务。但是，如果该用户使用基于 Kerberos 的程序（如 ftp 或 rlogin），则将以其自身身份执行此操作。
- Kerberos 会记录获取票证的用户，以便系统管理员在必要时查找在特定时间可以使用您的用户身份的人员。

使用 .k5login 文件的一种常见方法是将其放置在 root 的起始目录中，从而为列出的 Kerberos 主体提供对该计算机的 root 访问权限。此配置允许系统管理员成为本地 root，或以 root 身份远程登录，而不必公布 root 口令，并且不需要任何人通过网络键入 root 口令。

示例－使用 .k5login 文件授予对帐户的访问权限

假定 jennifer 决定以 root 身份登录到计算机 boston.example.com。由于在 boston.example.com 的 root 起始目录的 .k5login 文件中存在该用户的主体名称项，因此不必再次键入其口令。

```
% rlogin boston.example.com -l root -x

This rlogin session is using DES encryption for all data transmissions.

Last login: Thu Jun 20 16:20:50 from daffodil

SunOS Release 5.7 (GENERIC) #2: Tue Nov 14 18:09:31 EST 1998

boston[root]%
```

Kerberos 用户命令

Kerberos V5 产品是一个**单点登录**系统，这表示只需键入一次口令即可。Kerberos V5 程序将为您执行验证操作（并可以选择执行加密操作），因为 Kerberos 已内置在每个熟知的现有网络程序套件中。 Kerberos V5 应用程序是添加了 Kerberos 功能的现有 UNIX 网络程序的版本。

例如，使用基于 Kerberos 的程序连接到远程主机时，该程序、KDC 和远程主机将执行一组快速协商。完成这些协商后，该程序便已代表您向远程主机证明了您的身份，并且远程主机已授予您访问权限。

请注意，基于 Kerberos 的命令会首先尝试使用 Kerberos 进行验证。如果 Kerberos 验证失败，将会出现错误或尝试 UNIX 验证，具体取决于和命令一起使用的选项。有关更多详细信息，请参阅每个 Kerberos 命令手册页中的 Kerberos Security 部分。

基于 Kerberos 的命令概述

基于 Kerberos 的网络服务是一些连接到 Internet 中某个位置的其他计算机的程序。这些程序如下：

- `ftp`
- `rcp`
- `rdist`
- `rlogin`
- `rsh`
- `ssh`
- `telnet`

这些程序具有可透明地使用 Kerberos 票证与远程主机协商验证并选择协商加密的功能。在大多数情况下，您只会注意到不必再键入口令即可使用这些程序，因为 Kerberos 将为您提供身份证明。

Kerberos V5 网络程序包括可用于执行以下操作的选项：

- 将票证转发到其他主机（如果最初已获取可转发票证）。
- 加密在您的主机和远程主机之间传输的数据。

注 – 本节假定您已经熟悉这些程序的非 Kerberos 版本，并且将重点介绍 Kerberos V5 软件包添加的 Kerberos 功能。有关此处描述的命令的详细说明，请参阅其各自的手册页。

已将下列 Kerberos 选项添加至 `ftp`、`rcp`、`rlogin`、`rsh` 和 `telnet`：

`-a` 尝试使用现有票证自动登录。如果 `getlogin()` 返回的用户名与当前用户 ID 相同，则使用该用户名。有关详细信息，请参见 `telnet`(1) 手册页。

`-f` 将**不可重新转发的**票证转发到远程主机。此选项与 `-F` 选项互斥。在同一命令中不能同时使用这两个选项。

如果您有理由相信需要向第三台主机中其他基于 Kerberos 的服务验证您的身份，则应转发票证。例如，您可能要远程登录到另一台计算机，然后从该计算机远程登录到第三台计算机。

如果远程主机上的起始目录使用 Kerberos V5 机制挂载了 NFS，则必须使用可转发票证。否则，将无法访问起始目录。也就是说，假定您最初登录到系统 1，然后从系统 1 远程登录到您的主机（系统2），该主机从系统 3 挂载您的起始目录。在这种情况下，除非在 `rlogin` 中使用 `-f` 或 `-F` 选项，否则将无法访问起始目录，因为您的票证无法转发到系统 3。

缺省情况下，`kinit` 会获取可转发票证授予票证 (Ticket-Granting Ticket, TGT)。但是，您的配置在这方面可能会有所不同。

有关转发票证的更多信息，请参见第 497 页中的 “转发 Kerberos 票证” 。

`-F` 将 TGT 的**可重新转发**副本转发到远程系统。此选项与 `-f` 类似，但它允许访问更多（如第四台或第五台）计算机。因此，可将 `-F` 选项视为 `-f` 选项的超集。`-F` 选项与 `-f` 选项互斥。在同一命令中不能同时使用这两个选项。

有关转发票证的更多信息，请参见第 497 页中的 “转发 Kerberos 票证”。

`-k` *realm* 请求指定的 *realm* 中的远程主机的票证，而不是使用 `krb5.conf` 文件来确定领域本身。

`-K` 使用票证来向远程主机验证，但不自动登录。

`-m` *mechanism* 指定 `/etc/gss/mech` 文件中列出的要使用的 GSS-API 安全机制。缺省值为 `kerberos_v5`。

`-x` 加密此会话。

`-X` *auth_type* 禁用 *auth-type* 类型的验证。

下表显示具有特定选项的命令。"X" 表示命令包含该选项。

表 25–1 网络命令的 Kerberos 选项

	ftp	rcp	rlogin	rsh	telnet
`-a`					X
`-f`	X		X	X	X
`-F`			X	X	X
`-k`		X	X	X	X
`-K`					X
`-m`	X				
`-x`	X	X	X	X	X
`-X`					X

此外，`ftp` 还允许在其提示符下为会话设置保护级别：

clear 将保护级别设置为 "clear"（无保护）。此保护级别是缺省级别。

private 将保护级别设置为 "private"。通过加密保护数据传输的保密性和完整性。但是，并非所有 Kerberos 用户都可使用保密性服务。

safe 将保护级别设置为 "safe"。通过加密校验和保护数据传输的完整性。

也可以通过键入 `protect`，并后跟以上所示的任何保护级别（`clear`、`private` 或 `safe`），在 `ftp` 提示符下设置保护级别。

转发 Kerberos 票证

如第 495 页中的 “基于 Kerberos 的命令概述”中所述，某些命令允许您使用 -f 或 -F 选项转发票证。通过转发票证，可以“链接”网络事务。例如，可以远程登录到一台计算机，然后从该计算机远程登录到另一台计算机。使用 -f 选项可转发票证，而使用 -F 选项则可重新转发已转发的票证。

在图 25–2 中，用户 david 使用 kinit 获取了一个不可转发票证授予票证 (Ticket-Granting Ticket, TGT)。由于该用户未指定 -f 选项，因此该票证不可转发。在方案 1 中，该用户可以远程登录到计算机 B，但不能再登录到其他计算机。在方案 2 中，由于该用户尝试转发一个不可转发票证，因此 rlogin -f 命令失败。

1. （在 A 上）：kinit david@ACME.ORG

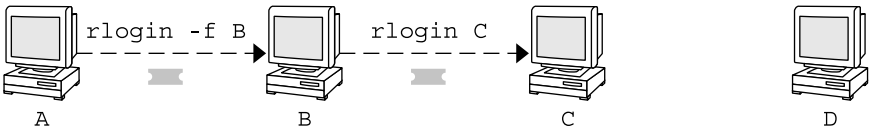
2. （在 A 上）：kinit david@ACME.ORG

图 25–2 使用不可转发票证

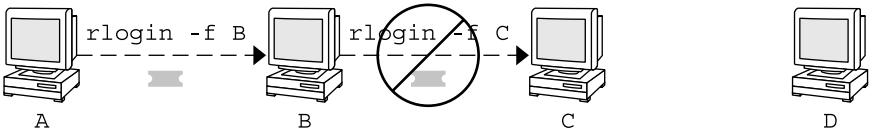
实际上，已设置了 Kerberos 配置文件，以便 kinit 在缺省情况下可获取可转发票证。但是，您的配置可能有所不同。为了便于说明，假定 kinit **不会**获取可转发 TGT，除非使用 kinit -f 调用该命令。另请注意，kinit 不包含 -F 选项。TGT 可以是可转发，也可以是不可转发。

在图 25–3 中，用户 david 使用 kinit -f 获取了可转发 TGT。在方案 3 中，由于该用户在 rlogin 中使用了可转发票证，因此可以访问计算机 C。在方案 4 中，由于票证不可重新转发，因此第二个 rlogin 失败。如方案 5 中所示，改用 -F 选项后，第二个 rlogin 将成功，并且票证可重新转发到计算机 D。

3.（在 A 上）：kinit -f david@ACME.ORG



4.（在 A 上）：kinit -f david@ACME.ORG



5.（在 A 上）：kinit -f david@ACME.ORG

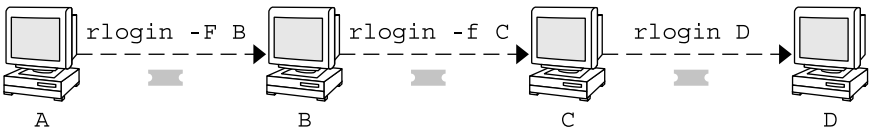


图 25–3 使用可转发票证

示例－使用基于 Kerberos 的命令

以下示例说明基于 Kerberos 的命令的选项的工作方式。

示例－将 -a、-f 和 -x 选项用于 telnet

在本示例中，用户 david 已经登录，并且要 telnet 到计算机 denver.example.com。该用户使用 -f 选项转发其现有票证，使用 -x 选项加密会话，并使用 -a 选项自动执行登录。由于该用户不打算使用第三台主机的服务，因此可使用 -f 而非 -F。

```
% telnet -a -f -x denver.example.com

Trying 128.0.0.5...

Connected to denver.example.com. Escape character is '^]'.

[ Kerberos V5 accepts you as "david@eng.example.com" ]

[ Kerberos V5 accepted forwarded credentials ]

SunOS 5.9: Tue May 21 00:31:42 EDT 2004  Welcome to SunOS

%
```

请注意，david 的计算机使用 Kerberos 来向 denver.example.com 进行自我验证，并且以自己的身份自动登录。该用户具有一个加密会话（即该用户已有的票证副本），因此永远不必键入其口令。如果该用户使用了非 Kerberos 版本的 telnet，则可能会提示其输入口令，并将此口令在未加密的情况下通过网络发送。如果入侵者在此期间一直在观察网络通信流量，则可能会知道 david 的口令。

如果转发 Kerberos 票证，则 telnet（以及此处讨论的其他命令）将会在退出时销毁这些票证。

示例－使用带有 -F 选项的 rlogin

在此示例中，用户 jennifer 希望登录到自己的计算机 boston.example.com。该用户使用 -F 选项转发其现有票证，并使用 -x 选项加密会话。由于该用户在登录到 boston 后，可能希望执行需要重新转发票证的其他网络事务，因此会选择 -F 而非 -f。另外，由于该用户要转发其现有票证，因此不必键入其口令。

```
% rlogin boston.example.com -F -x

This rlogin session is using encryption for all transmissions.

Last login Mon May 19 15:19:49 from daffodil

SunOS Release 5.9 (GENERIC) #2 Tue Nov 14 18:09:3 EST 2003

%
```

示例－在 ftp 中设置保护级别

假定 joe 要使用 ftp 从计算机 denver.example.com 的目录 ~joe/MAIL 中获取其邮件并加密该会话。交换过程如下所示：

```
% ftp -f denver.example.com

Connected to denver.example.com

220 denver.example.org FTP server (Version 6.0) ready.

334 Using authentication type GSSAPI; ADAT must follow

GSSAPI accepted as authentication type

GSSAPI authentication succeeded Name (daffodil.example.org:joe)

232 GSSAPI user joe@MELPOMENE.EXAMPLE.COM is authorized as joe

230 User joe logged in.
```

```
Remote system type is UNIX.

Using BINARY mode to transfer files.

ftp> protect private

200 Protection level set to Private

ftp> cd ~joe/MAIL

250 CWD command successful.

ftp> get RMAIL

227 Entering Passive Mode (128,0,0,5,16,49)

150 Opening BINARY mode data connection for RMAIL (158336 bytes).

226 Transfer complete. 158336 bytes received in 1.9 seconds (1.4e+02 Kbytes/s)

ftp> quit

%
```

要加密该会话，joe 需要将保护级别设置为 private。

◆ ◆ ◆ 第 26 章

Kerberos 服务（参考）

本章列出了许多属于 Kerberos 产品的文件、命令和守护进程。另外，本章还介绍了有关 Kerberos 验证的工作方式的详细信息。

以下是本章中参考信息的列表。

- 第 501 页中的 “Kerberos 文件”
- 第 502 页中的 “Kerberos 命令”
- 第 503 页中的 “Kerberos 守护进程”
- 第 504 页中的 “Kerberos 术语”
- 第 508 页中的 “Kerberos 验证系统的工作方式”
- 第 509 页中的 “使用 Kerberos 获取服务访问权限”
- 第 512 页中的 “使用 Kerberos 加密类型”
- 第 513 页中的 “使用 gsscred 表”
- 第 513 页中的 “Solaris Kerberos 和 MIT Kerberos 之间的显著差异”

Kerberos 文件

表 26–1 Kerberos 文件

文件名	说明
`~/.gkadmin`	用于在 SEAM 管理工具中创建新主体的缺省值
`~/.k5login`	授予 Kerberos 帐户访问权限的主体列表
`/etc/krb5/kadm5.acl`	Kerberos 访问控制列表文件，其中包含 KDC 管理员的主体名称及其 Kerberos 管理权限
`/etc/krb5/kadm5.keytab`	主 KDC 上的 `kadmin` 服务的密钥表文件
`/etc/krb5/kdc.conf`	KDC 配置文件

表 26–1 Kerberos 文件 （续）

文件名	说明
/etc/krb5/kpropd.acl	Kerberos 数据库传播配置文件
/etc/krb5/krb5.conf	Kerberos 领域配置文件
/etc/krb5/krb5.keytab	网络应用程序服务器的密钥表文件
/etc/krb5/warn.conf	Kerberos 票证到期警告和自动更新配置文件
/etc/pam.conf	PAM 配置文件
/tmp/krb5cc_*uid*	缺省凭证高速缓存，其中 *uid* 是用户的十进制 UID
/tmp/ovsec_adm.*xxxxxx*	口令更改操作生命周期的临时凭证高速缓存，其中 *xxxxxx* 是随机字符串
/var/krb5/.k5.*REALM*	KDC 存储文件，其中包含 KDC 主密钥的加密副本
/var/krb5/kadmin.log	kadmind 的日志文件
/var/krb5/kdc.log	KDC 的日志文件
/var/krb5/principal	Kerberos 主体数据库
/var/krb5/principal.kadm5	Kerberos 管理数据库，其中包含策略信息
/var/krb5/principal.kadm5.lock	Kerberos 管理数据库锁定文件
/var/krb5/principal.ok	Kerberos 数据库成功初始化时创建的 Kerberos 主体数据库初始化文件
/var/krb5/principal.ulog	Kerberos 更新日志，其中包含增量传播更新
/var/krb5/slave_datatrans	kprop_script 脚本用于传播的 KDC 备份文件
/var/krb5/slave_datatrans_*slave*	完全更新指定的 *slave* 时创建的临时转储文件

Kerberos 命令

本节列出了 Kerberos 产品中包括的部分命令。

表 26–2 Kerberos 命令

命令	说明
/usr/bin/ftp	文件传输协议程序
/usr/bin/rcp	远程文件复制程序
/usr/bin/rdist	远程文件分发程序

表 26–2 Kerberos 命令 （续）

命令	说明
/usr/bin/rlogin	远程登录程序
/usr/bin/rsh	远程 shell 程序
/usr/bin/telnet	基于 Kerberos 的 telnet 程序
/usr/lib/krb5/kprop	Kerberos 数据库传播程序
/usr/sbin/gkadmin	Kerberos 数据库管理 GUI 程序，用于管理主体和策略
/usr/sbin/kadmin	远程 Kerberos 数据库管理程序（运行时需要进行 Kerberos 验证），用于管理主体、策略和密钥表文件
/usr/sbin/kadmin.local	本地 Kerberos 数据库管理程序（运行时无需进行 Kerberos 验证，并且必须在主 KDC 上运行），用于管理主体、策略和密钥表文件
/usr/sbin/kclient	Kerberos 客户机安装脚本，可用于安装配置文件，也可不用于安装配置文件
/usr/sbin/kdb5_util	创建 Kerberos 数据库和存储文件
/usr/sbin/kproplog	列出更新日志中更新项的摘要

Kerberos 守护进程

下表列出了 Kerberos 产品使用的守护进程。

表 26–3 Kerberos 守护进程

守护进程	说明
/usr/sbin/in.ftpd	文件传输协议守护进程
/usr/lib/krb5/kadmind	Kerberos 数据库管理守护进程
/usr/lib/krb5/kpropd	Kerberos 数据库传播守护进程
/usr/lib/krb5/krb5kdc	Kerberos 票证处理守护进程
/usr/lib/krb5/ktkt_warnd	Kerberos 票证到期警告和自动更新守护进程
/usr/sbin/in.rlogind	远程登录守护进程
/usr/sbin/in.rshd	远程 shell 守护进程
/usr/sbin/in.telnetd	telnet 守护进程

Kerberos 术语

下一节介绍了 Kerberos 术语及其定义。这些术语可用于整个 Kerberos 文档。要理解 Kerberos 概念，必须先了解这些术语。

特定于 Kerberos 的术语

要管理 KDC，您需要了解本节中的术语。

Key Distribution Center, KDC（密钥分发中心）是负责颁发凭证的 Kerberos 组件。这些凭证是使用 KDC 数据库中存储的信息创建的。每个领域至少需要两个 KDC，一个主 KDC 以及至少一个从 KDC。所有 KDC 都可生成凭证，但仅有主 KDC 才能处理对 KDC 数据库所做的任何更改。

stash file（**存储文件**）包含 KDC 的主密钥。当重新启动服务器以便在启动 `kadmind` 和 `krb5kdc` 命令之前自动验证 KDC 时，将使用此密钥。由于此文件包含主密钥，因此应将此文件及其所有备份安全保存。此文件是使用 `root` 的只读权限创建的。要确保此文件安全，请勿更改相应的权限。如果此文件已被破坏，则其他用户可能会使用主密钥来访问或修改 KDC 数据库。

特定于验证的术语

要了解验证过程，您需要理解本节中的术语。程序员和系统管理员应熟悉这些术语。

client（**客户机**）是在用户工作站上运行的软件。在客户机上运行的 Kerberos 软件会在此过程中发出许多请求。因此，区分此软件的操作和用户非常重要。

术语 *server*（**服务器**）和 *service*（**服务**）通常可互换使用。具体而言，术语**服务器**用于定义运行 Kerberos 软件的物理系统。术语**服务**对应于服务器支持的特定功能（例如 `ftp` 或 `nfs`）。文档通常会将服务器描述为服务的一部分，但此定义会混淆了这些术语的含义。因此，术语**服务器**是指物理系统，而术语**服务**则是指软件。

Kerberos 产品使用两种类型的密钥。一种密钥类型是口令派生密钥。口令派生密钥会被指定给每个用户主体，并仅对该用户和 KDC 公开。 Kerberos 产品使用的另一种密钥类型是与口令无关联的随机密钥，因此不适合用户主体使用。随机密钥通常用于在密钥表中具有相应项并具有 KDC 生成的会话密钥的服务主体。服务主体可以使用随机密钥，因为服务可以访问密钥表中允许其以非交互方式运行的密钥。会话密钥由 KDC 生成，并在客户机和服务之间共享，可用于在两者之间提供安全事务。

ticket（**票证**）是一种信息包，用于将用户身份安全地传递到服务器或服务。一个票证仅对一台客户机以及某台特定服务器上的一项特殊服务有效。票证包含以下内容：

- 服务的主体名称
- 用户的主体名称
- 用户主机的 IP 地址

- 时间标记
- 定义票证生命周期的值
- 会话密钥的副本

所有此类数据都使用服务器的服务密钥进行加密。 请注意，KDC 可颁发嵌入在以下介绍的凭证中。 颁发票证之后，可重用票证直到其到期为止。

credential（**凭证**）是一种信息包，其中包含票证和匹配的会话密钥。凭证使用发出请求的主体的密钥进行加密。通常，KDC 会生成凭证以响应客户机的票证请求。

authenticator（**验证者**）是服务器用于验证客户机用户主体的信息。 验证者包含用户的主体名称、时间标记和其他数据。 与票证不同，验证者只能使用一次，通常在请求访问服务时使用。 验证者使用客户机和服务器共享的会话密钥进行加密。 通常，客户机会创建验证者，并将其与服务器或服务的票证一同发送，以便向服务器或服务进行验证。

票证类型

票证具有可管理其使用方式的属性。这些属性是在创建票证时指定给票证的，但稍后可修改票证的属性。例如，可将票证从 `forwardable` 更改为 `forwarded`。可以使用 `klist` 命令查看票证属性。请参见第 487 页中的 “查看 Kerberos 票证” 。

以下一个或多个术语对票证进行了描述：

Forwardable（可转发）/forwarded（已转发）
: 可转发票证可以从一台主机发送到另一台主机，从而使客户机无需对其自身进行重新验证。例如，如果用户 `david` 在用户 `jennifer` 的计算机上时获取了一个可转发票证，则前者可登录到自己的计算机，而不必获取新的票证（从而对自身进行重新验证）。有关可转发票证的示例，请参见第 486 页中的 “示例－创建 Kerberos 票证” 。

Initial（初始）
: 初始票证是一种直接颁发的票证，而并非基于票证授予票证的票证。某些服务（如用于更改口令的应用程序）可能会要求将票证标记为初始，以便向这些程序本身确保客户机可知道其私钥。初始票证表明客户机最近已进行了自我验证，而并非依赖于已长期使用的票证授予票证。

Invalid（未生效）
: 无效票证是一个尚未变为可用的未生效的票证。应用程序服务器会拒绝无效票证，直到其经过验证为止。要进行验证，必须在票证开始时间已过之后由客户机将设置了 `VALIDATE` 标志的票证放置在票证授予服务请求的 KDC 中。

Postdatable（可以后生效）/postdated（以后生效）
: 以后生效的票证是一种在其创建之后的某个指定时间之前不会生效的票证。例如，此类票证对于计划在深夜运行的批处理作业非常有用，因为如果该票证被盗，则在运行批处理作业之前，将无法使用该票证。颁发以后生效的票证时，该票证未生效，并在其开始时间已过且客户机请求 KDC 进行验证之前一直保持此状态。通常，以后生效的票证在票证授予票证的到期时间之前会一直有效。但是，如果将以后生效的票证标记为可更新，则通常会将其生命周期设置为等于票证授予票证的整个生命周期的持续时间。

Proxiable（可代理）/proxy（代理）
有时，主体需要允许服务代表其执行操作。创建该票证时，必须指定代理的主体名称。Solaris 发行版不支持可代理票证或代理票证。

可代理票证与可转发票证类似，但前者仅对单个服务有效，而可转发票证授予服务对客户机身份的完全使用权限。因此，可以将可转发票证视为一种超级代理。

Renewable（可更新）
由于拥有很长生命周期的票证存在安全风险，因此可将票证指定为可更新票证。可更新票证具有两个到期时间：票证当前实例的到期时间以及任意票证的最长生命周期（一周）。如果客户机要继续使用票证，则可在第一个到期时间之前更新票证。例如，某个票证的有效期可为一个小时，而所有票证的最长生命周期为 10 个小时。如果持有该票证的客户机要将该票证再保留几个小时，则此客户机必须在有效的小时数内更新票证。如果票证到达最长票证生命周期（10 个小时），则该票证将自动到期且无法更新。

有关如何查看票证属性的信息，请参见第 487 页中的 “查看 Kerberos 票证”。

票证生命周期

每当主体获取包括票证授予票证 (Ticket–Granting Ticket, TGT) 在内的票证时，都会将票证的生命周期设置为以下生命周期值中的最小值：

- 通过 `kinit` 的 `-l` 选项指定的生命周期值，前提是使用 `kinit` 获取票证。缺省情况下，`kinit` 使用最长生命周期值。
- `kdc.conf` 文件中指定的最长生命周期值 (`max_life`)。
- Kerberos 数据库中为提供票证的服务主体指定的最长生命周期值。如果使用 `kinit`，则服务主体为 `krbtgt/`*realm*。
- Kerberos 数据库中为请求票证的用户主体指定的最长生命周期值。

图 26–1 说明了如何确定 TGT 的生命周期以及四个生命周期值的来源。虽然该图说明的是如何确定 TGT 的生命周期，但所有主体获取票证时的情况基本相同。唯一的区别在于，`kinit` 不提供生命周期值，而提供票证的服务主体（非 `krbtgt/`*realm* 主体）会提供最长生命周期值。

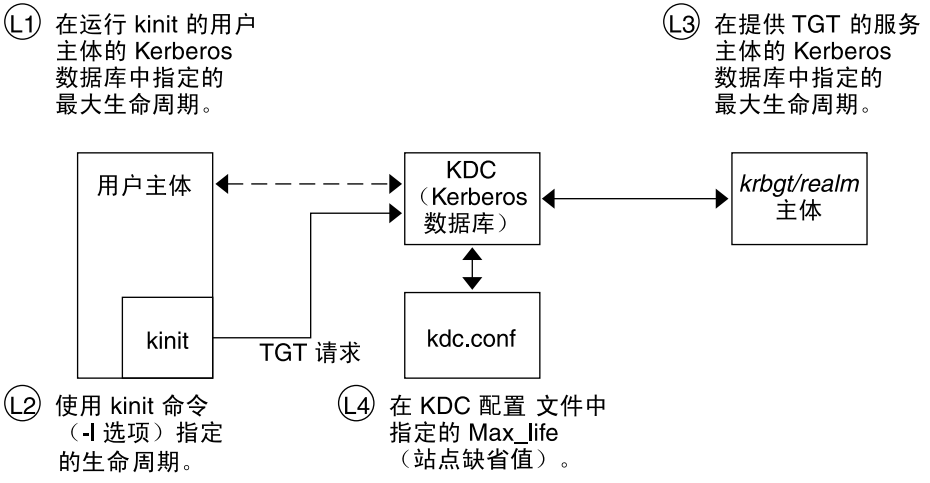


图 26–1 如何确定 TGT 的生命周期

可更新票证生命周期也是由四个值中的最小值确定的，但是使用的却是可更新的生命周期值，如下所示：

- 通过 `kinit` 的 `-r` 选项指定的可更新生命周期值，前提是使用 `kinit` 获取或更新票证。
- `kdc.conf` 文件中指定的最长可更新生命周期值 (`max_renewable_life`)。
- Kerberos 数据库中为提供票证的服务主体指定的最长可更新生命周期值。如果使用 `kinit`，则服务主体为 `krbtgt/`*realm*。
- Kerberos 数据库中为请求票证的用户主体指定的最长可更新生命周期值。

Kerberos 主体名称

每个票证都使用主体名称进行标识。主体名称可以标识用户或服务。以下是一些主体名称的示例：

表 26–4 Kerberos 主体名称示例

主体名称	说明
`changepw/kdc1.example.com@EXAMPLE.COM`	更改口令时，允许访问 KDC 的主 KDC 服务器的主体。
`clntconfig/admin@EXAMPLE.COM`	`kclient` 安装实用程序使用的主体。
`ftp/boston.example.com@EXAMPLE.COM`	`ftp` 服务使用的主体。此主体可用于替代 `host` 主体。

表 26–4 Kerberos 主体名称示例 （续）

主体名称	说明
host/boston.example.com@EXAMPLE.COM	基于 Kerberos 的应用程序（例如 klist 和 kprop）和服务（例如 ftp 和 telnet）使用的主体。此主体称为 host 主体或服务主体，用于验证 NFS 挂载。
K/M@EXAMPLE.COM	主密钥名称主体。一个主密钥名称主体可与每个主 KDC 关联。
kadmin/history@EXAMPLE.COM	一种主体，其中包含用于保存其他主体的口令历史记录的密钥。每个主 KDC 都具有这些主体之一。
kadmin/kdc1.example.com@EXAMPLE.COM	允许使用 kadmind 访问 KDC 的主 KDC 服务器的主体。
kadmin/changepw.example.com@EXAMPLE.COM	一种主体，用于接受来自未运行 Solaris 发行版的客户机的口令更改请求。
krbtgt/EXAMPLE.COM@EXAMPLE.COM	生成票证授予票证时使用的主体。
krbtgt/EAST.EXAMPLE.COM@WEST.EXAMPLE.COM	此主体是跨领域的票证授予票证的示例。
nfs/boston.example.com@EXAMPLE.COM	NFS 服务使用的主体。此主体可用于替代 host 主体。
root/boston.example.com@EXAMPLE.COM	与客户机的 root 帐户关联的主体。此主体称作 root 主体，用于向已挂载 NFS 的文件系统提供 root 访问权限。
username@EXAMPLE.COM	用户的主体。
username/admin@EXAMPLE.COM	admin 主体，可用于管理 KDC 数据库。

Kerberos 验证系统的工作方式

如果您可以提供证明您身份的票证和匹配的会话密钥，则应用程序允许您登录到远程系统。会话密钥包含特定于用户以及要访问的服务的信息。所有用户首次登录时，KDC 都会为其创建票证和会话密钥。票证和匹配的会话密钥可组成凭证。使用多个网络服务时，用户可以收集许多凭证。对于在特定服务器上运行的每个服务，用户都需要拥有一个凭证。例如，访问名为 boston 的服务器中的 ftp 服务需要凭证。访问其他服务器中的 ftp 服务需要对应于该服务的凭证。

创建和存储凭证的过程是透明的。凭证是由将凭证发送到请求程序的 KDC 创建的。收到凭证后，会将其存储在凭证高速缓存中。

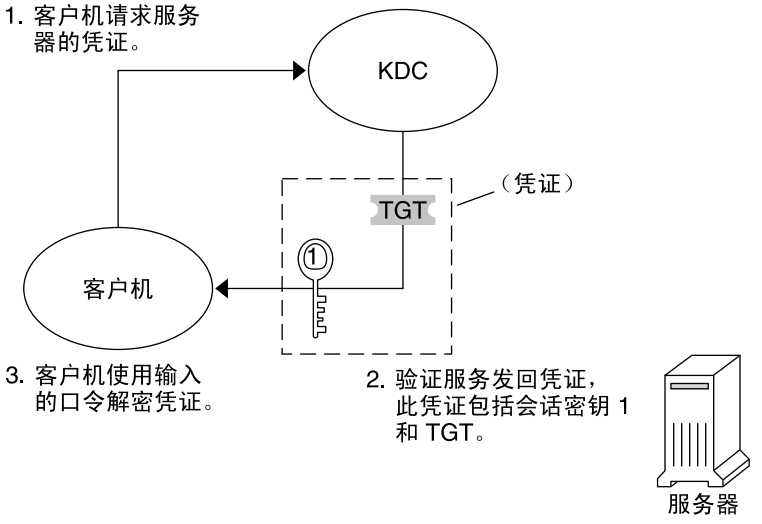
使用 Kerberos 获取服务访问权限

要访问特定服务器上的特定服务，用户必须获取两个凭证。第一个凭证用于票证授予票证（称为 TGT）。票证授予服务对此凭证进行解密之后，该服务即可为用户请求访问的服务器创建第二个凭证。然后，可使用第二个凭证来请求访问该服务器中的相应服务。该服务器成功解密第二个凭证后，便会授予用户访问权限。以下各节详细介绍了此过程。

获取用于票证授予服务的凭证

1. 要启动验证过程，客户机需要向验证服务器发送针对特定用户主体的验证请求。该请求在发送时未加密。由于请求中未包含安全信息，因此无需加密。
2. 验证服务收到该请求后，将在 KDC 数据库中查找该用户的主体名称。如果主体与数据库中的项匹配，则验证服务可获取该主体的私钥。然后，验证服务将生成一个供客户机和票证授予服务使用的会话密钥（称为会话密钥 1），以及一个用于票证授予服务的票证（票证 1）。此票证也称作**票证授予票证** (Ticket-Granting Ticket, TGT)。会话密钥和票证均使用该用户的私钥进行加密，并且会将信息发回客户机。
3. 客户机通过该用户主体的私钥，使用此信息对会话密钥 1 和票证 1 进行解密。由于该私钥仅对此用户和 KDC 数据库公开，因此包中的信息应是安全的。客户机将该信息存储在凭证高速缓存中。

在此过程中，通常会提示用户输入口令。如果用户指定的口令与用于生成存储在 KDC 数据库中的私钥的口令相同，则客户机可以成功解密验证服务发送的信息。现在，客户机便拥有了用于票证授予服务的凭证。客户机现在可以请求用于服务器的凭证。



TGT = 票证授予票证
KDC = 密钥分发中心

图 26–2 获取用于票证授予服务的凭证

获取用于服务器的凭证

1. 要请求访问特定服务器，客户机必须首先从验证服务获取用于该服务器的凭证。请参见第 509 页中的 “获取用于票证授予服务的凭证”。然后，客户机会向票证授予服务发送请求，其中包含服务主体名称、票证 1 以及使用会话密钥 1 加密的验证者。票证 1 最初是由验证服务使用票证授予服务的服务密钥加密的。
2. 由于票证授予服务的服务密钥对票证授予服务公开，因此可以解密票证 1。票证 1 中的信息包括会话密钥 1，因此票证授予服务可以解密验证者。此时，可使用票证授予服务验证用户主体。
3. 成功验证后，票证授予服务将为用户主体和服务器生成一个会话密钥（会话密钥 2），以及一个用于服务器的票证（票证 2）。然后，使用会话密钥 1 加密会话密钥 2 和票证 2。由于会话密钥 1 仅对该客户机和票证授予服务公开，因此此信息是安全的并可在网络上安全发送。
4. 客户机收到此信息包后，将使用存储在凭证高速缓存中的会话密钥 1 解密此信息。客户机即获取用于服务器的凭证。现在，客户机可以请求访问该服务器中的特定服务。

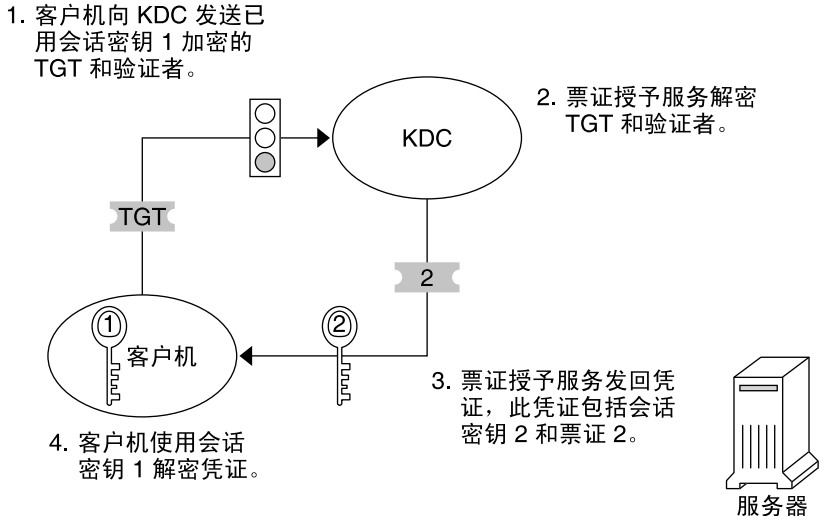


图 26–3 获取用于服务器的凭证

获取对特定服务的访问权限

1. 要请求访问特定服务，客户机必须首先从验证服务器获取用于票证授予服务的凭证，然后从票证授予服务获取服务器凭证。请参见第 509 页中的 “获取用于票证授予服务的凭证”和第 510 页中的 “获取用于服务器的凭证”。然后，客户机可将包含票证 2 和另一个验证者的请求发送到该服务器。该验证者使用会话密钥 2 进行加密。
2. 票证 2 是由票证授予服务使用该服务的服务密钥进行加密的。由于服务密钥对服务主体公开，因此该服务可以解密票证 2 并获取会话密钥 2。然后，可使用会话密钥 2 解密验证者。如果成功解密验证者，则可授予客户机对该服务的访问权限。

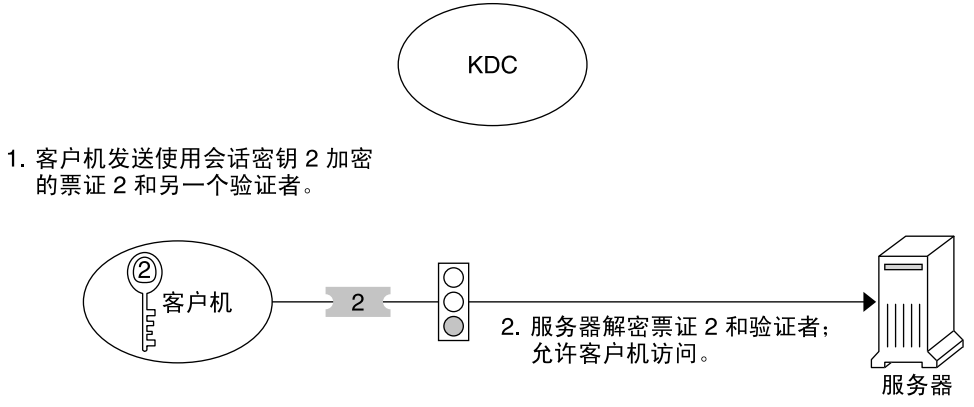


图 26–4 获取对特定服务的访问权限

使用 Kerberos 加密类型

加密类型可标识执行加密操作时要使用的加密算法和模式。使用 aes、des3-cbc-sha1 和 rc4–hmac 加密类型可以创建用于较高强度加密操作的密钥。这些较高强度的操作可增强 Kerberos 服务的整体安全性。

注 – 如果安装了非随附的强加密软件包，则可以将 `aes256-cts-hmac-sha1-96` 加密类型用于 Kerberos 服务。

客户机从 KDC 请求票证时，KDC 必须使用其加密类型与客户机和服务器兼容的密钥。尽管 Kerberos 协议允许客户机请求 KDC 对客户机的票证回复部分使用特定的加密类型，但该协议不允许服务器为 KDC 指定加密类型。

注 – 如果安装了未运行 Solaris 10 发行版的主 KDC，则在升级主 KDC 之前，必须将从 KDC 升级到 Solaris 10 发行版。Solaris 10 主 KDC 将使用新的加密类型，而较早版本的从 KDC 将无法处理这些加密类型。

以下列出了更改加密类型之前必须考虑的一些问题：

- KDC 假定服务器支持与主体数据库中的服务器主体项关联的第一个密钥/加密类型。
- 在 KDC 上，应确保生成的主体密钥与将验证主体的系统兼容。缺省情况下，`kadmin` 命令将为所有支持的加密类型创建密钥。如果验证主体的系统不支持该缺省加密类型集，则在创建主体时应限制加密类型。通过在 `kadmin addprinc` 中使用 `-e` 标志，或在 `kdc.conf` 文件中将 `supported_enctypes` 参数设置为此子集，可以限制加密类型。如果 Kerberos 领域中的大多数系统都支持缺省加密类型集的子集，则应使用 `supported_enctypes` 参数。如果设置 `supported_enctypes`，则将指定 `kadmin addprinc` 在为特定领域创建主体时使用的加密类型的缺省集。一般情况下，最好使用这两种方法之一来控制 Kerberos 使用的加密类型。
- 确定系统支持的加密类型时，请考虑系统中运行的 Kerberos 版本，以及为其创建服务器主体的服务器应用程序所支持的加密算法。例如，创建 `nfs/hostname` 服务主体时，应将加密类型限制为相应主机中的 NFS 服务器所支持的类型。请注意，在 Solaris 10 发行版中，NFS 服务器也支持所有受支持的 Kerberos 加密类型。
- `kdc.conf` 文件中的 `master_key_enctype` 参数可用于控制加密主体数据库中各项的主密钥的加密类型。如果已创建 KDC 主体数据库，请勿使用此参数。可在创建数据库时使用 `master_key_enctype` 参数，以便将缺省主密钥加密类型从 `des-cbc-crc` 更改为更强的加密类型。配置从 KDC 时，请确保所有从 KDC 都支持选择的加密类型，并且其 `kdc.conf` 中具有相同的 `master_key_enctype` 项。另外，如果在 `kdc.conf` 中设置了 `supported_enctypes`，则还应确保将 `master_key_enctype` 设置为 `supported_enctypes` 中的加密类型之一。如果未正确处理其中任一问题，则主 KDC 可能无法使用从 KDC。
- 在客户机上，可以控制客户机在通过 `krb5.conf` 中的多个参数从 KDC 获取票证时请求的加密类型。`default_tkt_enctypes` 参数用于指定客户机在通过 KDC 请求票证授予票证 (Ticket-Granting Ticket, TGT) 时要使用的加密类型。客户机可使用 TGT 以一种更有效的方式获取其他服务器票证。如果客户机使用 TGT 请求服务器票证（称为 TGS 请求），则设置 `default_tkt_enctypes` 将提供客户机对加密类型（用于保护客户机和 KDC 之间

的通信）的部分控制。请注意，default_tkt_enctypes 中指定的加密类型必须至少与 KDC 中存储的主体数据库中的一种主体密钥加密类型匹配。否则，TGT 请求将会失败。在大多数情况下，最好不要设置 default_tkt_enctypes，因为此参数会导致互操作性问题。缺省情况下，客户机代码会请求所有受支持的加密类型，而 KDC 会基于在主体数据库中找到的密钥来选择加密类型。

- default_tgs_enctypes 参数可限制客户机在其 TGS 请求（用于获取服务器票证）中请求的加密类型。另外，此参数还可限制 KDC 在创建客户机和服务器共享的会话密钥时使用的加密类型。例如，如果客户机要在执行安全 NFS 时仅使用 3DES 加密，则应将 default_tgs_enctypes 设置为 des3-cbc-sha1。请确保客户机主体和服务器主体在主体数据库中具有 des-3-cbc-sha1 密钥。与 default_tkt_enctype 一样，在大多数情况下，最好不要设置此参数，因为如果在 KDC 和服务器上未正确设置凭证，则此参数会导致互操作性问题。
- 在服务器上，可使用 kdc.conf 中的 permitted_enctypes 来控制服务器可接受的加密类型。此外，还可指定创建 keytab 项时使用的加密类型。同样，一般情况下最好不要使用其中任一方法来控制加密类型，而应由 KDC 来确定要使用的加密类型，因为 KDC 不会与服务器应用程序进行通信以确定要使用的密钥或加密类型。

使用 gsscred 表

如果缺省映射不满足要求，则 NFS 服务器尝试标识 Kerberos 用户时，将使用 gsscred 表。NFS 服务使用 UNIX ID 来标识用户。这些 ID 不属于用户主体或凭证。gsscred 表提供从 GSS 凭证到 UNIX UID 的其他映射（通过口令文件）。填充 KDC 数据库后，必须创建和管理该表。有关更多信息，请参见第 360 页中的 “将 GSS 凭证映射到 UNIX 凭证”。

接收到客户机请求时，NFS 服务便会尝试将凭证名称映射到 UNIX ID。如果映射失败，则会检查 gsscred 表。

Solaris Kerberos 和 MIT Kerberos 之间的显著差异

Kerberos 服务的 Solaris 10 版本基于 MIT Kerberos 版本 1.2.1。以下列出了 Solaris 10 发行版中提供的增强功能，在 MIT 1.2.1 版本中不会提供这些功能：

- Solaris 远程应用程序的 Kerberos 支持
- KDC 数据库的增量传播
- 客户机配置脚本
- 已本地化的错误消息
- BSM 审计记录支持
- Kerberos 通过 GSS-API 安全使用线程
- 使用密码学的加密框架

此版本还包括某些已发布的 MIT 1.2.1 错误修复。特别是，添加了 1.2.5 二叉树错误修复和 1.3 TCP 支持。

第 7 部分

Solaris 审计

本部分提供有关 Solaris 审计子系统的配置、管理和使用方法的信息。

◆ ◆ ◆ 第 27 章

Solaris 审计（概述）

Solaris 审计保留系统使用情况的记录。审计服务包括帮助分析审计数据的工具。

本章介绍在 Solaris 操作系统 中审计如何工作。以下是本章中信息的列表：

- 第 517 页中的 “什么是审计？”
- 第 518 页中的 “审计如何工作？”
- 第 519 页中的 “审计如何与安全相关？”
- 第 519 页中的 “审计术语和概念”
- 第 524 页中的 “Solaris 10 发行版中的 Solaris 审计增强功能”

有关规划建议，请参见第 28 章。有关在站点上配置审计的过程，请参见第 29 章。有关参考信息，请参见第 30 章。

什么是审计？

审计是指收集有关系统资源使用情况的数据。审计数据提供与安全相关的系统事件的记录。以后便可以使用此数据来指定主机上执行的操作的职责。成功的审计应包括两个安全功能：识别和验证。每次登录时，在用户提供用户名和口令之后，都将生成一个与此用户的进程关联的唯一审计会话 ID。登录会话期间启动的每个进程都会继承此审计会话 ID。即使用户在单个会话期间更改了身份，也会使用同一个审计会话 ID 跟踪所有的用户操作。有关更改身份的更多详细信息，请参见 su(1M) 手册页。

使用审计服务可以：

- 监视主机上发生的与安全相关的事件
- 记录网络范围内审计跟踪中的事件
- 检测误用或未经授权的活动
- 查看访问模式以及个人和对象的访问历史记录
- 发现绕过保护机制的尝试
- 发现用户更改身份时权限的扩展使用

在系统配置期间，可以预先选择要监视的审计记录类。还可以微调针对单个用户执行审计的程度。

收集完审计数据之后，便可使用后选工具来减少和检查所需的审计跟踪部分。例如，您可以选择查看单个用户或特定组的审计记录。可以在特定日期检查某个事件类型的所有记录。或者，可以选择在特定时间生成的记录。

安装非全局区域的系统可以从全局区域以相同的方式审计所有区域。还可以配置这些系统，使其收集非全局区域中的不同记录。有关更多信息，请参见第 579 页中的 “审计和 Solaris Zones” 。

审计如何工作？

审计在指定事件发生时生成审计记录。通常，生成审计记录的事件包括：

- 启动系统和关闭系统
- 登录和注销
- 创建进程或损毁进程，或者创建线程或损毁线程
- 打开、关闭、创建、销毁或重命名对象
- 使用权限功能或基于角色的访问控制 (role-based access control, RBAC)
- 识别操作和验证操作
- 由进程或用户执行的权限更改
- 管理操作，例如安装软件包
- 特定于站点的应用程序

审计记录从以下三个源生成：

- 应用程序
- 异步事件的结果
- 进程系统调用的结果

一旦捕获相关的事件信息，便会将此信息格式化为审计记录。然后将此记录写入审计文件。完整的审计记录以二进制格式存储。对于 Solaris 10 发行版，也可使用 `syslog` 实用程序记录审计记录。

以二进制格式存储的审计文件可以存储在本地分区中，也可以存储在挂载了 NFS 的文件服务器中。存储位置可以是同一系统上的多个分区、不同系统上的分区，或者相互链接的不同网络中系统上的分区。相互链接的审计文件的集合称为**审计跟踪**。审计记录在审计文件中按时间顺序累积。每条审计记录都包含识别事件的信息、导致事件的原因、事件发生的时间，以及其他相关信息。

审计记录还可以使用 `syslog` 实用程序进行监视。这些审计日志可以在本地存储。或者，可以通过 UDP 协议将这些日志发送到远程系统。有关更多信息，请参见第 522 页中的 “审计文件” 。

审计如何与安全相关？

Solaris 审计通过显示可疑的或异常的系统使用模式来帮助检测潜在的安全破坏。Solaris 审计还提供一种根据可疑操作追溯到特定用户的方法，因此可以起到一种威慑的作用。知道自己的操作正在被审计的用户很少会尝试执行恶意操作。

要保护计算机系统，特别是网络中的系统，需要在系统进程或用户进程开始之前具备控制活动的机制。安全性要求系统上安装有在活动发生时用于监视活动的工具，同时还要求在活动发生后报告活动。Solaris 审计的初始配置要求在用户登录或系统进程开始之前设置参数。大多数审计活动都涉及监视当前事件和报告符合指定参数的事件。Solaris 审计如何监视和报告这些事件的信息在第 28 章和第 29 章中详细介绍。

审计不能防止黑客未经授权的侵入。但是，审计服务可以报告特定用户在特定日期和时间执行了特定操作之类的信息。审计报告可以按登录路径和用户名来标识用户。此类信息可立即报告给终端和文件，以供以后分析。因此，审计服务提供的数据有助于确定以下内容：

- 系统安全如何受到威胁
- 需要关闭哪些漏洞来确保期望的安全级别

审计术语和概念

以下术语用于说明审计服务。某些定义包含更完整说明的链接。

表 27–1 Solaris 审计术语

术语	定义
Audit class（审计类）	一组审计事件。审计类提供选择一组要审计的事件的方法。有关更多信息，请参见第 521 页中的 “审计类和预选” 。
Audit directory（审计目录）	二进制格式的审计文件的仓库。有关审计目录类型的说明，请参见第 522 页中的 “审计文件” 。
Audit event（审计事件）	被审计的与安全相关的系统操作。为了便于选择，将事件分为各个审计类。有关可以审计的系统操作的介绍，请参见第 520 页中的 “审计事件” 。
Audit policy（审计策略）	一组可以在您的站点中启用或禁用的审计选项。这些选项包括是否记录某种审计数据，同时还包括是否在写满审计跟踪时暂停可审计的操作。有关更多信息，请参见第 531 页中的 “确定审计策略” 。
Audit record（审计记录）	存储在审计文件中的审计数据。一条审计记录描述一个审计事件。每条审计记录由多个审计标记组成。有关审计记录的更多信息，请参见第 521 页中的 “审计记录和审计标记” 。
Audit token（审计标记）	审计记录或审计事件字段。每个审计标记描述审计事件的一个属性，例如用户、程序或其他对象。有关所有审计标记的说明，请参见第 585 页中的 “审计标记格式” 。

表 27–1 Solaris 审计术语 （续）

术语	定义
Audit trail（审计跟踪）	一个或多个审计文件的集合，用于存储运行审计服务的所有系统上的审计数据。有关更多信息，请参见第 583 页中的 “审计跟踪” 。
Preselection（预选）	预选是指在启用审计服务之前选择要监视的审计类。预选的审计类的审计事件将会在审计跟踪中出现。由于不会审计未预选的审计类，因此这些审计类的事件将不会出现在审计跟踪中。后选工具 `auditreduce` 命令将从审计跟踪中选择记录。有关更多信息，请参见第 521 页中的 “审计类和预选” 。
Public object（公共对象）	公共对象是一个由 `root` 用户拥有且任何人都可读取的文件。例如，`/etc` 目录和 `/usr/bin` 目录中的文件就是公共对象。不能审计只读事件的公共对象。例如，即使预选了 `file_read (fr)` 审计类，也不会审计公共对象的读取。您可以通过更改 `public` 审计策略选项来覆盖缺省值。

审计事件

可以审计与安全相关的系统操作。这些可审计的操作称为**审计事件**。审计事件在 `/etc/security/audit_event` 文件中列出。在此文件中，会根据事件编号、符号名称、简短说明以及事件所属的审计类集定义每个审计事件。 有关 `audit_event` 文件的更多信息，请参见 `audit_event(4)` 手册页。

例如，以下项定义了 `exec()` 系统调用的审计事件：

```
7:AUE_EXEC:exec(2):ps,ex
```

当您预选审计类 `ps` 或审计类 `ex` 进行审计时，将在审计跟踪中记录 `exec()` 系统调用。

Solaris 审计可处理**可归属**事件和**无归属**事件。`exec()` 系统调用可归属到某个用户，因此可将该调用视为可归属事件。而在内核中断级别发生的事件则为无归属事件。对用户进行验证之前发生的事件也是无归属事件。`na` 审计类处理无归属审计事件。例如，引导系统便是一个无归属事件。

```
113:AUE_SYSTEMBOOT:system booted:na
```

预选某个审计事件所属的类以进行审计时，会在审计跟踪中记录此事件。例如，预选 `ps` 和 `na` 审计类以进行审计时，除其他事件之外，还会在审计跟踪中记录 `exec()` 系统调用和系统引导操作。

除 Solaris 审计服务定义的审计事件之外，第三方应用程序也可以生成审计事件。审计事件编号 32768 到 65535 适用于第三方应用程序。

审计类和预选

每个审计事件都属于一个或多个**审计类**。审计类是用于容纳大量审计事件的一种很方便的容器。**预选**要审计的类时，可以指定应在审计跟踪中记录该类中的所有事件。可以预选系统中的事件和特定用户启动的事件。运行审计服务之后，可以从预选类中动态添加或删除审计类。

- **系统范围预选**－在 `audit_control` 文件的 `flags`、`naflags` 以及 `plugin` 行中指定系统范围内的缺省值以进行审计。`audit_control` 文件在第 575 页中的 “audit_control 文件”中介绍。另请参见 `audit_control`(4) 手册页。
- **用户特定预选**－在 `audit_user` 数据库中针对个别用户指定除系统范围内的审计缺省值之外的其他值。

  审计预选掩码确定要针对用户审计的事件类。用户的审计预选掩码是系统范围内的缺省值和针对用户指定的审计类的组合。有关更多详细信息，请参见第 582 页中的 “进程审计特征”。

  `audit_user` 数据库可以从本地管理，也可以通过名称服务来管理。Solaris Management Console 可提供图形用户界面 (graphical user interface, GUI) 来管理此数据库。有关详细信息，请参见 `audit_user`(4) 手册页。
- **动态预选**－将审计类指定为 `auditconfig` 命令的参数，以便从进程或会话中添加或删除这些审计类。有关更多信息，请参见 `auditconfig`(1M) 手册页。

可以使用后选命令 `auditreduce` 从预选审计记录中选择记录。有关更多信息，请参见第 524 页中的 “检查审计跟踪”和 `auditreduce`(1M) 手册页。

审计类在 `/etc/security/audit_class` 文件中定义。每个项都包含类的审计掩码、类的名称，以及类的描述性名称。例如，在 `audit_class` 文件中 `ps` 和 `na` 类的定义为：

```
0x00100000:ps:process start/stop

0x00000400:na:non-attribute
```

有 32 种可能的审计类。其中包括两个全局类：`all` 和 `no`。这些审计类将在 `audit_class`(4) 手册页中介绍。

可以配置审计事件到类的映射。可以从类中删除事件、向类中添加事件，以及创建新类以包含选定事件。有关过程，请参见第 544 页中的 “如何更改审计事件的类成员关系”。

审计记录和审计标记

每条**审计记录**记录一个审计事件的发生。此记录包含操作执行者、受影响的文件、已尝试执行的操作以及操作发生的时间和位置等信息。以下示例给出了一条 `login` 审计记录：

```
header,81,2,login - local,,2003-10-13 11:23:31.050 -07:00

subject,root,root,other,root,other,378,378,0 0 example_system
```

```
text,successful login

return,success,0
```

为每个审计事件保存的信息类型由一组**审计标记**进行定义。每次为事件创建审计记录时，记录中都会包含为事件定义的部分或全部标记。事件的性质确定要记录的标记。在前面的示例中，每行都以审计标记的名称开头。名称后跟审计标记的内容。`login` 审计记录共由四种审计标记组成。

有关每个审计标记结构的详细说明和 `praudit` 输出示例，请参见第 585 页中的 “审计标记格式”。有关审计标记的二进制流的说明，请参见 `audit.log`(4) 手册页。

审计文件

审计记录收集在审计日志中。Solaris 审计为审计日志提供两种输出模式。称为**审计文件**的日志以二进制格式存储审计记录。系统或站点的审计文件组提供完整的审计记录。完整的审计记录称为**审计跟踪**。

`syslog` 实用程序收集并存储审计记录的文本版本的摘要。`syslog` 记录不是完整的记录。以下示例给出了 `login` 审计记录的 `syslog` 项：

```
Oct 13  11:24:11 example_system auditd: [ID 6472 audit.notice] \

        login - login ok session 378 by root as root:other
```

一个站点可以使用两种格式存储审计记录。可以配置站点中的系统，使其可以使用二进制模式，或者可以同时使用这两种模式。下表对二进制审计记录和 `syslog` 审计记录进行了比较。

表 27–2 二进制审计记录和 `syslog` 审计记录的比较

功能	二进制记录	`syslog` 记录
协议	写入文件系统	将 UDP 用于远程日志
数据类型	二进制	文本
记录长度	无限制	每条审计记录最多 1024 个字符
位置	存储在本地磁盘上以及使用 NFS 挂载的目录中	存储在 `syslog.conf` 文件中指定的位置
配置方式	编辑 `audit_control` 文件，并保护 NFS 挂载审计目录	编辑 `audit_control` 文件，编辑 `syslog.conf` 文件

表 27–2 二进制审计记录和 syslog 审计记录的比较 （续）

功能	二进制记录	syslog 记录
读取方式	通常，以批处理模式读取 XML 格式的浏览器输出	实时读取，或者通过为 syslog 创建的脚本进行搜索 纯文本输出
完整性	保证完整，并且以正确的顺序显示	不能保证完整
时间标记	格林威治标准时间 (Greenwich Mean Time, GMT)	被审计的系统时间

二进制记录提供最高的安全性和最大的范围。二进制输出满足安全证书的要求，例如公共标准受控制访问保护配置 (Controlled Access Protection Profile, CAPP)。这些记录将写入被保护不受窥探的文件系统中。在单独的系统上，将收集所有二进制记录并按顺序显示它们。当某个审计跟踪内的各系统跨时间区域分布时，便可使用二进制日志中的 GMT 时间标记进行精确比较。使用 praudit -x 命令可在浏览器中查看 XML 格式的记录。还可以使用脚本来解析 XML 输出。

相反，syslog 记录可以提供更大的便利性和灵活性。例如，您可以从各种源收集 syslog 数据。此外，当您监视 syslog.conf 文件中的 audit.notice 事件时，syslog 实用程序会记录一条带有当前时间标记的审计记录摘要。您可以使用为来自各种源（包括工作站、服务器、防火墙，以及路由器）的 syslog 消息开发的管理和分析工具。可以实时查看记录，并将其存储在远程系统中。

通过使用 syslog.conf 远程存储审计记录，可以保护日志数据免遭攻击者改动或删除。另一方面，远程存储审计记录时，这些记录容易遭受拒绝服务、伪装源地址等网络攻击。此外，UDP 会丢包或无序发送包。syslog 项的限制为 1024 个字符，所以可能截断日志中的某些审计记录。在单独的系统上，并非收集所有的审计记录。可能不会按顺序显示记录。由于每条审计记录都使用本地系统的日期和时间进行标记，因此不能根据时间标记为多个系统构造审计跟踪。

有关审计日志的更多信息，请参阅以下内容：

- audit_syslog(5) 手册页
- audit.log(4) 手册页
- 第 539 页中的 “如何配置 syslog 审计日志”

审计存储

审计目录以二进制格式保留审计文件。典型安装使用许多审计目录。所有审计目录的内容组成了**审计跟踪**。审计记录按以下顺序存储在审计目录中：

- **主审计目录** – 正常情况下放置系统审计文件的目录
- **辅助审计目录** – 主审计目录已满或不可用时放置系统审计文件的目录
- **最后使用的目录** – 主审计目录和所有辅助审计目录都不可用时使用的本地审计目录

这些目录在 audit_control 文件中指定。列表中早于此目录的目录已满时才会使用该目录。有关带有目录项列表的 audit_control 注释文件的信息，请参见示例 29–3。

检查审计跟踪

审计服务提供用于合并和减少审计跟踪文件的命令。auditreduce 命令可以合并审计跟踪中的审计文件。此命令还可以过滤文件以查找特定事件。praudit 命令读取二进制文件。praudit 命令的选项提供适合借助脚本和浏览器显示的输出。

Solaris 10 发行版中的 Solaris 审计增强功能

从 Solaris 9 发行版开始，已将以下功能引入 Solaris 审计中：

- Solaris 审计可以使用 syslog 实用程序以文本格式存储审计记录。有关介绍，请参见第 522 页中的 “审计文件”。有关设置 audit_control 文件以使用 syslog 实用程序的信息，请参见第 539 页中的 “如何配置 syslog 审计日志”。
- praudit 命令具有另外一种输出格式 XML。XML 是标准的可移植、可处理格式。XML 格式使得输出能够在浏览器中读取，并为报告的 XML 脚本提供数据源。praudit 命令的 -x 选项在第 572 页中的 “praudit 命令”中介绍。
- 已重新构造缺省的审计类集。审计元类为更加细分的审计类提供了一种保护。有关缺省类集列表的信息，请参见第 580 页中的 “审计类的定义”。
- bsmconv 命令不再禁用 Stop-A 键。可以审计 Stop-A 事件。
- 审计记录中的时间标记以 ISO 8601 格式报告。有关标准的信息，请访问 http://www.iso.org。
- 添加了三个审计策略选项：
  - **public** – 不再审计只读事件的公共对象。由于不再审计公共文件，因而审计日志的大小将显著减小。对敏感文件的读取尝试将更容易监视。有关公共对象的更多信息，请参见第 519 页中的 “审计术语和概念”。
  - **perzone** – perzone 策略具有广泛的影响。在每个区域中运行单独的审计守护进程。此守护进程使用特定于相应区域的审计配置文件。审计队列也特定于该区域。有关详细信息，请参见 auditd(1M) 和 auditconfig(1M) 手册页。有关区域的更多信息，请参见第 579 页中的 “审计和 Solaris Zones”。有关策略的更多信息，请参见第 528 页中的 “如何在区域中规划审计”。
  - **zonename** – 其中发生的审计事件包括在审计记录中的 Solaris 区域的名称。有关区域的更多信息，请参见第 579 页中的 “审计和 Solaris Zones”。有关何时使用选项的介绍，请参见第 531 页中的 “确定审计策略”。
- 添加了五个审计标记：
  - cmd 标记记录参数列表和与命令关联的环境变量的列表。有关更多信息，请参见第 589 页中的 “cmd 标记”。
  - path_attr 标记记录 path 标记对象下的属性文件对象的序列。有关更多信息，请参见第 594 页中的 “path_attr 标记”。

- `privilege` 标记记录进程中使用的权限。有关更多信息，请参见第 595 页中的“`privilege` 标记”。
- `uauth` 标记记录在命令或操作中使用的授权。有关更多信息，请参见第 600 页中的“`uauth` 标记”。
- `zonename` 标记记录发生审计事件的非全局区域的名称。`zonename` 审计策略选项确定 `zonename` 标记是否包括在审计记录中。有关更多信息，请参见第 601 页中的“`zonename` 标记”。

  有关概述信息，请参见第 579 页中的“审计和 Solaris Zones”。要了解有关区域的信息，请参见《System Administration Guide: Solaris Containers-Resource Management and Solaris Zones》中的第二部分，“Zones”。

◆ ◆ ◆ 第 28 章

规划 Solaris 审计

本章介绍如何针对 Solaris 安装设置审计服务。特别是，本章介绍了您在启用审计服务之前需要考虑的问题。以下是本章中规划信息的列表：

- 第 527 页中的 “规划 Solaris 审计（任务列表）”
- 第 531 页中的 “确定审计策略”
- 第 533 页中的 “控制审计成本”
- 第 534 页中的 “有效审计”

有关审计的概述，请参见第 27 章。有关在站点上配置审计的过程，请参见第 29 章。有关参考信息，请参见第 30 章。

规划 Solaris 审计（任务列表）

以下任务列表介绍了规划磁盘空间及要记录的事件时所需执行的主要任务。

任务	参考
确定非全局区域的审计策略	第 528 页中的 “如何在区域中规划审计”
规划审计跟踪的存储空间	第 529 页中的 “如何规划审计记录的存储”
确定要审计的对象及内容	第 529 页中的 “如何规划要审计的对象及内容”

规划 Solaris 审计（任务）

您需要选择审计的活动类型，同时需要收集有用的审计信息。审计文件不断增大，可能会很快占满可用空间，因此应该分配足够的磁盘空间。您还需要仔细规划要审计的对象及内容。

▼ 如何在区域中规划审计

如果您的系统实现了区域，则可以进行两种审计配置：

- 可以单独配置非全局区域。
- 可以在全局区域中为所有区域配置审计。

1 确定是否要在非全局区域中自定义审计。

- **如果不想在非全局区域中自定义审计，请转至步骤 2。**
- **如果要在非全局区域中自定义审计，请考虑以下事项：**
  - 还必须配置全局区域。

    要根据非全局区域中的审计配置文件收集审计记录，必须在全局区域中设置 perzone 审计策略。

    注 – 如果使用自定义名称服务文件实现非全局区域，则应设置 perzone 审计策略选项。名称服务文件包括 /etc/password、/etc/shadow 和 nsswitch.conf。有关不设置 perzone 选项的信息，请参见第 579 页中的 “审计和 Solaris Zones”。

  - 区域中的审计配置文件由此区域的审计守护进程读取。

    每个区域都运行自己的审计守护进程，具有自己的审计队列，并收集自己的审计日志。这些操作由计算机集中执行。
  - 各个区域都可以设置除 perzone 和 ahlt 以外的所有策略选项。这些策略选项在全局区域中设置。

  如果在每个区域中自定义审计配置文件，应使用第 529 页中的 “如何规划要审计的对象及内容”规划每个区域。可以跳过第一步。还必须使用第 529 页中的 “如何规划审计记录的存储”规划每个区域。

2 确定是否需要单映像审计跟踪。

单映像审计跟踪将正在审计的系统视为一台计算机。全局区域会在系统上运行唯一的审计守护进程，并收集每个区域的审计日志。仅可在全局区域中自定义审计配置文件。

此配置将所有区域视为单一系统的一部分。要区分区域审计记录，可以设置 zonename 策略。然后，可以使用 auditreduce 命令，按区域来选择审计事件。有关示例，请参见 auditreduce(1M) 手册页。

要规划单映像审计跟踪，请使用第 529 页中的 “如何规划要审计的对象及内容”进行规划。从第一步开始。还必须使用第 529 页中的 “如何规划审计记录的存储”规划每个区域。

▼ 如何规划审计记录的存储

审计跟踪需要专用文件空间。审计文件的专用文件空间必须可用且安全。各个系统都应具有多个为审计文件配置的审计目录。作为首要任务之一，在任何系统上启用审计之前，都应决定如何配置审计目录。以下过程介绍了规划审计跟踪存储时要解决的问题。

开始之前 如果要实现非全局区域，请在使用此过程之前完成第 528 页中的 “如何在区域中规划审计”。

1. **确定您站点所需的审计量。**

   针对审计跟踪平衡磁盘空间可用性和站点的安全需求。

   有关如何在保持站点安全的同时降低空间需求，以及如何设计审计存储的指南，请参见第 533 页中的 “控制审计成本”和第 534 页中的 “有效审计”。

2. **确定要审计的系统。**

   在这些系统上，至少为一个本地审计目录分配空间。有关如何指定审计目录的信息，请参见示例 29–3。

3. **确定要存储审计文件的系统。**

   确定要保存主审计目录和辅助审计目录的服务器。有关为审计目录配置磁盘的示例，请参见第 547 页中的 “如何创建审计文件的分区”。

4. **命名审计目录。**

   创建计划使用的所有审计目录的列表。有关命名约定，请参见第 583 页中的 “二进制审计文件名称约定”。

5. **确定哪些系统要使用哪些审计目录。**

   创建一个列表，显示哪个系统应使用哪个审计目录。此列表有助于您平衡审计活动。有关图解，请参见图 30–1和图 30–2。

▼ 如何规划要审计的对象及内容

开始之前 如果要实现非全局区域，请在使用此过程之前完成第 528 页中的 “如何在区域中规划审计”。

1. **确定是否需要单映像审计跟踪。**

   如果计划以不同的方式审计各个系统，请从下一步开始。应该针对每个系统完成其余的规划步骤。

   单映像审计跟踪将正在审计的系统视为一台计算机。要为某个站点创建单映像审计跟踪，安装中的每个系统都应进行如下配置：

   - 与其他所有系统使用相同的 `audit_warn`、`audit_event` 、`audit_class` 和 `audit_startup` 文件。

- 使用相同的 audit_user 数据库。此数据库可以位于名称服务中。
- 在 audit_control 文件中具有相同的 flags、naflags 和 plugin 项。

2 确定审计策略。

使用 auditconfig -lspolicy 命令查看可用策略选项的简短说明。缺省情况下，仅打开 cnt 策略。有关更全面的介绍，请参见步骤 8。

有关策略选项的影响，请参见第 531 页中的 “确定审计策略”。有关如何设置审计策略的信息，请参见第 550 页中的 “如何配置审计策略”。

3 确定是否要修改事件到类的映射。

在多数情况下，缺省映射便已够用。但是，如果添加新类、更改类定义，或者确定某特定系统调用的记录无用，则可能需要将某个事件移动到其他类。

有关示例，请参见第 544 页中的 “如何更改审计事件的类成员关系”。

4 确定要预选的审计类。

添加审计类或更改缺省类的最佳时机是在启动审计服务之前。

audit_control 文件中 flags、naflags 和 plugin 项的审计类值适用于所有用户和进程。预选类可以确定是只针对成功情况对审计类进行审计，还是只针对失败情况对其进行审计，或者同时针对两种情况对其进行审计。

有关如何预选审计类的信息，请参见第 536 页中的 “如何修改 audit_control 文件”。

5 确定系统范围预选审计类的用户例外情况。

如果决定使用系统范围预选审计类以外的方式来审计某些用户，请修改 audit_user 数据库中单个用户项。

有关示例，请参见第 541 页中的 “如何更改用户审计特征”。

6 确定最小可用磁盘空间。

当审计文件系统上的磁盘空间低于 minfree 百分比时，auditd 守护进程将切换到下一个可用审计目录。然后，此守护进程将发送一条警告，指出已超过软限制。

有关如何设置最小可用磁盘空间的信息，请参见示例 29–4。

7 决定如何管理 audit_warn 电子邮件别名。

只要审计系统需要通知您出现了需要管理干预的情况，就会运行 audit_warn 脚本。缺省情况下，audit_warn 脚本会向 audit_warn 别名发送电子邮件，并向控制台发送消息。

要设置别名，请参见第 550 页中的 “如何配置 audit_warn 电子邮件别名”。

8 决定当所有审计目录已满时需要执行的操作。

缺省情况下，当审计跟踪溢出时，系统还会继续工作。系统会对已删除的审计记录进行计数，但是不会记录事件。要获得更大的安全性，可以禁用 cnt 策略，同时启用 ahlt 策略。当审计跟踪溢出时，ahlt 策略将停止系统。

有关如何配置这些策略选项的信息，请参见示例 29–14。

9 决定是否收集 `syslog` 格式以及二进制日志格式的审计记录。

有关概述信息，请参见第 522 页中的 “审计文件”。

有关示例，请参见第 539 页中的 “如何配置 `syslog` 审计日志”。

确定审计策略

审计策略确定本地系统审计记录的特征。策略选项由启动脚本设置。启用审计服务的 `bsmconv` 脚本会创建 `/etc/security/audit_startup` 脚本。`audit_startup` 脚本执行 `auditconfig` 命令来建立审计策略。有关此脚本的详细信息，请参见 `audit_startup`(1M) 手册页。

缺省情况下，禁用大多数审计策略选项来最大程度地减少存储需求和系统处理需求。您可以使用 `auditconfig` 命令动态启用和禁用审计策略选项。可以使用 `audit_startup` 脚本永久启用和禁用策略选项。

使用下表确定您站点的需求是否可以解释启用一个或多个审计策略选项而造成的额外开销。

表 28–1 审计策略选项的影响

策略名称	说明	更改策略选项的原因
`ahlt`	此策略仅适用于异步事件。禁用后，此策略允许在不生成审计记录的情况下完成事件。 启用后，此策略会在审计文件系统已满时停止系统。需要管理干预才能清除审计队列、为审计记录提供空间，以及重新引导系统。只能在全局区域中启用此策略。此策略影响所有区域。	当系统可用性比安全性更重要时，适合使用禁用选项。 在安全性极为重要的环境中，适合使用启用选项。
`arge`	禁用后，此策略将从 exec 审计记录中忽略已执行程序的环境变量。 启用后，此策略会将已执行程序的环境变量添加到 exec 审计记录。与禁用此策略的情况相比，生成的审计记录包含更多详细信息。	与启用选项相比，禁用选项收集的信息要少很多。 当审计数个用户时，适合使用启用选项。怀疑 exec 程序中使用的环境变量有问题时，也可以使用此选项。
`argv`	禁用后，此策略将从 exec 审计记录中忽略已执行程序的参数。 启用后，此策略会将已执行程序的参数添加到 exec 审计记录。与禁用此策略的情况相比，生成的审计记录包含更多详细信息。	与启用选项相比，禁用选项收集的信息要少很多。 当审计数个用户时，适合使用启用选项。当您确信所运行的 exec 程序异常时，也可以使用此选项。

表 28–1 审计策略选项的影响 （续）

策略名称	说明	更改策略选项的原因
cnt	禁用后，此策略将阻止用户或应用程序运行。由于没有可用磁盘空间而导致审计记录无法添加到审计跟踪时，会发生阻止。 启用后，此策略允许在不生成审计记录的情况下完成事件。此策略维护已删除审计记录的计数。	在安全性极为重要的环境中，适合使用禁用选项。 当系统可用性比安全性更重要时，适合使用启用选项。
group	禁用后，此策略不会在审计记录中添加组列表。 启用后，此策略会在每条审计记录中添加组列表作为特殊标记。	禁用选项通常可以满足站点的安全要求。 当需要审计哪些组正在生成审计事件时，适合使用启用选项。
path	禁用后，此策略会在每条审计记录中最多记录一条在系统调用期间所使用的路径。 启用后，此策略会将与审计事件结合使用的每条路径记录到每条审计记录中。	禁用选项在审计记录中最多放置一条路径。 启用选项会将系统调用期间使用的每个文件名或路径输入到审计记录中，作为 path 标记。
perzone	禁用后，此策略针对每个系统只维护一个审计配置。全局区域中运行一个审计守护进程。通过预选 zonename 审计标记，可在审计记录中找到非全局区域中的审计事件。 启用后，此策略会为每个区域维护单独的审计配置、审计队列和审计日志。每个区域中运行单独的审计守护进程版本。只能在全局区域中启用此策略。	当您没有特殊的理由为每个区域维护单独的审计日志、队列和守护进程时，禁用选项很有用。 当您无法仅通过预选 zonename 审计标记来有效监视系统时，启用选项很有用。
public	禁用后，如果预选文件的读取，则此策略不会将公共对象的只读事件添加到审计跟踪。包含只读事件的审计类包括 fr、fa 和 cl。 启用后，如果预选了适当的审计类，则此策略会记录公共对象的每个只读审计事件。	禁用选项通常可以满足站点的安全要求。 启用选项很少有用。
seq	禁用后，此策略不会将序列号添加到每条审计记录中。 启用后，此策略会将序列号添加到每条审计记录中。sequence 标记保留序列号。	当审计顺利进行时，禁用选项便已够用。 当启用 cnt 策略后，适合使用启用选项。利用 seq 策略，可以确定废弃数据的时间。
trail	禁用后，此策略不会将 trailer 标记添加到审计记录中。 启用后，此策略会将 trailer 标记添加到每条审计记录中。	禁用选项会创建一条较短的审计记录。 启用选项会使用 trailer 标记清晰地标记每条审计记录的结束。trailer 标记经常与 sequence 标记结合使用。trailer 标记可以使审计记录的重新同步更简单、更精确。
zonename	禁用后，此策略不会在审计记录中包括 zonename 标记。 启用后，此策略会在非全局区域的每条审计记录中包括 zonename 标记。	当无需跨区域比较审计行为时，禁用选项很有用。 当需要区分各区域中的审计行为并对其进行比较时，启用选项很有用。

控制审计成本

由于审计会占用系统资源，因此必须控制所记录内容的详细程度。当您决定要审计的内容时，请考虑以下审计成本：

- 延长处理时间带来的成本
- 分析审计数据的成本
- 存储审计数据的成本

延长审计数据的处理时间带来的成本

延长处理时间带来的成本是审计成本中最不重要的部分。第一个原因是在执行需要大量运算的任务（例如映像处理、复杂计算等）时一般不会进行审计。另一个原因是单用户系统的成本通常会小到可以忽略。

分析审计数据的成本

分析成本大致上与收集的审计数据量成比例。分析成本包括合并与查看审计记录所需的时间，还包括将记录进行归档并保存在安全位置所需的时间。

生成的记录越少，分析审计跟踪数据所需的时间就越短。下面的第 533 页中的 “存储审计数据的成本”和第 534 页中的 “有效审计”部分介绍了高效进行审计的方法。有效的审计可减少审计数据量，同时仍保证足够的审计范围以实现站点的安全目标。

存储审计数据的成本

存储成本是最重要的审计成本。审计数据量取决于以下各项：

- 用户数
- 系统数
- 使用量
- 所需的可跟踪与可说明的程度

由于上述因素随站点不同而不同，因此没有公式可预先确定为审计数据存储预留的磁盘空间量。请遵循以下原则：

- 审慎地预选审计类，以减少生成的记录量。

  全部审计（即使用 `all` 类）会使磁盘空间很快被占满。即使编译程序之类的简单任务也可能生成很大的审计文件。一般的程序在一分钟之内可能会生成数以千计的审计记录。

  例如，通过忽略 `file_read` 审计类 `fr`，可以显著减少审计量。通过选择仅针对失败操作进行审计，有时也会减少审计量。例如，与针对所有 `file_read` 事件进行审计相比，针对失败的 `file_read` 操作（即 `-fr`）进行审计而生成的记录会少很多。
- 有效的审计文件管理也很重要。创建审计记录后，通过文件管理可减少所需的存储量。

- 了解审计类

  配置审计之前，应该了解类中包含的事件类型。可以更改审计事件到类的映射来优化审计记录的收集。

- 设计针对您站点的审计方案。

  以可获知的因素为基础，设计您的审计方案。此类度量包括站点所需的可跟踪量，以及管理的用户类型。

有效审计

以下技术可帮助您在更有效地进行审计的同时实现组织的安全目标。

- 任何时刻均只对特定百分比的用户同时进行随机审计。
- 通过合并、减少和压缩文件来降低审计文件的磁盘存储需求。制订对文件进行归档、将文件传送到可移动介质和脱机存储文件的过程。
- 实时监视审计数据有无异常行为。您可以扩展已经开发的管理和分析工具，以便处理 `syslog` 文件中的审计记录。

  您还可以设置监视某些活动的审计跟踪的过程。可以编写这样一个脚本，在它检测到异常事件时，以触发自动增加对特定用户或特定系统的审计作为响应。

  例如，可以编写执行以下操作的脚本：

  1. 监视所有审计文件服务器上审计文件的创建。
  2. 使用 `tail` 命令处理审计文件。

     通过对 `tail -0f` 命令输出内容实施 `praudit` 命令管道操作，可以在生成记录时产生审计记录流。有关更多信息，请参见 `tail(1)` 手册页。
  3. 分析此流以查看是否存在异常消息类型或其他指示符，并将分析结果提供给审计者。

     或者，可以使用脚本来触发自动响应。
  4. 经常监视审计目录，以查看是否有新的 `not_terminated` 审计文件出现。
  5. 如果仍在运行的 `tail` 进程不再向其文件中写入信息，请终止这些进程。

◆ ◆ ◆ 第 29 章

管理 Solaris 审计（任务）

本章介绍有助于设置和管理所审计的 Solaris 系统的过程，同时还包括管理审计跟踪的说明。以下是本章中信息的列表：

- 第 535 页中的 “Solaris 审计（任务列表）”
- 第 536 页中的 “配置审计文件（任务列表）”
- 第 546 页中的 “配置和启用审计服务（任务列表）”
- 第 557 页中的 “管理审计记录（任务列表）”

有关审计服务的概述，请参见第 27 章。有关规划建议的信息，请参见第 28 章。有关参考信息，请参见第 30 章。

Solaris 审计（任务列表）

以下任务列表介绍了管理审计所需执行的主要任务。这些任务是有序的。

任务	说明	参考
1. 规划审计	包含在配置审计服务之前要决定的配置问题。	第 527 页中的 “规划 Solaris 审计（任务列表）”
2. 配置审计文件	定义需要审计的事件、类和用户。	第 536 页中的 “配置审计文件（任务列表）”
3. 配置和启用审计	针对磁盘空间需求和其他审计服务要求配置每台主机。然后，启动审计服务。	第 546 页中的 “配置和启用审计服务（任务列表）”
4. 管理审计记录	收集并分析审计数据。	第 557 页中的 “管理审计记录（任务列表）”

配置审计文件（任务列表）

以下任务列表介绍了在站点上配置文件以自定义审计的过程。大多数任务是可选的。

任务	说明	参考
选择审计类，并自定义 `audit_control` 设置	涉及以下内容： ■ 预选系统范围的审计类 ■ 指定每个系统的审计目录 ■ 对审计文件系统设置磁盘空间限制	第 536 页中的 “如何修改 `audit_control` 文件”
（可选）以两种模式记录审计事件	除了存储二进制格式的审计记录之外，还可以实时监视审计事件。	第 539 页中的 “如何配置 `syslog` 审计日志”
（可选）更改用户的审计特征	设置系统范围的预选审计类的特定于用户的例外情况。	第 541 页中的 “如何更改用户审计特征”
（可选）添加审计类	通过创建用来保存事件的新审计类来减少审计记录数目。	第 543 页中的 “如何添加审计类”
（可选）更改事件到类的映射	通过更改事件到类的映射来减少审计记录数目。	第 544 页中的 “如何更改审计事件的类成员关系”

配置审计文件

在网络中启用审计之前，可以针对站点的审计要求自定义审计配置文件。另外，还可以在启用审计服务之后重新启动审计服务或重新引导本地系统，以读取已更改的配置文件。但是，推荐的做法是在启动审计服务前尽可能多地自定义审计配置。

如果已实现区域，则可以选择从全局区域审计所有区域。要区分审计输出中的区域，可以设置 `zonename` 策略选项。或者，可以在全局区域中设置 `perzone` 策略，并在非全局区域中自定义审计配置文件，以单独审计非全局区域。有关概述，请参见第 579 页中的 “审计和 Solaris Zones”。有关规划的信息，请参见第 528 页中的 “如何在区域中规划审计”。

▼ 如何修改 `audit_control` 文件

`/etc/security/audit_control` 文件配置系统范围的审计。此文件可确定审计的事件，发出审计警告的时间，以及审计文件的位置。

1 承担主管理员角色，或成为超级用户。

主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2 （可选的）保存 audit_control 文件的副本。

```
# cp /etc/security/audit_control /etc/security/audit_control.orig
```

3 修改站点的 audit_control 文件。

每一项都具有以下格式：

keyword:*value*

keyword 定义行的类型。类型包括 dir、flags、minfree、naflags 和 plugin。dir 行可以重复。

有关关键字的说明，请参见以下示例。有关 plugin 项的示例，请参见第 539 页中的 “如何配置 syslog 审计日志”。

value 指定与行类型相关联的数据。

示例 29–1 预选所有用户的审计类

audit_control 文件中的 flags 行定义了针对系统上所有用户审计的可归属事件的类。这些类用逗号分隔。允许使用空格。在本示例中，将为所有用户审计 lo 类中的事件。

```
# audit_control file

dir:/var/audit

flags:lo

minfree:20

naflags:lo
```

要查看哪些事件位于 lo 类中，请读取 audit_event 文件。另外，还可以使用 bsmrecord 命令，如示例 29–22 中所示。

示例 29–2 预选无归属事件

在本示例中，将审计 na 类中的所有事件，以及所有无归属的 login 事件。

```
# audit_control file

dir:/var/audit

flags:lo

minfree:20

naflags:lo,na
```

示例 29–3 指定二进制审计数据的位置

audit_control 文件中的 dir 行列出了要用于二进制审计数据的审计文件系统。在本示例中，定义了三个存放二进制审计数据的位置。

```
# audit_control file

#

# Primary audit directory - NFS-mounted from audit server

dir:/var/audit/egret.1/files

#

# Secondary audit directory - NFS-mounted from audit server

dir:/var/audit/egret.2/files

#

# Directory of last resort local directory

dir:/var/audit

flags:lo

minfree:20

naflags:lo,na

plugin:
```

要设置文件系统来保存二进制审计数据，请参见第 547 页中的 “如何创建审计文件的分区”。

示例 29–4 更改警告的软限制

在本示例中，设置了所有审计文件系统的最低空闲空间级别，以便在文件系统的可用率只有 10% 时发出警告。

```
# audit_control file

#

dir:/var/audit/examplehost.1/files
```

```
dir:/var/audit/examplehost.2/files

dir:/var/audit/localhost/files

flags:lo

minfree:10

naflags:lo,na
```

audit_warn 别名可接收警告。要设置别名，请参见第 550 页中的 “如何配置 audit_warn 电子邮件别名”。

▼ 如何配置 syslog 审计日志

可以指示审计服务只收集二进制审计数据，也可以指示审计服务收集二进制数据和文本数据。在以下过程中，将收集二进制审计数据和文本审计数据。收集的文本审计数据是二进制数据的子集。

开始之前 必须在 audit_control 文件的 flags 行或 naflags 行上指定预选审计类。文本数据是预选二进制数据的子集。

1. **承担主管理员角色，或成为超级用户。**

   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2. **（可选的）保存 audit_control 文件的副本。**

   ```
   # cp /etc/security/audit_control /etc/security/audit_control.save
   ```

3. **添加 plugin 项。**

   审计服务中的插件实现审计数据的二进制输出和 syslog 输出。未指定二进制插件。必须指定 syslog 插件。有关更多信息，请参见第 569 页中的 “auditd 守护进程”。

   plugin 项具有以下格式：

   plugin:name=*value*; p_flags=*classes*

   value 列出要使用的插件的名称。当前，唯一有效的值是 audit_syslog.so.1 插件。

   classes 列出在 flags 行和 naflags 行中指定的审计类的子集。

   有关 plugin 值的更多信息，请参见 audit_syslog(5) 手册页。

4 **在 syslog.conf 文件中添加 audit.notice 项。**

此项包括日志文件的位置。

```
# cat /etc/syslog.conf

...

audit.notice       /var/adm/auditlog
```

不应将文本日志和二进制审计文件存储在同一位置。auditreduce 命令假设审计分区中的所有文件都是二进制审计文件。

5 **创建日志文件。**

```
# touch /var/adm/auditlog
```

6 **刷新 syslog 服务的配置信息。**

```
# svcadm refresh system/system-log
```

7 **定期归档 syslog 日志文件。**

审计服务可生成大量输出。要管理日志，请参见 logadm(1M) 手册页。

示例 29–5 指定 syslog 输出的审计类

在以下示例中，syslog 实用程序收集预选审计类的子集。

```
# audit_control file

dir:/var/audit/host.1/files

dir:/var/audit/host.2/files

dir:/var/audit/localhost/files

flags:lo,ss

minfree:10

naflags:lo,na

plugin:name=audit_syslog.so.1; p_flags=-lo,-na,-ss
```

flags 和 naflags 项会指示系统收集所有二进制格式的登录/退出、无归属以及系统状态更改的审计记录。plugin 项会指示 syslog 实用程序仅收集失败登录、失败的无归属事件以及失败的系统状态更改。

示例 29–6 将 syslog 审计记录放在远程系统上

可以通过更改 syslog.conf 文件中的 audit.notice 项来指向远程系统。在本示例中，本地系统的名称为 example1。远程系统的名称为 remote1。

```
example1 # cat /etc/syslog.conf

...

audit.notice       @remote1
```

remote1 系统上 syslog.conf 文件中的 audit.notice 项指向日志文件。

```
remote1 # cat /etc/syslog.conf

...

audit.notice       /var/adm/auditlog
```

▼ 如何更改用户审计特征

每个用户的定义都存储在 audit_user 数据库中。这些定义为指定的用户修改 audit_control 文件中的预选类。nsswitch.conf 文件确定是否使用了本地文件或名称服务数据库。要计算用户的最终审计预选掩码，请参见第 582 页中的 “进程审计特征” 。

1 承担主管理员角色，或成为超级用户。

主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2 （可选的）保存 audit_user 数据库的副本。

```
# cp /etc/security/audit_user /etc/security/audit_user.orig
```

3 在 audit_user 数据库中添加新项。

在本地数据库中，每一项都具有以下格式：

username:*always-audit*:*never-audit*

username 选择要审计的用户的名称。

always-audit 选择总是应该针对指定用户进行审计的审计类列表。

never-audit 选择绝对不要针对指定用户进行审计的审计类列表。

可以通过用逗号分隔审计类来指定多个类。

audit_user 项将在用户下一次登录时生效。

示例 29–7 更改针对单个用户进行审计的事件

在本示例中，audit_control 文件包含系统的预选审计类：

```
# audit_control file

...

flags:lo,ss

minfree:10

naflags:lo,na
```

audit_user 文件显示了一个例外情况。当用户 jdoe 使用配置文件 shell 时，将审计此使用情况：

```
# audit_user file

jdoe:pf
```

jdoe 的审计预选掩码是 audit_user 设置和 audit_control 设置的组合。auditconfig -getaudit 命令显示 jdoe 的预选掩码：

```
# auditconfig -getaudit

audit id = jdoe(1234567)

process preselection mask = ss,pf,lo(0x13000,0x13000)

terminal id (maj,min,host) = 242,511,example1(192.168.160.171)

audit session id = 454
```

示例 29–8 只审计用户，而不审计系统

在本示例中，只在此系统上审计四个用户的登录和角色活动。audit_control 文件不预选系统的审计类：

```
# audit_control file

...

flags:

minfree:10
```

```
naflags:
```

audit_user 文件为四个用户预选两个审计类：

```
# audit_user file

jdoe:lo,pf

kdoe:lo,pf

pdoe:lo,pf

sdoe:lo,pf
```

以下 audit_control 文件可防止系统受到无担保的侵入。在与 audit_user 文件合并后，此文件比本示例中第一个 audit_control 文件更能保护系统安全。

```
# audit_control file

...

flags:

minfree:10

naflags:lo
```

▼ 如何添加审计类

当创建自己的审计类时，可以只将需要针对您所在站点审计的审计事件存放其中。在一个系统上添加该类时，应将此更改复制到正在审计的所有系统中。

1 **承担主管理员角色，或成为超级用户。**

主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2 **（可选的）保存 audit_class 文件的副本。**

```
# cp /etc/security/audit_class /etc/security/audit_class.orig
```

3 **在 audit_class 文件中添加新项。**

每一项都具有以下格式：

0x*number*:*name*:*description*

0x　将 *number* 标识为十六进制。

number　定义唯一审计类掩码。

name　定义审计类的字母名称。

description　定义审计类的说明性名称。

添加的项在该文件中必须唯一。请勿使用现有审计类掩码。

示例 29–9 创建新审计类

本示例会创建类来保存一个小型的审计事件组。`audit_class` 文件的添加项如下所示：

```
0x01000000:pf:profile command
```

此项创建名为 `pf` 的新审计类。示例 29–10 填充此新审计类。

▼ 如何更改审计事件的类成员关系

可能需要更改审计事件的类成员关系来减小现有审计类的大小，或者将事件放置在它自己的类中。在一个系统上重新配置审计事件到类的映射时，应将此更改复制到正在审计的所有系统中。

1. **承担主管理员角色，或成为超级用户。**

   主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2. **（可选的）保存 `audit_event` 文件的副本。**

   ```
   # cp /etc/security/audit_event /etc/security/audit_event.orig
   ```

3. **通过更改事件的 *class-list* 来更改特定事件所属的类。**

   每一项都具有以下格式：

   number:*name*:*description*:*class-list*

   number　审计事件 ID。

   name　审计事件的名称。

   description　通常为触发创建审计记录的系统调用或可执行文件。

   class-list　审计类的逗号分隔列表。

示例 29–10 将现有审计事件映射到新类

本示例将现有审计事件映射到在示例 29–9 中创建的新类。在 audit_control 文件中，二进制审计记录捕获 pf 类中事件的成功和失败信息。syslog 审计日志只包含 pf 类中事件的失败信息。

```
# grep pf | /etc/security/audit_class

0x01000000:pf:profile command

# vi /etc/security/audit_event

6180:AUE_prof_cmd:profile command:ua,as,pf

# vi audit_control

...

flags:lo,pf

plugin:name=audit_syslog.so.1; p_flags=-lo,-pf
```

示例 29–11 审计 setuid 程序的使用

本示例创建用来保存监视 setuid 和 setgid 程序调用事件的类。audit_control 项审计 st 类中事件的所有成功调用。

```
# vi /etc/security/audit_class

0x00000800:st:setuid class

# vi /etc/security/audit_event

26:AUE_SETGROUPS:setgroups(2):st

27:AUE_SETPGRP:setpgrp(2):st

40:AUE_SETREUID:setreuid(2):st

41:AUE_SETREGID:setregid(2):st

214:AUE_SETEGID:setegid(2):st

215:AUE_SETEUID:seteuid(2):st

# vi audit_control
```

```
...

flags:lo,+st

plugin:name=audit_syslog.so.1; p_flags=-lo,+st
```

配置和启用审计服务（任务列表）

以下任务列表介绍了配置和启用审计服务的过程。这些任务是有序的。

任务	说明	参考
1.（可选）更改审计配置文件	选择需要审计的事件、类和用户。	第 536 页中的 “配置审计文件（任务列表）”
2. 创建审计分区	创建审计文件的磁盘空间，并且使用文件权限来保护它们。	第 547 页中的 “如何创建审计文件的分区”
3. 创建 `audit_warn` 别名	定义需要关注审计服务时应收到电子邮件警告的对象。	第 550 页中的 “如何配置 `audit_warn` 电子邮件别名”
4.（可选）更改审计策略	定义站点所需的其他审计数据。	第 550 页中的 “如何配置审计策略”
5. 启用审计	启用审计服务。	第 553 页中的 “如何启用审计”
6.（可选）禁用审计	禁用审计服务。	第 555 页中的 “如何禁用审计”
7.（可选）重新读取审计配置更改	当 `auditd` 守护进程正在运行时，将审计配置更改读入内核。	第 556 页中的 “如何更新审计服务”
8.（可选）在非全局区域中配置审计	将策略设置为启用非全局区域，使其运行自己的审计守护进程	示例 29–16

配置和启用审计服务

为站点设置配置文件之后，需要为审计文件设置磁盘空间。还需要设置审计服务的其他属性，然后启用此服务。本节还包含更改配置设置时刷新审计服务的过程。

安装非全局区域后，可以采用与审计全局区域完全相同的方式来审计此非全局区域。或者，可以在非全局区域中修改审计配置文件，以便单独审计非全局区域。要自定义审计配置文件，请参见第 536 页中的 “配置审计文件（任务列表）” 。

▼ 如何创建审计文件的分区

以下过程说明如何创建审计文件及相应文件系统和目录的分区。根据需要可跳过一些步骤，具体取决于是否已经具有空分区，或者是否已经挂载了空文件系统。

1 承担主管理员角色，或成为超级用户。

主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2 确定所需的磁盘空间量。

至少为每台主机指定 200 MB 磁盘空间。但是，由于所需审计量决定磁盘空间需求，因此，所需的磁盘空间可能远大于此数字。请记住要包括一个用于最后使用的目录的本地分区。

3 根据需要创建专用的审计分区。

此步骤可以在服务器安装期间非常轻松地完成。另外，还可以在尚未挂载到服务器上的磁盘中创建分区。有关如何创建分区的完整说明，请参见《System Administration Guide: Devices and File Systems》中的第 12 章，“Administering Disks (Tasks)”。

```
# newfs /dev/rdsk/cwtxdysz
```

其中 `/dev/rdsk/cwtxdysz` 是分区的原始设备名称。

如果要审计本地主机，还要为本地主机创建最后使用的审计目录。

4 为每个新分区创建挂载点。

```
# mkdir /var/audit/server-name.n
```

其中 *server-name.n* 是服务器名称加一个标识各个分区的数字。数字是可选的，但是存在多个审计目录时，此数字非常有用。

5 添加自动挂载新分区的项。

将类似以下内容的行添加到 `/etc/vfstab` 文件：

```
/dev/dsk/cwtxdysz /dev/rdsk/cwtxdysz /var/audit/server-name.n   ufs  2  yes
```

6 （可选的）删除每个分区上的最小空闲空间阈值。

如果使用缺省配置，则会在目录的使用率达到 80% 时生成警告，出现此消息后，便无需再在分区上保留空闲空间。

```
# tunefs -m 0 /var/audit/server-name.n
```

7 挂载新的审计分区。

```
# mount /var/audit/server-name.n
```

8 在新分区上创建审计目录。

```
# mkdir /var/audit/server-name.n/files
```

9 更正挂载点和新目录的权限。

```
# chmod -R 750 /var/audit/server-name.n/files
```

10 在文件服务器上，定义其他主机可以使用的文件系统。

通常，会安装磁盘组来存储审计记录。如果一个审计目录要供多个系统使用，则必须通过 NFS 服务来共享此目录。在 `/etc/dfs/dfstab` 文件中针对每个目录添加类似以下内容的项：

```
share -F nfs /var/audit/server-name.n/files
```

11 在文件服务器上，重新启动 NFS 服务。

如果该命令是已启动的第一个或第一组 `share` 命令，则 NFS 守护进程可能未运行。

- **如果 NFS 服务处于脱机状态，请启用此服务。**

```
% svcs \*nfs\*

disabled       Nov_02   svc:/network/nfs/rquota:default

offline        Nov_02   svc:/network/nfs/server:default

# svcadm enable network/nfs/server
```

- **如果 NFS 服务正在运行，请重新启动此服务。**

```
% svcs \*nfs\*

online         Nov_02   svc:/network/nfs/client:default

online         Nov_02   svc:/network/nfs/server:default

# svcadm restart network/nfs/server
```

有关 NFS 服务的更多信息，请参阅《System Administration Guide: Network Services》中的“Setting Up NFS Services”。有关管理持久性服务的信息，请参见《System Administration Guide: Basic Administration》中的第 14 章“Managing Services (Overview)”和 `smf`(5) 手册页。

示例 29–12 创建最后使用的审计目录

运行审计服务的所有系统都应具有一个本地文件系统，当其他文件系统不可用时可以使用此本地文件系统。在本示例中，将在名为 `egret` 的系统上添加一个文件系统。由于此文件系统只在本地使用，因此无需执行任何关于文件服务器的步骤。

```
# newfs /dev/rdsk/c0t2d0

# mkdir /var/audit/egret

# grep egret /etc/vfstab

/dev/dsk/c0t2d0s1  /dev/rdsk/c0t2d0s1  /var/audit/egret ufs  2  yes  -
```

```
# tunefs -m 0 /var/audit/egret
# mount /var/audit/egret
# mkdir /var/audit/egret/files
# chmod -R 750 /var/audit/egret/files
```

示例 29–13 创建新的审计分区

在本示例中，将在由网络中其他系统使用的两个新磁盘上创建一个新的文件系统。

```
# newfs /dev/rdsk/c0t2d0
# newfs /dev/rdsk/c0t2d1
# mkdir /var/audit/egret.1
# mkdir /var/audit/egret.2
# grep egret /etc/vfstab
/dev/dsk/c0t2d0s1  /dev/rdsk/c0t2d0s1  /var/audit/egret.1 ufs  2  yes  -
/dev/dsk/c0t2d1s1  /dev/rdsk/c0t2d1s1  /var/audit/egret.2 ufs  2  yes  -
# tunefs -m 0 /var/audit/egret.1
# tunefs -m 0 /var/audit/egret.2
# mount /var/audit/egret.1
# mount /var/audit/egret.2
# mkdir /var/audit/egret.1/files
# mkdir /var/audit/egret.2/files
# chmod -R 750 /var/audit/egret.1/files /var/audit/egret.2/files
# grep egret /etc/dfs/dfstab
 share -F nfs /var/audit/egret.1/files
 share -F nfs /var/audit/egret.2/files
# svcadm enable network/nfs/server
```

▼ 如何配置 audit_warn 电子邮件别名

audit_warn 脚本会针对名为 audit_warn 的电子邮件别名生成邮件。要将此邮件发送到有效的电子邮件地址，可以执行步骤 2 中介绍的选项之一：

1 **承担主管理员角色，或成为超级用户。**

主管理员角色拥有主管理员配置文件。要创建角色并将角色指定给用户，请参见《System Administration Guide: Basic Administration》中的第 2 章，“Working With the Solaris Management Console (Tasks)”。

2 **配置 audit_warn 电子邮件别名。**

选择以下选项之一：

- **选项 1** – 使用 audit_warn 脚本中的另一个电子邮件帐户替换 audit_warn 电子邮件别名。

  在脚本的以下行中更改电子邮件别名：

  ```
  ADDRESS=audit_warn            # standard alias for audit alerts
  ```

- **选项 2** – 将 audit_warn 电子邮件重定向到另一个邮件帐户。

  在这种情况下，需要将 audit_warn 电子邮件别名添加到适当的邮件别名文件。可以将别名添加到本地 /etc/mail/aliases 文件或名称空间中的 mail_aliases 数据库。如果 root 邮件帐户成为 audit_warn 电子邮件别名的成员，则新项可能类似以下内容：

  ```
  audit_warn: root
  ```

▼ 如何配置审计策略

审计策略确定本地主机审计记录的特征。启用审计后，/etc/security/audit_startup 文件的内容将确定审计策略。

可以使用 auditconfig 命令来检查、启用或禁用当前审计策略选项。另外，还可以通过修改 audit_startup 脚本中 auditconfig 命令的策略选项来做出永久的审计策略更改。

1 **承担拥有审计控制配置文件的角色或成为超级用户。**

要创建拥有审计控制配置文件的角色并将该角色指定给用户，请参见第 186 页中的 “配置 RBAC（任务列表）”。

2 **查看审计策略。**

在启用审计之前，audit_startup 文件的内容将确定审计策略：

```
#!/bin/sh

/usr/bin/echo "Starting BSM services."

/usr/sbin/deallocate -Is
```

```
/usr/sbin/auditconfig -conf          配置事件类映射

/usr/sbin/auditconfig -aconf         配置无属性事件

/usr/sbin/auditconfig -setpolicy +cnt        对记录进行计数，而非删除
```

3 查看可用的策略选项。

```
$ auditconfig -lspolicy
```

注 – 只能在全局区域中设置 `perzone` 和 `ahlt` 策略选项。

---

4 启用或禁用选定的审计策略选项。

```
# auditconfig -setpolicy prefixpolicy
```

prefix *prefix* 值 + 会启用策略选项。*prefix* 值 - 会禁用策略选项。

policy 选择要启用或禁用的策略。

直到下次引导，或者由 `auditconfig -setpolicy` 命令修改此策略后，此策略才生效。

有关每个策略选项的说明，请参见第 531 页中的 “确定审计策略” 。

示例 29–14 设置 cnt 和 ahlt 审计策略选项

在本示例中，禁用了 `cnt` 策略，同时启用了 `ahlt` 策略。在此设置下，当审计分区已满时，会停止系统的使用。这些设置适合在安全性比可用性更重要时使用。有关设置此策略的限制的信息，请参见步骤 3。

以下 `audit_startup` 项会在重新引导后禁用 `cnt` 策略，同时启用 `ahlt` 策略：

```
# cat /etc/security/audit_startup

#!/bin/sh

/usr/bin/echo "Starting BSM services."

/usr/sbin/deallocate -Is

/usr/sbin/auditconfig -conf

/usr/sbin/auditconfig -aconf

/usr/sbin/auditconfig -setpolicy -cnt

/usr/sbin/auditconfig -setpolicy +ahlt
```

示例 29–15 临时设置 seq 审计策略

在本示例中，auditd 守护进程正在运行，并已设置 ahlt 审计策略。seq 审计策略添加到当前策略。seq 策略会在每条审计记录中添加 sequence 标记。在正在删除记录或审计记录已损坏时，此操作可以用来调试审计服务。

+ 前缀将 seq 选项添加到审计策略，而不是使用 seq 替换当前审计策略。auditconfig 命令直到下次调用此命令或下次引导时才使此策略生效。

```
$ auditconfig -setpolicy +seq

$ auditconfig -getpolicy

audit policies = ahlt,seq
```

示例 29–16 设置 perzone 审计策略

在本示例中，在全局区域的 audit_startup 脚本中设置 perzone 审计策略。引导区域时，非全局区域将根据其区域中的审计配置设置收集审计记录。

```
$ cat /etc/security/audit_startup

#!/bin/sh

/usr/bin/echo "Starting BSM services."

/usr/sbin/deallocate -Is

/usr/sbin/auditconfig -conf

/usr/sbin/auditconfig -aconf

/usr/sbin/auditconfig -setpolicy +perzone

/usr/sbin/auditconfig -setpolicy +cnt
```

示例 29–17 更改审计策略

在本示例中，审计守护进程正在运行，并已设置审计策略。auditconfig 命令会针对会话持续时间更改 ahlt 和 cnt 策略。在此设置下，当审计文件系统已满时，会删除审计记录，但也会对审计记录进行计数。有关设置 ahlt 策略的限制，请参见步骤 3。

```
$ auditconfig -setpolicy +cnt

$ auditconfig -setpolicy -ahlt
```

```
$ auditconfig -getpolicy

audit policies = cnt,seq
```

将更改置于 audit_startup 文件中后，更改后的策略将永久有效：

```
$ cat /etc/security/audit_startup

#!/bin/sh

/usr/bin/echo "Starting BSM services."

/usr/sbin/deallocate -Is

/usr/sbin/auditconfig -conf

/usr/sbin/auditconfig -aconf

/usr/sbin/auditconfig -setpolicy +cnt
```

不必在此文件中指定 -ahlt 选项，因为缺省情况下禁用 ahlt 策略选项。当可用性比审计记录提供的安全性更重要时，适合使用此设置。

▼ 如何启用审计

此过程在全局区域中启动审计服务。要在非全局区域中启用审计服务，请参见示例 29–18。

开始之前 应该在完成以下任务后执行此过程：

- 规划－第 527 页中的 “规划 Solaris 审计（任务列表）”
- 自定义审计文件－第 536 页中的 “配置审计文件（任务列表）”
- 设置审计分区－第 547 页中的 “如何创建审计文件的分区”
- 设置审计警告消息－第 550 页中的 “如何配置 audit_warn 电子邮件别名”
- 设置审计策略－第 550 页中的 “如何配置审计策略”

1 成为超级用户并使系统进入单用户模式。

```
% su

Password:      <键入超级用户口令>

# init S
```

有关更多信息，请参见 init(1M) 手册页。

2 运行启用审计服务的脚本。

转至 /etc/security 目录，并在其中执行 bsmconv 脚本。

```
# cd /etc/security

# ./bsmconv

This script is used to enable the Basic Security Module (BSM).

Shall we continue with the conversion now? [y/n] y

bsmconv: INFO: checking startup file.

bsmconv: INFO: move aside /etc/rc3.d/S81volmgt.

bsmconv: INFO: turning on audit module.

bsmconv: INFO: initializing device allocation files.



The Basic Security Module is ready.

If there were any errors, please fix them now.

Configure BSM by editing files located in /etc/security.

Reboot this system now to come up with BSM enabled.
```

有关此脚本的影响，请参见 bsmconv(1M) 手册页。

3 使系统进入多用户模式。

```
# init 6
```

启动文件 /etc/security/audit_startup 使 auditd 守护进程在系统进入多用户模式时自动运行。

此脚本的另一个影响是启用设备分配。要配置设备分配，请参见第 76 页中的 “管理设备分配（任务列表）” 。

示例 29–18 在非全局区域中启用审计

以下示例中，在全局区域中启用审计并引导非全局区域后，全局区域管理员启动了 perzone 策略。非全局区域的区域管理员配置了此区域的审计文件，随后在此区域中启动审计守护进程。

```
zone1# /usr/sbin/audit -s
```

▼ 如何禁用审计

如果在某一点不再需要审计服务，则此过程会使系统返回到启用审计之前的系统状态。如果正在审计非全局区域，则其审计服务也将禁用。

注意 – 此命令还禁用设备分配。如果需要分配设备，请勿运行此命令。要禁用审计并保留设备分配，请参见示例 29–19。

1. **成为超级用户并使系统进入单用户模式。**

```
% su

Password:       <键入超级用户口令>

# init S
```

有关更多信息，请参见 `init`(1M) 手册页。

2. **运行此脚本以禁用审计。**

转至 `/etc/security` 目录，并执行 `bsmunconv` 脚本。

```
# cd /etc/security

# ./bsmunconv
```

此脚本的另一个影响是禁用设备分配。

有关 `bsmunconv` 脚本的全部影响的信息，请参见 `bsmconv`(1M) 手册页。

3. **使系统进入多用户模式。**

```
# init 6
```

示例 29–19 禁用审计并保持设备分配

在本示例中，审计服务停止收集记录，但是设备分配继续工作。将删除 `audit_control` 文件中 `flags`、`naflags` 和 `plugin` 项的所有值，同时也将删除 `audit_user` 文件中所有用户项。

```
# audit_control file

...

flags:

minfree:10

naflags:
```

```
plugin:


# audit_user file
```

auditd 守护进程将运行，但是不保留任何审计记录。

▼ 如何更新审计服务

当在运行守护进程之后更改审计配置时，此过程将重新启动 auditd 守护进程。

1 **承担拥有审计控制权限配置文件的角色或成为超级用户。**

要创建拥有审计控制权限配置文件的角色并将该角色指定给用户，请参见第 186 页中的 “配置 RBAC（任务列表）”。

2 **选择适当的命令。**

- **如果修改 audit_control 文件中的 naflags 行，请更改无归属事件的内核掩码。**

  ```
  $ /usr/sbin/auditconfig -aconf
  ```

  也可以重新引导。

- **如果修改 audit_control 文件中的其他行，请重新读取 audit_control 文件。**

  审计守护进程内部存储 audit_control 文件的信息。要使用新信息，请重新引导系统或指示审计守护进程读取已修改的文件。

  ```
  $ /usr/sbin/audit -s
  ```

  注 – 将基于与每个进程相关联的审计预选掩码生成审计记录。执行 audit -s 不会在现有进程中更改掩码。要更改现有进程的预选掩码，必须重新启动此进程。也可以重新引导。

  ---

  audit -s 命令使审计守护进程从 audit_control 文件中重新读取 directory 和 minfree 值。此命令会更改后续登录所产生进程的预选掩码的生成。

- **如果在审计守护进程运行时修改 audit_event 文件或 audit_class 文件，请刷新审计服务。**

  将已修改事件到类的映射读入系统，并且确保正确审计使用此计算机的每个用户。

  ```
  $ auditconfig -conf
  ```

  ```
  $ auditconfig -setumask auid classes
  ```

  auid 用户 ID。

classes 预选审计类。

- **要在正在运行的系统上更改审计策略，请参见示例 29–15。**

示例 29–20 重新启动审计守护进程

在本示例中，系统降为单用户模式，随后回升到多用户模式。当系统进入多用户模式时，会将已修改的审计配置文件读入系统。

```
# init S

# init 6
```

管理审计记录（任务列表）

以下任务列表介绍了选择、分析和管理审计记录的过程。

任务	说明	参考
显示审计记录格式	显示为审计事件收集的信息类型和显示信息的顺序。	第 558 页中的 “如何显示审计记录格式”
合并审计记录	将多台计算机中的审计文件合并到一个审计跟踪。	第 561 页中的 “如何合并审计跟踪中的审计文件”
选择要检查的记录	选择要研究的特定事件。	第 563 页中的 “如何从审计跟踪中选择审计事件”
显示审计记录	可以查看二进制审计记录。	第 565 页中的 “如何查看二进制审计文件的内容”
清除错误命名的审计文件	向审计服务意外打开的审计文件提供结束时间标记。	第 567 页中的 “如何清除 `not_terminated` 审计文件”
防止审计跟踪溢出	防止写满审计文件系统。	第 568 页中的 “如何防止审计跟踪溢出”

管理审计记录

通过管理审计跟踪，可以监视网络中的用户操作。审计可以生成大量数据。以下任务显示如何使用所有这些数据。

▼ 如何显示审计记录格式

要编写可查找所需审计数据的脚本，需要了解审计事件中的标记顺序。bsmrecord 命令显示审计事件的审计事件编号、审计类、选择掩码和记录格式。

◗ **将所有审计事件记录格式置于 HTML 文件中。**

-a 选项列出所有审计事件记录格式。-h 选项以可在浏览器中显示的 HTML 格式显示此列表。

```
% bsmrecord -a -h > audit.events.html
```

在浏览器中显示 *html 文件时，请使用浏览器的“查找”工具来查找特定记录。

有关更多信息，请参见 bsmrecord(1M) 手册页。

示例 29–21 显示程序的审计记录格式

在本示例中，将显示由 login 程序生成的所有审计记录的格式。登录程序包括 rlogin、telnet、newgrp、Solaris Management Console 的角色登录，以及 Solaris 安全 Shell。

```
% bsmrecord -p login

terminal login

  program     /usr/sbin/login      See login(1)

              /usr/dt/bin/dtlogin  See dtlogin

  event ID    6152                 AUE_login

  class       lo                   (0x00001000)

      header

      subject

      text                         error message or "successful login"

      return



login: logout

  program     various              See login(1)

  event ID    6153                 AUE_logout
```

```
...


newgrp
  program     newgrp               See newgrp login
  event ID    6212                 AUE_newgrp_login
...


rlogin
  program     /usr/sbin/login      See login(1) - rlogin
  event ID    6155                 AUE_rlogin
...


SMC: role login
  program     SMC server           See role login
  event ID    6173                 AUE_role_login
...


/usr/lib/ssh/sshd
  program     /usr/lib/ssh/sshd    See login - ssh
  event ID    6172                 AUE_ssh
...


telnet login
  program     /usr/sbin/login      See login(1) - telnet
```

```
  event ID    6154                AUE_telnet

  ...
```

示例 29–22 显示审计类的审计记录格式

在本示例中，将显示 fd 类的所有审计记录的格式。

```
% bsmrecord -c fd


rmdir

  system call rmdir               See rmdir(2)

  event ID    48                  AUE_RMDIR

  class       fd                  (0x00000020)

      header

      path

      [attribute]

      subject

      [use_of_privilege]

      return


unlink

  system call unlink              See unlink(2)

  event ID    6                   AUE_UNLINK

  ...


unlinkat

  system call unlinkat            See openat(2)
```

```
event ID    286                 AUE_UNLINKAT

...
```

▼ 如何合并审计跟踪中的审计文件

通过合并所有审计目录中的所有审计文件，可以分析整个审计跟踪的内容。`auditreduce` 命令将其输入文件中的所有记录合并到单个输出文件中。然后可以删除输入文件。将输出文件置于名为 `/etc/security/audit`*server-name*`/files` 的目录中时，`auditreduce` 命令可以查找此输出文件，而无需指定全路径。

注 – 此过程仅适用于二进制审计记录。

---

1 承担拥有审计查看配置文件的角色或成为超级用户。

系统管理员角色拥有审计查看配置文件。另外，还可以创建拥有审计查看配置文件的单独用户。要创建角色并将其指定给用户，请参见第 186 页中的 “配置 RBAC（任务列表）”。

2 创建存储合并审计文件的目录。

```
# mkdir audit-trail-directory
```

3 限制对此目录的访问。

```
# chmod 700 audit-trail-directory

# ls -la audit-trail-directory

drwx------   3 root     sys          512 May 12 11:47 .

drwxr-xr-x   4 root     sys         1024 May 12 12:47 ..
```

4 合并审计跟踪中的审计记录。

转至 *audit-trail-directory* 目录并将审计记录合并到带有指定后缀的文件中。将合并在本地系统上 `audit_control` 文件中 `dir` 行列出的所有目录。

```
# cd audit-trail-directory

# auditreduce -Uppercase-option -O suffix
```

`auditreduce` 命令的大写选项处理审计跟踪中的文件。大写选项包括以下内容：

`-A` 选择审计跟踪中的所有文件。

`-C` 只选择完整文件。此选项会忽略带有后缀 `not_terminated` 的文件。

`-M` 选择带有特定后缀的文件。后缀可以是机器名，也可以是为摘要文件指定的后缀。

`-O` 在当前目录中创建一个带有开始时间和结束时间的 14 字符时间标记以及后缀 *suffix* 的审计文件。

示例 29–23 将审计文件复制到摘要文件

在以下示例中，系统管理员角色 sysadmin 将所有文件从审计跟踪复制到合并文件中。

```
$ whoami

sysadmin

$ mkdir /var/audit/audit_summary.dir

$ chmod 700 /var/audit/audit_summary.dir

$ cd /var/audit/audit_summary.dir

$ auditreduce -A -O All

$ ls *All

20030827183214.20030827215318.All
```

在以下示例中，只将完整文件从审计跟踪复制到合并文件中。

```
$ cd /var/audit/audit_summary.dir

$ auditreduce -C -O Complete

$ ls *Complete

20030827183214.20030827214217.Complete
```

在以下示例中，只将完整文件从 example1 计算机复制到合并文件中。

```
$ cd /var/audit/audit_summary.dir

$ auditreduce -M example1 -O example1summ

$ ls *summ

20030827183214.20030827214217.example1summ
```

示例 29–24 将审计文件移动到摘要文件

auditreduce 命令的 -D 选项会在您将审计文件复制到另一位置时将其删除。在以下示例中，会将一个系统中的完整审计文件复制到摘要目录以供以后检查。

```
$ cd /var/audit/audit_summary.dir
```

```
$ auditreduce -C -O daily_example1 -D example1

$ ls *example1

20030827183214.20030827214217.daily_example1
```

当此命令成功完成时，将删除 example1 系统中的审计文件，这些审计文件是 *daily_example1 文件的输入。

▼ 如何从审计跟踪中选择审计事件

可以过滤审计记录以便检查。有关过滤选项的完整列表，请参见 auditreduce(1M) 手册页。

1 承担拥有审计查看配置文件的角色或成为超级用户。

系统管理员角色拥有审计查看配置文件。另外，还可以创建拥有审计查看配置文件的单独用户。要创建角色并将其指定给用户，请参见第 186 页中的 “配置 RBAC（任务列表）”。

2 从审计跟踪或指定的审计文件中选择所需的记录类型。

auditreduce *-lowercase-option argument* [*optional-file*]

argument	小写选项所需的特定参数。例如，-c 选项需要审计类的 *argument*，例如 ua。
-d	选择特定日期的所有事件。*argument* 的日期格式为 *yyymmdd*。其他日期选项 -b 和 -a 选择特定日期之前和之后的事件。
-u	选择属于特定用户的所有事件。*argument* 是用户名。另一个用户选项 -e 选择属于有效用户 ID 的所有事件。
-c	选择预选审计类中的所有事件。*argument* 是审计类名。
-m	选择特定审计事件的所有实例。*argument* 是审计事件。
optional-file	审计文件的名称。

示例 29–25 合并和减少审计文件

auditreduce 命令可以在合并输入文件时删除不感兴趣的记录。例如，可以使用 auditreduce 命令仅保留审计文件中一个月以前的登录和退出记录。如果需要检索完整的审计跟踪，可以从备份介质中恢复此跟踪。

```
# cd /var/audit/audit_summary.dir

# auditreduce -O lo.summary -b 20030827 -c lo; compress *lo.summary
```

示例 29–26 将 na 审计记录复制到摘要文件

在本示例中，将审计跟踪中的所有无归属审计事件记录收集到一个文件中。

```
$ whoami

sysadmin

$ cd /var/audit/audit_summary.dir

$ auditreduce -c na -O nasumm

$ ls *nasumm

20030827183214.20030827215318.nasumm
```

将使用 na 记录的开始和结束日期为合并的 nasumm 审计文件添加时间标记。

示例 29–27 在指定的审计文件中查找审计事件

可以手动选择审计文件以仅搜索指定的文件组。例如，可以通过进一步处理前面示例中的 *nasumm 文件来查找系统引导事件。要执行此操作，可以指定文件名作为 auditreduce 命令的最后一个参数。

```
$ auditreduce -m 113 -O systemboot 20030827183214.20030827215318.nasumm

20030827183214.20030827183214.systemboot
```

20030827183214.20030827183214.systemboot 文件只包含系统引导审计事件。

示例 29–28 将一个用户的审计记录复制到摘要文件

在本示例中，将合并包含特定用户名称的审计跟踪中的记录。-e 选项查找有效用户。-u 选项查找审计用户。

```
$ cd /var/audit/audit_summary.dir

$ auditreduce -e tamiko -O tamiko
```

可以在此文件中查找特定事件。在以下示例中，将检查 2003 年 9 月 7 日（您的时间）用户登录和退出的时间。只检查那些以用户名作为文件后缀的文件。此日期的简捷形式为 *yyyymmdd*。

```
# auditreduce -M tamiko -O tamikolo -d 20030907 -u tamiko -c lo
```

示例 29–29 将选定的记录复制到单个文件

在本示例中，从审计跟踪中选择特定日期的登录和退出消息。将消息合并到目标文件。目标文件将写入除常规审计根目录以外的目录中。

```
# auditreduce -c lo -d 20030827 -O /var/audit/audit_summary.dir/logins

# ls /var/audit/audit_summary.dir/*logins

/var/audit/audit_summary.dir/20030827183936.20030827232326.logins
```

▼ 如何查看二进制审计文件的内容

使用 `praudit` 命令，可以查看二进制审计文件的内容。可以传输 `auditreduce` 命令的输出，也可以读取特定审计文件。`-x` 选项可用于进一步处理。

1. **承担拥有审计查看配置文件的角色或成为超级用户。**

   系统管理员角色拥有审计查看配置文件。另外，还可以创建拥有审计查看配置文件的单独用户。要创建角色并将其指定给用户，请参见第 186 页中的 “配置 RBAC（任务列表）”。

2. **使用以下 `praudit` 命令之一来生成最符合您需要的输出。**

   以下示例显示同一审计事件的 `praudit` 输出。审计策略已设置为包括 `sequence` 和 `trailer` 标记。

   - `praudit -s` 命令以短格式（每行一个标记）显示审计记录。使用 `-l` 选项在一行放置一条记录。

     ```
     $ auditreduce -c lo | praudit -s

     header,101,2,AUE_rlogin,,example1,2003-10-13 11:23:31.050 -07:00

     subject,jdoe,jdoe,staff,jdoe,staff,749,749,195 1234 server1

     text,successful login

     return,success,0

     sequence,1298
     ```

   - `praudit -r` 命令以原始格式（每行一个标记）显示审计记录。使用 `-l` 选项在一行放置一条记录。

     ```
     $ auditreduce -c lo | praudit -r

     21,101,2,6155,0x0000,192.168.60.83,1062021202,64408258
     ```

```
36,2026700,2026700,10,2026700,10,749,749,195 1234 192.168.60.17

40,successful login

39,0,0

47,1298
```

- `praudit -x` 命令以 XML 格式（每行一个标记）显示审计记录。使用 `-l` 选项在一行放置一条记录的 XML 输出。

```
$ auditreduce -c lo | praudit -x

<record version="2" event="login - rlogin" host="example1"

time="Wed Aug 27 14:53:22 PDT 2003" msec="64">

<subject audit-uid="jdoe" uid="jdoe" gid="staff" ruid="jdoe"

rgid="staff" pid="749" sid="749" tid="195 1234 server1"/>

<text>successful login</text>

<return errval="success" retval="0"/>

<sequence seq-num="1298"/>



</record>
```

示例 29–30 打印整个审计跟踪

通过 `lp` 命令的管道，整个审计跟踪的输出将转至打印机。打印机的访问应受到限制。

```
# auditreduce | praudit | lp -d example.protected.printer
```

示例 29–31 查看特定审计文件

在本示例中，在终端窗口中检查登录文件摘要。

```
# cd /var/audit/audit_summary.dir/logins

# praudit 20030827183936.20030827232326.logins | more
```

示例 29–32 放置 XML 格式的审计记录

在本示例中，审计记录转换为 XML 格式。

```
# praudit -x 20030827183214.20030827215318.logins > 20030827.logins.xml
```

可以在浏览器中显示 *xml 文件。可以使用脚本对文件内容进行操作，以便提取相关信息。

▼ 如何清除 not_terminated 审计文件

有时，审计守护进程退出，而其审计文件仍处于打开状态。或者，某服务器不可访问，并强制计算机切换到新服务器。在这种情况下，虽然审计文件不再用于审计记录，但此文件还是以字符串 not_terminated 作为结束时间标记。使用 auditreduce -O 命令为此文件提供正确时间标记。

1. **在审计文件系统上，按照创建顺序列出带有 not_terminated 字符串的文件。**

   ```
   # ls -R1t audit-directory*/files/* | grep not_terminated
   ```

   -R 列出子目录中的文件。

   -t 按照从最新到最旧的顺序列出文件。

   -1 将文件列成一列。

2. **清除旧的 not_terminated 文件。**

   将旧文件的名称指定到 auditreduce -O 命令。

   ```
   # auditreduce -O system-name old-not-terminated-file
   ```

3. **删除旧的 not_terminated 文件。**

   ```
   # rm system-name old-not-terminated-file
   ```

示例 29–33 清除关闭的 not_terminated 审计文件

在以下示例中，查找并重命名 not_terminated 文件，然后删除原文件。

```
ls -R1t */files/* | grep not_terminated

.../egret.1/20030908162220.not_terminated.egret

.../egret.1/20030827215359.not_terminated.egret

# cd */files/egret.1

# auditreduce -O egret 20030908162220.not_terminated.egret
```

```
# ls -1t

20030908162220.not_terminated.egret      当前审计文件

20030827230920.20030830000909.egret      输入（旧）审计文件

20030827215359.not_terminated.egret

# rm 20030827215359.not_terminated.egret

# ls -1t

20030908162220.not_terminated.egret      当前审计文件

20030827230920.20030830000909.egret      已清除的审计文件
```

新文件上的开始时间标记反映 not_terminated 文件中第一个审计事件的时间。结束时间标记反映此文件中最后一个审计事件的时间。

▼ 如何防止审计跟踪溢出

如果安全策略要求保存所有审计数据，则执行以下操作：

1. **设置计划以定期归档审计文件。**

   通过将文件备份到脱机介质来归档审计文件。另外，还可以将这些文件移动到归档文件系统。

   如果正在使用 syslog 实用程序收集文本审计日志，请归档文本日志。有关更多信息，请参见 logadm(1M) 手册页。

2. **设置计划以从审计文件系统中删除已归档审计文件。**

3. **保存和存储辅助信息。**

   归档解释审计记录和审计跟踪所需的信息。

4. **保留已归档审计文件的记录。**

5. **正确存储归档介质。**

6. **减少通过创建摘要文件存储的审计数据量。**

   可以使用 auditreduce 命令的选项从审计跟踪中提取摘要文件。摘要文件只包含指定类型的审计事件的记录。要提取摘要文件，请参见示例 29–25 和示例 29–29。

◆ ◆ ◆ 第 30 章

Solaris 审计（参考）

本章介绍重要的 Solaris 审计组件。以下是本章中参考信息的列表：

- 第 569 页中的 “审计命令”
- 第 574 页中的 “在审计服务中使用的文件”
- 第 579 页中的 “用于管理审计的权限配置文件”
- 第 579 页中的 “审计和 Solaris Zones”
- 第 580 页中的 “审计类”
- 第 582 页中的 “审计策略”
- 第 582 页中的 “进程审计特征”
- 第 583 页中的 “审计跟踪”
- 第 583 页中的 “二进制审计文件名称约定”
- 第 584 页中的 “审计记录结构”
- 第 585 页中的 “审计标记格式”

有关 Solaris 审计的概述，请参见第 27 章。有关规划建议，请参见第 28 章。有关在站点上配置审计的过程，请参见第 29 章。

审计命令

本节提供有关以下命令的信息：

- 第 569 页中的 “auditd 守护进程”
- 第 570 页中的 “audit 命令”
- 第 570 页中的 “bsmrecord 命令”
- 第 571 页中的 “auditreduce 命令”
- 第 572 页中的 “praudit 命令”
- 第 574 页中的 “auditconfig 命令”

auditd 守护进程

以下列表概述了 auditd 守护进程的功能。

- auditd 守护进程可打开和关闭 audit_control 文件内指定的目录中的审计文件。这些文件将按顺序打开。
- auditd 守护进程可装入一个或多个插件。Sun 提供了两个插件。/lib/security/audit_binfile.so.1 插件可将二进制审计数据写入文件。/lib/security audit_syslog.so.1 插件可将审计记录的文本摘要发送到 syslogd 守护进程。
- auditd 守护进程可使用 auditd 插件从内核读取审计数据并输出此数据。
- auditd 守护进程可执行 audit_warn 脚本来发出有关配置错误的警告。binfile.so.1 插件可执行 audit_warn 脚本。此脚本在缺省情况下将警告发送到 audit_warn 电子邮件别名以及控制台。syslog.so.1 插件无法执行 audit_warn 脚本。
- 缺省情况下，当所有的审计目录已满时，生成审计记录的进程便会暂停。此外，auditd 守护进程可将消息写入控制台以及 audit_warn 电子邮件别名。此时，只有系统管理员才可以修复审计服务。管理员可以登录以将审计文件写入脱机介质、从系统中删除审计文件，以及执行其他清除任务。

  可以使用 auditconfig 命令重新配置审计策略。

当系统进入多用户模式时，auditd 守护进程便会自动启动。也可以从命令行启动此守护进程。当 auditd 守护进程启动时，它会计算审计文件所需的空闲空间量。

auditd 守护进程使用 audit_control 文件中的审计目录列表作为创建审计文件的可能位置。此守护进程维护指向此目录列表的指针，该指针开始于第一个目录。每次在 auditd 守护进程需要创建审计文件时，都会将文件放入列表内的第一个可用目录中。列表开始于 auditd 守护进程的当前指针处。您可以运行 audit -s 命令将指针复位到列表的开始处。audit -n 命令指示此守护进程切换到新的审计文件。新的文件在当前文件所在的目录中创建。

audit 命令

audit 命令可控制 auditd 守护进程的操作。audit 命令可以执行以下任务：

- 启用和禁用审计功能
- 重置 auditd 守护进程
- 调整本地系统上的审计预选掩码
- 将审计记录写入其他审计文件

有关可用选项的介绍，请参见 audit(1M) 手册页。

bsmrecord 命令

bsmrecord 命令可显示在 /etc/security/audit_event 文件中定义的审计事件的格式。输出包括事件的审计 ID、审计类、审计标志以及按顺序列出的记录的审计标记。如果不使用任何选项，则 bsmrecord 输出将在终端窗口中显示。如果使用 -h 选项，则输出将适合于在浏览器中进行查看。有关 bsmrecord 命令的使用示例，请参见第 558 页中的 “如何显示审计记录格式”。另请参见 bsmrecord(1M) 手册页。

auditreduce 命令

auditreduce 命令可汇总以二进制格式存储的审计记录。此命令可以合并来自一个或多个输入审计文件的审计记录，还可以用于执行后选审计记录。这些记录仍保持二进制格式。要合并整个审计跟踪，请在审计服务器上运行此命令。审计服务器是指挂载了用于安装的所有审计文件系统的系统。有关更多信息，请参见 auditreduce(1M) 手册页。

使用 auditreduce 命令，可以从一个位置跟踪多个系统上的所有已审计的操作。此命令可以将所有审计文件的逻辑组合作为一个审计跟踪单元进行读取。您必须对站点上要进行审计的所有系统进行相同的配置，并为审计文件创建服务器和本地目录。auditreduce 命令会忽略记录的生成方式或记录的存储位置。如果不使用选项，则 auditreduce 命令将合并审计根目录内所有子目录中的所有审计文件的审计记录。通常，/etc/security/audit 为审计根目录。auditreduce 命令将已合并的结果发送到标准输出。 您也可以将这些结果放入按时间顺序排列的单个输出文件中。 此文件包含二进制数据。

auditreduce 命令还可以选择特定的记录类型进行分析。 auditreduce 命令的合并功能和选择功能在逻辑上是相互独立的。在系统合并输入文件并将其写入磁盘之前，auditreduce 命令将在系统读取记录时从这些文件中捕获数据。

通过为 auditreduce 命令指定选项，还可以执行以下操作：

- 请求已由指定的审计类生成的审计记录
- 请求已由某个特定用户生成的审计记录
- 请求已在特定日期生成的审计记录

如果不使用参数，则 auditreduce 命令将检查 /etc/security/audit 目录（缺省审计根目录）中的子目录，它还会检查 *start-time.end-time.hostname* 文件所在的 files 目录。auditreduce 命令对于审计数据位于不同目录的情况非常有用。图 30–1 显示了不同主机上不同目录中的审计数据。图 30–2 显示了不同审计服务器上不同目录中的审计数据。

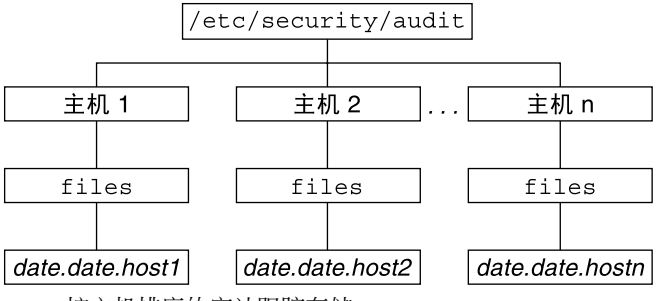


图 30–1 按主机排序的审计跟踪存储

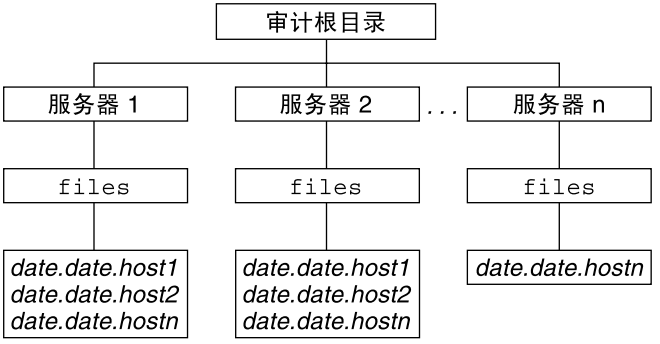


图 30–2 按服务器排序的审计跟踪存储

如果用于 `/etc/security/audit` 目录的分区非常小，则可能无法在缺省目录中存储审计数据。可以使用 `-R` 选项将 `auditreduce` 命令传递到其他目录：

```
# auditreduce -R /var/audit-alt
```

还可以使用 `-S` 选项指定特定的子目录：

```
# auditreduce -S /var/audit-alt/host1
```

有关其他选项和更多示例，请参见 `auditreduce`(1M) 手册页。

praudit 命令

`praudit` 命令可使 `auditreduce` 命令的二进制输出具有可读性。`praudit` 命令可从标准输入中读取二进制格式的审计记录，并以可显示的格式显示这些记录。输入既可以从 `auditreduce` 命令进行管道输出，也可以从单个审计文件进行管道输出。输入还可以使用 `cat` 命令生成以串联数个文件，或者针对当前审计文件使用 `tail` 命令生成。

`praudit` 命令可以生成四种输出格式。第五个选项，即 `-l`（长），可在每个输出行中列显一条审计记录。缺省设置为在每个输出行中放置一个审计标记。`-d` 选项可更改标记字段之间以及标记之间使用的分隔符。缺省分隔符为逗号。

- **缺省设置**－不带选项的 `praudit` 命令在每行显示一个审计标记。此命令将按审计事件的说明显示审计事件，例如 `ioctl(2)` 系统调用。任何可以显示为文本的值均以文本格式显示。例如，用户将显示为用户名，而不是用户 ID。
- **–r 选项**－原始选项将任何可以为数字的值显示为数字。例如，用户显示为用户 ID，Internet 地址以十六进制格式显示，模式以八进制格式显示。审计事件显示为其事件编号，例如 `158`。
- **–s 选项**－短选项将通过审计事件的表名显示审计事件，例如 `AUE_IOCTL`。此选项显示其他标记的方式与缺省选项显示这些标记的方式相同。
- **–x 选项**－XML 选项以 XML 格式显示审计记录。此选项可用于浏览器输入，或者可用于处理 XML 的脚本输入。

XML 通过审计服务提供的 DTD 进行说明。Solaris 软件还提供了样式表。DTD 和样式表位于 /usr/share/lib/xml 目录中。

在 praudit 命令的缺省输出格式中，可以轻松地将每条记录标识为一系列审计标记。每个标记都在单独的行中显示。每条记录都以 header 标记开始。例如，您可以使用 awk 命令进一步处理输出。

以下是 header 标记的 praudit -l 命令的输出：

```
header,173,2,settppriv(2),,example1,2003-10-13 13:46:02.174 -07:00
```

以下是同一 header 标记的 praudit -r 命令的输出：

```
121,173,2,289,0x0000,192.168.86.166,1066077962,174352445
```

示例 30–1 使用脚本处理 praudit 输出

您可能需要将 praudit 命令的输出作为多行文本处理。例如，您可能需要选择 auditreduce 命令无法选择的记录。您可以使用简单的 shell 脚本来处理 praudit 命令的输出。下面的简单示例脚本在每行中放置一条审计记录，搜索用户指定的字符串，然后将审计文件返回到其原始格式。

```
#!/bin/sh

#

## This script takes an argument of a user-specified string.

#  The sed command prefixes the header tokens with Control-A

#  The first tr command puts the audit tokens for one record

#  onto one line while preserving the line breaks as Control-A

#

praudit | sed -e '1,2d' -e '$s/^file.*$//' -e 's/^header/^aheader/' \\

| tr '\\012\\001' '\\002\\012' \\

| grep "$1" \\      查找用户指定的字符串

| tr '\\002' '\\012'      恢复原始的新换行符
```

请注意，脚本中的 ^a 为 Ctrl-A，而不是 ^ 和 a 这两个字符。前缀将 header 标记与可能显示为文本的字符串 header 区分开来。

auditconfig 命令

auditconfig 命令可提供命令行界面来检索并设置审计配置参数。auditconfig 命令可以执行以下任务：

- 显示、检查以及配置审计策略
- 确定审计功能处于打开还是关闭状态
- 关闭和打开审计功能
- 管理审计目录和审计文件
- 管理审计队列
- 获取并设置预选掩码
- 获取审计事件并将其设置为审计类映射
- 获取并设置配置信息，例如会话 ID 和审计 ID
- 配置进程、shell 以及会话的审计特征
- 重置审计统计信息

有关命令选项的介绍，请参见 auditconfig(1M) 手册页。

在审计服务中使用的文件

审计服务使用以下文件：

- 第 574 页中的 “system 文件”
- 第 575 页中的 “syslog.conf 文件”
- 第 575 页中的 “audit_class 文件”
- 第 575 页中的 “audit_control 文件”
- 第 576 页中的 “audit_event 文件”
- 第 576 页中的 “audit_startup 脚本”
- 第 577 页中的 “audit_user 数据库”
- 第 577 页中的 “audit_warn 脚本”
- 第 578 页中的 “bsmconv 脚本”

system 文件

/etc/system 文件包含内核在初始化期间读取以自定义系统操作的命令。用于激活和取消激活审计功能的 bsmconv 和 bsmunconv shell 脚本可修改 /etc/system 文件。 bsmconv shell 脚本将以下行添加到 /etc/system 文件：

```
set c2audit:audit_load=1
```

set c2audit:audit_load=1 项将导致在引导系统时装入用于审计的内核模块。bsmunconv shell 脚本将在重新引导系统时禁用审计功能。此命令可从 /etc/system 文件中删除 c2audit 行。

syslog.conf 文件

/etc/syslog.conf 文件可与 audit_control 文件结合使用以存储文本格式的审计记录。可以将 syslog.conf 文件配置为启用 syslog 实用程序来存储审计记录。有关示例，请参见第 539 页中的 “如何配置 syslog 审计日志”。

audit_class 文件

/etc/security/audit_class 文件可定义审计类。审计类是指多组审计事件。可以使用 audit_control 文件中的类名来预选要审计其事件的类。这些类可接受前缀以仅选择失败的事件或仅选择成功的事件。有关更多信息，请参见第 581 页中的 “审计类语法”。

超级用户或承担等效角色的管理员可以修改审计类的定义。此管理员可以通过在文本编辑器中编辑 audit_class 文件来定义新的审计类，重命名现有类，或更改现有类。有关更多信息，请参见 audit_class(4) 手册页。

audit_control 文件

每个系统上的 /etc/security/audit_control 文件都包含 auditd 守护进程的配置信息。使用此文件，每个系统都可以装入远程审计文件系统来存储其审计记录。

可以在 audit_control 文件中指定五种信息类型。每个信息行都以关键字开始。

- flags **关键字**－用作为系统上所有用户预选要审计的事件类的项的开头。此处指定的审计类将确定**系统范围的审计预选掩码**。审计类以逗号分隔。
- naflags **关键字**－用作当无法将某操作归属到特定用户时预选要审计的事件类的项的开头。审计类以逗号分隔。na 事件类应归入此项。naflags 项可以用于记录其他通常具有归属性但无法进行归属的事件类。例如，如果在引导系统时启动的某个程序可读取文件，则 naflags 项中的 fr 将针对此事件创建一条记录。
- minfree **关键字**－用作针对所有审计文件系统定义最小空闲空间级别的项的开头。minfree 的百分比必须等于 0 或大于 0。缺省值为 20%。当审计文件系统的空间使用率达到 80% 时，审计数据便会存储到下一个可用的审计目录中。有关更多信息，请参见 audit_warn(1M) 手册页。
- dir **关键字**－用作**目录定义**行的开头。每一行都定义了系统用于存储其审计文件的审计文件系统和目录。可以定义一个或多个目录定义行。dir 行的顺序非常重要。auditd 守护进程将按照指定顺序在目录中创建审计文件。第一个目录为系统的**主审计目录**。第二个目录为**辅助审计目录**，当第一个目录变满时， auditd 守护进程在此创建审计文件，以此类推。有关更多信息，请参见 audit(1M) 手册页。
- plugin **关键字**－指定 syslog 插件模块的**插件路径**和审计类。此模块可提供 Solaris 审计记录向文本的实时转换。plugin 行中的审计类必须为 flags 行和 naflags 行中的审计类的子集。

有关 audit_control 文件的更多信息，请参见 audit_control(4) 手册页。

示例 30–2 audit_control 文件样例

以下是系统 noddy 的 audit_control 文件样例。noddy 使用的两个审计文件系统位于审计服务器 blinken 上，第三个审计文件系统从第二台审计服务器 winken 中装入。仅当 blinken 上的审计文件系统变满或不可用时，才使用第三个文件系统。如果 minfree 的值为 20%，则指定当文件系统的空间使用率达到 80% 时运行警告脚本。这些设置指定可以对登录和管理操作进行审计。可以对操作进行审计以查看成功和失败的情况。可以对所有类型的失败进行审计（但创建文件系统对象失败除外），还可以对不具归属性的事件进行审计。syslog 审计日志将记录较少的审计事件。此日志包含失败的登录和管理操作的文本摘要。

```
flags:lo,am,-all,^-fc

naflags:lo,nt

minfree:20

dir:/etc/security/audit/blinken/files

dir:/etc/security/audit/blinken.1/files

#

# Audit filesystem used when blinken fills up

#

dir:/etc/security/audit/winken

plugin:name=audit_syslog.so.1; p_flags=-lo,-am
```

audit_event 文件

/etc/security/audit_event 文件包含缺省的审计事件到类的映射。可以编辑此文件来更改类映射。更改类映射时，必须重新引导系统或运行 auditconfig -conf 命令以将已更改的映射读入内核。有关更多信息，请参见 audit_event(4) 手册页。

audit_startup 脚本

当系统进入多用户模式时，/etc/security/audit_startup 脚本可自动配置审计服务。在 auditd 守护进程启动之前，此脚本要执行以下任务：

- 配置审计事件到类的映射
- 设置审计策略选项

有关更多信息，请参见 audit_startup(1M) 手册页。

audit_user 数据库

/etc/security/audit_user 数据库可针对单个用户修改系统范围的预选类。添加到 audit_user 数据库内用户项中的类可以通过两种方法修改 audit_control 文件中的设置：

- 通过指定始终针对此用户进行审计的审计类
- 通过指定从不针对此用户进行审计的审计类

audit_user 数据库中的每个用户项都包含三个字段：

username:*always-audit-classes*:*never-audit-classes*

审计字段将按顺序进行处理。*always-audit-classes* 字段打开对其中的类的审计。*never-audit-classes* 字段关闭对其中的类的审计。

注 – 应避免出现在 *never-audit-classes* 字段中放置 all 审计类这种常见错误。此错误将导致针对此用户关闭所有审计功能，从而覆盖 *always-audit-classes* 字段中的设置。此设置还覆盖 audit_control 文件中的系统范围的审计类设置。

用户的 *never-audit-classes* 设置将覆盖系统缺省设置。您可能不需要覆盖系统缺省设置。例如，假定您需要针对用户 tamiko 审计所有内容，但文件系统对象的成功读取除外。您还需要将系统缺省设置应用于 tamiko。请注意，在以下 audit_user 项中放置第二个冒号 (:)：

```
tamiko:all,^+fr:
```

正确的项

正确的项意味着“始终审计所有内容，但成功的文件读取除外”。

```
tamiko:all:+fr
```

错误的项

不正确的项意味着“始终审计所有内容，但从不审计成功的文件读取”。跟在第二个冒号之后的 *never-audit-classes* 字段将覆盖系统缺省设置。在正确的项中，*always-audit-classes* 字段包括 all 审计类的例外。由于 *never-audit-classes* 字段中没有审计类，因此，不会覆盖 audit_control 文件中的系统缺省设置。

注 – 成功的事件和失败的事件将分别进行处理。进程针对失败事件生成的审计记录数多于针对成功事件生成的审计记录数。

---

audit_warn 脚本

如果 auditd 守护进程在写入审计记录时遇到异常情况，则 /etc/security/audit_warn 脚本可通知电子邮件别名。您可以针对自己的站点自定义此脚本，以便在出现可能需要手动干预的情况时发出警告。也可以指定如何自动处理这些情况。对于所有错误情况，audit_warn 脚本将带有 daemon.alert 这一严重级别的消息写入 syslog。您可以使用 syslog.conf 来配置 syslog 消息的控制台显示。audit_warn 脚本还可将消息发送到 audit_warn 电子邮件别名。应在启用审计功能时设置此别名。

当 auditd 守护进程检测到以下情况时，便会调用 audit_warn 脚本。此脚本向 audit_warn 别名发送电子邮件。

- 审计目录的空间使用率已超过了 minfree 值所允许的限制。minfree 值或软限制是指某个审计文件系统上可用空间的百分比。

  将使用字符串 soft 以及可用空间低于最小值的目录的名称调用 audit_warn 脚本。auditd 守护进程可自动切换到下一个适当的目录。此守护进程会一直将审计文件写入新的目录，直到此目录达到其 minfree 限制为止。然后，auditd 守护进程按顺序转至 audit_control 文件中列出的每个剩余目录。此守护进程将一直写入审计记录，直到每个目录达到其 minfree 限制为止。

- 所有审计目录均已达到 minfree 阈值。

  将使用字符串 allsoft 调用 audit_warn 脚本。系统会向控制台写入一条消息，同时还向 audit_warn 别名发送电子邮件。

  当 audit_control 文件中列出的所有审计目录均已达到其 minfree 阈值时，auditd 守护进程便会切换回第一个目录。此守护进程将一直写入审计记录，直到此目录完全变满为止。

- 审计目录已完全变满，没有剩余空间。

  将使用字符串 hard 以及目录名称调用 audit_warn 脚本。系统会向控制台写入一条消息，同时还向 audit_warn 别名发送电子邮件。

  auditd 守护进程可自动切换到下一个具有可用空间的适当目录。 auditd 守护进程将按顺序转至 audit_control 文件中列出的每个剩余目录。此守护进程将一直写入审计记录，直到每个目录变满为止。

- 所有审计目录已完全变满。将使用字符串 allhard 作为参数来调用 audit_warn 脚本。

  缺省情况下，系统会向控制台写入一条消息，同时还向 audit_warn 别名发送电子邮件。生成审计记录的进程将继续执行，但会对审计记录进行计数。将不会生成审计记录。有关如何处理此情况的示例，请参见示例 29–14 和第 568 页中的 “如何防止审计跟踪溢出”。

- 出现了内部错误。以下是可能出现的内部错误：
  - ebusy－另一个 auditd 守护进程已在运行
  - tmpfile－无法使用临时文件
  - postsigterm－在关闭审计功能期间收到信号
- 发现了 audit_control 文件的语法存在问题。缺省情况下，系统会向控制台发送一条消息，同时还会向 audit_warn 别名发送电子邮件。

有关详细信息，请参见 audit_warn(1M) 手册页。

bsmconv 脚本

/etc/security/bsmconv 脚本可启用审计服务。 bsmunconv 命令可禁用审计服务。当 bsmconv 脚本运行之后，您可以配置审计目录和审计配置文件。重新引导系统时，将启用审计功能。

有关详细信息，请参见 bsmconv(1M) 手册页。

用于管理审计的权限配置文件

Solaris OS 提供了用于配置审计服务以及分析审计跟踪的权限配置文件。

- **审计控制** – 可使角色配置 Solaris 审计。此权限配置文件可授权对审计服务所使用的文件进行配置，还可使角色运行审计命令。拥有审计控制配置文件的角色可以运行以下命令：audit 、auditd、auditconfig、bsmconv 和 bsmunconv。
- **审计查看** – 可使角色分析 Solaris 审计记录。此权限配置文件可授权使用 praudit 和 auditreduce 命令读取审计记录。拥有此权限配置文件的角色还可以运行 auditstat 命令。
- **系统管理员** – 拥有审计查看权限配置文件。拥有系统管理员权限配置文件的角色可以分析审计记录。

有关配置角色以处理审计服务的信息，请参见第 186 页中的 “配置 RBAC（任务列表）” 。

审计和 Solaris Zones

区域就是在单个 Solaris 操作系统实例中创建的一个虚拟操作系统环境。可以在全局区域中针对所有以相同方式审计的区域设置审计策略。

以相同方式对所有区域进行审计时，只能针对审计服务自定义全局区域中的配置文件。+zonename 策略选项非常有用。当设置了此选项时，所有区域中的审计记录便会包含区域的名称。然后，可以按区域名称后选审计记录。要了解有关审计策略的信息，请参见第 531 页中的 “确定审计策略” 。有关示例，请参见第 550 页中的 “如何配置审计策略” 。

还可以分别对各区域进行审计。在全局区域中设置了策略选项 perzone 时，每个非全局区域便会运行自己的审计守护进程，处理自己的审计队列，并指定其审计记录的内容和位置。非全局区域还可以设置多数审计策略选项。由于非全局区域不能设置影响整个系统的策略，因此，它不能设置 ahlt 或 perzone 策略。有关进一步介绍，请参见第 528 页中的 “如何在区域中规划审计” 。

注 – 如果在非全局区域中自定义了名称服务文件，但没有设置 perzone 策略，则选择可用记录时必须慎用审计工具。某个区域中的用户 ID 可以指代其他区域中具有相同 ID 的其他用户。

要生成可用的记录，请在全局区域中设置 zonename 审计策略。在全局区域中，运行带有 zonename 选项的 auditreduce 命令。然后，在 zonename 区域中，针对 auditreduce 输出运行 praudit 命令。

要了解有关区域的信息，请参见《System Administration Guide: Solaris Containers-Resource Management and Solaris Zones》中的第二部分，“Zones”。

审计类

通过指定一个或多个事件类，可以预选 Solaris 审计的系统范围的缺省值。将在系统的 audit_control 文件中针对每个系统预选这些类。将针对这些事件类对使用系统的用户进行审计。此文件在第 575 页中的 “audit_control 文件”中介绍。

您可以配置审计类并创建新的审计类。审计类名称的长度最多为 8 个字符。类说明最多可包含 72 个字符。允许使用数字和非字母数字字符。

可以通过将审计类添加到 audit_user 数据库中某个用户的项，来修改针对该用户的审计内容。审计类还可用作 auditconfig 命令的参数。有关详细信息，请参见 auditconfig(1M) 手册页。

审计类的定义

下表显示了每个预定义的审计类、每个审计类的说明性名称，以及简短说明。

表 30–1 预定义的审计类

审计类	说明性名称	说明
all	all	所有类（元类）
no	no_class	用于关闭事件预选的空值
na	non_attrib	不可归属事件
fr	file_read	读取数据，打开进行读取
fw	file_write	写入数据，打开进行写入
fa	file_attr_acc	访问对象属性：stat、pathconf
fm	file_attr_mod	更改对象属性：chown、flock
fc	file_creation	创建对象
fd	file_deletion	删除对象
cl	file_close	close 系统调用
ap	application	应用程序定义的事件
ad	administrative	管理操作（旧的管理元类）
am	administrative	管理操作（元类）
ss	system state	更改系统状态
as	system-wide administration	系统范围的管理

表 30–1 预定义的审计类 （续）

审计类	说明性名称	说明
ua	user administration	用户管理
aa	audit administration	利用审计
ps	process start	启动和停止进程
pm	process modify	修改进程
pc	process	进程（元类）
ex	exec	执行程序
io	ioctl	ioctl() 系统调用
ip	ipc	系统 V IPC 操作
lo	login_logout	登录和注销事件
nt	network	网络事件：bind、connect、 accept
ot	other	杂项，例如设备分配和 memcntl()

可以通过修改 /etc/security/audit_class 文件来定义新类，还可以重命名现有类。有关更多信息，请参见 audit_class(4) 手册页。

审计类语法

可以只针对成功情况对事件进行审计，也可以只针对失败情况对事件进行审计，还可以同时针对两种情况对事件进行审计。如果不带前缀，则同时针对成功和失败情况对事件类进行审计。如果带有加号 (+) 前缀，则对事件类进行审计以仅查看是否成功。如果带有减号 (-) 前缀，则对事件类进行审计以仅查看是否失败。下表显示了某些可能的审计类表示。

表 30–2 审计类的加号和减号前缀

[*prefix*]*class*	说明
lo	审计所有成功的登录和注销尝试，以及所有失败的登录尝试。用户不会遇到失败的注销尝试。
+lo	审计所有成功的登录和注销尝试。
-all	审计所有失败的事件。
+all	审计所有成功的事件。

注意 – `all` 类会生成大量数据并快速填满审计文件系统。仅当具有特殊的理由审计所有活动时，才使用 `all` 类。

先前选择的审计类可以通过插入记号前缀 ^ 进一步修改。下表显示了插入记号前缀如何修改预选的审计类。

表 30–3 用于修改已指定的审计类的插入记号前缀

^[*prefix*]*class*	说明
`-all,^-fc`	审计所有失败的事件，但不审计失败的文件对象创建尝试
`am,^+aa`	审计所有管理事件以查看成功和失败的情况，但不审计成功的审计管理尝试
`am,^ua`	审计所有管理事件以查看成功和失败的情况，但不审计用户管理事件

可以在以下文件和命令中使用审计类及其前缀：

- 在 `audit_control` 文件的 `flags` 行中
- 在 `audit_control` 文件的 `plugin ...p_flags=` 行中
- 在 `audit_user` 数据库的用户项中
- 作为 `auditconfig` 参数选项的参数

有关在 `audit_control` 文件中使用前缀的示例，请参见第 575 页中的 “`audit_control` 文件”。

审计策略

审计策略可确定是否将其他信息添加到审计跟踪中。不同审计策略选项的效果在第 531 页中的 “确定审计策略”中介绍。

进程审计特征

以下审计特征在初始登录时设置：

- **进程预选掩码** – 来自 `audit_control` 文件和 `audit_user` 数据库的审计类的组合。当用户登录时，登录进程将合并预选类以便为用户进程建立**进程预选掩码**。进程预选掩码指定每个审计类中的事件是否生成审计记录。

  以下算法介绍了系统如何获取用户的进程预选掩码：

  `(flags` *line* `+` *always-audit-classes*`)` `-` *never-audit-classes*

  将来自 `audit_control` 文件中 `flags` 行的审计类与来自 `audit_user` 数据库内用户项的 *always-audit-classes* 字段中的类相加。然后，从总数中减去用户的 *never-audit-classes* 字段中的类。

- **审计 ID**－当用户登录时，进程便会获取审计 ID。由用户初始进程启动的所有子进程都会继承审计 ID。审计 ID 有助于履行职责。即使在用户变为 root 之后，审计 ID 仍保持不变。保存在每条审计记录中的审计 ID 始终允许根据操作追溯到已登录的初始用户。
- **审计会话 ID**－审计会话 ID 在登录时指定。此会话 ID 将由所有子进程继承。
- **终端 ID（端口 ID、计算机 ID）**－终端 ID 包含主机名和 Internet 地址，后跟标识用户登录的物理设备的唯一数字。通常是通过控制台登录。对应于控制台设备的数字为 0。

审计跟踪

审计跟踪包含二进制审计文件。此跟踪由 auditd 守护进程创建。一旦使用 bsmconv 命令启用了审计服务，auditd 守护进程便会在系统引导时启动。auditd 守护进程负责收集审计跟踪数据以及写入审计记录。

审计记录以二进制格式存储在专用于审计文件的文件系统中。即使可以实际将审计目录放在不是专用于审计的文件系统中，也**不要**这样做（唯一可用的目录除外）。唯一可用的目录是指仅当其他适当的目录都不可用时，可以保存审计文件的目录。

还有一种可以使审计目录位于不是专用的审计文件系统上的情况：即在是否进行审计都可以的软件开发环境下。充分利用磁盘空间可能比保存审计跟踪数据更为重要。但是，在注重安全性的环境中，将审计目录放在其他文件系统中是不可接受的。

管理审计文件系统时，还应该考虑以下因素：

- 主机应该至少具有一个本地审计目录。如果主机无法与审计服务器进行通信，则本地目录可以用作最后使用的目录。
- 使用读写 (rw) 选项挂载审计目录。远程挂载审计目录时，还应使用 intr 和 noac 选项。
- 列出审计文件系统所在的审计服务器上的审计文件系统。导出列表应该包含站点上要审计的所有系统。

二进制审计文件名称约定

每个二进制审计文件都是自包含的记录集合。文件的名称标识生成记录的时间跨度以及生成这些记录的系统。

二进制审计文件名称

已完成的审计文件具有如下名称格式：

start-time.end-time.system

start-time　审计文件中第一条审计记录的生成时间

end-time　最后一条记录写入文件的时间

system 生成文件的系统的名称

仍处于活动状态的审计具有如下名称格式：

start-time.not_terminated.*system*

有关 not_terminated 和关闭的审计文件名称的示例，请参见第 567 页中的 “如何清除 not_terminated 审计文件”。

二进制审计文件时间标记

文件名中的时间标记可供 auditreduce 命令用来查找特定时间范围内的记录。由于可以在线累积一个月或更长时间的审计文件，因此，这些时间标记非常重要。要搜索在过去 24 小时内生成的记录的所有文件将需要很大开销。

start-time 和 *end-time* 是具有 1 秒分辨率的时间标记，按照格林威治标准时间 (Greenwich Mean Time, GMT) 指定。格式为四位数字表示的年，后跟两位数字表示的月、日、小时、分钟和秒，如下所示：

YYYYMMDDHHMMSS

以 GMT 表示时间标记可确保即使跨不同的时区，也可以按照正确的顺序对这些标记排序。由于时间标记以 GMT 表示，因此，必须将日期和小时转换为当前时区才有意义。每当使用标准文件命令而不是 auditreduce 命令来处理这些文件时，都要切记这一点。

审计记录结构

审计记录是一系列审计标记。每个审计标记都包含事件信息，例如用户 ID、时间和日期。由 header 标记开始审计记录，可选的 trailer 标记结束记录。其他审计标记包含与审计事件相关的信息。下图显示了典型的审计记录。

图 30–3 典型的审计记录结构

审计记录分析

审计记录分析涉及从审计跟踪中后选记录。可以使用两种方法之一来分析收集的二进制数据。

- 可以分析二进制数据流。要分析数据流，需要了解每个标记中的字段顺序以及每条记录中的标记顺序，还需要了解审计记录的变体。例如，`ioctl()` 系统调用可针对“错误的文件名称”创建一条审计记录，此记录包含的标记与“无效的文件说明符”的审计记录中包含的标记不同。
  - 有关每个审计标记中的二进制数据顺序的说明，请参见 `audit.log`(4) 手册页。
  - 要了解审计记录中的标记顺序的说明，请使用 `bsmrecord` 命令。`bsmrecord` 命令的输出中包含在不同情况下出现的不同格式。方括号 (`[]`) 指示审计标记是可选的。有关更多信息，请参见 `bsmrecord`(1M) 手册页。有关示例，另请参见第 558 页中的 “如何显示审计记录格式”。
- 您可以使用 `praudit` 命令。此命令的选项可提供不同的文本输出。例如，`praudit -x` 命令可以为脚本和浏览器中的输入提供 XML。`praudit` 输出不包括仅用于帮助分析二进制数据的字段。输出不一定遵照二进制字段的顺序。此外，无法保证 Solaris 发行版之间 `praudit` 输出的顺序和格式完全相同。

  有关 `praudit` 输出的示例，请参见第 565 页中的 “如何查看二进制审计文件的内容”和 `praudit`(1M) 手册页。

  有关每个审计标记的 `praudit` 输出的说明，请参见第 585 页中的 “审计标记格式”部分中的各个标记。

审计标记格式

每个审计标记都有一个标记类型标识符，它后跟特定于标记的数据。每种标记类型都有自己的格式。下表显示了每个标记的标记名称以及简短说明。为了与先前的 Solaris 发行版兼容，将维护过时的标记。

表 30–4 用于 Solaris 审计的审计标记

标记名称	说明	更多信息
`acl`	访问控制列表 (Access Control List, ACL) 信息	第 587 页中的 “`acl` 标记”
`arbitrary`	具有格式和类型信息的数据	第 587 页中的 “`arbitrary` 标记（已过时）”
`arg`	系统调用参数值	第 588 页中的 “`arg` 标记”
`attribute`	文件 vnode 标记	第 588 页中的 “`attribute` 标记”
`cmd`	命令参数和环境变量	第 589 页中的 “`cmd` 标记”
`exec_args`	可执行的系统调用参数	第 589 页中的 “`exec_args` 标记”

表 30–4 用于 Solaris 审计的审计标记 （续）

标记名称	说明	更多信息
exec_env	可执行的系统调用环境变量	第 589 页中的 “exec_env 标记”
exit	程序退出信息	第 590 页中的 “exit 标记（已过时）”
file	审计文件信息	第 590 页中的 “file 标记”
group	进程组信息	第 591 页中的 “group 标记（已过时）”
groups	进程组信息	第 591 页中的 “groups 标记”
header	指示审计记录的开始	第 591 页中的 “header 标记”
in_addr	Internet 地址	第 592 页中的 “in_addr 标记”
ip	IP 数据包头信息	第 592 页中的 “ip 标记（已过时）”
ipc	系统 V IPC 信息	第 592 页中的 “ipc 标记”
ipc_perm	系统 V IPC 对象标记	第 593 页中的 “ipc_perm 标记”
iport	Internet 端口地址	第 593 页中的 “iport 标记”
opaque	无结构数据（未指定的格式）	第 594 页中的 “opaque 标记（已过时）”
path	路径信息	第 594 页中的 “path 标记”
path_attr	访问路径信息	第 594 页中的 “path_attr 标记”
privilege	权限集信息	第 595 页中的 “privilege 标记”
process	进程标记信息	第 595 页中的 “process 标记”
return	系统调用的状态	第 597 页中的 “return 标记”
sequence	序列号标记	第 597 页中的 “sequence 标记”
socket	套接字类型和地址	第 597 页中的 “socket 标记”
subject	主题标记信息（与 process 标记的格式相同）	第 598 页中的 “subject 标记”
text	ASCII 字符串	第 600 页中的 “text 标记”
trailer	指示审计记录的结束	第 600 页中的 “trailer 标记”
uauth	使用授权	第 600 页中的 “uauth 标记”
zonename	区域名称	第 601 页中的 “zonename 标记”

审计记录始终以 header 标记开始。header 标记指示审计记录在审计跟踪中的开始位置。对于可归属事件，subject 和 process 标记指代导致此事件发生的进程的值。对于不可归属事件，process 标记指代系统。

acl 标记

acl 标记可记录有关访问控制列表 (Access Control List, ACL) 的信息。此标记包含四个固定字段：

- 标记 ID 字段，将此标记标识为 acl 标记
- 指定 ACL 类型的字段
- ACL 值字段
- 列出与此 ACL 关联的权限的字段

praudit 命令可按如下方式显示 acl 标记：

```
acl,jdoe,staff,0755
```

arbitrary 标记（已过时）

arbitrary 标记可封装数据以进行审计跟踪。此标记包含四个固定字段以及一个数据组。固定字段如下所示：

- 标记 ID 字段，将此标记标识为 arbitrary 标记
- 建议的列显格式字段，例如十六进制
- 项大小字段，指定封装的数据大小（例如短）
- 计数字段，提供后续项的数量

此标记的其余部分由指定类型的 *count* 组成。praudit 命令可按如下方式显示 arbitrary 标记：

```
arbitrary,decimal,int,1

42
```

下表显示了列显格式字段的可能值。

表 30–5 arbitrary 标记的列显格式字段值

值	操作
AUP_BINARY	以二进制格式列显日期
AUP_OCTAL	以八进制格式列显日期
AUP_DECIMAL	以十进制格式列显日期
AUP_HEX	以十六进制格式列显日期
AUP_STRING	将日期列显为字符串

下表显示了项大小字段的可能值。

表 30–6 arbitrary 标记的项大小字段值

值	操作
AUR_BYTE	数据以 1 字节的字节单位列显
AUR_SHORT	数据以 2 字节的短单位列显
AUR_LONG	数据以 4 字节的长单位列显

arg 标记

arg 标记包含有关系统调用参数的信息：系统调用的参数号、参数值以及可选说明。此标记允许在审计记录中使用 32 位整数的系统调用参数。arg 标记具有五个字段：

- 标记 ID 字段，将此标记标识为 arg 标记
- 参数 ID 字段，告知标记所指的系统调用参数
- 参数值字段
- 说明性文本字符串长度字段
- 文本字符串字段

praudit 命令可按如下方式显示不包含第四个字段的 arg 标记：

```
argument,4,0xffbfe0ac,pri
```

praudit -x 命令包括显示的字段的名称：

```
<argument arg-num="4" value="0xffbfe0ac" desc="pri"/>
```

attribute 标记

attribute 标记包含文件 vnode 的信息。此标记具有七个字段：

- 标记 ID 字段，将此标记标识为 attribute 标记
- 文件访问模式和类型字段
- 属主用户 ID 字段
- 属主组 ID 字段
- 文件系统 ID 字段
- 节点 ID 字段
- 文件可能表示的设备 ID 字段

有关文件系统 ID 和设备 ID 的详细信息，请参见 statvfs(2) 手册页。

attribute 标记通常与 path 标记同时出现。attribute 标记在搜索路径期间生成。如果出现路径搜索错误，则没有可用的 vnode 来获取必需的文件信息。因此，attribute 标记不作为审计记录的一部分包括。praudit 命令可按如下方式显示 attribute 标记：

```
attribute,20666,root,root,247,4829,450971566127
```

cmd 标记

cmd 标记记录参数列表和与命令关联的环境变量的列表。

cmd 标记包含以下字段：

- 标记 ID 字段，将此标记标识为 cmd 标记
- 计数字段，用于命令参数
- 参数列表字段
- 下一个字段长度字段
- 参数内容字段
- 计数字段，用于环境变量
- 环境变量列表字段
- 下一个字段长度字段
- 环境变量内容字段

praudit 命令可按如下方式显示 cmd 标记：

```
cmd,argcnt,3,ls,-l,/etc,envcnt,0,
```

exec_args 标记

exec_args 标记可记录 exec() 系统调用参数。exec_args 标记具有两个固定字段：

- 标记 ID 字段，将此标记标识为 exec_args 标记
- 计数字段，表示传递给 exec() 系统调用的参数数量

此标记的其余部分由 *count* 字符串组成。praudit 命令可按如下方式显示 exec_args 标记：

```
exec_args,2,vi,/etc/security/audit_user
```

注 – 仅当 argv 审计策略选项处于活动状态时，才输出 exec_args 标记。

exec_env 标记

exec_env 标记可记录 exec() 系统调用的当前环境变量。exec_env 标记具有两个固定字段：

- 标记 ID 字段，将此标记标识为 exec_env 标记
- 计数字段，表示传递给 exec() 系统调用的参数数量

此标记的其余部分由 *count* 字符串组成。praudit 命令可按如下方式显示 exec_env 标记：

```
exec_env,25,
GROUP=staff,HOME=/export/home/jdoe,HOST=exm1,HOSTTYPE=sun4u,HZ=100,
```

```
LC_COLLATE=en_US.ISO8859-1,LC_CTYPE=en_US.ISO8859-1,LC_MESSAGES=C,

LC_MONETARY=en_US.ISO8859-1,LC_NUMERIC=en_US.ISO8859-1,

LC_TIME=en_US.ISO8859-1,LOGNAME=jdoe,MACHTYPE=sparc,

MAIL=/var/mail/jdoe,OSTYPE=solaris,PATH=/usr/sbin:/usr/bin,PS1=#,

PWD=/var/audit,REMOTEHOST=192.168.13.5,SHELL=/usr/bin/csh,SHLVL=1,

TERM=dtterm,TZ=US/Pacific,USER=jdoe,VENDOR=sun
```

注 – 仅当 `arge` 审计策略选项处于活动状态时，才输出 `exec_env` 标记。

`exit` 标记（已过时）

`exit` 标记可记录程序的退出状态。`exit` 标记包含以下字段：

- 标记 ID 字段，将此标记标识为 `exit` 标记
- 程序退出状态字段，传递给 `exit()` 系统调用时显示
- 返回值字段，描述退出状态或提供系统错误号

`praudit` 命令可按如下方式显示 `exit` 标记：

```
exit,Error 0,0
```

`file` 标记

`file` 标记是由 `auditd` 守护进程生成的特殊标记。当停用旧审计文件时，此标记可标记新审计文件的开始以及旧审计文件的结束。`auditd` 守护进程可生成包含此标记的特殊审计记录，以便将连续的审计文件同时“链接”到一个审计跟踪中。`file` 标记具有四个字段：

- 标记 ID 字段，将此标记标识为 `file` 标记
- 时间标记字段，标识创建或关闭文件的日期和时间
- 文件名长度字段
- 包含以空字符结尾的文件名的字段

`praudit -x` 命令可显示 `file` 标记的字段：

```
file,2003-10-13 11:21:35.506 -07:00,

    /var/audit/localhost/files/20031013175058.20031013182135.example1
```

group 标记（已过时）

此标记已由 groups 标记替换。请参见第 591 页中的 “groups 标记”。

groups 标记

groups 标记替换了 group 标记。groups 标记可记录进程凭证中的组项。groups 标记具有两个固定字段：

- 标记 ID 字段，将此标记标识为 groups 标记
- 计数字段，表示此审计记录中包含的组数

此标记的其余部分由 *count* 组项组成。praudit 命令可按如下方式显示 groups 标记：

```
groups,staff,admin
```

注 – 仅当 group 审计策略选项处于活动状态时，才输出 groups 标记。

---

header 标记

header 标记的特殊之处在于它标记审计记录的开始。header 标记与 trailer 标记组合使用以将记录中的所有其他标记括在一起。header 标记具有八个字段：

- 标记 ID 字段，将此标记标识为 header 标记
- 字节计数字段，表示审计记录的总长度（包括 header 和 trailer 标记）
- 版本号字段，标识审计记录结构的版本
- 审计事件 ID 字段，标识记录所表示的审计事件
- ID 修饰符字段，标识审计事件的特殊特征
- 地址类型字段，IPv4 或 IPv6
- 计算机 IP 地址字段
- 时间和日期字段，表示记录的创建时间和日期

在 64 位系统上，header 标记使用 64 位时间标记而不是 32 位时间标记显示。

praudit 命令可按如下方式显示 ioctl() 系统调用的 header 标记：

```
header,176,2,ioctl(2),fe,example1,2003-09-08 11:23:31.050 -07:00
```

ID 修饰符字段具有以下已定义标志：

```
0x4000            PAD_NOTATTR                       nonattributable event

0x8000            PAD_FAILURE                       failed audit event
```

in_addr 标记

in_addr 标记包含一个 Internet 协议地址。自 Solaris 8 发行版开始，Internet 地址可以用 IPv4 格式或 IPv6 格式显示。IPv4 地址使用 4 个字节。IPv6 地址使用 1 个字节来描述地址类型，使用 16 个字节来描述地址。in_addr 标记具有三个字段：

- 标记 ID 字段，将此标记标识为 in_addr 标记
- IP 地址类型字段，IPv4 或 IPv6
- IP 地址字段

praudit 命令可按如下方式显示不包含第二个字段的 in_addr 标记：

```
ip address,192.168.113.7
```

ip 标记（已过时）

ip 标记包含 Internet 协议数据包头的副本。ip 标记具有两个字段：

- 标记 ID 字段，将此标记标识为 ip 标记
- IP 数据包头副本字段，即全部 20 个字节

praudit 命令可按如下方式显示 ip 标记：

```
ip address,0.0.0.0
```

IP 数据包头结构在 /usr/include/netinet/ip.h 文件中定义。

ipc 标记

ipc 标记包含调用方用于标识特殊 IPC 对象的系统 V IPC 消息句柄、信号句柄或共享内存句柄。ipc 标记具有三个字段：

- 标记 ID 字段，将此标记标识为 ipc 标记
- 类型字段，指定 IPC 对象的类型
- 句柄字段，标识 IPC 对象

注 – IPC 对象标识符不符合 Solaris 审计标记的上下文无关性质。没有可唯一地标识 IPC 对象的全局“名称”。相反，IPC 对象由其句柄标识。这些句柄仅当 IPC 对象处于活动状态时才有效。但是，标识 IPC 对象应该不存在问题。很少用到系统 V IPC 机制，并且这些机制全部共享相同的审计类。

下表显示了 IPC 对象类型字段的可能值。这些值在 /usr/include/bsm/audit.h 文件中定义。

表 30–7 IPC 对象类型字段的值

名称	值	说明
AU_IPC_MSG	1	IPC 消息对象
AU_IPC_SEM	2	IPC 信号对象
AU_IPC_SHM	3	IPC 共享内存对象

praudit 命令可按如下方式显示 ipc 标记：

```
IPC,msg,3
```

ipc_perm 标记

ipc_perm 标记包含系统 V IPC 访问权限的副本。此标记将被添加到由 IPC 共享内存事件、IPC 信号事件和 IPC 消息事件生成的审计记录中。ipc_perm 标记具有八个字段：

- 标记 ID 字段，将此标记标识为 ipc_perm 标记
- IPC 属主的用户 ID 字段
- IPC 属主的组 ID 字段
- IPC 创建者的用户 ID 字段
- IPC 创建者的组 ID 字段
- IPC 访问模式字段
- IPC 序列号字段
- IPC 密钥值字段

praudit 命令可按如下方式显示 ipc_perm 标记：

```
IPC perm,root,sys,root,sys,0,0,0x00000000
```

值来自与 IPC 对象关联的 ipc_perm 结构。

iport 标记

iport 标记包含 TCP 或 UDP 端口地址。iport 标记具有两个字段：

- 标记 ID 字段，将此标记标识为 iport 标记
- TCP 或 UDP 端口地址字段

praudit 命令可按如下方式显示 iport 标记：

```
ip port,0xf6d6
```

opaque 标记（已过时）

opaque 标记包含作为一系列字节的未设置格式的数据。opaque 标记具有三个字段：

- 标记 ID 字段，将此标记标识为 opaque 标记
- 数据字节计数字段
- 字节数据组字段

praudit 命令可按如下方式显示 opaque 标记：

```
opaque,12,0x4f5041515545204441544100
```

path 标记

path 标记包含对象的访问路径信息。此标记包含以下字段：

- 标记 ID 字段，将此标记标识为 path 标记
- 路径长度字段
- 对象绝对路径字段，基于系统的实际根目录

praudit 命令可按如下方式显示不包含第二个字段的 path 标记：

```
path,/etc/security/audit_user
```

praudit -x 命令可按如下方式显示 path 标记：

```
<path>/etc/security/audit_user</path>
```

下图显示了 path 标记的格式。

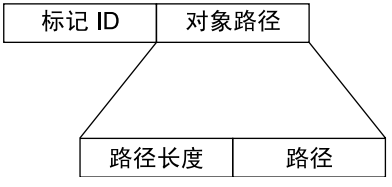


图 30–4 path 标记格式

path_attr 标记

path_attr 标记包含对象的访问路径信息。访问路径指定了 path 标记对象以下的属性文件对象的顺序。系统调用（例如 openat()）可访问属性文件。有关属性文件对象的更多信息，请参见 fsattr(5) 手册页。

path_attr 标记包含以下字段：

- 标记 ID 字段，将此标记标识为 path_attr 标记

- 计数字段，表示属性文件路径的段数
- *count* 空字符结尾字符串字段

praudit 命令可按如下方式显示 path_attr 标记：

```
path_attr,1,attr_file_name
```

privilege 标记

privilege 标记可记录针对进程的权限使用。并不针对基本集中的权限记录 privilege 标记。如果通过管理操作已从基本集中删除了权限，则会记录此权限的使用。有关权限的更多信息，请参见第 177 页中的 “权限（概述）”

privilege 标记包含以下字段：

- 标记 ID 字段，将此标记标识为 privilege 标记
- 后一字段长度字段
- 权限集名称字段
- 后一字段长度字段
- 权限列表字段

praudit 命令可按如下方式显示 privilege 标记：

```
privilege,effective,
```

process 标记

process 标记包含有关与进程关联的用户（例如信号接收者）的信息。process 标记具有九个字段：

- 标记 ID 字段，将此标记标识为 process 标记
- 审计 ID 字段
- 有效用户 ID 字段
- 有效组 ID 字段
- 实际用户 ID 字段
- 实际组 ID 字段
- 进程 ID 字段
- 审计会话 ID 字段
- 终端 ID 字段，包含设备 ID 和计算机 ID

审计 ID、用户 ID、组 ID、进程 ID 以及会话 ID 均为长字段，而不是短字段。

注 – process 标记的会话 ID、实际用户 ID 或实际组 ID 字段可能不可用。因此将值设置为 -1。

任何包含终端 ID 的标记都具有数个变体。 praudit 命令可隐藏这些变体。因此，对于任何包含终端 ID 的标记，均采用相同的方式处理终端 ID。终端 ID 为 IP 地址、端口号或设备 ID。设备 ID（例如连接到调制解调器的串行端口）可以为零。终端 ID 通过数种格式之一进行指定。

以设备编号表示的终端 ID 按如下方式指定：

- **32 位应用程序**－4 个字节表示设备编号，4 个字节未使用
- **64 位应用程序**－8 个字节表示设备编号，4 个字节未使用

在 Solaris 8 之前的发行版中，以端口号表示的终端 ID 按如下方式指定：

- **32 位应用程序**－4 个字节表示端口号，4 个字节表示 IP 地址
- **64 位应用程序**－8 个字节表示端口号，4 个字节表示 IP 地址

自 Solaris 8 发行版开始，以端口号表示的终端 ID 按如下方式指定：

- **使用 IPv4 的 32 位**－4 个字节表示端口号，4 个字节表示 IP 类型，4 个字节表示 IP 地址
- **使用 IPv6 的 32 位**－4 个字节表示端口号，4 个字节表示 IP 类型，16 个字节表示 IP 地址
- **使用 IPv4 的 64 位**－8 个字节表示端口号，4 个字节表示 IP 类型，4 个字节表示 IP 地址
- **使用 IPv6 的 64 位**－8 个字节表示端口号，4 个字节表示 IP 类型，16 个字节表示 IP 地址

praudit 命令可按如下方式显示 process 标记：

```
process,root,root,sys,root,sys,0,0,0,0.0.0.0
```

下图显示了 process 标记的格式。

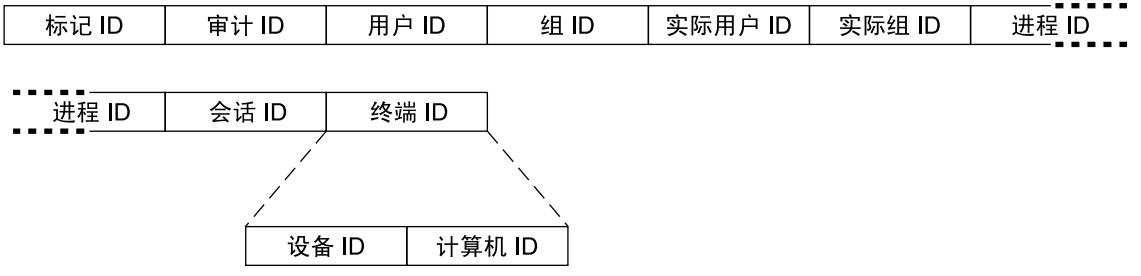


图 30–5 process 标记格式

return 标记

return 标记包含系统调用的返回状态 (u_error) 以及进程返回值 (u_rval1)。此标记具有三个字段：

- 标记 ID 字段，将此标记标识为 return 标记
- 系统调用错误状态字段
- 系统调用返回值字段

return 标记始终作为内核针对系统调用生成的审计记录的一部分返回。在应用程序审计中，此标记指示退出状态以及其他返回值。

praudit 命令可按如下方式显示系统调用的 return 标记：

```
return,failure: Operation now in progress,-1
```

praudit -x 命令可按如下方式显示 return 标记：

```
<return errval="failure: Operation now in progress" retval="-1/">
```

sequence 标记

sequence 标记包含一个序列号。此标记用于进行调试。sequence 标记具有两个字段：

- 标记 ID 字段，将此标记标识为 sequence 标记
- 32 位无符号长字段，包含序列号

每当向审计跟踪中添加审计记录时，序列号便会增加。praudit 命令可按如下方式显示 sequence 标记：

```
sequence,1292
```

praudit -x 命令可按如下方式显示 sequence 标记：

```
<sequence seq-num="1292"/>
```

注 – 仅当 seq 审计策略选项处于活动状态时，才输出 sequence 标记。

---

socket 标记

socket 标记包含介绍 Internet 套接字的信息。在某些情况下，此标记具有四个字段：

- 标记 ID 字段，将此标记标识为 socket 标记
- 套接字类型字段，指示所引用的套接字的类型（TCP、UDP 或 UNIX）
- 本地端口字段

- 本地 IP 地址字段

praudit 命令可按如下方式显示此 socket 标记实例：

```
socket,0x0002,0x83b1,localhost
```

在大多数情况下，此标记具有八个字段：

- 标记 ID 字段，将此标记标识为 socket 标记
- 套接字域字段
- 套接字类型字段，指示所引用的套接字的类型（TCP、UDP 或 UNIX）
- 本地端口字段
- 地址类型字段，IPv4 或 IPv6
- 本地 IP 地址字段
- 远程端口字段
- 远程 IP 地址字段

自 Solaris 8 发行版开始，Internet 地址可以用 IPv4 格式或 IPv6 格式显示。IPv4 地址使用 4 个字节。IPv6 地址使用 1 个字节来描述地址类型，使用 16 个字节来描述地址。

praudit 命令可按如下方式显示 socket 标记：

```
socket,0x0002,0x0002,0x83cf,example1,0x2383,server1.Subdomain.Domain.COM
```

praudit -x 命令介绍了 socket 标记字段。换行的目的是为了进行显示。

```
<socket sock_domain="0x0002" sock_type="0x0002" lport="0x83cf"

laddr="example1" fport="0x2383" faddr="server1.Subdomain.Domain.COM"/>
```

subject 标记

subject 标记可介绍执行或尝试执行某操作的用户。格式与 process 标记的格式相同。subject 标记具有九个字段：

- 标记 ID 字段，将此标记标识为 subject 标记
- 审计 ID 字段
- 有效用户 ID 字段
- 有效组 ID 字段
- 实际用户 ID 字段
- 实际组 ID 字段
- 进程 ID 字段
- 审计会话 ID 字段
- 终端 ID 字段，包含设备 ID 和计算机 ID

审计 ID、用户 ID、组 ID、进程 ID 以及会话 ID 均为长字段，而不是短字段。

注 – subject 标记的会话 ID、实际用户 ID 或实际组 ID 字段可能不可用。因此将值设置为 -1。

任何包含终端 ID 的标记都具有数个变体。 praudit 命令可隐藏这些变体。因此，对于任何包含终端 ID 的标记，均采用相同的方式处理终端 ID。终端 ID 为 IP 地址、端口号或设备 ID。设备 ID（例如连接到调制解调器的串行端口）可以为零。终端 ID 通过数种格式之一进行指定。

以设备编号表示的终端 ID 按如下方式指定：

- **32 位应用程序**－4 个字节表示设备编号，4 个字节未使用
- **64 位应用程序**－8 个字节表示设备编号，4 个字节未使用

在 Solaris 8 之前的发行版中，以端口号表示的终端 ID 按如下方式指定：

- **32 位应用程序**－4 个字节表示端口号，4 个字节表示 IP 地址
- **64 位应用程序**－8 个字节表示端口号，4 个字节表示 IP 地址

自 Solaris 8 发行版开始，以端口号表示的终端 ID 按如下方式指定：

- **使用 IPv4 的 32 位**－4 个字节表示端口号，4 个字节表示 IP 类型，4 个字节表示 IP 地址
- **使用 IPv6 的 32 位**－4 个字节表示端口号，4 个字节表示 IP 类型，16 个字节表示 IP 地址
- **使用 IPv4 的 64 位**－8 个字节表示端口号，4 个字节表示 IP 类型，4 个字节表示 IP 地址
- **使用 IPv6 的 64 位**－8 个字节表示端口号，4 个字节表示 IP 类型，16 个字节表示 IP 地址

subject 标记始终作为内核针对系统调用生成的审计记录的一部分返回。praudit 命令可按如下方式显示 subject 标记：

```
subject,jdoe,root,staff,root,staff,424,223,0 0 example1
```

下图显示了 subject 标记的格式。

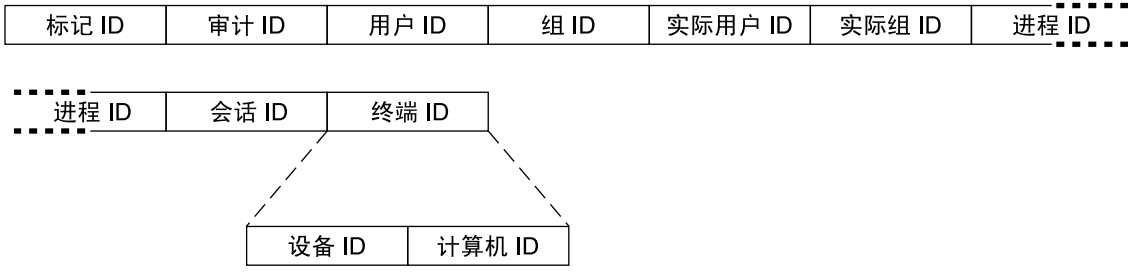


图 30–6 subject 标记格式

text 标记

text 标记包含一个文本字符串。此标记具有三个字段：

- 标记 ID 字段，将此标记标识为 text 标记
- 文本字符串长度字段
- 文本字符串自身字段

praudit 命令可按如下方式显示 text 标记：

```
text,logout jdoe
```

trailer 标记

header 和 trailer 这两个标记的特殊之处在于它们将审计记录的各个结束点区分开来，并将所有其他标记括在一起。header 标记可开始审计记录。trailer 标记可结束审计记录。trailer 标记是可选标记。仅当已经设置了 trail 审计策略选项时，才将 trailer 标记作为每条记录的最后一个标记添加。

在尾部打开的情况下生成审计记录时，auditreduce 命令可以检验尾部是否正确指回记录首部。trailer 标记支持向后查找审计跟踪。

trailer 标记具有三个字段：

- 标记 ID 字段，将此标记标识为 trailer 标记
- 填充编号字段，有助于标记记录结尾
- 审计记录中的字符总数字段，包括 header 和 trailer 标记

praudit 命令可按如下方式显示不包含第二个字段的 trailer 标记：

```
trailer,136
```

uauth 标记

uauth 标记记录在命令或操作中使用的授权。

uauth 标记包含以下字段：

- 标记 ID 字段，将此标记标识为 uauth 标记
- 后一字段中文本长度字段
- 授权列表字段

praudit 命令可按如下方式显示 uauth 标记：

```
use of authorization,solaris.admin.printer.delete
```

zonename 标记

zonename 标记可记录发生审计事件的区域。字符串 "global" 指示在全局区域中发生的审计事件。

zonename 标记包含以下字段：

- 标记 ID 字段，将此标记标识为 zonename 标记
- 后一字段中文本长度字段
- 区域名称字段

praudit 命令可按如下方式显示 zonename 标记：

```
zonename,graphzone
```

词汇表

Access Control List, ACL（访问控制列表） 与传统的 UNIX 文件保护相比，访问控制列表 (access control list, ACL) 可提供更为精细的文件安全性。例如，组使用 ACL 可以获取对某文件的读取权限，同时该组中仅有一个成员可向此文件进行写入。

admin principal（管理主体） 名称形式为 *username*/`admin` 的用户主体（如 `jdoe/admin`）。与普通用户主体相比，管理主体可以具有更多权限（例如，可以更改策略）。另请参见 principal name（主体名称）、user principal（用户主体）。

AES 高级加密标准。一种对称的 128 位块数据加密技术。美国政府在 2000 年 10 月采用算法的 Rijndael 变体作为加密标准，AES 从而取代了 user principal（用户主体）成为政府的加密标准。

algorithm（算法） 加密算法。这是一种已建立的递归计算过程，用于对输入执行加密或散列操作。

application server（应用程序服务器） 请参见 network application server（网络应用程序服务器）。

audit files（审计文件） 二进制审计日志。审计文件单独存储在一个审计分区中。

audit partition（审计分区） 配置用于保存审计文件的硬盘分区。

audit policy（审计策略） 决定要记录的审计事件的全局设置和按用户设置。通常，应用于审计服务的全局设置会影响审计跟踪所包括的可选信息。`cnt` 和 `ahlt` 这两种设置会影响系统在填充审计队列时执行的操作。例如，审计策略可能要求每条审计记录都包含一个序列号。

audit trail（审计跟踪） 来自所有主机的所有审计文件的集合。

authentication（验证） 验证主体所声明的身份的过程。

authenticator（验证者） 当客户机从 KDC 请求票证以及从服务器请求服务时，会传递验证者。这些验证者包含使用仅对客户机和服务器公开的会话密钥所生成的信息，这些信息可以作为最新来源进行检验，从而表明事务是安全的。验证者可与票证一起使用来验证用户主体。验证者中包括用户的主体名称、用户主机的 IP 地址，以及时间标记。与票证不同，验证者只能使用一次，通常在请求访问服务时使用。验证者是使用特定客户机和服务器的会话密钥进行加密的。

authorization（授权）

1. 在 Kerberos 中，是指决定主体是否可以使用服务，允许主体访问哪些对象，以及可对每个对象执行的访问操作类型的过程。

2. 在基于角色的访问控制 (role-based access control, RBAC) 中，是指可以指定给角色或用户（或嵌入权限配置文件中）的权限，此权限用于执行安全策略原本禁止的操作类。

Basic Security Module, BSM（基本安全模块）

Solaris 审计服务和设备分配。这些功能共同实现 C2 安全级别。

basic set（基本集）

登录时为用户进程指定的权限集。在未修改的系统上，每个用户的初始可继承集等同于登录时获取的基本集。

Blowfish

一种对称块加密算法，采用 32 位到 448 位的可变长度密钥。其作者 Bruce Schneier 声称 Blowfish 已针对密钥不经常更改的应用程序进行优化。

client principal（客户机主体）

(RPCSEC_GSS API) 是指使用受 RPCSEC_GSS 保护的网络服务的客户机（用户或应用程序）。客户机主体名称将以 `rpc_gss_principal_t` 结构的形式进行存储。

client（客户机）

狭义上讲，是指代表用户使用网络服务的进程，例如，使用 `rlogin` 的应用程序。在某些情况下，服务器本身即可是其他某个服务器或服务的客户机。

广义上讲，是指 a) 接收 Kerberos 凭证的主机，以及 b) 使用由服务器提供的服务的主机。

非正式地讲，是指使用服务的主体。

clock skew（时钟相位差）

所有参与 Kerberos 验证系统的主机上的内部系统时钟可以相差的最大时间量。如果任意两台参与主机之间的时间偏差超过了时钟相位差，则请求会被拒绝。 可以在 `krb5.conf` 文件中指定时钟相位差。

confidentiality（保密性）

请参见 privacy（保密性）。

consumer（使用者）

在 Solaris 加密框架中，使用者是指使用提供者提供的加密服务的用户。使用者可以是应用程序、最终用户或内核操作。Kerberos、IKE 和 IPsec 便是几个使用者示例。有关提供者的示例，请参见 provider（提供器）。

credential cache（凭证高速缓存）

包含从 KDC 接收的凭证的存储空间（通常为文件）。

credential（凭证）

包括票证及匹配的会话密钥的信息软件包。用于验证主体的身份。另请参见 ticket（票证）、session key（会话密钥）。

cryptographic algorithm（加密算法）

请参见 algorithm（算法）。

DES

数据加密标准。一种对称密钥加密方法，开发于 1975 年，1981 年由 ANSI 标准化为 ANSI X.3.92。DES 使用 56 位密钥。

device allocation（设备分配） 用户级别的设备保护。设备分配强制规定一次只能由一个用户独占使用一台设备。重用设备之前，将清除设备数据。可以使用授权来限制允许分配设备的用户。

device policy（设备策略） 内核级别的设备保护。设备策略在设备上作为两个权限集实现。第一个权限集控制对设备的读取权限，第二个权限集控制对设备的写入权限。另请参见 policy（策略）。

Diffie-Hellman protocol（Diffie-Hellman 协议） 也称为公钥密码学。Diffie 和 Hellman 于 1976 年开发的非对称密钥一致性协议。使用该协议，两个用户可以在没有任何以前秘密的情况下通过不安全的介质交换密钥。Diffie-Hellman 供 Kerberos 使用。

digest（摘要） 请参见 message digest（消息摘要）。

DSA 数字签名算法。一种公钥算法，采用大小可变（512 位到 4096 位）的密钥。美国政府标准 DSS 可达 1024 位。DSA 的输入依赖于 SHA1。

effective set（有效集） 当前对进程有效的权限集。

flavor（特性） 以前，**安全风格**和**验证特性**具有相同的含义，都是表示验证类型（AUTH_UNIX、AUTH_DES、AUTH_KERB）的特性。RPCSEC_GSS 也是安全风格，即使其除了验证之外还提供完整性和保密性服务也是如此。

forwardable ticket（可转发票证） 可供客户机在不需要完成远程主机上的完整验证过程的情况下用于请求此主机票证的票证。例如，如果用户 `david` 从用户 `jennifer` 的计算机上获取可转发票证，则他可以登录到自己的计算机，而不需要获取新票证（因此不需要再次进行自我验证）。另请参见 proxiable ticket（可代理票证）。

FQDN 全限定域名。例如，`central.example.com`（与简单的 `denver` 相对）。

GSS-API 通用安全服务应用程序编程接口。为各种模块化安全服务（包括 Kerberos 服务）提供支持的网络层。GSS-API 可用于安全验证服务、完整性服务和保密性服务。另请参见 authentication（验证）、integrity（完整性）和 privacy（保密性）。

hardening（强化） 为了删除主机中固有的安全漏洞而对操作系统的缺省配置进行的修改。

hardware provider（硬件提供器） 在 Solaris 加密框架中，是指设备驱动程序及其硬件加速器。硬件提供器使计算机系统不必执行开销很大的加密操作，从而可释放 CPU 资源以用于其他用途。另请参见 provider（提供器）。

host principal（主机主体） 服务主体的特殊实例，其中将主体（由主名称 `host` 表示）设置为提供一系列网络服务，如 `ftp`、`rcp` 或 `rlogin`。`host/central.example.com@EXAMPLE.COM` 便是一个主机主体示例。另请参见 server principal（服务器主体）。

host（主机） 可通过网络进行访问的计算机。

inheritable set（可继承集） 进程可以通过调用 `exec` 而继承的权限集。

initial ticket（初始票证） 直接颁发（即，不基于现有的票证授予票证）的票证。某些服务（如用于更改口令的应用程序）可能需要将票证标记为 *initial*，以便使其自身确信客户机知晓其私钥。这种保证非常重要，因为初始票证表明客户机最近已进行了自我验证（而并非依赖于存在时间可能较长的票证授予票证）。

instance（实例） 实例是主体名称的第二个部分，用于限定主体的主名称。对于服务主体，实例是必需的。`host/central.example.com` 便是主机全限定域名的实例。对于用户主体，实例是可选的。但是请注意，`jdoe` 和 `jdoe/admin` 都是唯一的主体。另请参见 primary（主）、principal name（主体名称）、service principal（服务主体）和 user principal（用户主体）。

integrity（完整性） 一种安全服务，除了用于用户验证之外，还用于通过加密校验和来计算验证传输数据的有效性。另请参见 authentication（验证）和 privacy（保密性）。

invalid ticket（无效票证） 尚未变为可用的以后生效的票证。应用程序服务器将拒绝无效票证，直到此票证生效为止。要使无效票证生效，必须在其开始时间已过后由客户机通过 TGS 请求将其提供给 KDC，同时设置 `VALIDATE` 标志。另请参见 postdated ticket（以后生效的票证）。

KDC 密钥分发中心。具有以下三个 Kerberos V5 组件的计算机：

- 主体和密钥数据库
- 验证服务
- 票证授予服务

每个领域都具有一个主 KDC，并且应该具有一个或多个从 KDC。

Kerberos 是指一种验证服务、此服务所使用的协议或者用于实现此服务的代码。

Solaris Kerberos 实现主要基于 Kerberos V5 实现。

虽然在技术方面有所不同，但是在 Kerberos 文档中经常会互换使用"Kerberos"和 "Kerberos V5"。

Kerberos（也可写成 Cerberus）在希腊神话中是指守护地狱之门的三头凶悍猛犬。

Kerberos policy（Kerberos 策略） 管理 Kerberos 服务中口令的使用的规则集。这些策略可以控制主体的访问权限或票证参数（如生命周期）。

keytab file（密钥表文件） 包含一个或多个密钥（主体）的密钥表文件。主机或服务使用密钥表文件的方式与用户使用口令的方式几乎相同。

key（密钥） 1. 通常是指以下两种主要密钥类型之一：

- **对称密钥**－与解密密钥相同的加密密钥。对称密钥用于对文件进行加密。
- **非对称密钥**或**公钥**－在公钥算法（如 Diffie-Hellman 或 RSA）中使用的密钥。公钥包括仅对一个用户公开的私钥、服务器或通用资源所使用的公钥，以及包含这两者的私钥/公钥对。私钥 (private key) 也称为**密**钥 (secret key)。公钥也称为**共享**密钥或**公用**密钥。

2. 密钥表文件中的项（主体名称）。另请参见 keytab file（密钥表文件）。

3. 在 Kerberos 中，是指加密密钥，此类密钥分为以下三种类型：

- **私钥**－由主体和 KDC 共享并在系统范围之外分发的加密密钥。另请参见 private key（私钥）。
- **服务密钥**－此密钥与私钥的用途相同，但是供服务器和服务使用。另请参见 service key（服务密钥）。
- **会话密钥**－在两个主体之间使用的临时加密密钥，其生命周期仅限于单个登录会话的持续期间。另请参见 session key（会话密钥）。

kvno

密钥版本号。按照生成顺序跟踪特殊密钥的序列号。最高的 kvno 表示的密钥最新。

limit set（限制集）

对可用于进程及其子进程的权限的外部限制。

MAC

1. 请参见 message authentication code, MAC（消息验证代码）。

2. 也称为标记操作。在政府安全术语中，MAC 是指强制访问控制。Top Secret 和 Confidential 之类的标签便是几个 MAC 示例。MAC 与 DAC 相对，后者是指自主访问控制。UNIX 权限便是一个 DAC 示例。

3. 在硬件中，是指 LAN 中的唯一计算机地址。如果计算机位于以太网中，则 MAC 是指以太网地址。

master KDC（主 KDC）

每个领域中的主 KDC，包括 Kerberos 管理服务器 `kadmind`，以及验证和票证授予守护进程 `krb5kdc`。每个领域至少都必须具有一个主 KDC，可以具有多个 KDC 副本或从 KDC，这些 KDC 为客户机提供验证服务。

MD5

一种重复加密散列函数，用于进行消息验证（包含数字签名）。该函数于 1991 年由 Rivest 开发。

mechanism（机制）

1. 指定加密技术以实现数据验证或保密性的软件包。例如：Kerberos V5、Diffie-Hellman 公钥。

2. 在 Solaris 加密框架中，是指用于特殊用途的算法的实现。例如，应用于验证的 DES 机制（如 CKM_DES_MAC）与应用于加密的 DES 机制（如 CKM_DES_CBC_PAD）不同。

message authentication code, MAC（消息验证代码）

MAC 可确保数据的完整性，并验证数据的来源。MAC 不能防止窃听。

message digest（消息摘要）

消息摘要是从消息中计算所得的散列值。此散列值几乎可唯一地标识消息。摘要对检验文件的完整性非常有用。

minimization（最小安装）

运行服务器所需的最小操作系统安装。与服务器操作不直接相关的任何软件或者不安装，或者在安装之后即删除。

name service scope（名称服务范围） 允许角色在其中执行操作的范围，即，由指定的名称服务（如 NIS、NIS+ 或 LDAP）提供服务的单个主机或所有主机。范围应用于 Solaris Management Console 工具箱。

network application server（网络应用程序服务器） 提供网络应用程序的服务器，如 `ftp`。一个领域可以包含若干网络应用程序服务器。

nonattributable audit event（无归属审计事件） 无法确定其触发者的审计事件，如 AUE_BOOT 事件。

NTP 网络时间协议。由特拉华大学开发的软件，可用于在网络环境中管理准确时间或网络时钟同步，或者同时管理这两者。可以使用 NTP 在 Kerberos 环境中维护时钟相位差。另请参见 clock skew（时钟相位差）。

PAM 可插拔验证模块。一种允许使用多种验证机制而不必重新编译运行这些机制的服务的框架。PAM 可用于在登录时初始化 Kerberos 会话。

passphrase（口令短语） 一种用于验证某个私钥是否是由口令短语用户创建的短语。理想的口令短语应包含 10-30 个字符，请混合使用字母和数字字符，并且避免简单的文本结构和名称。使用私钥对通信执行加密和解密操作时，系统会提示您提供口令短语以便验证。

password policy（口令策略） 可用于生成口令的加密算法，还可以指与口令有关的更普遍的问题，如必须对口令进行更改的频率，允许的错误项数以及其他安全注意事项。安全策略需要口令。口令策略可能要求使用 MD5 算法对口令进行加密，并可能对口令强度提出进一步要求。

permitted set（允许集） 可供进程使用的权限集。

policy in the cryptographic framework（加密框架中的策略） 在 Solaris 加密框架中，所实现的策略是禁用现有的加密机制。这样便不能再使用这些机制。加密框架中的策略可能会阻止使用提供器（如 DES）提供的特殊机制，如 `CKM_DES_CBC`。

policy（策略） 一般而言，是指影响或决定决策和操作的操作规划或操作过程。对于计算机系统，策略通常表示安全策略。站点的安全策略是规则集，可用于定义要处理的信息的敏感度并防止信息受到未经授权的访问。例如，安全策略可能要求对系统进行审计，使用相应权限保护设备，以及每六周更改一次口令。

有关 Solaris OS 特定区域中的策略实现的信息，请参见 audit policy（审计策略）、policy in the cryptographic framework（加密框架中的策略）、device policy（设备策略）、Kerberos policy（Kerberos 策略）、password policy（口令策略）和 RBAC policy（RBAC 策略）。

postdated ticket（以后生效的票证） 以后生效的票证直到创建之后的某一指定时间才能开始生效。此类票证对于计划在深夜运行的批处理作业等情况非常有用，因此如果票证被盗，则在运行批处理作业之前将无法使用此票证。颁发以后生效的票证时，将以 *invalid* 状态颁发该票证，并在出现以下情况之前一直保持此状态：a) 票证开始时间已过，并且 b) 使用 KDC 使客户机请求生效。通常，以后生效的票证在票证授予票证的到期时间之前会一直有效。但是，如果将以后生效的票证标记为 *renewable*，则通常会将其生命周期设置为等于票证授予票证的整个生命周期的持续时间。另请参见 invalid ticket（无效票证）和 renewable ticket（可更新票证）。

primary（主） 主体名称的第一部分。另请参见 instance（实例）、principal name（主体名称）和 realm（领域）。

principal name（主体名称） 1. 主体的名称，格式为 *primary/instance@REALM*。另请参见 instance（实例）、primary（主）和 realm（领域）。

2. (RPCSEC_GSS API) 请参见 client principal（客户机主体）和 server principal（服务器主体）。

principal（主体） 1. 参与网络通信活动并且具有唯一名称的客户机/用户或服务器/服务实例。Kerberos 事务涉及主体之间（服务主体与用户主体）或主体与 KDC 之间的交互。换句话说，主体是 Kerberos 可以为其指定票证的唯一实体。另请参见 principal name（主体名称）、service principal（服务主体）和 user principal（用户主体）。

2. (RPCSEC_GSS API) 请参见 client principal（客户机主体）和 server principal（服务器主体）。

privacy（保密性） 一种安全服务，其中传输的数据加密之后才会发送。保密性还包括数据完整性和用户验证。另请参见 authentication（验证）、integrity（完整性）和 service（服务）。

private-key encryption（私钥加密） 在私钥加密过程中，发送者和接收者使用相同的加密密钥。另请参见 public-key encryption（公钥加密）。

private key（私钥） 为每个用户主体提供的密钥，并且只对主体的用户和 KDC 公开。 对于用户主体，密钥基于用户的口令。另请参见 key（密钥）。

privilege model（权限模型） 计算机系统上比超级用户模型更为严格的安全模型。在权限模型中，进程需要具有相应的权限才能运行。系统管理可以分为多个独立的部分，这些部分基于管理员在其进程中所具有的权限。可以将权限指定给管理员的登录进程。或者，可以指定权限只对特定命令有效。

privilege set（权限集） 权限的集合。每个进程都有四个权限集，用于确定进程是否可以使用特定权限。请参见 limit set（限制集）、effective set（有效集）、permitted set（允许集）和 inheritable set（可继承集）。

此外，权限的 basic set（基本集）是指登录时为用户进程指定的权限集。

privileged application（特权应用程序） 可以覆盖系统控制的应用程序。此应用程序用于检查安全属性（如特定的 UID、GID、授权或权限）。

privilege（权限） Solaris 系统中的进程具有的独立权限。与 `root` 相比，权限可提供更为精细的进程控制。权限是在内核中定义和强制执行的。有关权限的完整说明，请参见 `privileges(5)` 手册页。

profile shell（配置文件 shell） 在 RBAC 中，角色（或用户）用户可在该 shell 中从命令行运行指定给角色权限配置文件的任何特权应用程序。配置文件 shell 包括 `pfsh`、 `pfcsh` 和 `pfksh`。它们分别对应于 Bourne shell (`sh`)、C shell (`csh`) 和 Korn shell (`ksh`)。

provider（提供器） 在 Solaris 加密框架中，是指为使用者提供的加密服务。PKCS #11 库、内核加密模块和硬件加速器便是几个提供器示例。提供器可插入到 Solaris 加密框架中，因此也称为**插件**。有关使用者的示例，请参见 consumer（使用者）。

proxiable ticket（可代理票证） 可供服务用于代表客户机执行客户机操作的票证。因此，可以说服务充当客户机的代理。使用该票证，服务便可具有客户机的身份。服务可以使用可代理票证来获取其他服务的服务票证，但是不能获取票证授予票证。可代理票证与可转发票证之间的区别在于可代理票证只对单项操作有效。另请参见 forwardable ticket（可转发票证）。

public-key encryption（公钥加密） 一种加密方案，其中每个用户都有两个密钥：一个是公钥，一个是私钥。在公钥加密过程中，发送者使用接收者的公钥对消息进行加密，而接收者则使用私钥对其进行解密。Kerberos 服务是一种私钥系统。另请参见 private-key encryption（私钥加密）。

QOP 保护质量。用于选择与完整性服务或保密性服务结合使用的加密算法的参数。

RBAC 基于角色的访问控制。全有或全无型超级用户模型的替代项。使用 RBAC，组织可以将超级用户的功能分离开来，并将各功能指定给特殊的用户帐户（称为角色）。可以根据个体的职责将角色指定给特定的个体。

RBAC policy（RBAC 策略） 与命令关联的安全策略。目前，`suser` 和 `solaris` 都是有效的策略。`solaris` 策略可识别权限和 `setuid` 安全属性。`suser` 策略仅识别 `setuid` 安全属性。可以与 Solaris 系统交互的 Trusted Solaris™ 系统提供了 `tsol` 策略，此策略可识别进程的权限、`setuid` 安全属性和标签。

realm（领域） 1. 由单个 Kerberos 数据库以及一组密钥分发中心 (Key Distribution Center, KDC) 提供服务的逻辑网络。

2. 主体名称的第三部分。对于主体名称 `jdoe/admin@ENG.EXAMPLE.COM`，领域为 `ENG.EXAMPLE.COM`。另请参见 principal name（主体名称）。

relation（关系） 在 `kdc.conf` 或 `krb5.conf` 文件中定义的配置变量或关系。

renewable ticket（可更新票证） 由于票证的生命周期过长会存在安全风险，因此可以将票证指定为 *renewable*。可更新票证具有两个到期时间：a) 票证的当前实例的到期时间，b) 任意票证的最长生命周期。如果客户机需要继续使用某票证，则可在首次到期之前更新此票证。例如，某个票证的有效期为 1 小时，所有票证的最长生命周期为 10 小时。如果持有票证的客户机希望保留此票证的时间长于 1 小时，则必须更新此票证。当某票证达到最长票证生命周期时，便会自动到期，并且无法更新。

rights profile（权限配置文件） 也称为权限或配置文件。是在可以指定给角色或用户的 RBAC 中使用的覆盖项的集合。一个权限配置文件可以包含授权、具有安全属性的命令以及其他权限配置文件。

role（角色） 一种用于运行特权应用程序的特殊身份，仅有指定用户才能使用此身份。

RSA 获取数字签名和公钥密码系统的方法。该方法于 1978 年首次由其开发者 Rivest、Shamir 和 Adleman 介绍。

SEAM Sun Enterprise 验证机制。是用于验证网络上的用户的初始系统版本的产品名称，基于麻省理工学院开发的 Kerberos V5 技术。此产品现在称为 Kerberos 服务。SEAM 指未包括在各种 Solaris 发行版中的那部分 Kerberos 服务。

secret key（私钥） 请参见 private key（私钥）。

Secure Shell（安全 Shell） 用于在不安全的网络中进行安全远程登录并提供其他安全网络服务的特殊协议。

security attributes（安全属性） 在 RBAC 中，是指当超级用户以外的用户运行管理命令时，可使此命令成功执行的安全策略覆盖项。在超级用户模型中，`setuid` 和 `setgid` 程序都是安全属性。将这些属性应用于某命令时，此命令便会成功执行，而与运行它的用户无关。在权限模型中，安全属性是指权限。为某命令提供权限后，此命令便会成功执行。权限模型与超级用户模型兼容，因为权限模型也可将 `setuid` 和 `setgid` 程序识别为安全属性。

security flavor（安全风格） 请参见 flavor（特性）。

security mechanism（安全机制） 请参见 mechanism（机制）。

security policy（安全策略） 请参见 policy（策略）。

security service（安全服务） 请参见 service（服务）。

seed（种子） 用于生成随机数的数字起动器。当起动器来自随机源时，种子称为**随机种子**。

server principal（服务器主体） (RPCSEC_GSS API) 提供服务的主体。服务器主体以 *service*@*host* 形式的 ASCII 字符串进行存储。另请参见 client principal（客户机主体）。

server（服务器） 为网络客户机提供资源的主体。例如，如果通过 `rlogin` 远程登录到计算机 `central.example.com`，则此计算机便是提供 `rlogin` 服务的服务器。另请参见 service principal（服务主体）。

service key（服务密钥） 由服务主体和 KDC 共享，并在系统范围之外分发的加密密钥。另请参见 key（密钥）。

service principal（服务主体） 为一项或多项服务提供 Kerberos 验证的主体。对于服务主体，主名称是服务的名称（如 `ftp`），其实例是提供服务的系统的全限定主机名。另请参见 host principal（主机主体）和 user principal（用户主体）。

service（服务） 1. 通常由多台服务器提供给网络客户机的资源。例如，如果通过 `rlogin` 远程登录到计算机 `central.example.com`，则此计算机便是提供 `rlogin` 服务的服务器。

2. 除验证之外，还提供其他保护级别的安全服务（完整性或保密性）。另请参见 integrity（完整性）和 privacy（保密性）。

session key（会话密钥） 由验证服务或票证授予服务生成的密钥。生成会话密钥是为了在客户机与服务之间提供安全事务。会话密钥的生命周期仅限于单个登录会话的持续期间。另请参见 key（密钥）。

SHA1 安全散列算法。该算法可以针对长度小于 2^{64} 的任何输入进行运算，以生成消息摘要。SHA1 算法是 DSA 的输入。

slave KDC（从 KDC） 主 KDC 的副本，可以执行主 KDC 的大多数功能。每个领域通常都具有若干从 KDC（但仅有一个主 KDC）。另请参见 KDC和 master KDC（主 KDC）。

software provider（软件提供器） 在 Solaris 加密框架中，是指提供加密服务的内核软件模块或 PKCS #11 库。另请参见 provider（提供器）。

stash file（存储文件） 存储文件包含 KDC 主密钥的已加密副本。当重新引导服务器以便在 KDC 启动 `kadmind` 和 `krb5kdc` 进程之前自动验证 KDC 时，会使用此主密钥。由于存储文件中包含主密钥，因此，应该保证存储文件及其任何备份的安全。如果加密受到威胁，则可以使用此密钥来访问或修改 KDC 数据库。

superuser model（超级用户模型） 计算机系统上的典型 UNIX 安全模型。在超级用户模型中，管理员对计算机具有全有或全无型控制权。通常，为了管理计算机，用户可成为超级用户 (`root`)，并可执行所有管理活动。

TGS 票证授予服务。负责颁发票证的那部分 KDC。

TGT 票证授予票证。由 KDC 颁发的票证，客户机可使用此票证来请求其他服务的票证。

ticket file（票证文件） 请参见 credential cache（凭证高速缓存）。

ticket（票证） 用于安全地将用户身份传递给服务器或服务的信息包。一个票证仅对一台客户机以及某台特定服务器上的一项特殊服务有效。票证包含服务的主体名称、用户的主体名称、用户主机的 IP 地址、时间标记以及定义此票证生命周期的值。票证是通过由客户机和服务使用的随机会话密钥创建的。一旦创建了票证，便可重复使用此票证，直到其到期为止。票证与新的验证者同时出现时，仅用于验证客户机。另请参见 authenticator（验证者）、credential（凭证）、service（服务）和 session key（会话密钥）。

user principal（用户主体） 属于特殊用户的主体。用户主体的主名称是用户名，其可选实例是用于说明相应凭证预期用法的名称（例如 `jdoe` 或 `jdoe/admin`）。也称为用户实例。另请参见 service principal（服务主体）。

virtual private network, VPN（虚拟专用网络） 通过使用加密和通道连接公共网络上的用户来提供安全通信的网络。

索引

数字和符号

. (.)
  路径变量项，44
  授权名称分隔符，216
[]（方括号），bsmrecord 输出，585
$$（双美元符号），父 shell 进程号，228
>（重定向输出），防止，44
>>（附加输出），防止，44
~/.gkadmin 文件，说明，501
~/.k5login 文件，说明，501
~/.rhosts 文件，说明，340
~/.shosts 文件，说明，340
~/.ssh/authorized_keys 文件
  覆盖，341
  说明，339
~/.ssh/config 文件
  覆盖，340
  说明，340
~/.ssh/environment 文件，说明，340
~/.ssh/id_dsa 文件，覆盖，341
~/.ssh/id_rsa 文件，覆盖，341
~/.ssh/identity 文件，覆盖，341
~/.ssh/known_hosts 文件
  覆盖，341
  说明，340
~/.ssh/rc 文件，说明，340
@（at 符号），device_allocate 文件，90
=（等号），文件权限符号，123
.（点），显示隐藏文件，128
\（反斜杠）
  device_allocate 文件，90
  device_maps 文件，89
;（分号）
  device_allocate 文件，90
;（分号）（续）
  安全属性的分隔符，222
+（加号）
  ACL 项，134
  sulog 文件，66
  审计类前缀，581
  文件权限符号，123
-（减号）
  sulog 文件，66
  审计类前缀，581
  文件类型符号，120
  文件权限符号，123
#（井号）
  device_allocate 文件，90
  device_maps 文件，89
?（问号），ASET 调优文件，158
*（星号）
  device_allocate 文件，90
  通配符
    在 ASET 中，156, 158
    在 RBAC 授权中，216, 219
  在 RBAC 授权中检查，210
3des-cbc 加密算法，ssh_config 文件，334
3des 加密算法，ssh_config 文件，334

A

-a 选项
  bsmrecord 命令，558
  digest 命令，260
  encrypt 命令，264
  getfacl 命令，138
  mac 命令，262

-a 选项（续）
smrole 命令，192-194
基于 Kerberos 的命令，495
-A 选项, auditreduce 命令，562
ACL
kadm5.acl 文件，457, 459, 462
对项进行检查，134
复制 ACL 项，137
复制项的限制，125
更改项，137-138
命令，126
目录的缺省项，126
目录项，126
任务列表，134-140
删除项，126, 138
设置文件，135
设置项，135-136
说明，47, 125-126
显示项，126, 138-140
项的格式，125-126
修改项，137
用户过程，134-140
有效的文件项，125-126
acl 审计标记, 格式，587
add_drv 命令, 说明，87
admin_server 部分, krb5.conf 文件，366
administrative (old) 审计类，580
administrative 审计类，580
AES 内核提供器，267
aes128-cbc 加密算法, ssh_config 文件，334
aes128-ctr 加密算法, ssh_config 文件，334
ahlt 审计策略
设置，551
说明，531
all, 在用户审计字段中，577
all 审计类
使用注意事项，582
说明，580
allhard 字符串, audit_warn 脚本，578
allocate 命令
磁带机，83
分配错误状态，89
使用，82-83
授权，89
说明，88
所需授权，225
allocate 命令（续）
用户授权，77
AllowGroups 关键字, sshd_config 文件，334
AllowTcpForwarding 关键字
sshd_config 文件，334
更改，319
AllowUsers 关键字, sshd_config 文件，334
allsoft 字符串, audit_warn 脚本，578
application 审计类，580
arbitrary 审计标记
格式，587
列显格式字段，587
项大小字段，587
arcfour 加密算法, ssh_config 文件，334
ARCFOUR 内核提供器，267
arg 审计标记, 格式，588
arge 审计策略
和 exec_env 标记，589
说明，531
argv 审计策略
和 exec_args 标记，589
说明，531
ASET
aset.restore 命令，154
aset 命令
-p 选项，162
交互式版本，160-161
启动，144
ASETDIR 变量，155
asetenv 文件，151, 152
ASETSECLEVEL 变量，156
CKLISTPATH_level 变量，157
NFS 服务和，154
PERIODIC_SCHEDULE 变量，153, 156
TASKS 变量，152, 157
UID_ALIASES 变量，151, 153, 157
uid_aliases 文件，151
YPCHECK 变量，153, 157
安排 ASET 执行，153, 156
别名文件
UID_ALIASES 变量，153
示例，159
说明，151
错误消息，164
调优文件，151, 153
调优文件示例，158

ASET（续）
定期运行，161-162
定期运行 ASET，161-162
工作目录，155
环境变量，155
环境文件，151
恢复初始系统状态，154
交互运行，160-161
配置，152-154, 154
任务列表，159-164
收集报告，162-164
说明，45, 143-159
停止定期运行，162
疑难解答，164
执行日志，147
主文件，145, 151
at 符号 (@), device_allocate 文件，90
at 命令, 所需授权，224
atq 命令, 所需授权，224
attribute 审计标记，588
audit administration 审计类，581
audit_class 文件, description 文件，575
audit_class 文件, 添加类，543-544
audit_control 文件
audit_user 数据库中的例外，577
flags 行
进程预选掩码，582
flags 行中的前缀，582
minfree 警告，578
编辑后的审计守护进程重新读取，556
概述，518
更改无归属事件的内核掩码，556
配置，536-539
示例，576
说明，575
项，575
项和区域，579
语法问题，578
audit_event 文件
更改类成员关系，544-546
说明，520
audit.notice 项, syslog.conf 文件，540
audit_startup 脚本
配置，550-553
说明，576
audit_user 数据库
进程预选掩码，582
类的前缀，582
系统范围审计类的例外，521
用户审计字段，577
指定用户例外情况，541-543
audit_warn 脚本
auditd 守护进程执行，570
配置，550
情况调用，578
说明，577
字符串，578
audit 命令
复位目录指针（-n 选项），570
更新审计服务，556-557
说明，570
现有进程的预选掩码（-s 选项），556
重新读取审计文件（-s 选项），570
auditconfig 命令
类的前缀，582
设置审计策略，552
说明，574
作为参数的审计类，521, 580
auditd 守护进程
audit_warn 脚本
说明，577, 578
执行，570
功能，569-570
审计跟踪创建，569-570, 583
审计文件的打开顺序，575
重新读取 audit_control 文件，556
重新读取内核的信息，556
auditlog 文件, 文本审计记录，540
auditreduce 命令，571
-c 选项，565
-O 选项，561-563
不带选项，571
过滤选项，563
合并审计记录，561-563
清除审计文件，567-568
时间标记的使用，584
使用大写选项，561
使用小写选项，563
示例，561-563
说明，571
尾部标记, 和，600

auditreduce 命令（续）
选项，571
选择审计记录，563-565
auth_attr 数据库
说明，219-220
摘要，216
AUTH_DES 验证，请参见AUTH_DH 验证
AUTH_DH 验证，和 NFS，285
authlog 文件，保存失败的登录尝试，58-59
authorized_keys 文件，说明，339
AuthorizedKeysFile 关键字，sshd_config 文件，334
AUTHS_GRANTED 关键字，policy.conf 文件，222
auths 命令，说明，223
auto_transition 选项，SASL 和，308
auxprop_login 选项，SASL 和，308

B

-b 选项，auditreduce 命令，563
Banner 关键字，sshd_config 文件，334
BART
安全注意事项，96
程序输出，117
概述，93-95
任务列表，95-96
详细输出，116
组件，94
bart compare 命令，94
bart create 命令，94，96
bart 命令，93
BART 中的引用语法，115
Batchmode 关键字，ssh_config 文件，334
BindAddress 关键字，ssh_config 文件，334
binding 控制标志，PAM，303
blowfish-cbc 加密算法，ssh_config 文件，334
Blowfish 加密算法
policy.conf 文件，63
ssh_config 文件，334
内核提供器，267
用于口令，63
Bourne shell，特权版本，176
bsmconv 脚本
创建 device_maps 文件，89-90
启用审计服务，553-554
说明，578
bsmrecord 命令
可选标记 ([])，585
列出程序的格式，558-560
列出类的格式，560-561
列出所有格式，558
示例，558
输出中的 []（方括号），585
说明，570
显示审计记录格式，558-561
bsmunconv 脚本，禁用审计服务，555-556

C

-c 选项
auditreduce 命令，564，565
bsmrecord 命令，560-561
-C 选项，auditreduce 命令，562
C shell，特权版本，176
c2audit:audit_load 项，system 文件，574
canon_user_plugin 选项，SASL 和，308
CD-ROM 驱动器
安全性，91-92
分配，84-85
cdrw 命令，所需授权，224
ChallengeResponseAuthentication 关键字，请参见KbdInteractiveAuthentication 关键字
changepw 主体，476
CheckHostIP 关键字，ssh_config 文件，334
chgrp 命令
说明，120
语法，130
chkey 命令，286，293
chmod 命令
更改特殊文件权限，133-134，134
说明，120
语法，133
chown 命令，说明，119
Cipher 关键字，sshd_config 文件，334
Ciphers 关键字，Solaris 安全 Shell，334
cklist.rpt 文件，145，149
CKLISTPATH_level 变量 (ASET)，157
clear 保护级别，496
ClearAllForwardings 关键字，Solaris 安全 Shell 端口转发，334

ClientAliveCountMax 关键字, Solaris 安全 Shell 端口转发，334
ClientAliveInterval 关键字, Solaris 安全 Shell 端口转发，334
clntconfig 主体, 创建，371
cmd 审计标记，524, 589
cnt 审计策略, 说明，532
Compression 关键字, Solaris 安全 Shell，334
CompressionLevel 关键字, ssh_config 文件，334
ConnectionAttempts 关键字, ssh_config 文件，334
crammd5.so.1 插件, SASL 和，308
cred 表
  DH 验证和，286
  由服务器存储的信息，288
cred 数据库
  DH 验证，286-288
  添加客户机凭证，290
  添加用户凭证，291
crontab 文件
  定期运行 ASET，144
  所需授权，225
  停止定期运行 ASET，162
CRYPT_ALGORITHMS_ALLOW 关键字, policy.conf 文件，39
CRYPT_ALGORITHMS_DEPRECATE 关键字, policy.conf 文件，39
crypt_bsdbf 口令算法，38
crypt_bsdmd5 口令算法，38
crypt.conf 文件
  第三方口令模块，64-65
  使用新口令模块更改，64-65
CRYPT_DEFAULT 关键字, policy.conf 文件，39
CRYPT_DEFAULT 系统变量，62
crypt_sunmd5 口令算法，38
crypt_unix 口令算法，38, 62-65
crypt 命令, 文件安全性，47
cryptoadm install 命令, 安装 PKCS #11 库，272
cryptoadm 命令
  -m 选项，272, 275
  -p 选项，273, 275
  安装 PKCS #11 库，272
  恢复内核软件提供器，275
  禁用加密机制，272, 275
  禁用硬件机制，279-281
  列出提供器，267
  说明，253
Cryptoki, **请参见**PKCS #11 库
csh 命令, 特权版本，176
.cshrc 文件, 路径变量项，44

D

-D 选项
  auditreduce 命令，562
  ppriv 命令，230
d_passwd 文件
  创建，60
  临时禁用拨号登录，61
  说明，41
-d 选项
  auditreduce 命令，564
  getfacl 命令，139
  praudit 命令，572
  setfacl 命令，138
deallocate 命令
  分配错误状态，89
  设备清理脚本和，92
  使用，85-86
  授权，89
  说明，88
  所需授权，225
decrypt 命令
  说明，254
  语法，264
default/login 文件, 说明，340
default_realm 部分, krb5.conf 文件，366
defaultpriv 关键字, user_attr 数据库，246
delete_entry 命令, ktutil 命令，481
DenyGroups 关键字, sshd_config 文件，334
DenyUsers 关键字, sshd_config 文件，334
DES 加密
  安全 NFS，286
  内核提供器，267
/dev/arp 设备, 获取 IP MIB-II 信息，75
/dev/urandom 设备，258-260
devfsadm 命令, 说明，87
device_allocate 文件
  格式，90
  说明，90-91
  样例，79, 90

device_maps 文件
格式，89
说明，89
项样例，89
dfstab 文件
安全模式，387
共享文件，47
DH 验证
共享文件，294-295
挂载文件，294
说明，286-288
为 NIS+ 客户机，290
为 NIS 客户机，292-293
在 NIS+ 中配置，290-291
在 NIS 中配置，292-293
DHCP 管理 (RBAC), 创建角色，190
dialups 文件, 创建，60
Diffie-Hellman 验证, **请参见**DH 验证
digest 命令
示例，261
说明，254
语法，260
digestmd5.so.1 插件, SASL 和，308
dir 行, audit_control 文件，575
dminfo 命令，89
DNS, Kerberos 和，359
domain_realm 部分
krb5.conf 文件，359, 366
DSAAuthentication 关键字, **请参见**PubkeyAuthentication 关键字
.dtprofile 脚本, 在 Solaris 安全 Shell 中使用，326
DynamicForward 关键字, ssh_config 文件，334

E

-e 选项
auditreduce 命令，564
ppriv 命令，230
ebusy 字符串, audit_warn 脚本，578
eeprom.rpt 文件，146, 149
eeprom 命令，36, 68-70
eject 命令, 设备清除和，92
elfsign 命令
说明，254
encrypt 命令
错误消息，265
说明，254
疑难解答，265
语法，258
env.rpt 文件，146, 149
EscapeChar 关键字, ssh_config 文件，334
/etc/d_passwd 文件
临时禁用拨号登录，61
创建，60
和 /etc/passwd 文件，41
/etc/default/kbd 文件，69-70
/etc/default/login 文件
Solaris 安全 Shell 和，337-338
登录缺省设置，58
说明，340
限制远程 root 访问，67-68
/etc/default/su 文件
监视 su 命令，66
监视访问尝试，67-68
显示 su 命令尝试，67-68
/etc/dfs/dfstab 文件
安全模式，387
共享文件，47
/etc/dialups 文件, 创建，60
/etc/group 文件, ASET 检查，145
/etc/hosts.equiv 文件, 说明，340
/etc/krb5/kadm5.acl 文件, 说明，501
/etc/krb5/kadm5.keytab 文件, 说明，501
/etc/krb5/kdc.conf 文件, 说明，501
/etc/krb5/kpropd.acl 文件, 说明，502
/etc/krb5/krb5.conf 文件, 说明，502
/etc/krb5/krb5.keytab 文件, 说明，502
/etc/krb5/warn.conf 文件, 说明，502
/etc/logindevperm 文件，40
/etc/nologin 文件
临时禁止用户登录，56-57
说明，340
/etc/nsswitch.conf 文件，37
/etc/pam.conf 文件
Kerberos 和，502
服务名称，302
控制标志，303
示例，304
说明，302
/etc/passwd 文件, ASET 检查，145

/etc/publickey 文件, DH 验证和，286
/etc/security/audit_event 文件, 审计事件和，520
/etc/security/audit_startup 文件，576
/etc/security/audit_warn 脚本，577
/etc/security/bsmconv 脚本，89-90
  说明，578
/etc/security/crypt.conf 文件
  第三方口令模块，64-65
  使用新口令模块更改，64-65
/etc/security/device_allocate 文件，90
/etc/security/device_maps 文件，89
/etc/security/policy.conf 文件, 算法配置，62-63
/etc/ssh_host_dsa_key.pub 文件, 说明，339
/etc/ssh_host_key.pub 文件, 说明，339
/etc/ssh_host_rsa_key.pub 文件, 说明，339
/etc/ssh/shosts.equiv 文件, 说明，340
/etc/ssh/ssh_config 文件
  覆盖，340
  关键字，334-338
  配置 Solaris 安全 Shell，333
  说明，340
  主机特定参数，337
/etc/ssh/ssh_host_dsa_key 文件, 说明，339
/etc/ssh/ssh_host_key 文件
  覆盖，341
  说明，339
/etc/ssh/ssh_host_rsa_key 文件, 说明，339
/etc/ssh/ssh_known_hosts 文件
  安全分发，338
  覆盖，341
  控制分发，338
  说明，339
/etc/ssh/sshd_config 文件
  关键字，334-338
  说明，339
/etc/ssh/sshrc 文件, 说明，340
/etc/syslog.conf 文件
  PAM 和，301
  可执行栈消息和，127
  审计和，540, 575
  失败的登录和，58-59
/etc/system 文件，574
exec_args 审计标记
  argv 策略和，589
  格式，589
exec_attr 数据库
  说明，221-222
  摘要，216
exec_env 审计标记, 格式，589
exec 审计类，581
exit 审计标记, 格式，590
EXTERNAL 安全机制插件, SASL 和，308

F

-f 选项
  setfacl 命令，137
  st_clean 脚本，92
  基于 Kerberos 的命令，495, 497-498
-F 选项
  deallocate 命令，89
  基于 Kerberos 的命令，496, 497-498
FallBackToRsh 关键字, ssh_config 文件，335
fd_clean 脚本, 说明，91
file_attr_acc 审计类，580
file_attr_mod 审计类，580
file_close 审计类，580
file_creation 审计类，580
file_deletion 审计类，580
file_read audit 类，580
file_write 审计类，580
FILE 权限，178
file 审计标记, 格式，590
find 命令, 使用 setuid 权限查找文件，140
firewall.rpt 文件，146, 149
flags 行
  audit_control 文件，575
  plugin 行和，539
  进程预选掩码，582
ForwardAgent 关键字, Solaris 安全 Shell 转发验证，335
ForwardX11 关键字, Solaris 安全 Shell 端口转发，335
FQDN（Fully Qualified Domain Name，全限定域名），
  在 Kerberos 中，359
ftp 命令
  Kerberos 和，495-496
  设置保护级别，496
  说明，502-503
ftpd 守护进程, Kerberos 和，503-504

G

GatewayPorts 关键字, Solaris 安全 Shell，335
getdevpolicy 命令, 说明，87
getfacl 命令
  -a 选项，138
  -d 选项，139
  检验 ACL 项，135
  示例，138-140
  说明，126
  显示 ACL 项，138-140
gkadmin 命令
  另请参见SEAM Administration Tool
  说明，503
.gkadmin 文件
  SEAM Administration Tool 和，447
  说明，501
GlobalKnownHostsFile 关键字, ssh_config 文件，335
GlobalKnownHostsFile2 关键字, **请参见**GlobalKnownHostsFile 关键字
group 审计标记, 由 groups 标记替换，591
group 审计策略
  和 groups 标记，532, 591
  说明，532
groups 审计标记，591
GSS-API
  Kerberos 和，346, 356
  Solaris 安全 Shell 中的凭证，332
  Solaris 安全 Shell 中的验证，312
  安全 RPC 中的凭证，290-291
gssapi.so.1 插件, SASL 和，308
GSSAPIAuthentication 关键字, Solaris 安全 Shell，335
GSSAPIDelegateCredentials 关键字, Solaris 安全 Shell，335
GSSAPIKeyExchange 关键字, Solaris 安全 Shell，335
GSSAPIStoreDelegatedCredentials 关键字, ssh_config 文件，335
gsscred 表, 使用，513
gsscred 命令, 说明，502-503
gssd 守护进程, Kerberos 和，503-504

H

-h 选项, bsmrecord 命令，558
hard 字符串, audit_warn 脚本，578
header 审计标记
  格式，591
  审计记录中的顺序，591
  事件修饰符字段标志，591
hmac-md5 算法, ssh_config 文件，336
hmac-sha1 加密算法, ssh_config 文件，336
Host 关键字
  ssh_config 文件，335, 337
host 主体, 创建，371
HostbasedAuthentication 关键字, Solaris 安全 Shell，335
HostbasedUsesNamesFromPacketOnly 关键字, sshd_config 文件，335
HostKey 关键字, sshd_config 文件，335
HostKeyAlgorithms 关键字, ssh_config 文件，335
HostKeyAlias 关键字, ssh_config 文件，335
hosts.equiv 文件, 说明，340

I

-i 选项
  bart create 命令，97, 104
  encrypt 命令，264
  st_clean 脚本，92
-I 选项
  bart create 命令，97
  st_clean 脚本，92
ID
  将 UNIX 映射到 Kerberos 主体，513
  审计
    概述，517-518
    机制，583
  审计会话，583
IdentityFile 关键字, ssh_config 文件，335
IgnoreRhosts 关键字, sshd_config 文件，335
IgnoreUserKnownHosts 关键字, sshd_config 文件，335
in_addr 审计标记, 格式，592
install 子命令, cryptoadm 命令，272
INTERNAL 插件, SASL 和，308
Internet 防火墙设置，50
ioctl 审计类，581
ioctl() 系统调用，581
  AUDIO_SETINFO()，92
IP MIB-II, 从 /dev/arp 获取信息，75
IP 地址, Solaris 安全 Shell 检查，334

ip 审计标记, 格式，592
ipc_perm 审计标记, 格式，593
ipc 类型字段值（ipc 标记），592
IPC 权限，178
ipc 审计标记，592
  格式，592
ipc 审计类，581
iport 审计标记, 格式，593

J

JASS 工具包, 链接，45

K

-k 选项
  encrypt 命令，264
  mac 命令，262
  基于 Kerberos 的命令，496
-K 选项
  usermod 命令，233
  基于 Kerberos 的命令，496
.k5.*REALM* 文件, 说明，502
.k5login 文件
  而不显示口令，493
  说明，493-494, 501
kadm5.acl 文件
  说明，501
  项的格式，462
  新主体和，457, 459
  主 KDC 项，367, 409
kadm5.keytab 文件
  说明，476, 501
kadmin.local 命令
  创建密钥表文件，368
  说明，503
  添加管理主体，368
  自动创建主体，451
kadmin.log 文件, 说明，502
kadmin 命令
  ktadd 命令，477-479
  ktremove 命令，479
  SEAM Administration Tool 和，445
  创建 host 主体，371
kadmin 命令（续）
  从密钥表中删除主体，479-480
  说明，503
kadmind 守护进程
  Kerberos 和，503
  主 KDC 和，504
kadmind 主体，476
kbd 文件，69-70
KbdInteractiveAuthentication 关键字, Solaris 安全 Shell，335
kcfd 守护进程，281-282
kclient 命令, 说明，503
kdb5_util 命令
  创建 KDC 数据库，367
  创建存储文件，376, 423
  说明，503
KDC
  备份和传播，412-413
  创建 host 主体，371
  创建数据库，367
  从，360
    定义，504
  从或主，350-351, 364
  端口，359
  规划，360
  将管理文件从从 KDC 复制到主 KDC，374, 421
  交换主和从，405-412
  配置从，372-377
  配置服务器，364-377
  配置主，365-372
  启动守护进程，377, 423
  数据库传播，361
  同步时钟，372, 377, 423
  限制对服务器的访问，428-429
  主
    定义，504
kdc.conf 文件
  票证生命周期和，506
  说明，501
kdc.log 文件, 说明，502
kdestroy 命令
  示例，489
  说明，502-503
KeepAlive 关键字, Solaris 安全 Shell，335
Kerberos
  dfstab 文件选项，387

Kerberos（续）
Kerberos V5 协议，345
参考，501-513
错误消息，431-441
访问服务器，509-512
概述
基于 Kerberos 的命令，495-496
验证系统，346-351, 508
管理，445-483
管理工具
请参见SEAM Administration Tool
规划，357-362
基于 Kerberos 的命令的选项，495
仅启用基于 Kerberos 的应用程序，428
口令管理，489-494
联机帮助，362
领域
请参见领域 (Kerberos)
命令，494-500, 502-503
配置 KDC 服务器，364-377
配置决策，357-362
使用，485-500
使用基于 Kerberos 的命令的示例，498-500
守护进程，503-504
授予对帐户的访问权限，493-494
网络命令选项表，496
文件，501-502
疑难解答，441
远程应用程序，349
术语，504-508
组件，352-353
Kerberos 命令，494-500
仅启用基于 Kerberos 的，428
示例，498-500
Kerberos 验证
`dfstab` 文件选项，387
和安全 RPC，286
`kern.notice` 项, `syslog.conf` 文件，127
`KEYBOARD_ABORT` 系统变量，69-70
`keylogin` 命令
检验 DH 验证设置，290
使用，287
`KeyRegenerationInterval` 关键字, `sshd_config` 文件，335
`keyserv` 守护进程，289
keyserver
启动，289
说明，287
`keytab` 选项, SASL 和，309
`kinit` 命令
`-F` 选项，486
票证生命周期，506
示例，486
说明，502-503
`klist` 命令
`-f` 选项，487-489
示例，487-489
说明，502-503
`known_hosts` 文件
控制分发，338
说明，340
Korn shell, 特权版本，176
`kpasswd` 命令
`passwd` 命令和，490
错误消息，490
示例，492
说明，502-503
`kprop` 命令, 说明，503
`kpropd.acl` 文件, 说明，502
`kpropd` 守护进程, Kerberos 和，503
`kproplog` 命令, 说明，503
`krb5.conf` 文件
`domain_realm` 部分，359
编辑，365
端口定义，359
说明，502
`krb5.keytab` 文件, 说明，502
`krb5cc_`*uid* 文件, 说明，502
`krb5kdc` 守护进程
Kerberos 和，503
启动，377, 423
主 KDC 和，504
`ksh` 命令, 特权版本，176
`ktadd` 命令
添加服务主体，476, 477-479
语法，477
`ktkt_warnd` 守护进程
Kerberos 和，503-504
`ktremove` 命令，479
`ktutil` 命令
`delete_entry` 命令，481

ktutil 命令（续）
list 命令，480, 481
read_kt 命令，480, 481
查看主体列表，479, 480-481
管理密钥表文件，476
说明，502-503

L

-l 选项
digest 命令，260
encrypt 命令，258
mac 命令，262
praudit 命令，572
-L 选项, ssh 命令，327-328
LDAP 名称服务
口令，37
指定口令算法，64
limitpriv 关键字, user_attr 数据库，246
list_devices 命令
授权，89
说明，88
所需授权，225
list 命令，480, 481
ListenAddress 关键字, sshd_config 文件，335
LocalForward 关键字, ssh_config 文件，335
log_level 选项, SASL 和，309
logadm 命令, 归档文本审计文件，568
.login 文件, 路径变量项，44
login_logout 审计类，581
login 环境变量, Solaris 安全 Shell 和，337-338
login 文件
登录缺省设置，58
限制远程 root 访问，67-68
LoginGraceTime 关键字, sshd_config 文件，336
loginlog 文件, 保存失败的登录尝试，57-58
logins 命令
显示没有口令的用户，56
显示用户的登录状态，54-55, 55
语法，54
LogLevel 关键字, Solaris 安全 Shell，336
LookupClientHostname 关键字, sshd_config 文件，336

M

-m 选项
cryptoadm 命令，272, 275
基于 Kerberos 的命令，496
-M 选项, auditreduce 命令，562
mac 命令
说明，254
语法，262
MACS 关键字, Solaris 安全 Shell，336
makedbm 命令, 说明，223
max_life 值, 说明，506
max_renewable_life 值, 说明，507
MaxAuthTries 关键字, sshd_config 文件，336
MaxAuthTriesLog 关键字, sshd_config 文件，336
MaxStartups 关键字, sshd_config 文件，336
MD5 加密算法
policy.conf 文件，62-63
内核提供器，267
mech_dh 机制
GSS-API 凭证，332
安全 RPC，290-291
mech_krb 机制, GSS-API 凭证，332
mech_list 选项, SASL 和，308
messages 文件, 可执行栈消息，127
minfree 行
audit_control 文件，575
audit_warn 情况，578
mount 命令, 带有安全属性，77
mt 命令, 磁带设备清除和，91

N

-n 选项
audit 命令，570
bart create 命令，97
naflags 行
audit_control 文件，575
plugin 行和，539
NET 权限，178
network 审计类，581
newkey 命令
生成密钥，286
为 NIS 用户创建密钥，293-294
NFS 服务器, 配置 Kerberos，382-384

NFS 文件系统
  ASET 和，154
  提供客户机-服务器安全性，286-288
  通过 AUTH_DH 进行安全访问，294
  验证，285
NIS+ 名称服务
  ASET 检查，153
  cred 表，286
  cred 数据库，291
  口令，37
  添加经过验证的用户，291-292
  验证，285
  指定口令算法，63-64
NIS 名称服务
  口令，37
  验证，285
  指定口令算法，63
nisaddcred 命令
  生成密钥，286
  添加客户机凭证，290
no_class 审计类，580
nobody 用户，47
noexec_user_stack_log 变量，127, 142
noexec_user_stack 变量，127, 142
NoHostAuthenticationForLocalHost 关键字，
  ssh_config 文件，336
nologin 文件, 说明，340
non_attrib 审计类，580
nscd（名称服务高速缓存守护进程）
  使用，223
  使用 svcadm 命令启动，189
nsswitch.conf 文件, 登录访问限制，36
NTP
  Kerberos 和，361
  从 KDC 和，377, 423
  主 KDC 和，372
null 审计类，580
NumberOfPasswordPrompts 关键字, ssh_config 文
  件，336

O

-o 选项, encrypt 命令，264
-O 选项, auditreduce 命令，561-563
od 命令, 生成密钥，258-260
opaque 审计标记, 格式，594
OpenSSH, **请参见**Solaris 安全 Shell
optional 控制标志, PAM，304
other 审计类，581
ovsec_adm.*xxxxx* 文件, 说明，502

P

-p 选项
  aset 命令，162
  bart create，104
  bsmrecord 命令，558-560
  cryptoadm 命令，273, 275
  logins 命令，55
PAM
  /etc/syslog.conf 文件，301
  Kerberos 和，353, 355
  堆叠，298
  服务名称，302
  概述，297
  规划，300
  控制标志，303
  模块，304
  模块类型，302
  配置文件，302, 304
  任务列表，299
  添加模块，300-301
pam_*.so.1 文件, 说明，304
pam.conf 文件
  Kerberos 和，502
  服务名称，302
  控制标志，303
  示例，304
  说明，302
pam_roles 命令, 说明，223
PAMAuthenticationViaKBDInt 关键字, sshd_config 文
  件，336
passwd 命令
  和 kpasswd 命令，490
  和名称服务，37
passwd 文件
  ASET 检查，145
  和 /etc/d_passwd 文件，41
PasswordAuthentication 关键字, Solaris 安全
  Shell，336

path_attr 审计标记，524, 594
PATH 环境变量
  和安全性，44
  设置，44
path 审计标记, 格式，594
path 审计策略, 说明，532
PERIODIC_SCHEDULE 变量 (ASET)，153, 156
PermitEmptyPasswords 关键字, sshd_config 文件，336
PermitRootLogin 关键字, sshd_config 文件，336
PermitUserEnvironment 关键字, sshd_config 文件，336
perzone 审计策略
  设置，552
  使用，579
  说明，532
pfcsh 命令, 说明，176
pfexec 命令, 说明，224
pfksh 命令, 说明，176
pfsh 命令, 说明，176
PKCS #11 库
  Solaris 加密框架中，252
  添加为提供器，272
pkcs11_kernel.so 用户级提供器，267
pkcs11_softtoken.so 用户级提供器，267
pkgadd 命令
  安装第三方软件，64
  安装第三方提供器，271
plain.so.1 插件, SASL 和，308
plugin_list 选项, SASL 和，309
plugin 行
  audit_control 文件，575
  flags 行和，539
policy.conf 文件
  关键字
    用于 RBAC 授权，222
    用于口令算法，39
    用于权限，222, 246
    用于权限配置文件，222
  基本 Solaris 用户权限配置文件，214
  说明，222-223, 224
  添加口令加密模块，64-65
  指定加密算法，62-63
  指定口令算法，62-63
    在名称服务中，63
Port 关键字, Solaris 安全 Shell，336
postsigterm 字符串, audit_warn 脚本，578
ppriv 命令
  列出权限，228
  用于调试，230
praudit 命令
  XML 格式，567
  不带选项，572
  查看审计记录，565-567
  将 auditreduce 输出传输到，566
  将审计记录转换为可读格式，566, 572
  输出格式，572
  选项，572
  用于 -x 选项的 DTD，573
  在脚本中使用，573
PreferredAuthentications 关键字, ssh_config 文件，336
principal.kadm5.lock 文件, 说明，502
principal.kadm5 文件, 说明，502
principal.ok 文件, 说明，502
principal.ulog 文件, 说明，502
principal 文件, 说明，502
PrintMotd 关键字, sshd_config 文件，336
priv.debug 项, syslog.conf 文件，246
PRIV_DEFAULT 关键字
  policy.conf 文件，222, 246
PRIV_LIMIT 关键字
  policy.conf 文件，222, 246
private 保护级别，496
privilege 审计标记，525, 595
privileges 文件, 说明，179
PROC 权限，178
process modify 审计类，581
process start 审计类，581
process 审计标记, 格式，595
process 审计类，581
prof_attr 数据库
  说明，220-221
  摘要，216
.profile 文件, 路径变量项，44
profiles 命令, 说明，224
PROFS_GRANTED 关键字, policy.conf 文件，222
PROM 安全模式，68-70
Protocol 关键字, ssh_config 文件，336
ProxyCommand 关键字, ssh_config 文件，336
PubkeyAuthentication 关键字, Solaris 安全 Shell，336
public 审计策略
  说明，532

public 审计策略（续）
只读事件，532
publickey 映射, DH 验证，286-288
pwcheck_method 选项, SASL 和，308

R

-r 选项
bart create，104
passwd 命令，37
praudit 命令，572
-R 选项
bart create，97, 104
ssh 命令，327-328
RBAC
编辑权限配置文件，205-207
更改用户属性
通过命令行，209
管理命令，223-224
规划，187-188
基本概念，171-173
检查脚本或程序中的授权，210
名称服务和，218
配置，186-198
配置文件 shell，176
权限配置文件，175
权限配置文件数据库，220-221
确保脚本安全，210
审计角色，195-196
审计配置文件，579
使用特权应用程序，201-202
授权，174
授权数据库，219-220
数据库，216-223
数据库关系，217-218
添加角色，188-190
添加新的权限配置文件，206
添加自定义角色，192-194
通过命令行添加角色，191-194
修改角色，203-205
修改用户，207-209
用于管理的命令，223-224
与超级用户模型比较，169-171
元素，171-173
RC4, **请参见**ARCFOUR 内核提供器
rcp 命令
Kerberos 和，495-496
说明，502-503
read_kt 命令，480, 481
reauth_timeout 选项, SASL 和，308
rem_drv 命令, 说明，87
RemoteForward 关键字, ssh_config 文件，336
required 控制标志, PAM，303
requisite 控制标志, PAM，303
return 审计标记, 格式，597
rewoffl 选项
mt 命令
磁带设备清除和，91
.rhosts 文件, 说明，340
RhostsAuthentication 关键字, Solaris 安全 Shell，336
RhostsRSAAuthentication 关键字, Solaris 安全 Shell，336
rlogin 命令
Kerberos 和，495-496
说明，502-503
rlogind 守护进程, Kerberos 和，503-504
roleadd 命令
使用，191
说明，224
roledel 命令, 说明，224
rolemod 命令
更改角色的属性，203
说明，224
roles 命令
使用，199
说明，224
root 角色 (RBAC), 承担角色，200
root 用户
登录帐户
说明，40
跟踪登录，43
更改为 root 角色，196-198
监视 su 命令尝试，43, 66
限制访问，47
限制远程访问，67-68
在 RBAC 中替换，176
在控制台上显示访问尝试，67-68
root 主体, 添加至主机的 keytab，476
RPCSEC_GSS API, Kerberos 和，356
RSA 内核提供器，267
RSAAuthentication 关键字, Solaris 安全 Shell，336

rsh 命令
  Kerberos 和，495-496
  说明，502-503
rsh 命令（受限 shell），44
rshd 守护进程, Kerberos 和，503-504
rstchown 系统变量，130
rules 文件 (BART)，95
rules 文件格式 (BART)，114-115
rules 文件规范语言, **请参见**引用语法
rules 文件属性, **请参见**关键字

S

-s 选项
  audit 命令，570
  praudit 命令，572
-S 选项, st_clean 脚本，92
safe 保护级别，496
SASL
  插件，308
  概述，307
  环境变量，308
  选项，308-309
saslauthd_path 选项, SASL 和，309
scp 命令
  复制文件，328-329
  说明，341
SCSI（小型计算机系统接口）设备, st_clean 脚本，91
SEAM Administration Tool
  gkadmin 命令，445
  .gkadmin 文件，447
  kadmin 命令，445
  帮助，447
  帮助内容，447
  被修改的文件，447
  查看策略列表，464-466
  查看策略属性，466-468
  查看主体列表，452-454
  查看主体属性，454-456
  创建新策略，456, 468-469
  创建新主体，456-459
  登录窗口，448
  等效命令行，446-447
  复制主体，459
  概述，446-450
SEAM Administration Tool（**续**）
  关联说明，447
  过滤器模式字段，453
  和 X 窗口系统，446-447
  和列表权限，475
  和有限的管理权限，474-475
  或 kadmin 命令，446
  联机帮助，447
  面板表，472-474
  面板说明，472-474
  启动，448-450
  权限，475
  缺省值，449
  如何受权限影响，475
  删除策略，471-472
  删除主体，460-461
  设置主体缺省值，461-462
  显示主体子列表，453
  修改策略，470-471
  修改主体，459-460
SEAM Administration Tool 的等效命令行，446-447
sendmail 命令, 所需授权，225
seq 审计策略
  和 sequence 标记，532, 597
  说明，532
sequence 审计标记
  格式，597
  和 seq 审计策略，597
ServerKeyBits 关键字, sshd_config 文件，337
setfacl 命令
  -d 选项，138
  -f 选项，137
  示例，137-138
  说明，126
  语法，135-136
setgid 权限
  安全风险，122
  符号模式，123
  绝对模式，124, 134
  说明，122
setuid 权限
  安全风险，45, 122
  符号模式，123
  绝对模式，124, 134
  使用设置的权限查找文件，140
  说明，121-122

sftp 命令, 说明，342
sh 命令, 特权版本，176
SHA1 内核提供器，267
shell, 特权版本，176
shell 脚本, 写入特权，235
shell 进程, 列出其权限，228-230
shell 命令
  /etc/d_passwd 文件项，41
  传递父 shell 进程号，228
shosts.equiv 文件, 说明，340
.shosts 文件, 说明，340
slave_datatrans_*slave* 文件, 说明，502
slave_datatrans 文件
  KDC 传播和，412-413
  说明，502
smattrpop 命令, 说明，224
smexec 命令, 说明，224
smmultiuser 命令, 说明，224
smprofile 命令
  更改权限配置文件，205
  说明，224
smrole 命令
  更改角色的属性，204
  使用，192-194
  说明，224
smuser 命令
  更改用户的 RBAC 属性，208
  说明，224
socket 审计标记，597
soft 字符串, audit_warn 脚本，578
solaris.device.revoke 授权，89
Solaris 安全 Shell
  scp 命令，328-329
  TCP 和，319
  本地端口转发，327-328, 328
  创建密钥，320-323
  当前发行版中的更改，314
  到防火墙外部的连接
    通过命令行，330
    通过配置文件，329-330
  登录到远程主机，323-324
  登录环境变量和，337-338
  典型会话，331-333
  复制文件，328-329
  更改口令短语，323
  公钥验证，312
Solaris 安全 Shell（续）
  关键字，334-338
  管理，331-333
  管理员任务列表，314, 315
  基于 OpenSSH，314
  跨防火墙的连接，329
  命令执行，333
  命名身份文件，339
  配置端口转发，318-319
  配置服务器，333
  配置客户机，333
  软件包，339
  生成密钥，320-323
  使用端口转发，327-328
  数据转发，333
  说明，311
  添加到系统，339
  文件，339
  协议版本，311
  验证
    要求，312-313
  验证步骤，332
  验证方法，312-313
  以较少的提示登录，324-326
  用户过程，319-320
  远程端口转发，328
  在不提供口令的情况下使用，324-326
  转发邮件，327-328
Solaris 安全 Shell 中的 ALTSHELL，338
Solaris 安全 Shell 中的 CONSOLE，337
Solaris 安全 Shell 中的 PASSREQ，338
Solaris 安全 Shell 中的 PATH，338
Solaris 安全 Shell 中的 RETRIES，338
Solaris 安全 Shell 中的 SUPATH，338
Solaris 安全 Shell 中的 TIMEOUT，338
Solaris 安全 Shell 中的 TZ，338
solaris 安全策略，221
Solaris 加密框架, **请参见**加密框架
Solaris 审计任务列表，535
sr_clean 脚本, 说明，92
ssh-add 命令
  存储私钥，324-326
  示例，324-326
  说明，341
ssh-agent 命令
  脚本中，326

ssh-agent 命令（续）
配置，326
说明，341
通过命令行，324-326
.ssh/config 文件
覆盖，340
说明，340
ssh_config 文件
覆盖，340
关键字，334-338
请参见特定关键字
配置 Solaris 安全 Shell，333
主机特定参数，337
.ssh/environment 文件, 说明，340
ssh_host_dsa_key.pub 文件, 说明，339
ssh_host_dsa_key 文件, 说明，339
ssh_host_key.pub 文件, 说明，339
ssh_host_key 文件
覆盖，341
说明，339
ssh_host_rsa_key.pub 文件, 说明，339
ssh_host_rsa_key 文件, 说明，339
.ssh/id_dsa 文件，341
.ssh/id_rsa 文件，341
.ssh/identity 文件，341
ssh-keygen 命令
使用，320-323
说明，341
ssh-keyscan 命令, 说明，341
ssh-keysign 命令, 说明，341
.ssh/known_hosts 文件
覆盖，341
说明，340
ssh_known_hosts 文件，339
.ssh/rc 文件, 说明，340
ssh 命令
端口转发选项，327-328
覆盖关键字设置，342
使用，323-324
使用代理命令，330
说明，341
sshd_config 文件
/etc/default/login 项的覆盖，337-338
关键字，334-338
请参见特定关键字
说明，339
sshd.pid 文件, 说明，339
sshd 命令, 说明，341
sshrc 文件, 说明，340
st_clean 脚本
说明，91
用于磁带机，91
sticky 位权限
符号模式，123
绝对模式，124, 134
说明，122
StrictHostKeyChecking 关键字, ssh_config 文件，337
StrictModes 关键字, sshd_config 文件，337
su 命令
监视使用，66
在角色承担中，199-201, 201-202
在控制台上显示访问尝试，67-68
su 文件, 监视 su 命令，66
subject 审计标记, 格式，598
Subsystem 关键字, sshd_config 文件，337
sufficient 控制标志, PAM，304
sulog 文件，66
监视内容，66
suser 安全策略，221
svcadm 命令
管理加密框架，253, 254
启用 keyserver 守护进程，289
启用加密框架，281-282
刷新加密框架，270-272
重新启动 NFS 服务器，548
重新启动 Solaris 安全 Shell，319
重新启动 syslog 守护进程，59, 540
重新启动名称服务，189
svcs 命令
列出 keyserver 服务，289
列出加密服务，281-282
SYS 权限，178
sysconf.rpt 文件，146, 149
syslog.conf 文件
audit.notice 级别，540
kern.notice 级别，127
priv.debug 项，246
保存失败的登录尝试，58-59
和审计，575
可执行栈消息，127
审计记录，518

SYSLOG_FAILED_LOGINS
  系统变量，58
  在 Solaris 安全 Shell 中，338
syslog 格式, 审计记录，575
SyslogFacility 关键字, sshd_config 文件，337
system state 审计类，580
system-wide administration 审计类，580
system 文件, bsmconv 影响，574

T

tail 命令, 用法示例，534
TASKS 变量 (ASET)，152, 157
taskstat 命令 (ASET)，145, 148
TCP
  Solaris 安全 Shell 和，319, 333
  地址，593
telnet 命令
  Kerberos 和，495-496
  说明，502-503
telnetd 守护进程, Kerberos 和，503-504
text 审计标记, 格式，600
TGS, 获取凭证，509-510
TGT, 在 Kerberos 中，346-347
/tmp/krb5cc_*uid* 文件, 说明，502
/tmp/ovsec_adm.*xxxxx* 文件, 说明，502
tmpfile 字符串, audit_warn 脚本，578
TMPFS 文件系统, 安全性，122
trail 审计策略
  和 trailer 标记，532
  说明，532
trailer 审计标记
  praudit 显示，600
  格式，600
  审计记录中的顺序，600
truss 命令, 用于权限调试，230-231
tune.rpt 文件，145, 149

U

-U 选项
  allocate 命令，89
  list_devices 命令，88
uauth 审计标记，525, 600
UDP
  Solaris 安全 Shell 和，319
  地址，593
  端口转发和，319
  用于远程审计日志，522
UID_ALIASES 变量 (ASET)，151, 153, 157
uid_aliases 文件 (ASET)，151, 153
umask 值
  典型设置，122
  和文件创建，122-123
umount 命令, 带有安全属性，77
UNIX 文件权限, **请参见**文件, 权限
update_drv 命令
  使用，73-74
  说明，87
use_authid 选项, SASL 和，309
UseLogin 关键字, sshd_config 文件，337
user administration 审计类，581
user_attr 数据库
  defaultpriv 关键字，246
  limitpriv 关键字，246
  RBAC 关系，217-218
  说明，216, 218-219
User 关键字, ssh_config 文件，337
useradd 命令
  说明，224
  添加本地用户，196
userdel 命令, 说明，224
UserKnownHostsFile 关键字, ssh_config 文件，337
UserKnownHostsFile2 关键字, **请参见**UserKnownHostsFile 关键字
usermod 命令
  更改用户的 RBAC 属性，208
  说明，224
  用于指定角色，194-195
/usr/aset/asetenv 文件，151, 152
/usr/aset/masters/tune 文件
  规则，158
  说明，151
  修改，153
/usr/aset/masters/uid_aliases 文件，151
/usr/aset/reports/latest 目录，149
/usr/aset/reports 目录, 结构，149
/usr/aset/reports 目录结构，148
/usr/aset 目录，144
/usr/lib/kprop 命令, 说明，503

/usr/lib/krb5/kadmind 守护进程, Kerberos 和，503
/usr/lib/krb5/kpropd 守护进程, Kerberos 和，503
/usr/lib/krb5/krb5kdc 守护进程, Kerberos 和，503
/usr/lib/krb5/ktkt_warnd 守护进程, Kerberos 和，503
/usr/lib/libsasl.so 库, 概述，307
/usr/sbin/gkadmin 命令, 说明，503
/usr/sbin/kadmin.local 命令, 说明，503
/usr/sbin/kadmin 命令, 说明，503
/usr/sbin/kclient 命令, 说明，503
/usr/sbin/kdb5_util 命令, 说明，503
/usr/sbin/kproplog 命令, 说明，503
/usr/share/lib/xml 目录，573
usrgrp.rpt 文件
  示例，150
  说明，146, 149
uucico 命令, 登录程序，60

V

-v 选项
  digest 命令，260
  mac 命令，262
  ppriv 命令，228
v1 协议, Solaris 安全 Shell，311
v2 协议, Solaris 安全 Shell，311
/var/adm/auditlog 文件, 文本审计记录，540
/var/adm/loginlog 文件, 保存失败的登录尝试，57-58
/var/adm/messages 文件, 可执行栈消息，127
/var/adm/sulog 文件, 监视内容，66
/var/krb5/.k5.*REALM* 文件, 说明，502
/var/krb5/kadmin.log 文件, 说明，502
/var/krb5/kdc.log 文件, 说明，502
/var/krb5/principal.kadm5.lock 文件, 说明，502
/var/krb5/principal.kadm5 文件, 说明，502
/var/krb5/principal.ok 文件, 说明，502
/var/krb5/principal.ulog 文件, 说明，502
/var/krb5/principal 文件, 说明，502
/var/krb5/slave_datatrans_*slave* 文件, 说明，502
/var/krb5/slave_datatrans 文件, 说明，502
/var/log/authlog 文件, 失败的登录，58-59
/var/run/sshd.pid 文件, 说明，339
VerifyReverseMapping 关键字, ssh_config 文件，337
vnode 审计标记, 格式，588
vold 守护进程, 由设备分配关闭，78

W

warn.conf 文件, 说明，502

X

-X 选项, 基于 Kerberos 的命令，496
X 窗口系统, 和 SEAM Administration Tool，446-447
-x 选项
  praudit 命令，572
  基于 Kerberos 的命令，496
X11 转发
  在 Solaris 安全 Shell 中，333
  在 ssh_config 文件中配置，335
X11DisplayOffset 关键字, sshd_config 文件，337
X11Forwarding 关键字, sshd_config 文件，337
X11UseLocalHost 关键字, sshd_config 文件，337
xauth 命令, X11 转发，337
XAuthLocation 关键字, Solaris 安全 Shell 端口转发，337
XML 格式, 审计记录，567
XML 选项, praudit 命令，572
Xylogics 磁带机设备清理脚本，91

Y

YPCHECK 变量 (ASET)，153, 157

Z

zonename 审计标记，525, 601
zonename 审计策略
  使用，579
  说明，532

安

安全 NFS，285
安全 RPC
  keyserver，287
  概述，49-50
  和 Kerberos，286
  实现，286-288
  说明，285

安全 RPC（续）
替代项，49
安全策略, 缺省 (RBAC)，216
安全服务, Kerberos 和，351
安全机制, 使用 -m 选项指定，496
安全连接
登录，323-324
跨防火墙，329
安全模式, 设置环境，386-388
安全属性
打印机管理权限配置文件，173
检查，174
命令的特殊 ID，174
命令权限，174
说明，171
用于挂载已分配设备，77
直接指定时的注意事项，177
安全性
BART，96
DH 验证，286-288
Kerberos 验证，387
NFS 客户机-服务器，286-288
保护 PROM，68-70
保护设备，91-92
保护硬件，68-70
策略概述，32
防止拒绝服务，45
计算文件的 MAC，261-263
计算文件摘要，260-261
加密文件，263-266
禁止远程登录，67-68
口令加密，38
跨不安全的网络，329
设备，41-43
审计和，519
系统硬件，68-70
指向 JASS 工具包的链接，45
阻止特洛伊木马程序，44
安装
加密框架中的提供器，255
口令加密模块，64-65

帮

帮助
SEAM Administration Tool，447
联机 URL，362
帮助内容, SEAM Administration Tool，447

包

包传送
包粉碎，50-51
防火墙安全性，50

保

保存, 失败的登录尝试，57-58
保护
BIOS, 指针，68-69
PROM，68-69
使用加密框架保护文件，257-258
保护级别
clear，496
private，496
safe，496
在 ftp 中设置，496
保护文件
任务列表，127
使用 ACL，125-126, 134-140
使用 ACL 保护文件的任务列表，134
使用 UNIX 权限，119-125, 128-134
使用 UNIX 权限任务列表，128
用户过程，128-134
保密性
Kerberos 和，345
安全服务，351
可用性，496

报

报告
ASET，149, 150, 155
BART，93
比较 (ASET)，151
目录 (ASET)，149

报告工具, **请参见**bart compare

备
备份
  Kerberos 数据库，412-413
  从 KDC，360

变
变量
  ASET 环境变量
    ASETDIR，155
    ASETSECLEVEL，156
    CKLISTPATH_level，151, 153, 157
    PERIODIC_SCHEDULE，153, 156
    TASKS，152, 157
    UID_ALIASES，151, 153, 157
    YPCHECK，153, 157
    摘要，155
  KEYBOARD_ABORT，69-70
  login 和 Solaris 安全 Shell，337-338
  noexec_user_stack，127
  noexec_user_stack_log，127
  rstchown，130
  代理服务器和端口，330
  审计与命令关联的内容，589
  添加到审计记录，531, 589
  在 Solaris 安全 Shell 中设置，338

标
标准清除, st_clean 脚本，92

表
表, gsscred，513

病
病毒
  拒绝服务攻击，45
  特洛伊木马程序，44

拨
拨号口令
  /etc/d_passwd 文件，41
  安全性，40-41
  创建，59-61
  禁用，41
  临时禁用，61

不
不可归属类，575

操
操作员 (RBAC)
  创建角色，189
  建议的角色，170
  权限配置文件的内容，213

测
测试清单，94

策
策略
  SEAM Administration Tool 面板，472-474
  Solaris OS 中的定义，32
  查看列表，464-466
  查看属性，466-468
  创建 (Kerberos)，456
  概述，32
  管理，445-483
  加密框架中的定义，253
  口令和，490

策略（续）
删除，471-472
设备上，72-73
新建 (Kerberos)，468-469
修改，470-471
用于管理的任务列表，464
用于审计，531-533
指定口令算法，61-62

插

插槽, 加密框架中的定义，253
插件
SASL 和，308
加密框架中，252
审计服务中，539

查

查看
ACL 项，138-140
shell 中的权限，228-229, 237-238
XML 审计记录，566, 572
策略列表，464-466
策略属性，466-468
二进制审计文件，565-567
加密机制
可用的，269, 275
现有的，267, 268, 275
进程的权限，228
可用的加密机制，269, 275
没有口令的用户，55-56
票证，487-489
权限配置文件的内容，215
设备策略，72-73
设备分配信息，78
审计记录格式，558-561
使用 `list` 命令查看密钥列表缓冲区，480, 481
文件的 MAC，263
文件权限，128-129
文件摘要，261
现有的加密机制，268, 275
用户的登录状态，54-55
直接指定的权限，237
查看（续）
主体列表，452-454
主体属性，454-456

超

超级用户
监视访问尝试，67-68
权限模型中的差别，179
与 RBAC 模型比较，169-171
与权限模型比较，177-183
在 RBAC 中删除，176

成

成本控制, 和审计，533
成功
关闭审计类，582
审计类前缀，581

承

承担角色
`root`，200
如何，198
系统管理员，201
在 Solaris Management Console 中，201-202
在终端窗口中，199-201
主管理员，199-200

程

程序
检查 RBAC 授权，210
可识别权限，180, 181

初

初始票证, 定义，505

处
处理时间成本, 审计服务，533

窗
窗口检验器，287

创
创建
- d_passwd 文件，60
- /etc/d_passwd 文件，60
- root 用户作为角色，196-198
- Solaris 安全 Shell 密钥，320-323
- 安全相关的角色，189-190
- 本地用户，196
- 拨号口令，59-61
- 操作员角色，189
- 存储文件，376, 423
- 二进制审计文件的分区，547-549
- 角色
  - 具有有限范围，190
  - 为特定配置文件，188-190
  - 在命令行上，191-194
- 临时用户的口令，60
- 密钥
  - 用于加密，258-260
- 密钥表文件，368
- 凭证表，384
- 权限配置文件，205-207
- 审计跟踪
  - auditd 守护进程，583
  - auditd 守护进程的角色，569-570
- 使用 kinit 创建票证，486
- 使用 Solaris Management Console 的权限配置文件，206
- 文件摘要，260-261
- 系统管理员角色，189
- 新策略 (Kerberos)，456, 468-469
- 新设备清理脚本，92
- 新主体 (Kerberos)，456-459
- 自定义角色，192-194

磁
磁带机
- 分配，83
- 清理数据，91
- 设备清理脚本，91

磁盘分区, 二进制审计文件，547-549
磁盘空间需求，533

从
从 KDC
- 定义，504
- 规划，360
- 或主，364
- 配置，372-377
- 与主 KDC 交换，405-412
- 主 KDC 和，350-351

从不审计类, audit_user 数据库，577

存
存储
- 口令短语，264
- 审计文件，529, 547-549

存储成本, 和审计，533
存储文件
- 创建，376, 423
- 定义，504

存储溢出防止, 审计跟踪，568

错
错误
- 分配错误状态，89
- 内部错误，578
- 审计目录已满，570, 578

错误消息
- encrypt 命令，265
- Kerberos，431-441
- 使用 kpasswd，490

打
打印, 审计日志，566
打印机管理权限配置文件，213

代
代理票证, 定义，506
代理守护进程, Solaris 安全 Shell，324-326

单
单点登录系统，494-500
  Kerberos 和，345

登
登录
  root 登录
    跟踪，43
    限制到控制台，67-68
    帐户，40
  安全性
    保存失败的尝试，57-58
    对设备的访问控制，40
    访问限制，37
    跟踪 root 登录，43
    系统访问控制，36
  和 AUTH_DH，287
  监视失败，57-58
  临时禁止，56-57
  任务列表，54
  失败的登录日志，58-59
  使用 Solaris 安全 Shell，323-324
  系统登录，40
  显示用户的登录状态，54-55, 55
  用户的基本权限集，180

等
等号 (=), 文件权限符号，123

低
低 ASET 安全级别，144

第
第三方口令算法, 添加，64-65

点
点 (.)
  路径变量项，44
  授权名称分隔符，216
  显示隐藏文件，128

调
调试, 权限，230
调试序列号，597
调优文件 (ASET)
  规则，158
  示例，158
  说明，151
  修改，153

读
读权限, 符号模式，123

端
端口, 用于 Kerberos KDC，359
端口转发
  Solaris 安全 Shell，327-328, 328
  在 Solaris 安全 Shell 中配置，318-319

短
短 praudit 输出格式，572

堆
堆叠, 在 PAM 中，298

对
对话密钥
  在安全 RPC 中解密，288
  在安全 RPC 中生成，287
对提供器进行签名, 加密框架，255
对象重用要求
  设备清理脚本
    编写新脚本，92
    磁带机，91
  用于设备，91-92

范
范围 (RBAC), 说明，176

方
方括号 ([]), bsmrecord 输出，585

防
防火墙系统
  ASET 设置，146
  安全的主机连接，329
  安全性，50
  包粉碎，50-51
  包传送，50-51
  使用 Solaris Secure Shell 的外部连接
    通过命令行，330
    通过配置文件，329-330
  受信任主机，50
  外部连接，330
防止
  可执行文件危及安全，127
  审计跟踪溢出，568
  系统受到危险程序的破坏，140-142

访
访问
  root 访问
    防止登录 (RBAC)，196-198
    监视 su 命令尝试，43, 66
    限制，47, 67-68
    在控制台上显示尝试，67-68
  安全 RPC 验证，285
  安全性
    ACL，47, 125-126
    NFS 客户机-服务器，286-288
    PATH 变量设置，44
    root 登录跟踪，43
    setuid 程序，45
    保存失败的登录，57-58
    报告问题，51
    登录访问限制，37
    登录控制，36
    登录验证，324-326
    防火墙设置，50
    监视系统使用情况，46
    控制系统使用情况，43-46
    设备，72
    外围设备，41
    网络控制，48-51
    文件访问限制，44
    物理安全性，36
    系统硬件，68-70
    远程系统，311
  对 KDC 服务器的限制，428-429
  访问服务器
    使用 Kerberos，509-512
  共享文件，47
  控制列表
    请参见ACL
  使用 Solaris 安全 Shell 进行登录验证，324-326
  授予帐户，493-494
  为特定服务获取，511-512
  系统登录，40
  限制
    设备，41-43, 72
  限制对
    系统硬件，68-70
访问控制列表
  请参见ACL
访问控制列表 (Access Control List, ACL), **请参见**ACL

非
非分层领域, 在 Kerberos 中，350-351

分
分层领域
  配置，377-378
  在 Kerberos 中，350-351, 358
分号 (;)
  device_allocate 文件，90
  安全属性的分隔符，222
分配错误状态，89
分配设备
  强制，78-79
  任务列表，81-82
  由用户，82-83
分析, praudit 命令，572

服
服务
  Kerberos 中的定义，504
  获取对特定服务的访问权限，511-512
  在主机上禁用，481-483
服务管理工具
  启用 keyserver，289
  刷新加密框架，271
  重新启动 Solaris 安全 Shell，319
  重新启动加密框架，281-282
服务密钥
  Kerberos 中的定义，504
  密钥表文件和，476-483
服务名称, PAM，302
服务器
  AUTH_DH 客户机-服务器会话，286-288
  Kerberos 中的定义，504
  获取凭证，510-511
  领域和，350-351
  配置 Solaris 安全 Shell，333
  使用 Kerberos 访问，509-512
服务主体
  从密钥表文件中删除，479-480
  规划名称，359
  说明，349
服务主体（续）
  添加至密钥表文件，476, 477-479

符
符号链接, 文件权限，121
符号模式
  更改文件权限，123, 131
  说明，123

辅
辅助审计目录，575

复
复制
  ACL 项，137
  使用 Solaris 安全 Shell 复制文件，328-329
  主体 (Kerberos)，459

附
附加箭头 (>>), 防止附加，44

高
高 ASET 安全级别，144
高速缓存, 凭证，508

更
更改
  ACL 项，137-138
  audit_class 文件，543-544
  audit_control 文件，536-539
  audit_event 文件，544-546
  root 用户成为角色，196-198
  Solaris 安全 Shell 的口令短语，323
  角色的属性，203-205

更改（续）
可分配设备，79-81
口令算法任务列表，61-62
密钥，286
命令行权限配置文件，206
命令行中的用户属性，209
权限配置文件内容，205-207
缺省口令算法，61-62
设备策略，73-74
使用 kpasswd 更改口令，490
使用 passwd 更改口令，490
特殊文件权限，133-134
文件的组拥有权，130-131
文件权限
符号模式，131
绝对模式，132-133
特殊，133-134
文件属主，129-130
域的口令算法，63
更新, 审计服务，556-557

公

公共对象, 审计，520
公共目录
sticky 位和，122
审计，520
公钥
DH 验证和，286-288
Solaris 安全 Shell 身份文件，339
更改口令短语，323
生成公钥/私钥对，320-323
公钥密码学
AUTH_DH 客户机-服务器会话，286-288
更改公钥和密钥，286
公钥数据库，286
公用密钥
计算，288
密钥，286
生成密钥
对话密钥，287
公共和秘密，286
公钥验证, Solaris 安全 Shell，312
公用密钥
DH 验证和，286-288
公用密钥（续）
计算，288

共

共享文件
和网络安全性，47
通过 DH 验证，294-295

故

故障排除
list_devices 命令，78
分配设备，83
挂载设备，85

挂

挂载
审计目录，583
通过 DH 验证的文件，294
已分配的 CD-ROM，84-85
已分配的软盘，84
已分配的设备，83-85

关

关键字
另请参见特定关键字
BART 中的属性，115
Solaris 安全 Shell，334-338
Solaris 安全 Shell 中的命令行覆盖，342
关联说明, SEAM Administration Tool，447

管

管理
另请参见管理
ACL，134-140
Kerberos
策略，463-472

管理, Kerberos（续）
密钥表，476-483
主体，450-463
NFS 客户机-服务器文件安全性，286-288
RBAC 任务列表，202-203
RBAC 属性，205-207
Solaris 安全 Shell
服务器，333
概述，331-333
客户机，333
任务列表，315
安全 RPC 任务列表，289
拨号登录，60
不具有权限，179
加密框架，254
加密框架和区域，255
加密框架任务列表，266
角色，188-190
角色的属性，203-205
口令算法，61-62
权限，227
权限配置文件，205-207
权限任务列表，227
设备，76
设备策略，71-72
设备分配，76
设备分配任务列表，76
审计，535
auditreduce 命令，561-563
在区域中，579
成本控制，533
降低存储空间需求，533
进程预选掩码，570
任务列表，535
审计跟踪溢出防止，568
审计记录，521
审计类，521, 580
审计事件，520
审计文件，565-567
说明，518
效率，534
审计跟踪溢出，568
审计记录任务列表，557
审计文件，561-563, 568
使用 Kerberos 管理口令，489-494
使用 Solaris 安全 Shell 进行远程登录，320-323
管理（续）
替换超级用户的角色，187-188
文件权限，127-128, 128-134
元插槽，254
在区域中审计，528, 579

归

归档, 审计文件，568
归档磁带机设备清理脚本，91

规

规划
Kerberos
从 KDC，360
端口，359
客户机名称和服务主体名称，359
领域，358
领域分层结构，358
领域名称，358
领域数，358
配置决策，357-362
时钟同步，361
数据库传播，361
PAM，300
RBAC，187-188
审计，527-531
审计任务列表，527
在区域中审计，528

合

合并, 二进制审计记录，561-563
合并审计文件
auditreduce 命令，561-563, 571
从不同的区域，579

环

环境变量
另请参见变量

环境变量（**续**）
ASETDIR (ASET)，155
ASETSECLEVEL (ASET)，156
CKLISTPATH_level (ASET)，153, 157
PATH，43
PERIODIC_SCHEDULE (ASET)，153, 156
Solaris 安全 Shell 和，337-338
TASKS (ASET)，152, 157
UID_ALIASES (ASET)，151, 153, 157
YPCHECK (ASET)，153, 157
覆盖代理服务器和端口，330
审计标记，589
位于审计记录中，531, 585
用于 ssh-agent 命令，341
摘要 (ASET)，155

恢

恢复, 加密提供器，275

会

会话 ID, 审计，583
会话密钥
Kerberos 验证和，508
Kerberos 中的定义，504

获

获取
访问特定服务，511-512
进程的权限，228-230
可转发票证，486
权限，181, 233-234
使用 kinit 获取票证，486
特权命令，203-205
用于 TGS 的凭证，509-510
用于服务器的凭证，510-511

基

基本 Solaris 用户权限配置文件，214
基本安全模块 (Basic Security Module, BSM)
请参见设备分配
请参见审计
基本权限集，180
基本审计报告工具, **请参见**BART
基于 Kerberos 的命令的选项，495
基于角色的访问控制, **请参见**RBAC
基于主机的验证
说明，312
在 Solaris 安全 Shell 中配置，315-317

机

机制
加密框架中的定义，253
禁用硬件提供器的所有机制，279-281
启用硬件提供器的一些，280

计

计算
DH 密钥，292
密钥，258-260
文件的 MAC，261-263
文件摘要，260-261
计算机安全性
请参见系统安全
计算机应急响应组/协调中心 (Computer Emergency Response Team/Coordination Center, CERT/CC)，51

加

加号 (+)
ACL 项，134
sulog 文件中的项，66
审计类前缀，581
文件权限符号，123
加密
DES 算法，286
encrypt 命令，263-266
NIS 用户的私钥，293
安全 NFS，286
安装第三方口令模块，64-65

加密（续）
保密性服务，345
口令，61-62
口令算法，38
口令算法列表，38
生成对称密钥，258-260
使用 -x 选项，496
使用用户级命令，254
文件，47, 257-258, 263-266
在 policy.conf 文件中指定口令算法，38
在 ssh_config 文件中指定算法，334
指定口令算法
本地，61-62
主机之间的通信，324
主机之间的网络通信，311-314
加密服务, **请参见**加密框架
加密管理 (RBAC)
创建角色，195
使用权限配置文件，272, 275
加密框架
cryptoadm 命令，253, 254
elfsign 命令，254
PKCS #11 库，252
安装提供器，255
错误消息，265
对提供器进行签名，255
交互，253-254
连接提供器，255
列出提供器，266-270
区域和，255, 281-282
任务列表，257
使用角色管理，195
使用者，252
刷新，281-282
说明，252
提供器，252
用户级命令，254
重新启动，281-282
术语定义，252
注册提供器，255

监

监视
su 命令尝试，43, 66
监视（续）
超级用户访问尝试，67-68
超级用户任务列表，66
失败的登录，57-58
实时审计跟踪，534
使用特权命令，195-196
系统使用情况，46

减

减号 (-)
sulog 文件中的项，66
审计类前缀，581
文件类型的符号，120
文件权限符号，123
减少, 审计文件，561-563
减小, 审计文件，571

检

检验器
窗口，287
返回到 NFS 客户机，288
说明，287

将

将审计消息复制到单个文件，565

降

降低, 审计文件的存储空间需求，534

交

交互运行 ASET，160-161
交换主 KDC 和从 KDC，405-412

脚

脚本
audit_startup 脚本，576
audit_warn 脚本，577
bsmconv 脚本，578
bsmconv 以启用审计，553-554
bsmconv 影响，574
处理 praudit 输出，573
监视审计文件示例，534
检查 RBAC 授权，210
权限使用，235-236
确保安全，210
设备分配的 bsmconv，76
设备清理脚本
另请参见设备清理脚本
使用权限运行，182
用于清理设备，91-92

角

角色
承担，199-201, 201-202
承担 root 角色，200
承担系统管理员角色，201
承担主管理员角色，199-200
创建
DHCP 管理角色，190
root 角色，196-198
安全相关的角色，189-190
操作员角色，189
加密管理角色，195
具有有限范围的角色，190
设备安全角色，190
网络安全角色，190
为特定配置文件，188-190
系统管理员角色，189
在命令行上，191-194
自定义操作员角色，192-194
从命令行添加，191-194
登录之后承担，176
更改属性，203-205
建议的角色，169
列出本地角色，199, 224
确定角色的特权命令，240-243
确定直接指定的权限，237-238
角色（续）
审计，195-196
使 root 用户成为角色，196-198
使用 usermod 命令指定，194-195
使用指定的角色，199-201, 201-202
说明，175-176
添加自定义角色，192-194
为特定配置文件添加，188-190
修改，203-205
修改用户指定，190
疑难解答，190
用于访问硬件，68-69
在 RBAC 中使用，169
在 Solaris Management Console 中承担，201-202
在终端窗口中承担，176, 199-201
摘要，171
指定权限给，233-234

结

结束, 在关闭审计功能期间收到的信号，578

解

解除分配
麦克风，86
强制，79
设备，85-86
解密
对话密钥，288
密钥，286, 287
文件，264

禁

禁用
拨号口令，61
程序使用可执行栈，142
记录可执行栈消息，142
加密机制，272
键盘关闭，69-70
键盘中止，69-70
可执行栈，142

禁用（续）
临时拨号登录，61
设备分配，555
审计策略，550-553
审计服务，555-556
危及安全的可执行文件，127
系统中止序列，69-70
硬件机制，279-281
远程 root 访问，67-68
中止序列，69-70
主机上的服务 (Kerberos)，481-483
禁止
访问系统硬件，68
临时禁止登录，56-57
使用内核软件提供器，275-278
使用硬件机制，279-281
用户登录，56-57

进

进程权利管理，**请参见**权限
进程权限，178
进程审计特征
进程预选掩码，582
审计 ID，583
审计会话 ID，583
终端 ID，583
进程预选掩码，说明，582

井

井号 (#)
device_allocate 文件，90
device_maps 文件，89

绝

绝对模式
更改特殊文件权限，133-134
更改文件权限，123，132-133
设置特殊权限，124
说明，123

可

可插拔验证模块，**请参见**PAM
可代理票证，定义，506
可读的审计记录格式
将审计记录转换为，566，572
可更新票证，定义，506
可继承权限集，180
可执行栈
防止，127，142
禁止记录消息，142
日志消息，127
可转发票证
定义，505
使用 -F 选项，496，497-498
使用 -f 选项，495，497-498
示例，486
说明，346

客

客户机
AUTH_DH 客户机-服务器会话，286-288
Kerberos 中的定义，504
配置 Kerberos，388-403
配置 Solaris 安全 Shell，331，333
客户机名称，在 Kerberos 中规划，359

控

控制
访问系统硬件，68
系统访问，53
系统使用情况，43-46
控制标志，PAM，303
控制清单 (BART)，93
控制台，显示 su 命令尝试，67-68

口

口令
LDAP，37
指定新的口令算法，64
NIS，37

口令, NIS（续）
指定新的口令算法，63
NIS+，37
指定新的口令算法，63-64
PROM 安全模式，36, 68-70
Solaris 安全 Shell 中的验证，312
UNIX 和 Kerberos，489-494
安装第三方加密模块，64-65
本地，37
拨号口令
/etc/d_passwd 文件，41
临时禁用，61
策略和，490
查找没有口令的用户，55-56
创建拨号，59-61
登录安全性，36, 37
管理，489-494
加密算法，38
临时禁用拨号，61
密钥解密，287
任务列表，54
使用 Blowfish 加密算法，63
使用 kpasswd 命令更改，490
使用 MD5 加密算法，62-63
使用 passwd -r 命令更改，37
使用 passwd 命令更改，490
使用新算法，63
授予访问权限但不显示，493-494
系统登录，37, 40
显示没有口令的用户，56
修改主体口令，460
选择建议，489-490
要求硬件访问，68-69
硬件访问和，68-69
在 Solaris 安全 Shell 中删除，324-326, 326
指定算法，62-63
本地，61-62
在名称服务中，63
口令短语
encrypt 命令，264
mac 命令，262
安全存储，264
更改 Solaris 安全 Shell，323
示例，324
用于 MAC，262-263
在 Solaris 安全 Shell 中使用，322, 324-326
口令验证, Solaris 安全 Shell，312

库

库, 用户级提供器，267

跨

跨领域验证, 配置，377-380

类

类, **请参见**审计类

联

联机帮助
SEAM Administration Tool，447
URL，362
联机帮助的 URL, SEAM Tool，362

列

列表权限, SEAM Administration Tool 和，475
列出
加密框架提供器，278
加密框架中的可用提供器，266-270
加密框架中的提供器，266-270
可以承担的角色，199, 224
没有口令的用户，55-56
设备策略，72-73
硬件提供器，278
列显格式字段, arbitrary 标记，587

领

领域 (Kerberos)
分层，377-378
分层或非分层，350-351
分层结构，358

领域 (Kerberos)（续）
服务器和，350-351
将主机名映射到，359
名称，358
内容，350
配置决策，358
配置跨领域验证，377-380
请求特定票证，496
数目，358
在主体名称中，349-350
直接，378-380

令

令牌, 加密框架中的定义，253

麦

麦克风
分配，82
解除分配，86

密

密钥
Kerberos 中的定义，504
创建，258-260
服务密钥，476-483
会话密钥
Kerberos 验证和，508
生成，258-260, 286
生成对称密钥，258-260
生成用于 Solaris 安全 Shell，320-323
为 NIS 用户创建 DH 密钥，293-294
为 Solaris 安全 Shell 创建，320-323
用于 MAC，263
密钥表文件
创建，368
管理，476-483
将服务主体添加至，476, 477-479
将主 KDC 的主机主体添加到，371
删除服务主体从，479-480
使用 delete_entry 命令禁用主机的服务，481
密钥表文件（续）
使用 ktremove 命令删除主体，479
使用 ktutil 命令查看内容，479, 480-481
使用 ktutil 命令进行管理，476
使用 list 命令查看密钥列表缓冲区，480, 481
使用 read_kt 命令读入密钥表缓冲区，480, 481
密钥分发中心, **请参见**KDC

面

面板, SEAM Administration Tool 表，472-474

名

名称
设备名称
device_maps 文件，89, 90
审计类，580
审计文件，584
名称服务
另请参见各种名称服务
范围和 RBAC，176

命

命令
另请参见单个命令
ACL 命令，126
Kerberos，502-503
RBAC 管理命令，223-224
Solaris 安全 Shell 命令，341-342
安全 RPC 命令，286
加密框架命令，254
检查权限，175
确定用户的特权命令，238-240
设备策略命令，87
设备分配命令，88
审计命令，569
文件保护命令，119
用户级加密命令，254
用于管理权限，245
指定权限，181
命令执行, Solaris 安全 Shell，333

命名约定
RBAC 授权，216
Solaris 安全 Shell 身份文件，339
设备，78
审计目录，538, 576
审计文件，583

模

模块
PAM，304
PAM 中的类型，302
口令加密，38
模式, 加密框架中的定义，253

目

目录
另请参见文件
ACL 项，126
`audit_control` 文件定义，575
`auditd` 守护进程指针，570
报告 (ASET)，149
公共目录，122
工作目录 (ASET)，155, 160-161
挂载审计目录，583
核对表任务设置 (ASET)，153, 157
权限
缺省值，122-123
说明，120-121
审计目录已满，570, 578
显示文件和相关信息，119, 128-129
主文件 (ASET)，151

内

内核提供器, 列出，267

配

配置
`ahlt` 审计策略，551
配置（**续**）
ASET，152-154, 154
`audit_class` 文件，543-544
`audit_control` 文件，536-539
`audit_event` 文件，544-546
`audit_startup` 脚本，550-553
`audit_user` 数据库，541-543
`audit_warn` 脚本，550
`auditconfig` 命令，574
Kerberos
NFS 服务器，382-384
从 KDC 服务器，372-377
概述，363-429
客户机，388-403
跨领域验证，377-380
任务列表，363-364
添加管理主体，368
主 KDC 服务器，365-372
NIS+ 用户的 DH 密钥，291-292
NIS+ 中的 DH 密钥，290-291
NIS 用户的 DH 密钥，293-294
NIS 中的 DH 密钥，292-293
`perzone` 审计策略，552
RBAC，186-198
RBAC 任务列表，186
`root` 用户作为角色，196-198
Solaris 安全 Shell，314
服务器，333
客户机，333
Solaris 安全 Shell 的基于主机的验证，315-317
Solaris 安全 Shell 任务列表，315
Solaris 安全 Shell 中的端口转发，318-319
`ssh-agent` 守护进程，326
拨号登录，60
角色，188-190, 203-205
通过命令行，191-194
临时审计策略，552
名称服务，198
命令行权限配置文件，206
权限配置文件，205-207
设备策略，71-72
设备分配，76
设备任务列表，71
审计策略，550-553
审计服务任务列表，546
审计跟踪溢出防止，568

配置（续）
审计文件，536-546
审计文件任务列表，536
文本审计日志，539-541
硬件安全性，68-70
硬件访问口令，68-69
在区域中审计，579
自定义角色，192-194
配置决策
Kerberos
从 KDC，360
端口，359
将主机名映射到领域，359
客户机名称和服务主体名称，359
领域，358
领域分层结构，358
领域名称，358
领域数，358
时钟同步，361
数据库传播，361
口令算法，38
审计
策略，531-533
区域，528
文件存储，529
要审计的对象及内容，529-531
配置文件
请参见权限配置文件
ASET，144
audit_class 文件，575
audit_control 文件，536-539, 570, 575
audit_event 文件，576
audit_startup 脚本，576
audit_user 数据库，577
device_maps 文件，89
nsswitch.conf 文件，36
pam.conf 文件，302, 304
用于口令算法，38
policy.conf 文件，38, 62-63, 224
Solaris 安全 Shell，331
syslog.conf 文件，58-59, 246, 575
system 文件，574
包含权限信息，246-247
配置文件 shell, 说明，176
配置应用程序服务器，380-381

票

票证
-F 选项或 -f 选项，496
-k 选项，496
Kerberos 中的定义，504
klist 命令，487-489
查看，487-489
初始，505
创建，485
代理，506
定义，346
或凭证，346
获取，485
可代理，506
可更新，506
可以后生效，505
可转发，346, 486, 497-498, 505
类型，505-508
请求特定领域，496
生命周期，506-507
使用 kinit 创建，486
未生效，505
文件
请参见凭证高速缓存
销毁，489
以后生效的，346
有关失效的警告，398
最长可更新生命周期，507
票证类型，505-508
票证生命周期, 在 Kerberos 中，506-507
票证授予服务, **请参见**TGS
票证授予票证, **请参见**TGT
票证文件, **请参见**凭证高速缓存

凭

凭证
高速缓存，508
或票证，346
说明，287, 505
为 TGS 获取，509-510
为服务器获取，510-511
映射，360
凭证表, 添加单项，384-385

其

其他 ACL 项, 说明，125-126

启

启动
- ASET 交互，160-161
- KDC 守护进程，377, 423
- shell 中的 ASET，144
- 安全 RPC keyserver，289
- 定期运行 ASET，161-162
- 设备分配，76-77
- 审计，553-554
- 审计守护进程，557

启用
- 加密机制，274
- 键盘中止，69-70
- 仅基于 Kerberos 的应用程序，428
- 设备分配，76-77
- 审计，553-554
- 审计服务，553-554
- 审计服务任务列表，546
- 使用内核软件提供器，275
- 硬件提供器的机制和功能，280

强

强制清除, `st_clean` 脚本，92

清

清除, 二进制审计文件，567-568

清单
- **另请参见**`bart create`
- 测试，94
- 控制，93
- 文件格式，113-114
- 自定义，99-104

区

区域
- `perzone` 审计策略，579
- `zonename` 审计策略，579
- 规划审计，528
- 加密服务和，281-282
- 加密框架和，255
- 设备和，42
- 审计和，579
- 在全局区域中配置审计，551

权

权限
- **请参见**权限配置文件
- ACL 和，47, 125-126
- ASET 处理，144, 145
- `setgid` 权限
  - 符号模式，123
  - 绝对模式，124, 134
  - 说明，122
- `setuid` 权限
  - 安全风险，122
  - 符号模式，123
  - 绝对模式，124, 134
  - 说明，121-122
- sticky 位，122
- `umask` 值，122-123
- 保护内核进程，177
- 查找缺少的，231-232
- 超级用户模型中的差别，179
- 程序可识别权限，181
- 从基本集删除，234
- 从限制集删除，235
- 从用户中删除，182
- 调试，183, 230
- 调优文件 (ASET)，151, 153, 154
- 对 SEAM Administration Tool 的影响，475
- 更改文件权限
  - `chmod` 命令，120
  - 符号模式，123, 131
  - 绝对模式，123, 132-133
- 管理，227
- 具有指定权限的进程，181
- 类别，178

权限（续）
命令，245
目录权限，120-121
缺省值，122-123
确定直接指定的权限，236-238
任务列表，227
如何使用，236
设备和，182-183
审计和，247
升级，247
使用 setuid 权限查找文件，140
使用权限执行命令，182
说明，171, 178
特殊文件权限，121-122, 122, 124
添加到命令，232-233
文件，246-247
文件权限
符号模式，123, 131
更改，123-125, 131
绝对模式，123, 132-133
说明，120-121
特殊权限，122, 124
限制用户或角色的使用，234-235
用户类和，120
由进程继承，181
与超级用户模型比较，177-183
在 shell 脚本中使用，235-236
在集中实现，180
在进程上列出，228-230
指定给脚本，182
指定给命令，181
指定给用户，181
指定给用户或角色，233-234
权限工具, 说明，205-207
权限集
基本，180
将权限添加到，182
可继承，180
列出，180
删除权限，182
限制，180
有效，180
允许，180
权限检查, 在应用程序中，174
权限配置文件
用于审计服务，579
权限配置文件（续）
查看内容，215
创建
在 Solaris Management Console 中，206
在命令行上，205
创建方法，205-207
创建角色，188-190
典型内容，211
更改内容，205-207
排序，215
使用系统管理员配置文件，68
数据库
请参见 prof_attr 数据库和 exec_attr 数据库
说明，171, 175
通过命令行更改，206
修改，205-207
疑难解答，207
主要权限配置文件说明，211

缺

缺省值
audit_startup 脚本，576
policy.conf 文件中的权限设置，246
policy.conf 文件中的系统范围，38
praudit 输出格式，573
umask 值，122-123
目录的 ACL 项，126
系统范围审计，580

确

确保安全
登录任务列表，54
脚本，210
口令任务列表，54
确定
进程的权限，228-230
权限任务列表，236
使用 setuid 权限确定文件，140
文件是否具有 ACL，134

任

任务列表
ASET，159-164
Kerberos 配置，363-364
Kerberos 维护，364
PAM，299
Solaris 安全 Shell，314
保护文件，127
保护系统硬件，68
保证登录和口令的安全，54
保证系统安全，53
防止程序受到安全风险，140
分配设备，81-82
更改口令加密的缺省算法，61-62
管理 RBAC，202-203
管理安全 RPC，289
管理策略 (Kerberos)，464
管理和使用权限，227
管理加密框架，266
管理设备策略，71-72
管理设备分配，76
管理审计记录，557
管理主体 (Kerberos)，451
规划审计，527
加密框架，257
监视和限制超级用户，66
控制对系统硬件的访问，68
配置 Kerberos NFS 服务器，382
配置 RBAC，186
配置 Solaris 安全 Shell，315
配置设备，71
配置设备策略，71-72
配置审计服务，546
配置审计文件，536
启用审计服务，546
设备，71
设备策略，71-72
设备分配，76
审计，535
使用 ACL 保护文件，134
使用 BART 任务列表，95-96
使用 RBAC，185
使用 Solaris 安全 Shell，319-320
使用 UNIX 权限保护文件，128
使用加密机制保护文件，257-258
使用加密框架，257
任务列表（续）
使用角色，198
使用设备分配，81-82
系统访问，53
运行 ASET，159-164

日

日志文件
BART
程序输出，116-117
详细输出，116-117
syslog 审计记录，575
监视 su 命令，66
检查审计记录，571
配置审计服务，539-541
审计记录，522, 566
审计记录所用的空间，570
失败的登录尝试，58-59
执行日志 (ASET)，147

软

软件包, Solaris 安全 Shell，339
软盘驱动器
分配，84
设备清理脚本，91-92
软限制
audit_warn 情况，578
minfree 行说明，575

散

散列, 文件，257-258

删

删除
ACL 项，126, 138
not_terminated 审计文件，567-568
策略 (Kerberos)，471-472
基本集中的权限，234

删除（续）
加密提供器，274, 275
密钥表文件中的服务主体，479-480
权限配置文件，205
软件提供器
临时，276
永久，277
设备策略，74
设备的策略，74
审计文件，561
使用 ktremove 命令删除主体，479
限制集中的权限，235
已归档审计文件，568
主机的服务，481
主体 (Kerberos)，460-461

设

设备
/dev/urandom 设备，258-260
安全性，41-43
不需要授权就可使用，80
策略命令，87
查看分配信息，78
查看设备策略，72-73
超级用户模型和，182-183
登录访问控制，40
防止使用某些，80
防止使用所有，80-81
分配使用，81-82
更改可分配的，79-81
更改设备策略，73-74
挂载已分配的设备，83-85
管理，71-72
管理分配，76
获取 IP MIB-II 信息，75
解除设备分配，85-86
列出，72-73
列出设备名称，78
强制分配，78-79
强制解除分配，79
区域和，42
权限模型和，182-183
删除策略，74
设备（续）
设备分配
请参见设备分配
审计策略更改，74-75
审计分配，81
使可分配，76-77
授权用户来分配，77-78
添加设备策略，73-74
通过设备分配保护，41
卸载已分配的设备，86
在内核中保护，41
设备安全 (RBAC), 创建角色，190
设备策略
add_drv 命令，87
update_drv 命令，73-74, 87
查看，72-73
从设备中删除，74
概述，41-43
更改，73-74
管理设备，71-72
命令，87
内核保护，87-92
配置，72-75
任务列表，71-72
审计更改，74-75
设备分配
allocate 命令，88
deallocate 命令，88
设备清理脚本和，92
使用，85-86
device_allocate 文件，90-91
device_maps 文件，89-90
不需要授权，80
查看信息，78
防止，80-81
分配错误状态，89
分配设备，82-83
更改可分配设备，79-81
挂载设备，83-85
管理设备，76
机制的组成，87
解除设备分配，85-86
禁用，555
可分配设备，91
命令，88
命令授权，88-89

设备分配（**续**）
配置文件，89
启用，76-77
强制分配设备，78-79
强制解除设备分配，79
任务列表，76
设备清理脚本
CD-ROM 驱动器，91-92
编写新脚本，92
磁带机，91
软盘驱动器，91-92
说明，91-92
选项，92
音频设备，92
审计，81
使设备可分配，76-77
使用，81-82
使用 `allocate` 命令，82-83
示例，83
授权用户来分配，77-78
添加设备，76
卸载已分配的设备，86
需要授权，79-81
用户过程，81-82
设备管理，**请参见**设备策略
设备清理脚本
CD-ROM 驱动器，91-92
编写新脚本，92
磁带机，91
和对象重用，91-92
软盘驱动器，91-92
说明，91-92
选项，92
音频设备，92
设置
审计策略，550-553
主体缺省值 (Kerberos)，461-462

身

身份文件（Solaris 安全 Shell），命名约定，339

审

审计
当前发行版中的更改，524-525
更新信息，556-557
规划，527-531
角色，195-196
禁用，555-556
启用，553-554
区域和，579
权限和，247
权限配置文件，579
设备策略中的更改，74-75
设备分配，81
预选定义，520
在区域中规划，528
在全局区域中配置，528, 551
审计 ID
概述，517-518
机制，583
审计标记
另请参见单独的审计标记名称
当前发行版中的新增内容，524
格式，585
列表，585
审计记录格式，584
说明，519, 522
审计策略
`public`，532
缺省值，531-533
设置，550-553
设置 `ahlt`，551
设置 `perzone`，552
说明，519
影响，531-533
在全局区域中设置，579
审计查看权限配置文件，579
审计跟踪
包括的事件，521
查看不同区域中的事件，579
查看事件，565-567
创建
`auditd` 守护进程的角色，570
防止溢出，568
分析成本，533
概述，518
合并所有文件，571

审计跟踪（续）
清除未终止文件，567-568
审计策略的影响，531
实时监视，534
使用 praudit 命令进行分析，572
说明，520
无公共对象，520
选择事件，563-565
审计会话 ID，583
审计记录
syslog.conf 文件，518
/var/adm/auditlog 文件，540
标记系列，584
概述，521
格式，584
格式设置，558
合并，561-563
减少审计文件，561-563
审计目录已满，570, 578
生成事件，518
说明，519
显示，565-567
显示程序的格式，558-560
显示格式
过程，558-561
摘要，570
显示审计类的格式，560-561
以 XML 格式显示，567
转换为可读格式，566, 572
审计记录格式, bsmrecord 命令，558
审计控制权限配置文件，579
审计类
audit_control 文件中的项，575
audit_user 数据库中的例外，577
定义，580
概述，521
进程预选掩码，582
前缀，581
设置系统范围，580
说明，519, 520
添加，543-544
系统范围，575
系统范围设置的例外，521
修改缺省值，543-544
映射事件，521
语法，581, 582
审计类（续）
预选，520, 536-539
审计类的前缀，581
审计类前缀中的 ^（插入记号），582
审计类前缀中的插入记号 (^)，582
审计类预选, 对公共对象的作用，520
审计目录
创建，548-549
分区，547-549
结构样例，571
说明，519
审计配置文件, **请参见** audit_control 文件
审计日志
另请参见 审计文件
比较二进制和文本，522
模式，522
配置文本审计日志，539-541
文本格式，575
审计事件
audit_event 文件，520
从二进制文件查看，565-567
从区域内的审计跟踪中进行选择，579
从审计跟踪选择，563-565
更改类成员关系，544-546
说明，520
映射到类，521
摘要，519
审计守护进程, **请参见** auditd daemon
审计特征
会话 ID，583
进程，582
进程预选掩码，570
审计 ID，583
用户进程预选掩码，582
终端 ID，583
审计文件
auditreduce 命令，571
打开顺序，575
打印，566
分区磁盘，547-549
管理，568
合并，561-563, 571
减少，561-563
减小，571
将消息复制到单个文件，565
降低存储空间需求，533, 534

审计文件（续）
可用于文件系统的最小空闲空间，575
名称，584
配置，536-546
切换到新文件，570
时间标记，584
审计文件大小
减少，561-563
减小，571
降低存储空间需求，534
审计消息, 复制到单个文件，565
审计阈值，575
审计预选掩码, 针对各个用户进行修改，541-543
审计中的预选, 审计，520

生

生成
Solaris 安全 Shell 密钥，320-323
密钥，286
用于 Solaris 安全 Shell 的密钥，320-323
用于加密的对称密钥，258-260

失

失败
关闭审计类，582
审计类前缀，581
失败的登录尝试
loginlog 文件，57-58
syslog.conf 文件，58-59

实

实例, 在主体名称中，349-350

时

时间标记
ASET 报告，149
审计文件，584
时钟同步
Kerberos 和，361, 372, 377, 423
时钟相位差
Kerberos 和，361, 404-405

使

使用
ACL，135-136
allocate 命令，82-83
ASET，159-164
BART，96
cryptoadm 命令，266
deallocate 命令，86
digest 命令，260-261
encrypt 命令，263-266
mac 命令，261-263
mount 命令，84
ppriv 命令，228
RBAC 任务列表，185
smrole 命令，233
Solaris 安全 Shell 任务列表，319-320
ssh-add 命令，324-326
ssh-agent 守护进程，324-326
truss 命令，230-231
umount 命令，86
usermod 命令，233
加密框架任务列表，257
角色，198
角色任务列表，198
权限，236
权限任务列表，236
设备分配，81-82, 82-83
文件权限，127-128
新的口令算法，63
使用者, 加密框架中的定义，252

始

始终审计类
audit_user 数据库，577
进程预选掩码，582

事

事件, 说明，520
事件修饰符字段标志（header 标记），591

守

守护进程
- auditd，569-570
- kcfd，254
- Kerberos 表，503-504
- keyserv，289
- nscd（名称服务高速缓存守护进程），189, 223
- rpc.nispasswd，64
- ssh-agent，324-326, 326
- sshd，331-333
- vold，78
- 使用权限运行，179

受

受限 shell (rsh)，44
受信任主机，50

授

授权
- Kerberos 和，345
- 类型，49-50

授权 (RBAC)
- solaris.device.allocate，77, 88
- solaris.device.revoke，89
- 不需要授权就可进行设备分配，80
- 定义，174
- 分配设备，77-78
- 检查通配符，210
- 粒度，216
- 命名约定，216
- 数据库，216-223
- 说明，171, 215-216
- 委托，216
- 要求授权的命令，224-225
- 用于设备分配，88-89
- 在特权应用程序中检查，175

授予对帐户的访问权限，493-494

属

属性, BART 中的关键字，115

数

数据加密标准, **请参见**DES 加密
数据库
- audit_user，577
- auth_attr，219-220
- exec_attr，221-222
- KDC 传播，361
- prof_attr，220-221
- RBAC，216-223
- user_attr，218-219
- 安全 NFS 的 cred，286, 290
- 安全 NFS 的 publickey，286
- 包含权限信息，246-247
- 备份和传播 KDC，412-413
- 创建 KDC，367
- 密钥，286

数据转发, Solaris 安全 Shell，333

刷

刷新, 加密服务，281-282

双

双美元符号 ($$), 父 shell 进程号，228

私

私钥
- **另请参见**密钥
- Kerberos 中的定义，504
- Solaris 安全 Shell 身份文件，339

算

算法
- 加密框架中的定义，252
- 口令
  - 配置，62-63
- 口令加密，38
- 在加密框架中列出，266-270

随

随机数, od 命令，258-260

所

所有 (RBAC), 权限配置文件，214-215

特

特洛伊木马程序，44
特权端口, 安全 RPC 的替代项，49
特权应用程序
- ID 检查，174
- 权限检查，174
- 授权检查，175
- 说明，171

特殊权限
- `setgid` 权限，122
- `setuid` 权限，121-122
- sticky 位，122

提

提供器
- 安装，255
- 定义为插件，252
- 恢复使用内核软件提供器，275
- 加密框架中的定义，253
- 禁用硬件机制，279-281
- 禁止使用内核软件提供器，275-278
- 连接到加密框架，255
- 列出硬件提供器，278
- 签名，255

提供器（续）
- 添加库，272
- 添加软件提供器，270-272
- 添加用户级软件提供器，272
- 在加密框架中列出，266-270
- 注册，255

替

替换, 超级用户使用角色，187-188

添

添加
- ACL 项，135-136
- DH 验证的密钥，290-291
- PAM 模块，300-301
- RBAC 属性到传统应用程序，209-210
- 安全相关的角色，189-190, 195
- 安全性到设备，73-74, 76-81
- 安全性属性到传统应用程序，209-210
- 本地用户，196
- 拨号口令，59-61
- 操作员角色，189
- 服务主体至密钥表文件 (Kerberos)，477-479
- 管理主体 (Kerberos)，368
- 加密管理角色，195
- 加密框架插件，270-272
- 角色
  - 具有有限范围，190
  - 通过命令行，191-194
  - 为特定配置文件，188-190
  - 至用户，190
- 角色审计，195-196
- 可分配设备，76-77
- 口令加密模块，64-65
- 库插件，272
- 区域审计，527-531
- 权限到命令，232-233
- 权限配置文件的属性，205-207
- 权限直接给用户或角色，233-234
- 软件提供器，270-272
- 审计策略，552
- 审计类，543-544

添加（续）
审计目录，547-549
使用 Solaris Management Console 的权限配置文件，206
系统管理员角色，189
系统硬件安全性，68-69
新权限配置文件，205-207
已挂载文件系统的 DH 验证，289
硬件提供器机制和功能，280
用户级软件提供器，272
自定义角色，192-194
自定义角色 (RBAC)，192-194

停
停止, 临时拨号登录，61

通
通配符
Solaris 安全 Shell 中的主机，329
在 ASET 调优文件中，158
在 ASET 文件中，156
在 RBAC 授权中，216
通用安全服务 API, **请参见**GSS-API

同
同步时钟，372, 377, 404-405, 423

透
透明, Kerberos 中的定义，346

完
完整性
Kerberos 和，345
安全服务，351

网
网关, **请参见**防火墙系统
网络, 相关权限，178
网络安全 (RBAC), 创建角色，190
网络安全性
报告问题，51
防火墙系统
包粉碎，50-51
受信任主机，50
需要，50
概述，48
控制访问，48-51
授权，49-50
验证，49-50
网络时间协议, **请参见**NTP

伪
伪 tty, 在 Solaris 安全 Shell 中使用，333

委
委托, RBAC 授权，216

文
文件
ACL 项
检查，134
删除，126, 138
设置，135-136
添加或修改，137-138
显示，126, 138-140
有效项，125-126
ASET 检查，145
BART 清单，113-114
`kdc.conf`，506
Kerberos，501-502
`syslog.conf` 文件，575
安全性
ACL，47
`umask` 缺省值，122-123
UNIX 权限，119-125

文件, 安全性（续）
访问限制，44
更改权限，123-125, 131
更改拥有权，129-130
加密，47, 257-258
目录权限，120-121
特殊文件权限，124
文件类型，120
文件权限，120-121
显示文件信息，119, 128-129
用户类，120
包含权限信息，246-247
复制 ACL 项，137
更改 ACL，137-138
更改特殊文件权限，133-134
更改拥有权，119, 129-130
更改组拥有权，130-131
公共对象，520
管理 Solaris 安全 Shell，339
计算 MAC，261-263
计算摘要，260-261, 261
加密，257-258, 263-266
解密，264
清单 (BART)，113-114
权限
setgid，122
setuid，121-122
sticky 位，122
umask 值，122-123
符号模式，123, 131
更改，120, 123-125, 131
绝对模式，123, 132-133
缺省值，122-123
说明，120-121
确定是否具有 ACL，134
散列，257-258
删除 ACL，138
设置 ACL，135-136
使用 ACL 保护，134-140
使用 digest 验证完整性，260-261
使用 setuid 权限查找文件，140
使用 Solaris 安全 Shell 复制，328-329
使用 UNIX 权限保护，128-134
特殊文件，121-122
通过 DH 验证共享，294-295
通过 DH 验证挂载，294
文件（续）
文件类型，120
文件类型的符号，120
显示 ACL 项，138-140
显示文件信息，128-129
显示信息，119
显示隐藏文件，128
相关权限，178
拥有权
和 setgid 权限，122
和 setuid 权限，121-122
摘要，260-261
文件 vnode 审计标记，588
文件的拥有权
ACL 和，47, 125-126
更改，119, 129-130
更改组拥有权，130-131
文件的用户类，120
文件权限模式
符号模式，123
绝对模式，123
文件系统
NFS，285
TMPFS，122
安全性
TMPFS 文件系统，122
验证和 NFS，285
共享文件，47

问

问号 (?), 在 ASET 调优文件中，158

无

无效票证, 定义，505

物

物理安全性, 说明，36

系

系统, 防止受到危险程序的破坏，140-142
系统 V IPC
  ipc_perm 审计标记，593
  ipc 审计标记，592
  ipc 审计类，581
  权限，178
系统安全
  ACL，125-126
  root 访问限制，47, 67-68
  su 命令监视，43, 66
  保存失败的登录尝试，57-58
  拨号登录和口令，40-41
  拨号口令
    临时禁用，61
  登录访问限制，37
  防火墙系统，50
  防止受到危险程序的破坏，140-142
  概述，36
  基于角色的访问控制 (role-based access control, RBAC)，43, 169-171
  计算机访问，36
  口令，37
  口令加密，38
  权限，177-183
  任务列表，140
  受限 shell，44
  特殊的登录，40
  显示
    没有口令的用户，56
    用户的登录状态，54-55, 55
  限制远程 root 访问，67-68
  硬件保护，36, 68-70
系统变量
  另请参见变量
  CRYPT_DEFAULT，62
  KEYBOARD_ABORT，69-70
  noexec_user_stack，142
  noexec_user_stack_log，142
  rstchown，130
  SYSLOG_FAILED_LOGINS，58
系统调用
  arg 审计标记，588
  close，580
  exec_args 审计标记，589
  exec_env 审计标记，589
系统调用（**续**）
  ioctl()，581
  ioctl 来清理音频设备，92
  return 审计标记，597
系统管理员 (RBAC)
  保护硬件，68
  承担角色，201
  创建角色，189
  建议的角色，170
  权限配置文件，212
系统属性, 相关权限，178
系统硬件, 控制访问，68-70

显

显示
  ACL 项，126, 134, 138-140
  ASET 任务状态，145, 148
  root 访问尝试，67-68
  su 命令尝试，67-68
  XML 格式的审计记录，567
  加密框架中的提供器，266-270
  可分配设备，78
  可以承担的角色，199, 224
  没有口令的用户，55-56
  设备策略，72-73
  审计策略，550
  审计记录，565-567
  审计记录格式，558-561
  文件和相关信息，119
  文件信息，128-129
  选定的审计记录，561-563
  用户的登录状态，54-55, 55
  主体子列表 (Kerberos)，453

限

限制
  超级用户任务列表，66
  用户或角色的权限使用，234-235
  用户权限，234
限制对 KDC 服务器的访问，428-429
限制权限集，180

项

项大小字段, arbitrary 标记，587

消

消息验证代码 (message authentication code, MAC), 计算
文件，261-263

销

销毁, 使用 kdestroy 销毁票证，489

效

效率, 审计和，534

写

写权限, 符号模式，123

卸

卸载
加密提供器，274
已分配的设备，86

新

新功能
BART，93-117
Kerberos 增强功能，353-354
PAM 增强功能，298-299
SASL，307
Solaris 安全 Shell 增强功能，314
Solaris 加密框架，251-255
加密框架，251-255
进程权利管理，177-183
命令
bart compare，94
bart create，94

新功能, 命令（续）
cryptoadm，266
decrypt，264
digest，260-261
encrypt，263-266
getdevpolicy，72-73
kcfd，281-282
kclient，353
kpropd，353
mac，261-263
ppriv，228-230
praudit -x，566
ssh-keyscan，341
ssh-keysign，341
强口令加密，38
权限，177-183
设备策略，42
审计增强功能，524-525
系统安全增强，35-36
元插槽，251

星

星号 (*)
device_allocate 文件，90
通配符
在 ASET 中，156, 158
在 RBAC 授权中，216, 219
在 RBAC 授权中检查，210

修

修改
策略 (Kerberos)，470-471
角色 (RBAC)，203-205
用户 (RBAC)，207-209
用户的角色指定，190
主体 (Kerberos)，459-460
主体口令 (Kerberos)，460

选

选择
- 您的口令，489-490
- 审计跟踪中的事件，563-565
- 审计记录，563-565
- 审计类，536-539

掩

掩码 ACL 项
- 目录的缺省项，126
- 设置，135-136
- 说明，125-126

掩码（审计）
- 进程预选说明，582
- 系统范围的进程预选，575

验

验证
- AUTH_DH 客户机-服务器会话，286-288
- DH 验证，286-288
- Kerberos 概述，508
- Kerberos 和，345
- Solaris 安全 Shell
  - 方法，312-313
  - 进程，332
- 安全 RPC，285
- 类型，49-50
- 名称服务，285
- 配置跨领域，377-380
- 使用 -x 选项禁用，496
- 说明，49-50
- 网络安全性，49-50
- 已挂载 NFS 文件，294
- 与 NFS 一起使用，285
- 术语，504-505

验证方法
- Solaris 安全 Shell，312-313
- Solaris 安全 Shell 中的 GSS-API 凭证，312
- Solaris 安全 Shell 中的公钥，313
- Solaris 安全 Shell 中的键盘交互，313
- Solaris 安全 Shell 中的口令，313
- Solaris 安全 Shell 中基于主机，312, 315-317

验证者
- 在 Kerberos 中，505, 510

疑

疑难解答
- ASET 错误，164
- encrypt 命令，265
- Kerberos，441
- 角色功能，190
- 权限配置文件，207
- 权限要求，230-232
- 缺少权限，230-232

以

以后生效的票证
- 定义，505
- 说明，346

溢

溢出防止, 审计跟踪，568

音

音频设备, 安全性，92

应

应用程序服务器, 配置，380-381

映

映射
- Kerberos 主体的 UID，513
- 事件到类（审计），521
- 主机名到领域 (Kerberos)，359

映射 GSS 凭证，360

硬

硬件
  保护，36, 68-70
  列出连接的硬件加速器，278
  要求访问口令，68-69
硬件提供器
  禁用加密机制，279-281
  列出，278
  启用机制和功能，280
  装入，278
硬盘, 用于审计的空间需求，533

用

用户
  初始可继承权限，180
  创建本地用户，196
  分配设备，82-83
  挂载已分配的设备，83-85
  基本权限集，180
  计算文件的 MAC，261-263
  计算文件摘要，260-261
  加密文件，263-266
  解除设备分配，85-86
  禁止登录，56-57
  没有口令，55-56
  确定直接指定的权限，236-238
  确定自己的特权命令，238-240
  添加本地用户，196
  通过命令行更改属性，209
  显示登录状态，54-55
  限制基本权限，234
  卸载已分配的设备，86
  修改审计预选掩码，541-543
  修改属性 (RBAC)，207-209
  指定 RBAC 缺省值，222-223
  指定分配授权给，77-78
  指定权限给，233-234
用户 ACL 项
  目录的缺省项，126
  设置，135-136
  说明，125-126
用户 ID
  审计 ID 和，517-518, 583
  在 NFS 服务中，384
用户过程
  chkey 命令，294
  保护文件，128-134
  承担角色，198
  分配设备，81-82
  计算文件的 MAC，261-263
  计算文件摘要，260-261
  加密 NIS 用户的私钥，293
  加密文件，257-258
  解密文件，263-266
  生成对称密钥，258-260
  使用 ACL，134-140
  使用 Solaris 安全 Shell，319-320
  使用指定的角色，198
用户脚本, 配置 ssh-agent 守护进程，326
用户审计字段, audit_user 数据库，577
用户数据库 (RBAC), **请参见** user_attr 数据库
用户帐户
  另请参见用户
  ASET 检查，145
  显示登录状态，54-55, 55
用户帐户工具, 说明，207-209
用户主体, 说明，349
用于 praudit 命令的 DTD，573

邮

邮件, 使用 Solaris 安全 Shell，327-328

有

有关票证失效的警告，398
有效权限集，180

与

与 Internet 相关的标记
  in_addr 标记，592
  ip 标记，592
  iport 标记，593
  socket 标记，597

预
预选, 审计类，536-539
预选掩码（审计）
　降低存储成本，570
　说明，582
　系统范围，575

元
元插槽
　管理，254
　加密框架中的定义，253

原
原始 praudit 输出格式，572

远
远程登录
　安全性和，288
　禁止远程用户，67-68
　授权，49-50
　验证，49-50

允
允许权限集，180

运
运行 ASET 任务列表，159-164

在
在关闭审计功能期间收到的信号，578

摘
摘要
　计算文件，260-261
　文件，260-261, 261

执
执行权限, 符号模式，123
执行日志 (ASET)，147

直
直接领域，378-380

指
指定
　给权限配置文件中的命令的权限，232-233
　角色给本地用户，194-195
　角色给用户，189, 190
　权限给脚本中的命令，235-236
　权限给用户或角色，233-234

智
智能卡文档, 指针，31

中
中 ASET 安全级别，144

终
终端 ID, 审计，583
终止, 在关闭审计功能期间收到的信号，578

重
重定向箭头 (>), 防止重定向，44

重放的事务，288
重新启动
ssh 服务，319
sshd 守护进程，319
加密服务，281-282
审计守护进程，556

术

术语
Kerberos，504-508
特定于 Kerberos 的，504
特定于验证的，504-505

主

主 KDC
从 KDC 和，350-351, 364
定义，504
配置，365-372
与从 KDC 交换，405-412
主管理员 (RBAC)
承担角色，199-200
建议的角色，170
权限配置文件内容，212
主机
Solaris 安全 Shell 主机，312
禁用 Kerberos 服务，481-483
受信任主机，50
主机名, 映射到领域，359
主机主体, DNS 和，359
主名称, 在主体名称中，349-350
主审计目录，575
主体
Kerberos，349-350
SEAM Administration Tool 面板，472-474
查看列表，452-454
查看属性，454-456
查看主体子列表，453
创建，456-459
创建 clntconfig，371
创建 host，371
从密钥表文件中删除，479
从密钥表中删除服务主体，479-480
主体（续）
服务主体，349
复制，459
管理，445-483
将服务主体添加至 keytab，476, 477-479
删除，460-461
设置缺省值，461-462
添加管理，368
修改，459-460
用户 ID 比较，384
用户主体，349
用于管理的任务列表，451
主体名称，349-350
自动创建，451-452
主文件 (ASET)，145, 151

注

注册提供器, 加密框架，255

转

转换
审计记录为可读格式，566, 572

传

传播
KDC 数据库，361
Kerberos 数据库，412-413

自

自定义, 清单，99-104
自定义报告 (BART)，111-113
自定义操作员 (RBAC), 创建角色，192-194
自动安全性增强工具, **请参见**ASET
自动创建主体，451-452
自动登录
禁用，496
启用，495
自动启用审计功能，576

组

组, 更改文件拥有权，130-131
组 ACL 项
    目录的缺省项，126
    设置，135-136
    说明，125-126
组 ID (group ID, GID) 号, 特殊登录和，40
组成部分, Solaris 安全 Shell 用户会话，333
组件
    BART，94
    RBAC，171-173
    设备分配机制，87

最

最低权限, 原则，178
最低权限的原则，178